

THE MISOSYS QUARTERLY

In this issue:

- ☛ Five Twelve K: A better way, by Richard King
- ☛ Multi-Command, by Frank Durda, IV
- ☛ Touch/cc, by Kenneth Peck
- ☛ Expanded 800-MISOSYS Order Line
- ☛ Fixes for LS-DOS 6.3.1
- ☛ DoubleDuty Version 2.6.0 released

☛ and **MISOSYS** releases **SAID-86** for **MS-DOS**

**MISOSYS will be closed August
6th through 10th**

**It's time
to renew
your sub-
scription for
Volume V**





GOT A MONKEY ON YOUR BACK
WHEN PEOPLE ASK FOR TRS-80 SUPPORT!

Tell them about Computer News 80

Computer News 80 is the only MONTHLY magazine with a worldwide readership that supports the Tandy Radio Shack TRS-80 Computers Model I III 4/4P/4D. Now in the third year of publication Computer News 80 offers hints, tips, programs, problem solving and how-to-do-it articles. There is an open forum for readers questions and answers. There is an advertising section with those companies that still sell and support TRS-80 products and a classified section for reader ads, both of which creates a source guide for the TRS-80 user. 48 to 56 pages of solid information monthly!

Subscriptions for 12 Months \$24 in US, \$35.50 Canada & Mexico, \$36 all other countries.....If you want a sample issue send \$2.

In addition to the many software products we have available for the TRS-80 computers and the FILE CABINET public domain library of over 15,000 programs, we also have printer ribbons for your Radio Shack printer at reasonable prices, fan fold labels, A-B switches, DRAM memory upgrade kits and printer cables - all at reasonable prices.

We also have MS-DOS PROGRAMS

****PACK** by David Goben MS-DOS version....\$17.95 + \$4.00 S&H
A Basic Program Packer, Unpacker and Compression Utility, packs your basic program in the minimum amount of space on your disk to save disk space.

****The BIBLE** King James version for MS-DOS Computers.
Double Sided or Single Sided - 5-1/4 or 3-1/2 disks.
New Testament...\$19.95 Old Testament...\$39.95 Both for...\$49.00
Add \$4.00 S&H in the US, \$6.00 Canada, \$8.00 all other countries.

****xT.CAD** by Microdex MS-DOS Version....\$95.00 + \$4.50 S&H
Computer Aided Drafting program, professional drafting quality, precisely scaled technical drawings up to 24"x36" on your plotter. Will do detail sheets on your dot matrix printer. ****xT.CAD Demo Disk and Manual....\$20.00 + \$4.50 S&H** Purchase price of the Demo Disk and Manual is applied to the purchase the xT.CAD program.

Soon to be announced, the CN80
MS-DOS public domain/shareware
library and the CN80 CPM public
domain library for TRS-80 and
MS-DOS computers.

Computer News 80

307-265-6483

P.O. BOX 680/CASPER, WYOMING 82602-0680

The MISOSYS Quarterly is a publication of MISOSYS, Inc., PO Box 239, Sterling, VA 22170-0239, 703-450-4181.

The MISOSYS Quarterly is published four times a year in February, May, August, and November. Published by Roy Soltoff

THE MISOSYS QUARTERLY subscription rate information

Each issue of TMQ has information on MISOSYS products, programs and utilities, patches, significant messages from our CompuServe forum, and articles on programming. Not only that, TMQ will keep you up to date with information, news, and announcements concerning our entire product line and related machine environments. Subscription cost varies by rate zone as follows:

A = \$25; United States via 3rd class bulk mail
B = \$30; Canada, Mexico, via 1st Class
C = \$32; Colombia, Venezuela, Central America via AO Air
D = \$35; South America, Europe, & North Africa via AO Air
E = \$40; Asia, Australia, Africa, Middle East via AO Air

TMQ Toolbox

The MISOSYS Quarterly is published using the following facilities:

The hardware used for development of the "camera ready" copy consists of an AST Premium/386 computer (20 MHz) equipped with 5 Megabytes of RAM, a Seagate ST4096 80-Megabyte hard drive, a Colorado Memory Systems DJ10 tape backup device, a NEC Multisync II color monitor driven by a Video Seven VGA card, an AST TurboScan scanner (Microtek MS300), and a NEC LC-890 PostScript laser printer.

Text is developed, edited, spell-checked, and draft formatted using Microsoft WORD Version 5.0; Submissions on paper and letters are scanned and converted to text using Read-Right optical character recognition software by OCR Systems. Final page composition is developed using PageMaker 3.0 by Aldus. Cover art and clip art comes from CLIPPER, a product of Dynamic Graphics.

Table of Contents

The Blurb	2
Announcing TMQ Volume V	2
Enhanced 800-MISOSYS order line	3
PD Software Librarian	4
Out of print TMQ's	4
DISK NOTES 4.4	4
DoubleDuty Version 2.6.0 released	5
SAID-86 Version 1.0, by Mark Allen Reed, released	6
Letters to the Editor	7
Correction to "Upgrading the 4P"	11
Resource: Companies	15
Applications for the User	16
Multi-Command, by Frank Durda, IV	16
TOUCH.CCC, by Ken Peck	19
DOS Subjects	20
Model III DOS	20
Model III Programs	23
Model 4 DOS	23
Model 4 Programs	26
LS-DOS 6.3.1 notes	27
Patches to 6.3.1	32
Model 4 Programs, Part II	35
MS-DOS Topics	39
The Hardware Corner	42
Five Twelve K: A better way, by Richard King	42
Miscellaneous hardware topics	55

List of Advertisers

Computer News 80	IFC
MISOSYS, Inc.	41,63-64,IRC,RC
Pacific Computer Exchange	IRC
Flaming Sparrow Press	18
Empac, Inc.	18
TRSTimes magazine	6

List of Patches in this Issue

DSORT51/FIX	23
DIR1/FIX correction	31
FIX631A through FIX631E for LS-DOS 6.3.1	32
GMAP01/FIX to GO:MTC MAPPER/CMD	39
XLBOOTK/FIX, XLBOOTS/FIX, XLSYS0S/FIX	48-50
SDFORM61/FIX	56
MSCRES51/FIX, MSCRES61/FIX for MSCSI	57,58
MSCSI52/FIX, MSCSI62/FIX for MSCSI	59

MAJOR ANNOUNCEMENT

Due to the tremendous appeal to continue publishing lodged by my readers, *THE MISOSYS QUARTERLY* will move on to Volume V. I just couldn't stop after hearing and reading so many encouraging remarks. But TMQ will take a slightly different tack so that it is less of a burden on my time, but still significant in content for the reader. Current plans are to reduce it to about 48 pages. I will eliminate the plastic wrap for both environmental concerns and for the cost of labor to implement the wrap. I will eliminate class B for domestic U.S. which will reduce the manual efforts needed to get it out the door. I have been studying the Postal rates. In spite of the reduction in size, I cannot reduce the TMQ rates due to current and projected postal rates and costs of preparation. My current rate base was originally established four years ago and was based on a Quarterly magazine of 64 pages. After numerous postage increases, and constant weight overruns due to TMQ exceeding its size target, 48 pages seems like a good target. But in this way, I can continue to publish what I consider to be a publication highly useful to my readers.

Most of you currently under a subscription need to renew for the next volume. Please continue your encouraging support by getting that renewal in promptly.

Points to Ponder

Floppy drives: Citizen Watch Company has announced what they claim to be the first 20 megabyte floppy disk drive with true read/write downward capability using conventional media. Production quantities at \$200 in OEM volumes are expected this summer. Don't forget that we sell 720K 3.5" drives in a 5.25" 1/2 height form factor housing at competitive prices.

Floppy Diskettes: Speaking of drives, apparently Eastman Kodak has sold its

The Blurb by Roy Soltoff

Verbatim Corp subsidiary to Mitsubishi Kasei for some \$200 million. Verbatim, as you recall, was one of the early floppy diskette manufacturers here in the States; all of my Model I diskettes were Verbatim. Incidentally, Verbatim filed a formal complaint to the U.S. Commerce Department a few years ago claiming Japanese firms dumping of floppies at below cost. This resulted in anti-dumping duties of 28-51 percent against Japanese diskette manufacturers. Now that Verbatim is owned by a Japanese firm, I wonder how that will impact those duties?

Just the fax: The chip wars are heating up as far as fax chips are concerned. Exar now has available a 2-chip set implementing a 9600 baud send/receive combination fax/modem. Fax transmission supports CCITT V.29 standard Group III at 9600 baud. Modem transmission supports V.22 at 2400/1200/300 baud. With the chip set priced at \$36.82 in 100 quantities, look for fax/modem boards to continue their downward trend in pricing. Silicon Systems, Rockwell, Yamaha, Sierra Semi, and now Exar all are manufacturing fax/modem chip sets with differing capabilities.

Starving Bill Gates: After the 2 for 1 split in Microsoft stock, Bill winds up with slightly more than 40 million shares. That gives him a net worth of over \$2 billion. Not baaaaad!

Computers get bigger: Now comes news that the Boston Computer Museum opened its "Walk-Through Computer" exhibit on June 23rd. This two-story, two-room model of a 486 PC with 16 megabytes of memory features a 25-foot keyboard you can walk on, a trackball the size of a bumper car, a wall-sized motherboard, and a 486 chip blown up 50 times its size to show electron flows and interconnections. Of course the whole thing is just a model controlled and driven by a Mac IIFX.

The squeeze is on you: InfoChips Sys-

tems has recently released a new data compression chip. The IC-105 implements a noiseless, non-lossy data-compression algorithm that can effectively triple storage capacity in magnetic disk drives and boost throughput. Considering that most compression techniques apply only to sequential data transmission, InfoChip's device claims to be able to support random access!

The IC-105 coprocessor chip uses a fast-converging string-matching adaptive algorithm to build a dictionary of the data on the fly as the data is compressed and then stored in 32K of static RAM. Compression takes place at up to 2 Mbytes/s, and decompression at 5 Mbytes/s; this avoids any degradation of throughput even on high speed LANs.

I've already seen ads for a short board using the IC-105 chip; the board is designed to plug into a standard PC bus. This is just what I need to expand the storage capability of my 80 Megabyte drive which always appears to be full. More on this hot board later.

There's space in Tucson: Finally on the news front, if you have a spare few hundreds of million dollars, IBM has about 2.1 million square feet of plant in Tucson up for sale. The complex, covering 1,350 acres with 11 buildings, was just built about a dozen years ago.

Credit Card Orders: Just a reminder to anyone submitting an order using a credit card, I need the expiration date! Also, don't forget to include all of the digits: MasterCard account numbers are 16-digits in four groups of four; the last four digits are usually in the position of the hologram and sometimes hard to see, but they're there. VISA account numbers are either four groups of four or one group of four followed by three groups of three. Errors in account numbers usually delay your order. Also, please include a daytime telephone number (that's daytime

referenced to Eastern Time). Phone numbers are absolutely essential for foreign hard drive orders.

Tandy Software Products: Tandy is getting closer to a distribution procedure for the "discontinued" software for which they own rights. Their Consumer Mail department has already taken over the LS-DOS 6.3.1 product. According to my source, Consumer Mail has already received a considerable quantity of other merchandise from TEW - the Tandy Electronic Warehouse. Expectations are now for some major activity in the mid-summer time frame. So hang in there a little longer if you are trying to find some older product not currently available. Most likely I'll be able to address specific products and prices in our next issue of TMQ.

TMQ IV.iii lateness: Sorry about the lateness of TMQ 4.3. The problem was that I was about a month late preparing it due to the workload of 6.3.1. LS-DOS 6.3.1 had to be finished, tested, and released before I started on TMQ. I just cannot do two big projects simultaneously. I'm a Touring machine! Additionally, unknown to me, my printing company was physically relocating. I normally get a week turnaround from them from the time they pick up the masters until they drop off boxes of completed magazines. On the IV.iv issue, I got about a 3-4 week turnaround. They've completed their move into larger quarters with more capabilities. So I shouldn't run into that snag again. TMQ IV.iii was mailed on April 3rd, 1990.

As I write this sentence, it's Friday the 15th of June. This issue will be finished up this weekend and off to the printers on Monday. If all goes as it should, it should get into the mail about one week shy of the three-month interval between issues. That's not bad! Look for TMQ V.i in September.

800 ORDER LINE

Well, AT&T's deal on 800 service was too hard to pass up. The latest direct service is similar to *Readyline* in capability - able to handle all incoming 800 calls into one direct line. Previously, to handle Vir-

ginia, I would have needed a second access line; to handle Canada, a third. With this service, everything comes into the one access line; I pay only for the additional usage. And there was no setup or signup charges. I've wanted to handle Virginia for some time; now it's done.

For those uneducated in the nuances of telecommunications, let me shed a little light on the subject. Up until about 1970, all telephone communications in the United States was handled by regulated telephone companies. There was the Bell System handling perhaps up to 75-80% of the *traffic*, GTE and United Telecom handling maybe 15%, and the remainder handled by a few thousand small "independent" phone companies. Franchised jurisdictions were carved out by law once the Federal Government came to decide that multiple servers within one territory were detrimental to the provision of service. It's the same way with electric utilities.

Then came the Carterphone decision which is historic to the field of data telecommunications. Carterphone was a company which manufactured an acoustically-coupled modem. Remember them? The Bell System, in infinite opposition to telephone attachment devices, lost the court case in which they litigated Carterphone's right to illegally connect to the telephone network. Carterphone was the first. If my memory serves me correctly, MCI became the second with court approval of a microwave communications private line link between Chicago and St Louis. Although I believe that AT&T was correct in their analysis that the link was "skimming the cream", AT&T nevertheless lost that argument as well. Those decisions, were the basis for the tremendous growth in the cross country private line microwave networks built by the entities known as Special Common Carriers (SCCs).

Throughout this time, though, there were still such things as *interstate* service (calls originating and terminating in different States) and *intrastate* service (both ends of a call within one State). Intrastate service was considered the domain of the local telephone company. Interstate service was the domain of Long Lines, a wholly owned subsidiary of AT&T. Even-

tually, the SSCs came to provide interstate long distance service through extensive litigation and FCC rule changes. Local service was, and is, still the domain of the telephone companies (Telcos).

Now what does all of this have to do with my 800 service? Well 800 service was originated by AT&T years ago. But AT&T could legally provide only interstate 800 service; if you wanted to also support in-State 800 calls, that service had to be provided by your local Telco. That's why a company wanting 800 service had to have two different 800 phone numbers. One number is for all out of state calls while another number is for in-State calls. A dedicated local access line is needed to connect each 800 service. Eventually, AT&T added 800 service to Canada at the cost of another access line and monthly service fee. To have acquired all three services, a user would have had to have three access lines and pay three service fees. I know it seems kind of stupid with all the computerization used for routing telephone calls. But that was the law.

A few years back, AT&T initiated what they called *Readyline* service. That allowed then to handle interstate 800 calls to a non-dedicated access line by routing the call to your regular number. But I then would have had to give up my "MISOSYS" number and would also be unable to determine whether an incoming call was through the "caller-pays" or "answerer-pays" telephone networks. That was not too useful.

I checked into adding Canada about a year ago and it would have cost me about another \$150/month just for the monthly service charge - no calls. I don't have room for another access line.

So when I heard about the new *Masterline* service, I jumped! This new service will make it a little easier for more folks to call in an order - as long as the 800-line doesn't get abused, it will benefit my customers. Effective May 1st, our ORDER LINE - 800-MISOSYS - began accepting calls incoming from all 50 states (including in-state Virginia), DC, and from CANADA. Please note that this expansion of service can only be continued if folks ensure that calls placed to our ORDER

LINE are for orders, questions on an order, or for a brief question about a product which may lead to an order. This service cannot be continued if it is abused with questions involving a need for technical support. I thank you for your consideration; we can work together on this.

It's summer vacation time at MISOSYS

If I don't shut down for at least a week during the summer, the Soltoff's will start to fade. Brenda suggested the first full week in August. So MISOSYS will be closed from August 6th until August 10th (plus surrounding weekends, of course). We will resume normal business operations on Monday August 13th.

TMQ Schedule

Our **target** for mailing the *THE MISOSYS QUARTERLY* is the last week of the respective month as follows: Spring issue in February, Summer issue in May, Fall issue in August, and **Winter issue in November**. Due to the lateness of the last issue, this issue is not on target, but it was mailed less than three months since the last mailing!

Note that your mailing label usually has the expiration date of your subscription. For instance, those with "90/08" **complete their subscription with this issue**. If you want to save me the cost of mailing a renewal notice, send in your renewal fee quickly. I usually wait about a month after TMQ is mailed before sending out renewal notices.

TMQ advertising

If you are interested in reaching a dedicated TRS-80 audience, consider *THE MISOSYS QUARTERLY*. If you have a TRS-80 Model III or 4 related product to sell, you can reach these buyers by placing your advertisement in our publication. TMQ is read world-wide. Our subscribers are predominantly in the United States; however, we do have a significant number in Canada, Europe, and Australia.

.....	
• Current space rates are as follows: •	
•	
• Full page	\$125
• Half page	\$75
• Quarter page	\$50
• Ninth page	\$20
•	
.....	

I compose the ninth-page ad layout so you have no artwork charge. Just submit your text. We accept only black & white ads; however, ads for our inside covers are printed in the same color as the cover (TMQ alternates between PMS colors: green 354, purple 266, blue 293, and red 199). If you would like to place your ad in *THE MISOSYS QUARTERLY*, call me.

PD Software Librarian

Vic McClung has volunteered to be the librarian for the collection of TRS-80 public domain diskettes. Henceforth all requests and contributions be directed directly to him at:

Vic McClung
914 Crescent
Sikeston, MO 63801
USA

Note that if you upload a "public domain" file to our CompuServe forum [PCS-49], and want it to receive general distribution, please also mail a copy on disk to Vic. There is no legal provision for downloading files from Compuserve and redistributing them. Some of our readers who do not have access to our forum have an interest in those submissions. So if you want to help out the most numbers of fellow users, don't limit your submissions to just one source.

MISOSYS Forum

In case you have been off the planet for some time now and just returned, don't forget that MISOSYS sponsors a forum on CompuServe. You can reach many "experts" on TRS-80 and MS-DOS subjects by dialing in. The forum is reached via GO PCS49, or GO LDOS. The forum name is the LDOS/TRSDOS 6 forum. MISOSYS took it over from Logical Systems over four years ago. If you have any questions concerning access, get on

and leave a message to SYSOP. Joe Kyle-DiPietropaolo will get to you. Please don't call me here at MISOSYS because I cannot answer any questions as to its operation.

The forum contains a great deal of programs which you can download, as well as enter into the lively discussions which thread through the message system. If you do programming on a PC, the forum also contains the listings from *Programmer's Journal*. If you want to direct a message to me, my user ID is 70140,310. Post a message in private if you don't want it "broadcast"; some folks even send me orders via a PRIVATE message.

DISK NOTES 4.4

Each issue of *THE MISOSYS QUARTERLY* usually contains program listings, patch listings, and other references to files we have placed onto a disk. DISK NOTES 4.4 corresponds to this issue of TMQ. If you want to obtain all of the patches and all of the listings, you may conveniently purchase a copy DISK NOTES is priced at \$10 Plus S&H. The S&H charges are \$2 for US, Canada, and Mexico, \$3 elsewhere.

Out of print TMQ's

For out of print issues, we are providing back issues of *THE MISOSYS QUARTERLY* via copier reprint. The price is \$12.50 plus \$2.75 S&H in the U.S. and CANADA. For foreign zone D, the S&H rate is \$5.50; zone E is \$6.50. The price for regular back issues still in print is \$10 + S&H. We are currently out of print on all issues of Volume I and Volume II. **Special deal now in effect for all four issues of Volume III: just \$24 + S&H \$5 (US), \$6 (CAN), \$14) ZoneD, \$20 (ZoneE).** Here's a synopsis of past issues:

Volume I See the index in issue III.i.

Volume II See the index in issue III.iii.

III.i Reading NEWDOS/80 disks; An LB archival utility; Popup Application Window; XMODEM in C; Getting into computer math, part

I; TMQ Volume I index.

III.ii Getting into computer math Part 2; Writing interactive RATFOR/FORTRAN programs; PRO-EnhComp: a review; Desktop publishing and the Model 4; A better TERM/APP; adding floppy drives; and a new XLR8er interface.

III.iii The CRC program; PG: a page display program; Locating high memory routines; FIXMA3; Jumbo tape backup for PC clones; New style for TMQ using Pagemaker; and an Index to Volume II.

III.iv Checking for a file from Model 4 BASIC; Surviving the Hard Disk crash; An "interview" with Niklaus Wirth; Keep your printer clean and oiled; On-line HELP with PRO-WAM; MISOSYS announces availability of Hard Drives; Logic in the C language.

IV.i Cataloging files with a word processor; Page display PRO-WAM application; File undating with FUNDATE; Array load routine for BASIC; XLR8er and the GT-180 graphics board.

IV.ii Printing from BASIC without cutting words; LOAD100 for Model 100; Generating date/time stamp; Favorite recipes; Some BASIC routines.

IV.iii Fast in-memory sort using XLR8er RAM; XLR8er RAM as graphics video RAM; Upgrade your 4P with external floppy drives; Doubling of files solved; SuperScripts document file format; FELSWOOP PRO-WAM export utility.

XLR8er support memory shrinks again!

If you are using the XLR8er adapter and the Houdé patches, you may be interested in a utility available from M. A. D. Software. It dynamically applies the patches to the system each time the system boots and locates the first available location in low memory to load them. The modifications XLR8SET applies are functionally identical to the Houdé patches, but only use 35 bytes of low memory, instead of the 125 bytes the Houdé patches require.

How does M.A.D. Software accomplish this? Well, it turns out there are two things they do which are different from other XLR8er @BANK interfaces. First, XLR8SET does away with memory resident code to check on the amount of RAM installed since the check is done by the AUTO'd command installation; it won't permit installation of its memory-resident code if it doesn't find the needed extra memory. Second, M.A.D. has apparently swapped around the two 32K banks of RAM previously designated as banks 1 and 2 in order to simplify the Z80 code necessary to create the memory management port bank images used to image the two banks. The only caution I can suggest is that programs which use the DOS @BANK to determine memory availability but then turn around and use their own hardware twiddling may cause problems if the programs are to use but one of the two banks. That's a small price to pay for another great reduction in low-memory use.

To be transparent, XLR8SET treats the remainder of the command line as a subsequent command. This allows XLR8SET to be specified in the AUTO command along with the command that was already there. For example, if your AUTO line was DO = STARTUP/JCL, you can change it to XLR8SET DO = STARTUP/JCL and both commands will be executed. XLR8SET is priced at \$10 plus shipping and tax (Texas only) direct from M.A.D. Software. In the U. S., the shipping is \$2. See the Resource:Companies listing for the address.

Product News

DRAM and PALs:

I keep a supply of 256K-150ns DRAMs on hand for our XLR8er board, and usually 64K-150ns DRAMs for motherboard replacement. These chips are available for separate purchase. Note that I generally stock DRAM "Pulls"; chips used previously in sockets but pulled, refurbished, and tested. All DRAMs are 100% tested again by us before shipping. I also have a BP Microsystems Logic Programmer to program the Programmable Logic Device (PLD) for our SCSI host adaptor.

This piece of hardware can also handle a wide assortment of PALs and PLDs, such as the memory expansion PAL (U72) needed to upgrade a 64K 26-1069 Model 4 to 128K. Prices as follows:

64K-15 DRAM @ \$2/chip
256K-15 DRAM @ \$4/chip
U72 PAL16L8-25CN @ \$8/chip

Ribbon Cable Assemblies

MISOSYS has also acquired a Cirris Systems cable tester. This interesting piece of equipment performs a 100% test for shorts and opens on cables. It handles many different kinds of connectors based on switchable test assemblies. I have acquired an assortment of test assemblies to support the kinds of connectors typically associated with the TRS-80 micro-computer. Thus, I'm custom fabricating low-volumes of cables according to your specifications, as well as providing standard replacement cables for your needs. These are all using unshielded ribbon cable. I can provide cables using DB-25 M/F, 34-pin edgcard M/F, 36-pin printer, 50-pin edgcard F, 50-pin SCSI M/F, 34-pin Header M/F, as well as DB9 Male. Need a replacement RS232 or printer cable? Probably about \$10-\$20 depending on the connectors used.

Kel-AM 34-pin male edgcard @ \$8
Kel-Am 34-pin female edgcard @ \$5

DoubleDuty Version 2.6.0

DoubleDuty has been revised to work with all expanded memory known to the DOS. I have also added a BANK parameter so that you can have DoubleDuty use any two adjacent pair of memory banks for its "second 64K Model 4". For instance, if you have an XLR8er board installed with 256K additional memory (banks 3-11), you could install DoubleDuty using banks 10 and 11. In this way, you would have a 128K Model 4 partition and a 64K Model 4 partition still with six more memory banks available. Programs which use extra memory but are not "expanded memory literate", such as Multiplan or enhanced Visicalc, could be installed in one DoubleDuty partition and use 128K, still having another program

installed in the other DoubleDuty partition. I have even had PRO-WAM and DoubleDuty both installed above bank 2 in expanded memory. What a machine!

If you already own DoubleDuty and have expanded memory known to the DOS (see the MEMORY command of LS-DOS 6.3.1), then you really ought to have this version of DoubleDuty. Just send your original DoubleDuty master disk to MISOSYS for a disk refresh; the fee is \$10 plus S&H (\$2 in U.S. & Canada, \$3 elsewhere). It doesn't matter if your DoubleDuty was purchased from Radio Shack or direct from MISOSYS; I'll take it either way and convert it into a 2.6.0 "expanded memory literate" version.

If you don't already own DoubleDuty, maybe now is to time to think about

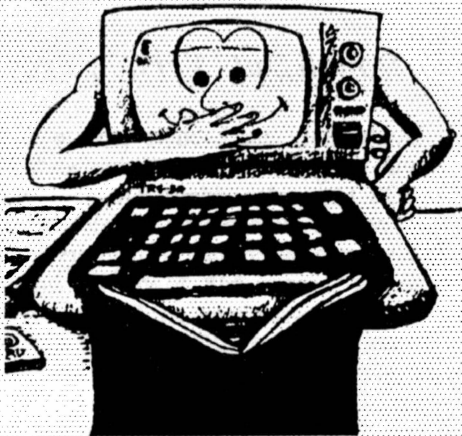
acquiring it. In case you don't know what it does, it turns a 128K (or greater) machine into two partitions with a third able to invoke DOS library commands. This lets you run two programs concurrently switching between either at the flick of a function key. It doesn't support multitasking, so only the foreground application receives CPU time. But DoubleDuty does let you run two programs such as Multiplan and Superscript without having to terminate one to run the other. The regular price of DoubleDuty is \$49.95 + \$2 S&H (U.S.).

SAID-86 Version 1.0

Finally, after almost two years of work by Mark Allen Reed, MISOSYS releases his SAID-86, a text editor for MS-DOS

machines (version 3.x or greater). SAID-86 can be easily used to edit batch files, CONFIG.SYS files, program source files, plain text files, and all other ASCII files you can think of. SAID-86 sports quick operation, nine editing buffers, nine levels of UNDO, mouse support, color support, macro support, DOS SHELL support, and Wordstar-command operation. If you use MS-DOS and don't have a text editor, or are struggling to edit text files while trying to use a word processor, SAID-86 is for you! At just \$29.95, you can't afford to go editing without it. Check out our ad for SAID-86 in this issue.

TRSTimes magazine



TRSTimes is the bi-monthly magazine devoted exclusively to the TRS-80 Models I, III & 4/4P/4D.

We are in our third year of publication and each issue typically features:
 'Type-in' programs in Basic and Assembly language,
 Hands-on tutorials, Hints & Tips, Reviews,
 CP/M coverage, Questions & Answers, Letters,
 Nationwide ads, Humor and more.

1990 calendar year subscription rates:
 U.S. & Canada: \$18.00
 Other Countries: \$23.00 surface mail or \$29.00 air mail
 (all payments in U.S. currency, please)

TRSTimes magazine
20311 Sherman Way. Suite 221
Canoga Park, CA. 91306

Fm Ken Strickler, Stanwood WA: Roy, [see *New Harddisk System for Radio Shack Model III/4*, by Ken Strickler printed in TMQ IV.ii pp 59-60] I really did have the problem with not being able to find the clock, and having the disk errors, although I sure can't repeat the problem now!

As for running the Low-Level Format, I did this for a couple of reasons. First, since I think that you are a great programmer, I like to run your software, just to see it work, the messages that are given, screen presentation and all of the stuff that many users don't care about. Doing some programming myself, I appreciate the huge amount of effort that is required to write and debug quality software. The other reason, is that since my 40 MEGGER left your TLC hands and got to my TLC hands, the shipping box was converted from NEW to USED. I wonder how far the box had to DROP to SMASH one of the CORNERS into a ROUND shape with about a 1 inch radius? The fact that the unit RATTLED when rotated slowly in my hands removing the Bubble Packing, prompted me to look inside and find that the Power Supply card with connecting Clock/Joystick interface card had been shaken loose at one end! I wondered if the heads might possibly be jarred slightly, so I felt that "Laying" down a new low-level format was probably a pretty good idea! (I needed that "CUP-O-COFFEE" anyway!

As for the problem with not being able to create a 720K disk under LDOS 5.3, that problem disappeared also! I am having a bit of a problem with the real time clock in MOD III mode, in that after it is initially set, it doesn't update. The problem doesn't occur until AFTER I have sysgened the system and re-booted! I will still investigate it some more, to see if I can pin down the exact location, before I write any more WILD comments.

Since I wrote last, I have had the opportunity to FIRE up the DIALER portion of PRO-WAM. The application that I am using is to just place phone calls to REAL PEOPLE, no machines. Even though I hit the <H> key (for hangup) immediately after I get the Dial Complete message, if the distant party answers on the first ring, and the modem detects anything that triggers it, it will send the hand-shake tone.

Letters to the Editor

Most disturbing to say the least. Have you a PATCH which would force the sending of the ACK (hex 06) code. I am using the RS internal 300 BAUD modem, with the patches provided by MISOSYS applied to the DIALER program.

Another interesting bit that I have encountered is that when I switch to the MOD III mode, (I have an XLR8er) and run the M3BOOT program, I have to use an old version of MODELA/III. The MODELA/III on LSDOS6.3 doesn't work. I does work if I am BOOTING up one of my regular 4P's into MOD III mode however! I have composed a FLIPPY disk which allows me to boot up ANY Model III or IV, with or without an XLR8er, but I have to have 2 versions of MODELA/III to do it. I could sure use the extra space if I could eliminate one version.

I have purchased an Anitek 3/4 Meg kit (that is all that will fit in the Mod 4P, and even then I have to keep the "PAN" from closing all the way to protect the PC graphics mod from Anitek) and seem to recall that you had some patches to make ERAMDISK work on the Anitek? Is that true, or has my mind slipped again? (I'm getting so old now that I have to have my "Old Man Glasses" to read for very long! Arms are getting too short!) I have searched through my TMQ but haven't been able to locate just what I'm looking for, which is to make use of ERAMDISK to look at the 768K Anitek memory. For those who have one, I find that I have to RESET my 4P several times before it will load properly. This particular 4P only has the Anitek modifications in it, including the SPEED-UP, PC Character Set, and 768K of memory plus the 1200 baud modem (internal) from MISOSYS.

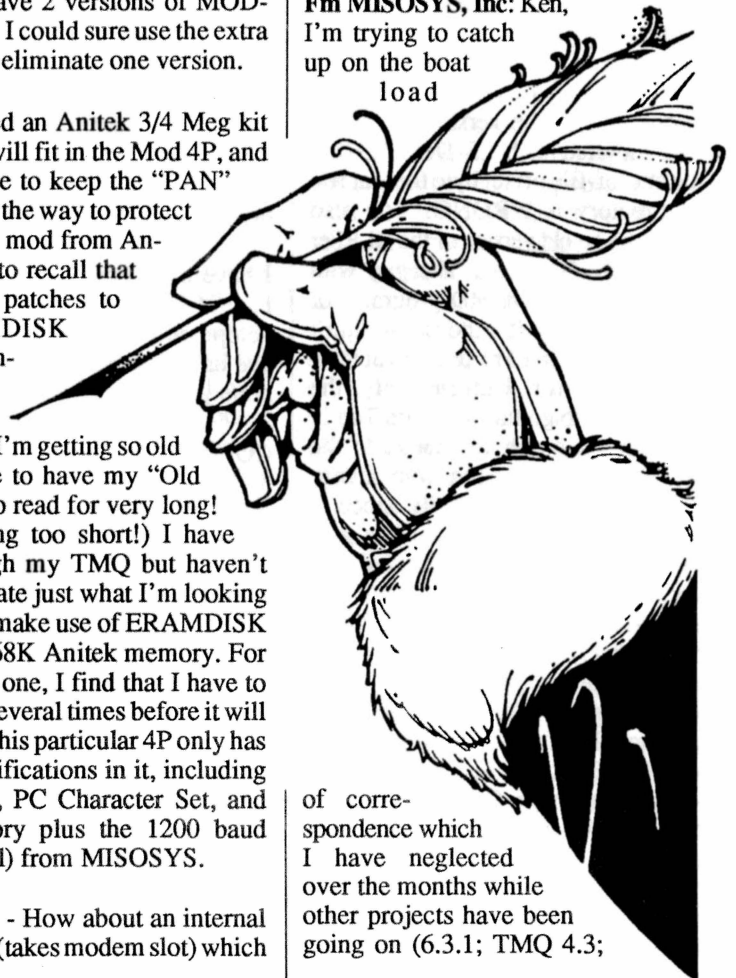
Another subject - How about an internal board for the 4P (takes modem slot) which

could have the real time clock? I don't know how this would interface through the RS-232, but just an idea! (Nothing is TOO tough for the person who doesn't have to DO IT!)

One last question, I asked about the cost of adding another 40 MEG bubble to mine, and hadn't heard. Does that have to be done at the FACTORY, or can the FIELD do it?

Hope that this letter finds you and the family in fine spirits for the holidays, and keep up the great work.

Fm MISOSYS, Inc: Ken, I'm trying to catch up on the boat load



of correspondence which I have neglected over the months while other projects have been going on (6.3.1; TMQ 4.3;

etc...). Thought I would surprise you with a letter. Incidentally, since my community is beginning to push recycling, I thought I would go ahead and put together a package of all the disks you have sent me. In that way, you can recycle them back with more letters/articles/whatnots... That's why this is a bundle.

The only other person with a clock problem in the Model III mode still had the remains of a NEWCLOCK80 patch in his system disk which interfered with the MSCSI clock patch. I also have no intentions of trying to design any other standalone clock boards; if you need something else, look into a Smartwatch.

Turning to the Tandy 4P 300-baud modem, I know of no patch which would cause the kind of modem disconnect which you would prefer. I believe it is a design flaw in the modem itself since it doesn't support the usual "Hayes-type" feature of any character terminating the connection unless modem tone has been established. I do still sell the 1200-baud internal 4P modem...

The problem with the version of MODEL/III supplied with LS-DOS 6.3.0 is that the end-of-file offset byte for that file in the directory is WRONG! You also probably have an old version of the XLR8er Model III interface disk. The guy who wrote that M3BOOT/CMD program for HiTech really blew it - the thing didn't even trap disk errors! I re-did it to at least abort with an error. Unfortunately, the logistics I have to go through with Tandy to make even a one-byte change to the Model 4 disk prevented me from changing the directory - especially since it impacted on so few folks. Those who were affected could either ZAP the directory or use an older MODEL/III file. Actually, I had even recommended that as the 6.3.0's MODEL/III file had a correction for the Model III mode clock timer, but LDOS 5.3 also corrected that; thus, using LDOS 5.3 on a 4P with the newer MODEL/III file resulted in a half-speed time clock. On the other hand, I did correct the offset byte in the 6.3.1 release.

Anyone can add a second drive to their MHDT34 package; it is your equipment,

isn't it? I believe page 6 (3rd through 5th paragraphs) of the MHDT34 manual explains what has to be done to add a second drive. I do recommend that you may want to keep that spot open for a tape drive backup device. That may materialize later this year.

Fm Ken Strickler, Stanwood, WA: Roy, This letter is "away" behind. I meant to get to this, like in JANUARY! So much is changing in the field and at such a rapid pace, I find that I am running full speed ahead just to stay even! I'm sure that you recognize the feeling!

I see that in the current CN80 that you have released another version of LS-DOS with some new, neat features and I was wondering if the updates were going to be available as a SITE LICENSE as well as REGULAR version? I have both, and would like to keep the SITE version screen. In either case I will be ordering one or more when I get the next *MISOSYS QUARTERLY*!

Along the same line (spending money), I was wondering how much it would cost together to add another 40 Meg bubble to my Hard Disk System?

I have been doing some reminiscing of late, in conjunction with the current developments on the micro-computer field, and have put together this little history. I hope you will enjoy it!

COMPUTERS — THEN AND NOW

30 Years ago, when I began working on computers, a computer room was fairly large, say 20 by 50 feet, and housed 1 (count'em one) computer! Main Memory was a whopping 100,000 octal (65535 decimal) addresses and the main CORE ARRAY was housed in a cabinet 4-5 feet square, standing 6 feet high. The support hardware to access this 'BABY' was mounted in 'wings' extending from 2 sides, about 8 feet long! Then, there was the ADDER/ACCUMULATOR, another MONSTER 12-14 feet long and 6 feet high. Two more 12-14 foot frames held the 'DRUM' storage, one MAIN DRUMS, one AUX DRUMS - each having 6 drums

containing 12 - 4K (10,000 octal) words. Total storage 576K - 33 bit words. (4 bytes+ parity). And so on! As for power consumption, we (USAF) generated our own using 4 - 970 Brake Horsepower, 6 cylinder diesels. Yes sir, folks, a mere 3,000,000 watts of power to run a 'pair' of computers and the peripheral equipment! How would you like to have one of these 'BABIES' all of your own? Had a 'CREW' of 18 or so to maintain them too. About 50,000 vacuum tubes, as I recall! No HEAT required, but plenty of air conditioning! The blinding speed came in at 166,666 additions per second, I remember it well - STATE OF THE ART. The whole system was housed in a 4 story building! Total cost for the complex, about \$30,000,000!

27 Years ago, thanks to the transistor, computer technology had been significantly reduced in size and power consumption, in addition to increased reliability. Still the cost was approaching \$1,000,000! The SPACE program was in full swing, and one of the great spin-offs was the shrinking of computers for the rockets.

25 Years ago, when I started with IBM, the System 360 was all the rage! The capacity had risen to 512K of 64 bit words (Model 50) supporting tape drives (6 feet tall) and disk drives (a 'CUBE' about 3 feet on an edge). While you couldn't exactly "PLUG'ER IN TO THE WALL", a 50 amp - 208 volt - 3 phase circuit would handle the load! Cost - still \$500,000 to \$1,000,000 - with a substantial increment in computing power.

20 Years ago, MINI computers were making in-roads into the main frame computer domain, by offering machines with quite a lot of computing power, and no longer requiring huge amounts of special power wiring. Cost is down to a mere \$100,000-\$200,000 depending on what was required.

All of the above machines were time shared due to the cost, and would support many users, or run programs, one after the other, in a BATCH mode of processing.

15 Years ago, MICRO computers made their arrival. Memory configurations from

4K to 64K ram were available running various operating systems, I remember running at a 2 Mhz clock rate, and thinking that while it wasn't breaking any speed records, at least I could plug it into any HOUSEHOLD outlet, and I didn't have to call the power company when I 'FIRED' it up! It still wasn't exactly cheap, by the time I had finished with my MODEL I (TANDY) computer system, I had over \$3,000 invested, but it only cost about \$.02 per hour to run! We hadn't heard of RFI or EMI in the HOBBY WORLD, but we were soon to find out! Little SOFTWARE companies started appearing, writing programs for making the HOME computer do something! Graphic presentations left a little to be desired, and real animated stuff - wasn't real animated! Remember PAC-MAN? Any software programmer could fill the available memory in a minute, and be complaining about writing modules which had to load 1 or more additional modules in order to complete a task! (We were just lucky to have our 'own' system!) Still, it was a HACKERS world at home. I was building a VECTOR 1+ - s100 bus machine!

10 Years ago, with the memory limit pressed to the 640K (nobody will ever need more) limit - the IBM personal computer or CLONE had started to make real progress. Businesses started to see the value of what the computer could do. We could now provide pallets of printed paper that nobody cared about or read! The paper companies were glad to feed our machines, at a profit too! Old TELETYPE printers had been replaced by DOT MATRIX printers, and the price had fallen to where only a 2nd MORTGAGE was required! The market, for me at least, had started to leave me in the DUST. I had spent a 'SMALL FORTUNE' getting to where I was for I feared that the interest would not be great enough to sustain continued growth, and as a computer fanatic, I wanted to get one before it was too late. As a matter of fact, I got 2, just in case I had to fix one!

5 Years ago, RFI and EMI problems were well in hand, and even PORTABLE computers were available. PORTABLE meant that there was a HANDLE mounted on the machine. Running through an airport to the next interconnecting plane,

that 29 pound MODEL 4P that I carried didn't feel so PORTABLE! But, all things considering, let's see - 128K, dual 186K floppy drives, monitor, and running at 4 Mhz, internal modem (300 baud), hi-res graphics adapter (640 x 400 pixel). The software operating systems had come quite a long way, and the very-limited original operating systems had long been replaced by fairly complete systems allowing the user a large amount of flexibility! (You could do a lot of 'STUFF')! I think that I had about \$1450 invested - through a mail order discount place.

TODAY - 1990 - the 640K boundary is gone - machines with up to 24 Meg of 32 bit memory - running at up to 33 Mhz, with floppy disk drives of 1.2 or 1.44 Mbyte capacities, Hard disks of up to 4.6 Gigabytes, modems at 19,200 baud, laser printers - yes-sir-re-bob - step right up - we can let you have this for ONLY \$10,000 to \$15,000 (without the \$20,000 4.6 Gig harddisk!). Excuse me! I had to scrimp and save for the \$1450 machine, and now this! NO WAY! In reading about the latest and greatest, since I can't afford one, I note with some interest that the programmers are still complaining about the lack of memory, and the cost, and the speed and whatever else. What with their memory caching, and disk caching software/hardware, trying to get bigger and faster! The latest complaining is the amount of time that it takes to read these HUGE modules into memory, that now smaller, say 32K modules are written, so that the wait on load time can be reduced. Loading a 640K program to run a 5K module isn't too efficient! However, with memory sharing between Terminate and Stay Resident programs (TSR), and application Window programs (WINDOWS 286/386) the 'ole computer sure stays busy. Have you ever wondered what it would be like to have your 'ole MODEL IV running some of this neat stuff? Could it be possible? Wouldn't it be fun! - Now Let's See ...

Now it ain't 33 Mhz, but you could get 8 Mhz (throughput) plus 256K ram board. Makes a nice RAM-DISK - instant access - no waiting - easy to install, gives a total of 320K in the Model IV - cost \$182.00 + S&H - source MISOSYS, Inc. (price from MISOSYS QUARTERLY Volume IV.ii)

If you want more than just a RAM-DISK, how about a Terminate and Stay Resident program like a Window! Handy for a calculator, calendar, dialer - you know a paperless desk! (You would never know it by looking at my desk!) Give PRO-WAM a shot! Only takes 1 bank of 32K memory. You can even run it if you only have a 128K machine. Use a 'HOT-KEY' to call it up, and 'BREAK' returns you to your current program. Just like the BIG GUYS! - cost \$74.95 + S&H - source MISOSYS, Inc. (price from MISOSYS QUARTERLY Volume IV.ii)

How about a little more speed on the telephone lines! Now it ain't 19,200 but it is 4 times faster than the 300 baud modem that RS sold. A 300 - 1200 baud modem - internal mount for \$89.95 + S&H - source MISOSYS, Inc. (price from MISOSYS QUARTERLY Volume IV.ii) [note: the TT512P modem is currently priced at \$79.95 + S&H]

How about all this RAM-CACHE and DISK CACHE that we are all hearing about, and all of the wonderful things that is supposed to do - I know it is hard to believe, but MISOSYS, Inc. had a program SEVERAL YEARS AGO, which is out of production, I suppose BECAUSE NOT ENOUGH OF US BOUGHT IT had some very similar features! Since I have been trying to get EVERYTHING possible, I was fortunate enough to obtain as copy of OVERDRIVE in the MARK IV collection! Requiring 1 bank of memory (32 K) system files SYS1 thru SYS5, SYS9 thru SYS12 are loaded for INSTANT ACCESS by the operating system, and BUFFER (Read that DISK CACHE) space is allocated for 2 drives. (Allows complete tracks to be read into the RAM, and repeated access to the disk is reduced, significantly reducing some types of programs data access times.)

Ah - yes, but what about MASS STORAGE - you mean like hard disks - How about that 80 MEGGER from MISOSYS. How Much do you need? We are, after all talking about a HOME COMPUTER on a limited budget! Just How much data do you think that you need at one time? It is true that the device is a EXTERNAL connection, but when equipped with the CLOCK/CALENDAR option and JOYS-

TICK interface not only keeps track of the time and date, interfaces the KRAFT Joystick, but allows storage of 1250 - 64K blocks of storage. That's 16,666 - 8 1/2 x 11 pages, each with 60 lines of 80 characters, with each space containing a character - no blank lines! - cost (20 Meg w/out clock/joystick \$495 - additional as capabilities increase) - source MISOSYS, Inc. - (price source MISOSYS QUARTERLY Volume IV.ii)

So - how can I use all of the enhancements just mentioned. Well, you are limited to your desires! I can only tell you what I do! In the GOLDEN OLDIES Series is a program called DOCONFIG which allows changing the way a system looks to the user [note: DOCONFIG is currently available in GO:SYS]. Currently, for writing articles or letters, I reserve banks 1 and 2 for my ALLWRITE word processor, bank 3 is the printer spooler, bank 4 contains PROWAM containing a calculator, card filer, calendar, and dialer programs as the 'HOT KEY' and the remainder of PROWAM and MR ED applications by Universal call. Banks 5 thru 10 are used for a RAM-DISK allowing for 184K formatted storage. It is true that in my 4P, I don't have a modem since I used that slot for my XLR8er card, but I live way out in the boondocks, and downloading by mail is a lot cheaper!

NOW INTO THE FUTURE ...

The industry magazines are all aglow with the new 25 MHZ 80486 machines touting here-to-fore unheard of capabilities. Local Area Networks (LAN), Unix Networks, Micro-Channels and related software abound at prices that make bankers SMILE! Artificial Intelligence (AI) and Neural Networks (You will be hearing more about this in the future) are being developed. Both of these technologies require HUGE amounts of SPEED and MEMORY to say nothing of the MASS \$\$\$\$\$ to buy the stuff! Certainly not cost effective in the HOME MARKET NOW, but HOW ABOUT IN 25 YEARS!

From NOW until NEXT TIME - I'll be trying to move the utilization of my MODEL IV from 11% to 11.1%!

Well Roy, I guess that was a little longer

than I expected, but I feel better having put it into words! I can easily see that I would not be having near as much fun with my computer system if it were not for your efforts and continued support. The 40 Megger hums along beautifully (knock on wood) without any problem. Daylight Savings time was on the right day [note: the real time clock used in the MISOSYS MSCSI host adaptor correctly adjusts for daylight savings time]. The auto park feature allows me to forget (I did anyway) about having to park the heads. By being able to power down the disk drive, I am not running it continuously adding unnecessary wear to the bearings. Thanks again.

Just a passing question, could DDUTY be structured to run with ERAMDISK, PROWAM and the other LOW MEMORY stuff by committing another XLR8er bank to it. (Probably not enough interest - but a neat idea!) Any plans for re-releasing OVERDRIVE again?

Fm MISOSYS, Inc: Double Duty can run with PRO-WAM, and perhaps ERAMDISK - but not in addition to a hard disk driver all in low memory. See the information about the latest 2.6 release of Double Duty elsewhere in this TMQ. I am also considering a review of Overdrive for restructuring to use less main memory; it uses too much for my tastes.

As far as adding a second 40-Meg drive, I would hold off for now for two reasons. I do have under development, albeit slow progress, a TAPE backup/restore program for use with the QIC-40 40-Meg tape drive available for MS-DOS. Another reason is that I will be shifting over to 3.5" Kalok 40-megabyte drives as soon as they start shipping them and I have had the time to evaluate them. The 3.5" drive uses less power and generates less heat. That would be a useful ingredient when adding a second drive bubble.

If you have already purchased a Site License for LS-DOS 6.3.0, then all you need to purchase is one copy of the 6.3.1 replacement disk. I have no expectations of creating different signon screens.

Fm Ken Strickler, Stanwood, WA: Roy, Thank you for returning all my disks that

I submitted over the years! I will try to put them to good use by submitting more information.

I will keep the second slot open for a possible tape drive later this year. What a neat idea! I know that when we discussed that possibility early in the development of the external hard disk box, you indicated that the tape controller used something like a 765 chip ... et al, and that we probably wouldn't be able to do it. For whatever reason that you changed your mind, I am glad! (How about one of the NEW 20MEG FLOPPY 3 1/2 inches?)

I received the TMQ in the mail yesterday, and have of course, read it now a couple of times. I trust that all the listings will be on the DISK NOTES 4.3 disk, as always. I have a machine with the AT (768K) memory, and am anxious to try the patches!

I missed the recapping of the COMPUSERVE information, as I am not a subscriber! I live so far out in the "boonies" that every call is LONG DISTANCE, and the amount of time that I would have to spend getting the information would pay for my TMQ subscription! Besides that, in the TMQ you have sifted the information, and I know that I will be getting the HOT STUFF! I do understand that the time involved is too much, and for the subscription base that you are currently supporting, just doesn't put bread and butter on the table. None the less, I will miss that part of the TMQ.

In reading the BLURB, I see that even the continuation of the TMQ seems to be in question! I want to say that I appreciate the effort that you have put into the TRS 1,3,4 market over the past years, and fully understand your wish to move on to something else, especially with a growing family! (Have you seen the price of shoes, clothes, not to speak about FOOD!) Does that mean that you will be moving into the PC machines, or will you be supporting the TRS 1,3,4 and advertising in another magazine (CN80)?

I understand that the market is rapidly moving to the 3 1/2 inch disks, and it will be a whole lot easier as the manufacturers "DRY UP" production of the 5 1/4 inch drives! I have around 1000 - 5 1/4 disks,

many configured to 720K, and I buy them for \$.25 each (200 at a time). The best price that I can receive on the 3 1/2 inch disks is \$.49 each, so it would cost \$500.00 in disks plus 11 computers X 2 drives each X \$88.00, which equals \$2436.00! A little steep for me! (I would take donations! I might buy 1 (one) 3 1/2 inch to try it. I sure like the way the disk is protected. If I didn't have to support so many systems, and had it to do over again, I would go with the 3 1/2 inch drive!)

I enjoyed the article on upgrading the Model 4P with the new drives, and it is very similar to the procedure that was published in 80 MICRO some years back. A month after the 80 MICRO was published, some knowledgeable person (I assume) wrote to 80 MICRO and said that the 'FAILURE' mode of the 'MOV' is too 'SHORT'! He recommended that the 'MOV' be inserted AFTER THE FUSE instead of BEFORE THE FUSE, as indicated in the TMQ. I have structured machines both ways, and the 'MOV' legs are long enough to reach the other land patterns! AS INSTALLED, A SHORTED 'MOV' WOULD NOT BE FUSE PROTECTED, AND WOULD HAVE TO POP THE HOUSE BREAKER (15 OR 20AMPS). THERE MIGHT BE SOME ACCOMPANYING 'SMOKE'! I haven't had any failures of the 'MOV's yet, and I have only WIRED ONE COMPUTER PAST THE FUSE. I'm sure that sooner or later I will have wished that I had wired them PAST THE FUSE!

Well, since I have just written to you, (our letters passed in the mail) I will get my order together and get this out for TMQ IV.iv (MAY DEADLINE for 'MOV' warning!) I will be recycling one of those disks that I sent you before!

Fm MISOSYS, Inc: Thanks for the warning, Ken. You're not the only one who advised me of that.

Although I previously stated that the continuation of TMQ was "questionable", that in no way implied that I was going to stop selling TRS-80 software. First of all, remember that TMQ was a support vehicle of "all MISOSYS software products

and related machine environments". I was considering stopping TMQ because it was no longer a profitable use of my time. Let it be said here and now that as long as MISOSYS is still in business selling software, MISOSYS will continue to sell TRS-80 software. True, it may take a different form over the years, but we will still remain in that business. TRS-80 software may be sold, "as is". But that's the future, and this is now.

Excellent TMQ Article

Fm Peter Van Caesele: Roy, The article by Rich King in the last issue of TMQ (All the way to 512 K) was excellent. It solved all my memory refresh problems. Just for your information, I have a Pre revision A board and the modification works fine. With the revised patches that Rich provided, my system has never run better. I must congratulate Rich and extend my deepest thanks to him for providing me with the helpful personal support.

TMQ IV.iii

Fm Gary Phillips: Roy, the new (slightly slimmer) TMQ looks fine. The CompuServe extracts were fun, but if taking them out eases your workload a bit I won't complain. The detective work on the "doubling" of file names on some gate-array model 4 systems was great reading, sort of a techie whodunit. I do want to point out that Scott Toennissen is giving credit to the wrong person for the direct boot patches for the 4P. Adam Rubin was the author of those fixes. All I did was adapt them from TRSDOS 6.2 to LS-DOS 6.3. And since then, Adam has improved on them even further, correcting some oddities encountered on certain gate array machines, in spite of the fact that neither he nor I have gate array machines ourselves. Now who is going to go on and adapt them to 6.3.1? (Adam, are you reading this?) Or do we just hold off until M.A.D. software produces an improved boot ROM for us?

Fm MISOSYS, Inc: Nothing in low memory changed - that includes the boot track (except for the patch to the floppy driver); thus, I would not think that those would be impacted. Anything which patches SYS0/SYS or any other module would be affected. Note also that anything which tried to open any /SYS file using the password of "LSIDOS" would give an error of "Illegal access attempted to protected file". You could easily counter that by ATTRIBing the password of the system file back to LSIDOS from SYS-TEM6 for the duration of the program's implementation. For instance, a simple:

```
ATTRIB BOOT/SYS.SYSTEM6
(O="LSIDOS")
```

reverts that system file's password to the one used all releases of 6.x prior to the 6.3.1 release.

Removing the forum excerpts saves me about a week and helps to slim down the issue. With postal rates going up 20% next year, printing costs escalating, and my time becoming mucho valuable for trying to get revenue generating work done, keeping a slim TMQ is essential. If it is to be continued, it could well remain as slim as 4.3 (or slimmer), probably saddle stitched to save a few bucks but to make it easier to bind (perfect binding is tough on 30 sheets), and probably remove 1st class mailing for US folks to make it easier for me to get out the door. Also, continuing to get "article input" is essential. I just don't have the time to write all of TMQ; have to do things to keep putting food on the table and try to save a little to put three kids through school. But then, most folks are in that boat these days.

Fm Louis Clarke, Sycamore MS: I enjoy your "Quarterly" very much. I have a Model 4P and an upgrade cassette based Model III that's now a dual floppy Model 4 e/w 128K. I enjoyed the article "Upgrade your 4P with external floppy drives" which was a reprint of "On the upgrade". There was a subsequent correction printed in 80 Microcomputing June 1986 page 27. There are a few female connectors that are supposed to be male connectors, and that the last physical drive should be terminated with a resistor pack. But most importantly, there was a change in the

design of MOV placement to the power pack.

Fm MISOSYS, Inc: Thanks for the tip. Here's the correction as published in the 80-Microcomputing you noted.

According to Tsun Tam, the instructions for upgrading the 4P contained a few errors. In photos 8 and 9, the internal cables should show male, not female connectors. On the external cable in Fig. 2, the pin labels for the two female edge connectors (right end of the diagram) were reversed.

In addition, the last physical drive should be terminated with a resistor pack so that the signal reaching the drive is the proper voltage.

Finally, because blown MOVs short out, they should be installed after the fuse. The Figure below, which replaces Photo 6 of the original article, indicates proper placement.

Although many 4P owners have successfully completed the upgrade, a few have had problems getting the disk drives to read double-sided disks. Tsun's upgrade requires that the drives be configured according to the industry standard - a standard that Radio Shack doesn't follow. To ensure proper operation of the drives, follow the directions for modifying the old cables carefully or purchase new internal (and external) cables.

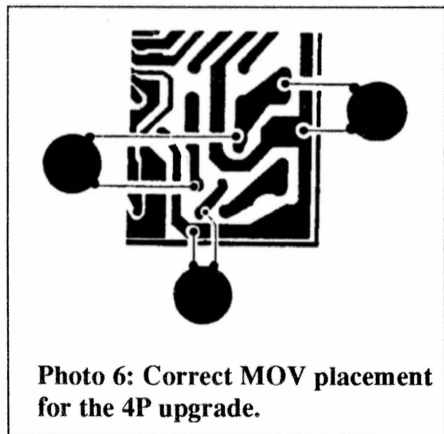


Photo 6: Correct MOV placement for the 4P upgrade.

Read on for more updates on power supplies...

Fm Frank Durda, IV., Fort Worth, TX: Roy, The article on upgrading your 4P (TMQ IV.iii, p. 27) makes me want to bring up a couple of additional points. Radio Shack used two different power supplies in the Model 4 and 4P systems. One was made by Astec and the other was built by Tandy. The MOV modification shown is correct for the Tandy supply. MOVs can be applied to the Astec supply, but the PC board etch arrangement is slightly different. (See attached tracing.)

The second item is a problem with the Tandy supply, which if left uncorrected, could damage a system. On many of the Tandy supplies, a voltage-adjust potentiometer at R15 was sealed in place with a material (orange-colored) that was improper for this task. Over time a solvent leaches out of the sealing material and attacks the carbon conductor in the potentiometer itself. The result is that the value of the potentiometer can change gradually with humidity and temperature changes, or quite suddenly. The visible signs of this problem are blooming or shrinking of the display, sometimes accompanied by the system rebooting itself. In a few cases, the voltage levels have gone high enough to damage the computer. (Apparently this power supply has also been banned from Radio Shack hard disk systems because of these problems.)

In photo 7 on page 30 of TMQ IV.iii, the Tandy power supply can be seen. It is readily identified by the wing-style heat sink on the switching transistor (upper-left corner). (The Astec power supply uses a well-style heat sink and has a 115V / 230V option jumper, usually marked with an orange label.)

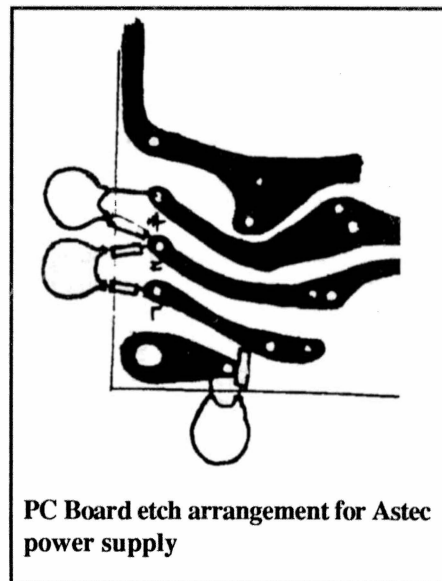
When you bring a machine with these symptoms to Radio Shack, they either replace the Tandy power supply with an Astec or replace the potentiometer with a pair of resistors that provide equivalent resistance. The way you do this is to remove the potentiometer (intact and without changing the setting), and then measure the resistance between each outer contact and the inner contact. The two values are the resistor values you should try to obtain and install in place of the potentiometer. 1/2 or 1/4 watt resistors are suggested. Or you can obtain a new

potentiometer, adjust it so it has the same resistance values as the original, and then install it.

Finally, if you want to check the power supply 5 volt level, a load must be attached for the power supply to work properly. Check at V1, and you want to have between 4.95 and 5.25 VDC. The voltages at the CPU board are usually lower.

Fm MISOSYS, Inc: Thanks for the tip, Frank. I know a lot has been said over the years about those Tandy supplies.

For those who may not know too much about Frank, recollect that he was the one who ran down that obscure hardware problem in the gate array Model 4 which caused random problems with floppy diskette recognition. Frank Durda IV has been with Tandy Corporation since 1982 and has designed hardware or developed software for the Model 16, III, 100, 12, 4, 4P, 4D, 2000, 6000, and several of the MS-DOS-based systems. He also is an instructor of assembly language, C and computer system design at Tarrant County Junior College.



PC Board etch arrangement for Astec power supply

Tandy Consumer Mail

Fm Donald P. Vincent Jr.: If anyone else needs it, here is the address for TANDY's Consumer Mail Center Service that Roy mentioned to me. TANDY CONSUMER MAIL CENTER 401 NE 38th St. Ft. Worth TX 76106

Registration Cards?

Fm Gary Phillips: Roy, 6.3.1 and Mr. Ed received. Both look good, though I didn't get them installed because Comm Ed saw fit to allow TWO power failures in my neighborhood last night. At that point I shut everything down so as to reduce risks to the hardware. I noticed that there were no registration cards to return. Does this mean that you are no longer keeping a database or just that you're automatically registering the products? Thanks for the good software and quick service in any case.

Fm MISOSYS, Inc: I am no longer using registration cards for TRS-80 products except those sold through Tandy - the only outside entity. All other products are sold direct and I already have the information from the invoice. I also no longer keep product specific records; there turned out to be no economic reason for doing so. When was the last time you had to fill out a registration card for a non-hardware consumer product which was < \$100. I didn't even get a reg card the last time I bought a car? I do get them for toasters, woks, etc., but that's only to build up customer lists for selling to other direct mail merchants.

On TMQ Contributions

Fm Frank Slinkman, Glen Allen, VA: Dear Roy, As you know, my knowledge of computers and programming is somewhat narrow, to say the least, so I don't know what I can offer in the way of contributing to the preservation of TMQ.

Specific application code to solve specific problems is really what I "do", so, given a stream of suggestions/requests, I could come up with some code and an explanatory article for every issue. I'm getting to the point now where I'm running out of ideas for new programs, anyway.

Fm MISOSYS, Inc: Frank, As far as contributions to TMQ, I have found over the years that the total production of TMQ is a burden on my time; it leaves too little left over for productive program development and management. Something has to give. To continue to publish TMQ, I have to minimize my involvement. But I am concerned as to its appearance. That means that I want to have control over final production; I just don't have the time to do a lot of writing. I need articles. I am no longer going to include forum excerpts. For one thing, they are too time confusing to groom. The appearance of the excerpts is also less than what I and my readers want. The correspondence I receive concerning TMQ requests more articles; after all, TMQ is a magazine! I believe that words with program listings as illustrations, are what folks want to see. Your articles, for the most part, are exactly what is needed. Your articles contain a lengthy discussion covering the technique with a code stream providing an implementation of that technique. Ainsworth's articles are also the kind people want to read. So if you have anything else along those lines of format, that would be superb.

I believe there are still users wishing to read about programming technique; i.e. a regular column on BASIC, assembler, C, etc., would be useful. There are a few folks who have an interest in hardware who could generate a hardware column. With sufficient input, I could spend just a little time in formatting the magazine. I would also reduce its size a little further than the last issue - maybe to 48 pages. It would be saddle stitched, sent only via bulk mail, and probably not plastic wrapped (Gotta save the environment).

I have no final decision at this point, but don't forget that CN80 is not the only other publication - there's also TRSTimes, TRSLINK, and club newsletters. Support of MISOSYS products will be by MISOSYS.

Fm Danny C. Mullen, Ft Polk, LA: Anyway, on a different subject; life or death for TMQ. Some observations and comments follow:

I really like the professional way it's been put together since I subscribed. I will somewhat miss the COMPUSERVE excerpts - somewhat because at times they did seem to drag on about obscure/uninteresting subjects to me; however I did pick up some REAL niceties, notably the SMARTWATCH installation for MODEL 4, which was immediately installed.

If TMQ can't continue in present format and/or size, PLEASE try smaller or cheaper version; disk version (maybe with limited HI-RES ads for HI-RES board owner & {much work?}) {scanned with your page scanner} {high duplication cost, high distribution cost} {long duplication time?}) (maybe on a common word processor {LeScript} format or TED for ASCII and no graphics). Loss of TMQ may signal death knell for TRS-80 even with CN-80, TRSTIMES, et al.

Many of us TRS-80 users have indicated its cheap costs to operate/maintain and have no desire for those costly 'other-than-8 bit machines'. Makes me wish 16 bits were illegal for home use! In this respect, I hope you can attain economic possession of older TANDY software rights/software so it can be sold at TRS-80 affordable prices and allow you profit. Though what I'll call the 'old TRS-80 stable' will probably possess several of the titles, I'm sure there are TRS-80 neophytes who could certainly benefit from it.

Again on TMQ life or death. What about instead of or in addition to on-line with COMPUSERVE to you and LDOS forum folks, a MISOSERVE BBS for TMQ alternative. Possibly multiline board with subscription costs to keep out the less-

than-desirable types. Ads, info, patches, updates, etc could be posted, and could even be run by..(Vic Mclung or ...)? Just a thought. Get some use out of that AST Premium/386! <grin> and some of those hard drives!<<grin>>.

You need input/columns to assist in continuing, and may never reach the goal of 'enough' input. Some of the potential contributors may feel a little intimidated by the quality/technicality of some subjects covered, however queer that may seem. I'm talking about those who may have one- or two-time tidbits to share, but are fearful of being embarrassed by publication of their tip/program/observations/experiences, etc.

I, myself, am pondering right now about submitting a short series of a couple of the programs I did for PD - that is, doing a section at a time until complete; with comments as you go. Though, admittedly, this could be frustrating for a quarterly (a la TRSTIMES bimonthly stuff). I'll do anything to try to assist TMQ longevity. The only thing that could possibly interfere with my doing such, is the US Army and its sometime erratic training requirements (ie my summer this year is pretty well shot) but I can always find time for dabbling.

This pretty well exhausts some thoughts I had about possible demise of TMQ.
PLEASE DON'T DO IT

Some unrelated subjects now, no reply necessary.

Some of the undocumented flags in THE SOURCE (ie G,H,Q,U,X,Z): are they used for propriety reasons or are they available for possible use by developers?

The infamous CTL-255 routine: I understand SCRIPSIT uses it for keyboard stuff, but is there a key combination available from DOS that could utilize that routine? Looks like good patch space for non-SCRIPSIT users:

Several years ago in the old 'Northern Bytes' a writer was alluding to changing the INITCRT table to allow 80x25 display but there was a question of whether the CRT table needed to be changed to an

older or newer version. There was never a follow-up to the article to answer the question, that I know of. I thought I read that even though this routine is in THE SOURCE, it actually had no effect if you change the values in the table. Any thoughts/input to that? This is in the bootup code.

This letter needs to get in the mail, so will end it so I can get on with the disk you sent. Again, I really appreciate your support to us users and the responsiveness you've given to my questions/problems in the past - and present as evidenced with aforementioned update disk.

Roy, I had some follow-up thoughts to explore about suggestion of a BBS or MISOSERVE, MISOSOURCE, (MISOSYSOP ?), MISONLINE, if you will. The 'BBS' wouldn't have to be in the typical sense a BBS like normal. I know BBS operations usually take a lot of SYSOP time/effort/\$, but some advantages I envision follow:

Less time used to do layups (layouts?) for hard copy to printers. Reduced cost of printing/mailling. You can be working on coding/writing new or revising old code, developing patches, replies to letters, etc., and when certain thoughts/ideas/solutions came to mind, you could easily post this to the database of the BBS. Any news/ideas received over the phone or wherever could LIKEWISE be immediately posted. Us users would have relatively instant access to the most current information possible. Searching for data for replies, routines needed, or referencing this info would make your job easier. Board could be split into separate areas/interests (Model 3, 4, MS-DOS, tips, patches, new products, merchandise, etc) that could be accessed by SYSOP or user. Instant price changes could be posted; products ordered and so forth. I'd rather pay you \$12,00 per hour than COMPUSERVE since I don't foresee my use of their service, other than to access PCS-49, very often. Plus, my 'local' access phone costs more because it's in-state rather than out-of-state which really adds up quickly. I think you could still get advertisers to buy some space and they would get access to instantly update offers, prices, etc. too. Users could still upload their routines and they could be checked-out

easier by you (don't know if you do that now or not). Downloads could be available in measured amounts. Maybe some surcharge after so many? Contributors could do their stuff at least as easily as now. It would need to be able to serve several users at a time since the old 80 MICRO one was impossible to log onto most of the time. Security would be tough, I know, but some system must be available that could ensure pirates/viruses, etc were filtered out. Might be a good forum to show off your hard drive system's capabilities. Might be good forum for possible demos of your products to potential buyers. Interfaced with your 800 order line, your bookkeeping may be enhanced; and tracking product sales (slow movers vs. HOT ones) may assist you. (If orders could be placed through BBS). All of the above, I understand, hinges on sales strength in a dwindling TRS-80 market, but a bonus is possible product sales boost for your MS-DOS line once they got the word. What I mean is, I can't recall seeing any of your MS-DOS line advertised in any other current publications. I would think this to be a cheaper vehicle for advertisements. Don't know if any of this is economically feasible, but just some thoughts I wanted to cover.

Fm MISOSYS, Inc: When I was laying out the design of LS-DOS 6.0, I wanted to bundle the addressing of specific data elements from a base register. Under LDOS, many bytes of bit-addressable data came to be known as "flags": the system flag, the keyboard flag, etc. This then developed as the "flags" table under LS-DOS. Space was established for 26 flags labeled A-Z. All flags were not initially assigned for use. But all are reserved for SYSTEM use. Don't be foolish to think that low-system memory is for application developers; the DOS cannot grow with features without using additional memory space. The DOS memory partitioning was established from the outset. Folks who abuse that partitioning just make it difficult for users down stream. Want some free memory. Get one of my hard drives with a Real Time Clock and use the 50-bytes of memory in it. Only the first 14 out of 64 are reserved for "system" use.

As far as the CTL-255 routine, I am aware that SuperScripsit uses it - not Scripsit. Probably ScripsitPRO uses it as well. Key

combination? I'm not sure what your implication is. The CTL-255 *KI control call returns the keyboard's matrix scan into the caller's "row" array. There is no combination of keyboard keys which utilize the code executed on that service call. What do you need patch space for?

Now here's the "skinny" on the INITCRTC table. When I received the initial specs for 6.0, it appeared that the Model 4 was going to use a standard software programmable video controller chip (CRTC). The BOOT routine was provided to configure the chip for 80x24 video. But I was not able to get the CRTC cursor addressing to operate the cursor. I later was advised that the specific CRTC chip used was mask programmed (firmware encoded) so that it would not be able to be programmed by the DOS. Supposedly, the reason was that there was insufficient room in the Model III C-ROM to implement the CRTC initialization programming; thus, the dual-registers on the chip were factory programmed for Model III and Model 4 modes. Too bad, as a programmable chip's ability to implement hardware scrolling would have made the video fly. The BOOT/SYS code to "program" the CRTC remained even though it served no purpose because I didn't need the space for anything else.

As far as a BBS, I already sponsor one - on CompuServe. It has the resources I could not provide. Joe does an admirable job as SYSOP. I can't see supporting another service. Besides, I couldn't put another phone line into the house - I have four already!

Fm Henry A. Blumenthal, Jacksonville, FL: Dear Mr. Soltoff, Success! Thanks to your Spring TMQ (which arrived two days after I dispatched a postcard asking why I hadn't gotten my copy, despite a first-class rate), I fixed the Error 7H corruption that would occur whenever I would try a backup to or from hard drive 0.

It was your answer to David J. Kelton on directory pointers. I inspected that byte and found a 14 rather than a 4C. 14, of course, is where the directory of the booting floppy is; how it worked its way over to your MISOSYS hard drive system files months after the partitions were installed, I have no idea. But that was the problem!

I also have the answer for Patrick H. Larkin, of Bedford, Texas, on corruption of VisiCalc data by the spooler. Yes, it does happen. Here is why, as explained by Mercedes Silver in the Feedback column of 80 Micro magazine (she was answering my question):

"Fixing the bank overwrite problem would require a major patch. ... VisiCalc does not respect the TRS/LS-DOS 6 banking scheme, simply because it requires the extra 64K of memory space for operation. Its approach also allows it to speed up bank switching without going through the slightly slower supervisory call selection techniques."

Ultimately, I bought an outboard printer spooler. But until then I reserved as little memory for spooling as I thought I could get away with when using VisiCalc, and kept a close eye on remaining KB in the upper righthand corner.

Now, here's a question about maintenance of your hard drive: Do I need to clean the fan area from time to time? Anything else I need to know? I can't find anything in the paperwork that came with the 20-meg machine about maintenance, other than replacing the clock-and-calendar battery someday.

Fm MISOSYS, Inc: Folks continue to have problems with their BOOT/SYS file as noted in this issue's DOS column.

Since our hard drive fan does not have a fine mesh screen, there is nothing to clog up in a short time frame. In the long run, you may want to vacuum dust and dirt periodically from the hard drive fan - as well as from your keyboard and monitor screen.

Resource: Companies

Aerocomp, PO Box 223957, Dallas, TX 75212 [214-637-5400]

Anitek Software Products, PO Box 361136, Melbourne, FL 32936 [407-259-9397]

Computer News 80, PO Box 680, Casper, WY 82602

Computer Reset, PO Box 461782, Garland, TX 75046 [214-276-8072]

Cornucopia Software, Inc., 1625 Beverly Place, Berkeley, CA 94707 [415-528-7000]

GRL Software, Suite 209, 1051 KLO Rd., Kelowna, BC V1Y 4X6, CANADA

Howe Software, 64 Windmill Road, Armonk, NY 10504 [914-273-3998]

Hypersoft, PO Box 51155, Raleigh, NC 27609 [919-847-4779]

M.A.D. Software, P.O. Box 331323, Fort Worth, TX 76163

Microdex Corp., see Computer News 80

Micro-Labs, Inc., 7309 Campbell Road, Dallas, TX 75248 [214-702-8654]

MISOSYS, Inc., PO Box 239, Sterling, VA 22170 [703-450-4181: Orders to 800-MISOSYS]

Pacific Computer Exchange, 1031 SE Mill, Suite B, Portland, OR 97214, [503-236-2949]

Powersoft: See MISOSYS, Inc.

Practical Programs, 1104 Aspen Drive, Toms River, NJ 08753 [201-349-6070]

RANTECH Computer Systems, PO Box 1101, Clackamas, OR 97015 [503-771-0390]

Storage Power, 10391 Oakhaven Dr., Stanton, CA 90680 [714-952-2700]

Tandy National Parts - Hardware [817-870-5600]

T/Maker Research Company, 812 Pollard Road, Suite 8, Los Gatos, CA 95030, [408-866-0127]

TRSTimes magazine, 20311 Sherman Way, Suite 221, Canoga Park, CA 91306

Try-o-Byte, 1008 Alton Circle, Florence, SC 29501 [803-662-9500]

MULTI-COMMAND

Frank Durda IV
5951 Ashford Ct.
Ft. Worth, TX 76133

As a long-time writer of code for the LDOS/TRSDOS/LS-DOS environment, I have been frustrated from time to time with the difficulty of writing a program that has to be run when the system is first booted. The problem is, how do you explain the steps that are needed to add your program to someone else's system (that you have never seen), without inadvertently wrecking their existing configuration? For example, if you knew they already had an AUTO command, the logical thing to do would be to have them write a Job Control Language (JCL) file that contains the command I wanted performed and the command they wanted performed. Then that JCL file would be invoked with the AUTO command. But, since there are many people who never learned how to use JCL at all, this seemingly simple task could take some time to set up correctly. And what if their command invoked a JCL file already? As you can imagine, covering all the possibilities becomes a problem.

Similarly, sometimes it would be very useful to perform several commands while using very little of the screen. I frequently find myself displaying a directory and then executing a series of commands on files in that list. Invariably, the directory listing scrolls off the screen before I have finished using it, so I have to issue the directory command again and again. Most of the scrolling is caused by the DOS Ready prompts and the individual command lines.

The following program, which is called MCM (Multi-CoMmand), helps solve both of these problems by allowing more than one command to be typed on the same line. Each command is separated by a semicolon and the commands are exe-

Applications for the User

```

title      'Multicmd (MCM) - Executes numerous commands from
one line'
print      short,nomac          ;Source packed for compact
printing.
psect      7800h                ;Below 32K and well out of the
way.
;-----
greet: defm      'Multicmd (MCM) - Multiple command executor - '
      defm      'Version 1(1) 9-Jun-90 '
      defb      0ah              ;For TRSDOS 6 / LS-DOS 6
      defm      'Copyright 1981, 1990 Frank Durda IV, All Rights
Reserved.'
      defb      0dh
;      You are free to reproduce and modify this program provided
the copy-
;      right strings and related code are not disturbed. This
program is
;      provided "as is". The author can be reached electronically
via
      USENET at:      ...texbell!rwsys!nemesis!z80mail
;      or      ...microsoft!trsvax!hal6000!nemesis!z80mail
006091
;-----
; OVERVIEW:
;      This program accepts one or more commands and executes them
via the
;      @cmndr svc. After each command completes, the next command
is execu-
;      ted until all are performed. The usage lines for this
program is:
usage: defm      'Usage: MCM [-v] command 1 [: command 2] [: command
3]'
      defb      0dh
;      The -v echos each command as it is executed for debugging
purposes.
;-----
stackr: equ      1              ;A '1' equ places program on stack.
;MCM can't be overwritten, but it
;does not leave very much stack left.
;A '0' equ disables this feature.
@dsply: equ      10             ;Display a string.
@abort: equ      21             ;Abort execution.
@exit: equ      22              ;Exit this program.
@cmndi: equ      24             ;Issue a command and don't come back.
@cmndr: equ      25             ;Issue a command and come back.
lsdos macro      #1
      ld      a,#1
      rst     28h
endm

;We begin here. LS-DOS gives us HL pointing at the first item after
;the program name in the system command buffer. First, we make a
copy.

begin: ld      de,ourbuf        ;Make copy of system command buffer.
      ld      bc,80            ;Copy entire length, although some
;of the data copied is useless.
      ld      hl,ourbuf        ;Use our buffer from now on.
begina: ld      a,(hl)          ;Start test for options.
      cp      ' '              ;Test for and skip any spaces.
      jr      z,skipsp         ;Spaces are ignored.
      cp      '-'              ;Now that we have a non-space, is it
;an option flag? If not, handle
;the first command.
      inc     hl
      ld      a,(hl)           ;Get option code

```

cuted from left to right. So if you wanted your AUTO command to set the clock, load the RS-232 driver, and then run BASIC, your AUTO command could look like this:

MCM CLK4 ; SET *CL COM ; BASIC

and all three commands would be performed without having to use JCL.

You can execute as many commands as you can fit on a line. The spaces around each command are not required. MCM also has an option (-v) that displays each command as it is about to be executed. As written, MCM will stop when a command returns an error. So, if COM was spelled wrong in the above example and a File not found error occurred, the BASIC command would not be performed.

MCM can be used in the AUTO command or at the DOS Ready prompt. If you desire, you can place MCM in your SYS13/SYS file and invoke it like this:

*DATE;LIB;TIME;FREE;DEVICE

MCM actually mimics one of the capabilities of the command interpreter (shell) that exists in UNIX and XENIX systems.

The code was written for the Radio Shack ALDS assembler, but can be easily converted to other assemblers. Note that MCM can be assembled in two different ways. In one form, the program places itself on the system stack so that none of the commands or programs you execute can overwrite it. This uses a lot of stack space, and depending on the number of drivers and filters you have and the programs you want MCM to execute, this strategy may not be practical. When MCM is assembled in its second configuration, it uses RAM in the range 0x7800 - 0x79ff, and you must make sure that the commands you want MCM to execute don't try to write into that area of memory. (This address range was chosen so that MCM could be run on XLR8er-based systems before the Memory Management Unit (MMU) is initialized.) The last command on the line is the exception to this rule and can use any part of memory.

```
;Here we read the option(s) that have been located. If you want to
;add other option switches, the tests go here.
```

```
cp      'v'      ;Test for Verbose option.
jr      z,goverb ;We have a match.
cp      'v'      ;Check for lowercase since it can
jr      nz,badopt ;be received in either case.
```

```
goverb:ld      (verbose),a ;Set verbose flag to non-zero value.
skipsp:inc     hl          ;Advance to the next character.
jr      begina  ;Look for other options.
```

```
;Here we have a switch that we don't know about, so display an
error.
```

```
badopt:ld      hl,usage    ;Point at usage-line message.
        lsdos  @dsply      ;Display it.
        lsdos  @abort      ;Abort execution.
```

```
;We have processed all the switches, so start handling the commands.
```

```
endopt:ld      a,(verbose) ;Determine if we should print the
or          a              ;titles.
jr          z,movcore      ;No, skip printing that.
push       hl              ;Save command pointer for a second.
ld         hl,greet        ;Point at title.
lsdos      @dsply          ;Display it.
pop        hl              ;Get back command pointer.
```

```
;This code copies the part of the program that processes commands
;into an area below the current stack pointer, then moves the stack
;pointer below that point. This provides a safe base to execute the
;program from, without having to worry about what part of memory a
;given command or program may use. For example, one of your com-
mands
```

```
;could be MEMORY (CLEAR) and MCM would continue working. The down-
side
```

```
;of this is that we use about 145 bytes of stack, leaving very
;little for other programs. At the top of the program is a switch
that
```

```
;disables placing the program in the stack.
```

```
movcore:
ift      stackr
push     hl          ;Save command pointer.
ld       hl,0ffffh   ;Subtract 1 from address given.
add      hl,sp        ;Get current stack position -1.
ld       de,edcore    ;Get address of last byte of core.
ex       de,hl        ;HL = Last byte of core region.
;DE = SP start (last) of stack area.
ld       bc,incore    ;Get size of core area.
di       ;-----Interrupts must be disabled
;while we manipulate the stack.
lddr     de           ;Transfer core pgm into the stack.
inc      de           ;Get entry address to relocated mod.
pop      hl           ;Get back command buffer pointer.
ld       bc,stcore    ;Compute address of copy of buffer
or       a            ;that is now in the stack area.
sbc      hl,bc        ;orig ptr - orig core base = offset
add      hl,de        ;offset + new core base = new pointer
ex       de,hl        ;HL = start of relocated core.
;DE = pointer to first command to do.
ld       sp,hl        ;Set stack to protect our code.
ei       ;-----Interrupts can now be enabled.
jp       (hl)         ;Jump to relocated code (below).
```

```
;This is the running core which breaks the commands apart and
;passes them to the O/S. It uses relative addressing so it can
;be relocated on-the-fly.
```

```
stcore:equ     $      ;Start of code to be relocated.
ex          de,hl     ;Move command pointer to HL.
```

```

endif
skipblk:ld    a,(hl)      ;Pick up the next character.
cp           ' '         ;Is it a leading space?.
jr          nz,fstchar   ;No, this is first char of command.
skiplp:inc   hl          ;Skip byte and move to next.
jr          skipblk

fstchar:push  hl          ;Save start of command. Now find end.
looknxt:ld   a,(hl)      ;Get next character.
cp          0dh          ;Have we reached the end of line?
jr          z,fndlcmd     ;End of command has been found.
cp          ','          ;Is this the end of a single command?
jr          z,docmd       ;Yes,end of single command.
inc         hl           ;Keep looking, bump to next character
jr          looknxt

docmd:ld     (hl),0dh      ;Change ; into a RETURN.
ex         (sp),hl        ;Get back starting point for this
                        ;command and save end of this command.

ld         a,(verbose)    ;Should we echo the command.
or         a              ;we are about to execute.
jr          z,quiet1      ;No, don't display anything.
push      hl              ;Save HL (start of command).
lstdos    @dsply          ;Display it.
pop       hl              ;Restore HL (start of command).
quiet1:lstdos @cmndr       ;Execute the command we separated.
ld        a,h             ;Did the command return a NZ error
or        l               ;code in HL? If so, don't continue.
pop       hl              ;Get back pointer to end of previous
                        ;command (now start of next command).

jr          z,skiplp      ;Handle next command.
lstdos    @exit           ;Pass error code to our invoker.

;Execute the last command on the line and exit this program.

fndlcmd:pop  hl           ;Get back pointer to start of cmd.
ld          a,(verbose)   ;Should we echo the command.
or          a             ;we are about to execute.
jr          z,quiet2      ;No, don't display anything.
push      hl              ;Save HL (start of command).
lstdos    @dsply          ;Display it.
pop       hl              ;Restore HL (start of command).
quiet2:lstdos @cmndi       ;Execute the last command.
;=====The @cmndi call does not return.

ourbuf: defb    80%0      ;Our command buffer.
verbose: defb    0        ;Quiet or noisy execution flag.
ift      stackr
lncore: equ    $-stcore   ;Size of area to relocate.
edcore: equ    $-1        ;End of area to relocate.
endif

end      begin           ;begin is first instruction to do.

```

About the author...

For those who may not know too much about Frank, recollect that he was the one who ran down that obscure hardware problem in the gate array Model 4 which caused random problems with floppy diskette recognition. Frank Durda IV has been with Tandy Corporation since 1982 and has designed hardware or developed software for the Model 16, III, 100, 12, 4, 4P, 4D, 2000, 6000, and several of the MS-DOS-based systems. He also is an instructor of assembly language, C and computer system design at Tarrant County Junior College.

There are a lot of things that could be added to MCM. For example, you could add an option switch to indicate if you wanted MCM to load itself into the stack or not. It could also be used as part of a utility that performs its own tasks and then allows trailing commands to be executed. If you were very ambitious, you could even write an extended command interpreter that allows multiple commands all the time. Have fun!

ROBOTICS VIII

The ROBOTICS VIII program was inspired by John Webster's BYTE Magazine article. The program has a task, the robot has eyes and tries to go around objects with minimum moves. You can manually set a task. Program on 5.25" disk for TRS-80 in BASIC for Model III 32K, Model 4 64K at \$45. Consulting services for Heat Transfer Analysis and Math programs available. Program has been sold internationally and is available from EMPAC, INC., 439 Edgewood Place, Rutherford, N.J. 07070 [201-939-8304].

TRS-80 NOSTALGIA?

you need a copy of

**WHAT I DID
WITH MY TRASH:**
Ten years with a TRS-80

A new book by Eric Bagai. Essays and parodies, weird rumors, mythic hacks, and just plain country sense. Send \$5.95 now to Flaming Sparrow Press, Box 9747, North Hollywood CA 91609. Order yours today!

TOUCH

The following program was submitted back in early 1989 by Ken Peck of McGregor, Texas [see reference to touch/ccs in TMQ III.iv pp 67-69]. Although I mentioned at one time that Ken's touch program was printed in TMQ, it apparently never was. "Touch", under UNIX, is a command to alter the date of a file to a selected date. Version 6.3.1 of LS-DOS provides the capability via the RESET command to alter a file's date to current system date and time; this touch/ccs allows arbitrary dates.

```
#include <stdio.h>
#include <unistd.h>
#include <utime.h>
#include <time.h>

#option FIXBUFS ON
#option REDIRECT OFF

char *prgname = "touch";

enum err_types {NO_ERR, CMD_ERR, ACC_ERR,
FMT_ERR, MOD_ERR, ERR_ERR};

void abend(err_no, path)
int err_no;
char *path;
{
    extern char *sys_errlist();
    static char *msg[] =
    {
        "no error",
        "touch filespec [mm/dd/yy[
hh:mm]]",
        "can't access",
        "invalid date or time",
        "can't update mod date on",
        "unknown error"
    };

    switch (err_no)
    {
        case CMD_ERR:
            fprintf(stderr, "usage:
%s.\n", msg[err_no]);
            break;
        case ACC_ERR:
        case MOD_ERR:
            fprintf(stderr, "%s: %s
%s.\n", prgname, msg[err_no], path);
            fprintf(stderr, "sys-
tem: %s\n", sys_errlist(errno));
            break;
        case NO_ERR:
        case FMT_ERR:
        default:
            fprintf(stderr, "%s:
%s.\n", prgname, msg[err_no]);
            break;
    }
    exit(1);
}
```

```
char *getint(p, i)
char *p;
int *i;
{
    int value = 0;
    while (isdigit(*p))
        value = value * 10 + *p++ -
'0';
    *i = value;
    return ++p;
}

main(argc, argv)
int argc;
char **argv;
{
    char *path;
    char *p;
    int month, day, year, hour, minute;
    struct utimbuf times, *ptimes;
    struct tm tmbuf;

    fprintf(stderr,
        "%s - by Ken Peck -
%8.8s\n", prgname, _DATE_);
    if (argc == 1)
        abend(CMD_ERR, NULL);
    path = *++argv;
    if (access(path, W_OK))
        abend(ACC_ERR, path);
    if (argc > 2)
    {
        p = getint(*++argv, &month);
        p = getint(p, &day);
        p = getint(p, &year);
        if (argc == 4)
        {
            p = getint(*++argv,
&hour);
            p = getint(p, &minute);
        }
        else
            hour = minute = 0;
        tmbuf.tm_year = year;
        tmbuf.tm_mon = -month;
        tmbuf.tm_mday = day;
        tmbuf.tm_hour = hour;
        tmbuf.tm_min = minute;
        tmbuf.tm_sec = 0;
        if ((times.modtime =
mktime(&tmbuf)) == -1)
            abend(FMT_ERR, NULL);
        ptimes = &times;
    }
    else
    {
        ptimes = NULL;
    }
}
```

Model III DOS

LDOS 5.3 Timestamping

Fm Dan Berky: Is there available, a utility comparable to the UNIX and MSDOS command TOUCH that allows the user to change the time stamping on files, either to the current system time/date or a user definable time/date? Is there a work around within LDOS itself to do this?

Fm MISOSYS, Inc: (1) The new LS-DOS 6.3.1 release includes the command, RESET filespec (DATE) to set the file date/time to current system date/time. (2) Look for a touch command written in C in this issue. I expect to be releasing a version of 5.3.1 in the future later this year.

LDOS Boot step rate

Fm Kevin R. Parris: I have an LDOS boot disk that changes step rates after 'CONFIGURING'; it was formatted while the system was running on a backup of the original master system disk, which has a slow step rate (quite understandable, considering the variety of equipment configurations it must support). Where can I ZAP the floppy step rate that is loaded during boot time, before the system reaches the sysgen processing phase, to make the whole thing go a little bit faster? I know the "correct" way to do this is just format another disk, with the desired step rate, and rebuild the boot setup on it, but I just don't want to take that much trouble right now. I am using version 5.3 on a 4/p. I just need to know which byte on which sector has the step rate, and what values I can stick in there.

Fm MISOSYS, Inc: Two things effect the boot rate. One: When you format a disk, you can specify the boot-strap-step-rate. This is the rate used for disk access during the boot process. That's after the

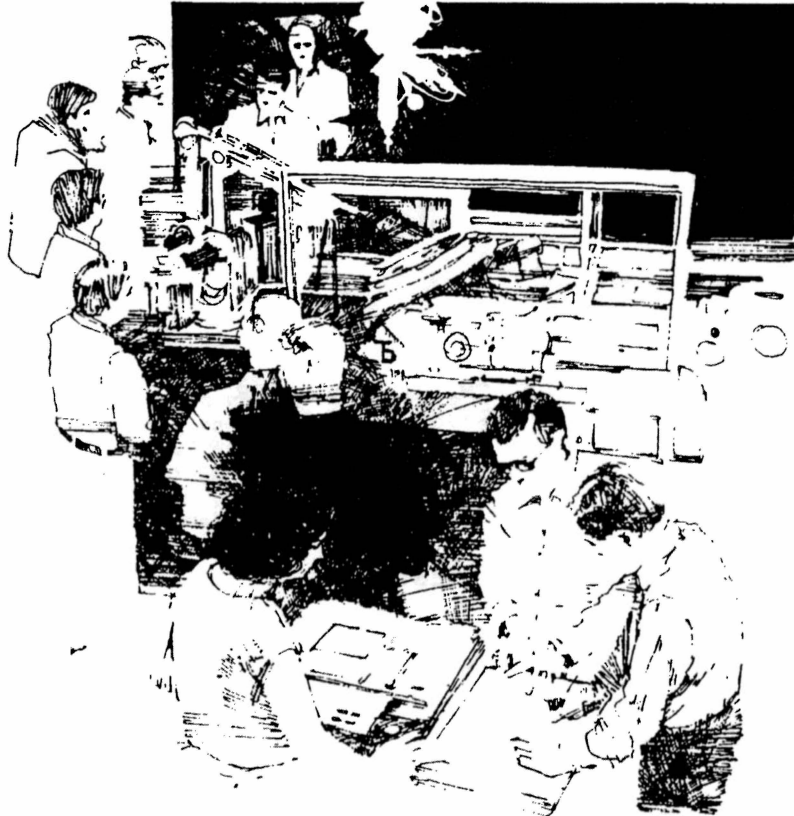
ROM reads the BOOT/SYS sector for continuing the booting by the DOS. After you are at LDOS Ready, you can change the drive stepping rate by using the SYSTEM command i.e. SYSTEM (STEP=s,Drive=d) where s is 0-3 (0 fastest). You can change the step on all enabled floppies by omitting the Drive=d parameter. It is this value of the SYSTEM step setting which takes hold after a CONFIG file is installed. So don't patch anything; just invoke the SYSTEM command, then invoke another SYSTEM (SYSGEN) - which can be included in the same SYSTEM command. i.e. SYSTEM (STEP=0,SYSGEN) changes all floppy step rates to 6ms and then SYSGENs the result.

Fm Kevin R. Parris: Roy, The disk already has a SYSGEN file on it, and after

the CONFIGURING message goes away, the 6ms step rate is in effect on the boot drive. But during the time before it gets to CONFIGURING it uses the 30ms step rate that was put on it when I formatted it. Which byte of what sector do I ZAP to change the step rate that it uses prior to loading the SYSGEN config file? My experience is that the SYSGEN does not change the step rate used during the initial boot stages.

Fm MISOSYS, Inc: Sorry, I misunderstood. The easy way, without knowing exactly where the boot step rate is encoded, is to format a new disk specifying boot step rate as 6ms (0 as a parameter). Then compare the boot sector (track 0, sector 1) of each disk. There should be a

DOS Subjects



one byte change in the seek command code of the FDC. Finding it yourself that way is better for you than me telling you where to look.

LDOS 5.3.1 Wish List

Fm Lloyd Evans, Arlington, TX: Roy, Since it appears that you have fulfilled most of my 'wish list' for an update to 6.3 please send me one. You better be careful or you are going to put the hackers out of business.

Just in case, here is my 'wish list' for 5.3. (1) Single key escape from Basic <SHIFT !>; (2) Repeat key at Dos Ready <CTRL R>; (3) Single key parms for MOD, OLD, and NEW in BACKUP; (4) Remove the : from FREE :d; (5) Update TED to 6.3.1 workalike; (6) Add UNKILL to one of the libraries; (7) Make the DOS commands as much alike as possible.

Thank you for your efforts on a product where the rewards in dollars must be small.

One of our club members gave a demo of 6.3.1 at the last meeting and I like the information given by the new MEMORY command but if Sys6 ever has to be reassembled please consider giving us the old screen back with a parameter of 'SWITCH' that would give us the new screen.

Fm MISOSYS, Inc: Some of your requests are reasonable; some aren't. Off the top of my head, <SHIFT !> could probably be added. LDOS already has a repeat key at DOS Ready - it's <CLEAR SHIFT R> available when the MiniDOS filter is resident. The <CTRL-R> facility in LS-DOS is part of the @KEYIN service call which in Model I/III mode is in ROM. Since the ROM "can't be changed", an add-on must suffice. Use MiniDOS; too many other programs may use ^R and conflicts would invariably result. Abbreviations should also be looked at to be uniform across both DOS releases. TED will be updated to 6.3.1 capabilities. The underlying thrust of LDOS 5.3 was to

make the commands as compatible with 6.3 as possible. I'll look more closely at that grooming for a 5.3.1 release. Finally, UNKILL is available as part of the UTILITY DISK 1; a small price of \$19.95 gets you UNKILL along with COMP, DCT, DIRCHECK, MAP, RAMTEST, READ40, and TYPEIN.

Fm Michael E. Webb, Gainesville, FL: I am writing as a TRS-80 Model III user, and as a very satisfied registered user of LDOS 5.3. I am appreciative of your continued support of the Radio Shack TRS-80 computers.

The new "TRSTimes" reports the release of LS-DOS 6.3.1 for the Model 4, which demonstrates your continued work to support the TRS-80 community with new software. As a Model III user, I am writing to urge the upgrading of LDOS 5.3 to a new version that also allows for continued date access (I may still be using this thing in the year 2000!). I would particularly encourage you to consider a change in the dating scheme that will allow infinite use—such as a four-digit year figure and elimination of the day-of-week feature of the dating scheme. It seems to me that it has been the day-of-week algorithms that have been the perpetual bugaboo of extending the dating scheme. Frankly, I could live without the day of the week, especially if it means that I would have a perpetual use of LDOS - as long as I choose to use a TRS-80 and as long as I can find one to use.

As the market for an enhanced LDOS 5.3 (5.3.1 or 5.4?) would encompass an even larger audience than that for LS-DOS 6.3.1 (since both the Model III and Model 4 will run LDOS), I urge you to consider an enhanced LDOS 5.3, with perpetual dating, as your next TRS-80 software project. I, for one, would be ready to buy one as soon as it is available.

Fm MISOSYS, Inc: Michael, To set the record straight, it has not been the day-of-the-week algorithm which has been the "bugaboo of extending the dating scheme". It was the simple fact that in the original

design of the DOS - which was inherited by LDOS and subsequently LS-DOS, the directory allowed only 3 bit positions to store the file year. In three bits you can represent only one of eight values. That's the origin of the original range of 1980-1987, a range of eight years. When LSI expanded the date support of the directory, they chose to provide both a 5-bit field (capable of supporting a 32-year range for new dating) and retained the old 3-bit field to provide complete downward compatibility. LSI furthermore chose to restrict the year support to just 20 years - from 1980-1999. That was most likely done to avoid the complications of rollover from 1999 to 2000.

Perpetual use? That's argumentative. In any system implementation given finite space, one always compromises between range support versus space consumption. For a data base to support 65,536 records, it requires only 16-bit pointers. To support more records, it would require a larger pointer size; each pointer would take up more space. How many folks will be using their TRS-80 past 2011? That's almost 22 years from now. I doubt that the audience will be substantial. Besides, even those folks who would still be using their TRS-80 typically would not have files more than 10 years old. It is always feasible to change the base year from 1980.

I hope that you now understand that day-of-the-week programming is not what hinders the year range support of LDOS and LS-DOS. But the new 32-year range in LS-DOS, and the same treatment to be given to LDOS as soon as I can get to that, will provide near perpetual use of your systems.

Confusion on "wildcarding"

Fm Michael E. Webb: Roy, Thank you for your explanation of how the LDOS/LS-DOS dating scheme works. As a regular (and generally satisfied) user of LDOS 5.3 (this letter is being prepared using PowerScript under LDOS), I appreciate the fact that you continue to support our machines.

In your letter, you stated that you would be working on LDOS again soon. I look forward to hearing of LDOS 5.3.1 or whatever you will be calling it. I write again to pass on a suggestion to review while evaluating for the upgrade.

One "bug" that has occasionally caused me trouble in LDOS (both 5.1.3/R - which I started out on - as well as 5.3) is in the wildcard functions, which sometimes wildcard when I don't want it to. It seems to work consistently, so its different manifestations are probably in the same part of the code.

How it works is this; when using the BACKUP and PURGE commands, as well as when I am using Richard VanHouten's DEARC and DEZIP utilities, I find that LDOS has a tendency to include files that I had not intended. For example, I regularly create what I call a "microsystem disk" to allow maximum data storage when copying files to and from MS-DOS disks using SuperCross/XT. I place only system files 1, 2, 3, 4, 8, 10 and 12 on the disk. However, I found it quite disconcerting that when the command "BACKUP SYS1/SYS:0 :1 (S)" is given, the computer responds by copying SYS1/SYS, SYS10/SYS, SYS11/SYS and SYS12/SYS all from that one command! Of course, that means that I have to go back and KILL or PURGE SYS11/SYS. Likewise, if I (for whatever reason) want to use the PURGE command to kill a certain file (let's call it "FILE1/TXT"), I cannot use the (Q=N) parameter if I have files that start with the same letters in the filename (such as "FILE1/TXT" and "FILE2/TXT") or else the program will delete them as well. Using DEARC and DEZIP, before they supported the (Q=Y) parameter, I was occasionally surprised to see it extract more files than I wanted it to, likewise because the file(s) I didn't want had the same first letters as the filename proper of the file I did want; for example, an ARC or ZIP file might have a file called "KZ/DOC" (made-up name) as a short 4K README-type file and "KZ43/DOC" as a longer 55K manual file. If I entered "DEZIP KZ/ZIP KZ/DOC (LIST)", the computer would not only list KZ/DOC, but KZ43/DOC as well. Once I tried this where the longer file came before the short file, and it was very irritating to have

to scan through the longer document to get to the one I wanted to review.

I don't know if anyone else has complained about this (or if you may have already prepared a fix for this and published it in *The MISOSYS Quarterly*), but I know for me that I would really like to be rid of this little complication. You have rightly pointed out that a major change in the dating scheme probably would be unnecessary considering how few TRS-80's will be in operation past 2011. I do think, however, that this particular adjustment would be worth making, or at least noted so that people don't get unnecessarily surprised when extra files get backed up, or worse, wanted files get PURGED by people who don't know about how to un-PURGE files by using SuperUtility or the PD program RESTORE/CMD. And there ARE new TRS-80 users coming on board - I myself was introduced to TRS-80 Model III computing only a little over two years ago (recent for TRS-80) when my father got himself an XT clone and passed his TRS-80 on to me for my home computing use.

Regardless, I do truly enjoy using LDOS, and whether or not you incorporate this suggestion, I am sure I will continue to be a faithful and satisfied user.

Fm MISOSYS, Inc: You have misunderstood the implementation of a "partial filespec" (partspec) in LDOS. The User Manual notes one syntax of BACKUP as follows:

BACKUP partspec w/wcc:s TO
:d (parm,parm)

Further text states, "The use of partspecs, -partspecs (not partspecs), and the wcc (wildcard character) will let you choose files based on their filename and extension." Many examples are provided. Also noted is, "you can see exactly what files will be moved by a particular BACKUP command by doing a 'DIR' command of the source disk using the same partspec and/or parameters you intend to use with the BACKUP." The *General Information* section includes a subtopic entitled "Entering LDOS Commands". This section

should make it abundantly clear that all full file extensions which have a partial character string which matches the partspec are part of the class of files identified and designated by that partspec for those DOS commands which support partspecs. Incidentally, DIR, PURGE, and BACKUP share the partspec support which is to operate identically across all three commands.

What I gather you would prefer is that the implementation of "wildcarding" in LDOS (and subsequently LS-DOS) would match identically to the wildcarding implementation in CP/M and subsequently MS-DOS. In that implementation, there is no such thing as a partial file specification; the use of wild card characters, asterisk and dollar sign, introduce respectively match all remaining or match position only. But that's another DOS. LDOS and LS-DOS behave according to their implementation which matches their documentation.

DOS Error Codes

Fm Neil L. Van Leuvan, Abington, MA: We purchased a LDOS 5.3 Model 3 upgrade kit in January of 1988. We sent in the registration card, but I have enclosed a copy of the invoice to verify this. Since this time we have not received any patches or notices of upgraded versions.

Although the program works fine under most conditions, we have recently encountered a possibly serious bug. We are running LDOS 5.3 on a Model 4 with a 15 meg Radio Shack hard drive partitioned into 7 logical drives (5 on the hard drive and 2 floppies). Under LDOS we are running the Tandy Network 3 host program, so this hardware and software configuration serves as a host for 16 Model 3/4 stations. Usually the directory of a logical drive fills up with small student programs way before the entire drive space is used. When this occurs, LDOS indicates that the disk space is full. However, when the available space on a logical drive is full but there are still directory slots open, LDOS puts the file name in the directory and does not indicate that it cannot store

the data on the drive because there is no room. Thus, the student thinks that his program has saved when in fact nothing has been stored under that directory entry. In fact, even though the file name is in the directory, a DIR reveals that the file has 0 records and the date field is empty.

I would appreciate any patches that may have been written since we purchased our version of LDOS. 5.3 and any information you may have regarding this apparent bug.

Fm MISOSYS, Inc: Neil, I beg to differ with your opinion, but LDOS does indeed return an error code when a WRITE SECTOR service function is requested but the disk drive referenced has no available space. There is never any error returned when creating a file directory entry when an unused directory slot is available but free space is not because that scenario is NOT AN ERROR. All programs using DOS service functions are expected to recognize and handle all DOS error return codes. You don't state what program was in execution when your problem was experienced, so I cannot comment on the particulars.

Incidentally, when a program continues to issue WRITE SECTOR commands but chooses to ignore error codes, the resulting directory data is exactly as you indicated - a file specification with zero records. This data could also be generated if a program attempted to create a new file on a disk which had an unused directory entry but no free space and then recognized the error code returned from the first WRITE SECTOR but neglected to KILL the file created!

Finally, any patches we have released for LDOS were printed in *THE MISOSYS QUARTERLY*, our support publication available on a subscription basis. As a rule, we don't provide direct notification of changes to products that retail for \$34.95.

Model III Programs

Note on DSM51

There was a short fix to DSM4, DSORT61/FIX, printed in *THE MISOSYS QUARTERLY*, issue I.ii, back in Fall of 1986. The fix was to correct DSORT's index file generation when virtual memory was used for sorting and the number of records was a multiple of 128 (i.e. was divisible by 128 with a remainder of zero). It's taken close to four years for the same bug to surface on the model III version of DSORT.

Scott Ranstrom, of Portland Oregon recently called that to my attention. I researched the problem he was having with DSM51 when the index file generated was associated with a data set with a number of records evenly divisible by 128. I believe the following patch will correct that.

```
PATCH DSORT51 (DOC, 96=00
00:F0C, 96=28 03)
```

Model 4 DOS

FIX Format

Fm Carl Berger, Largo, FL: Roy, I still have to use DATECONV after using DDFORM. I thought the patches I applied would eliminate the need to use DATECONV. I applied DDFORM6A/FIX and DDFORM63/FIX. I tested DDFORM in an all floppy environment to make sure my hard drive configuration was not the culprit. Are you sure that after doing a DDFORM and then a DD 7 FILENAME:1, then BACKUP :0 :7 is supposed to give me the correct DATE/TIME stamps on the files? My DDFORM still gives me TRSDOS 6.2 directory structure until I use DATECONV (Please explain) I am

using LS-DOS 06.03.00 Level - H, DDFORM/CMD Version 1.0.1(c) 1983, DD/CMD Version 1.0.1 also.

Fm MISOSYS, Inc: Suffice it to say that Carl and I went round and round on this subject until he sent me a copy of his "patched" DDFORM. I was then able to discern the problem. The fixes for both Model III DiskDISK and Model 4 LS-DiskDISK printed in TMQ issue II.i altered the programs to support the DOS x.3 extended dating structure. A sample patch line from one of the fix files was:

```
D02, A5=63; F02, A5=62; D02, B8=4F;
F02, B8=47; D03, 7D=00 00; . . .
```

Where fix files of this nature were printed, accompanying text stated, "In order to conserve space in *THE MISOSYS QUARTERLY*, we may logically print more than one FIX line on a printed line; HOWEVER, ALWAYS USE A HARD <ENTER> FOR THE [semi-]COLON WHEN TYPING IN A FIX FILE for TRSDOS 6." The reason for this is that the Model 4 PATCH command doesn't support the ability to have more than one logical line-end on a physical line. Thus, it will essentially ignore any subsequent D-type records after the first. Carl's problem was that he forgot to restructure the fix; thus, the patch was only partially applied. Note that the LDOS PATCH command does support multiple logical line-end characters on a physical line as it has additional code to convert all logical carriage returns to physical ones. Too bad I didn't recognize that when 6.3.1 was implemented.

Fm Carl Berger, Largo, FL: Dear Roy; Thank you for solving my DiskDISK patch problem. I typed in the patch verbatim as it was printed in the magazine not realizing I was violating patch syntax by putting too much into one patch line. My DDFORM formats with LS-DOS 6.3 date convention now.

LS-DOS 6.3.1 Coming

Fm MISOSYS, Inc: Prior to the formal release of LS-DOS 6.3.1 version in March of this year, I let it be known via our CompuServe forum (PCS-49) that MISOSYS was soon to be releasing the update. I encouraged readers to submit any requests for enhancements to be evaluated for adoption. Herewith follows some dialogue received both on-line and from letters concerning the release. It demonstrates that users do indeed have a valued opinion which helps shape the operating system they use.

Fm John Tollini: Roy, I have been meaning to check on a quirk (maybe even bug) that has been in at least 6.3 from the start. When a directory is requested either from BASIC (system "dir") or through PRO-WAM, it is not sorted as it would be directly from DOS.

Fm MISOSYS, Inc: What do you mean quirk? DIR can only sort the directory if sufficient buffer space exists for the sort. When you invoke DIR from @CMNDR (as is the case with BASIC's SYSTEM"command" or PRO-WAM's LIB function), the only free buffer space is the DOS library overlay region - up to 2FFFH. Sorting cannot be done in that space as DIR occupies most of the overlay region with its own code.

Fm Shane Dawalt: Roy, Who (software level) is actually responsible for a 'basic' interface. (A basic interface would be command-line editing with, perhaps, a simple review facility. Full-screen editing and/or cut & paste ability is not needed as far as I am concerned, that is, not if command review is implemented.) In most systems, you have the DOS, the shell and the applications. The DOS shouldn't be bothered by user input, it should only provide system services. The shell, on the other hand, is the communications inter-

face between the files/devices and the user. Therefore, the shell should provide the basic interface with allowances for applications to take over command entry when needed.

The main problem with add-ons over shell-provided interfaces is "extra baggage." I use CED under MSDOS. It supports excellent command line editing and command review abilities. It also supports other features that I just cannot grasp what they should be used for and haven't really paid all that much attention to them. Unfortunately, these unused "other features" are still using their weight in RAM whether I want them or not. A shell wouldn't provide all this "extra baggage." Just a straight editing/review implementation would be provided which allows add-ons from applications or utilities. My problem with CED is that some programs need that extra 14K (with CED installed I have 547K free). To run these other programs, I must write specialized startup procedures which are pains to write and run.

The same basic problem holds for the M4 ... if you don't have enough memory, you need to write messy specialized environment disks OR write nasty JCL things. Yuck! I know the history idea would be hard to fit into a Model 4 with limited memory ... but couldn't SYS1 provide the line editor?

Fm MISOSYS, Inc: The "who" or "what" of responsibility for providing a given set of features is the designer of the DOS. The design has to take into consideration an implementation of features to serve the widest audience recognizing that memory space for system code is finite. There is no room in the present design of the DOS to include command line editing. I also question whether such a feature should be part of the basic set of functions.

You claim that you don't want to include extra baggage. But why would you just assume that an add-on, such as DOEDIT, would add any more baggage than just implementing the command at the DOS level? I am puzzled by your remark. Yes, I could have added DOEDIT as another module - like FORMS/FLT and PR/FLT

and KSM/FLT - but why should I add any thing just because some folks think it should be part of the DOS. Personally, there is too much in the DOS product now that should not be there. What other DOS provides a communications program? At the price of the Model 4 DOS, it already goes overboard.

Fm Adam Rubin: Roy, After much thought (and considerable editing), here are a few things I'd like to see in LS-DOS 6.3.1.

Housekeeping: 1) Fix overlay lengths (at 23FE) of SYS2, SYS5, and SYS12. 2) Fix EOF byte offset in directory for MOD-ELA/III. 3) Shortening FORMAT/CMD by eight bytes would free up another granule on the distribution disk. (Reword a message or two?) Reducing SYS8 by fifteen bytes would also free up a granule.

Documentation: Document the parameters CREATE (FILL), DATECONV (CS), PATCH (O=N), and perhaps SYSTEM (CKDRV).

Bug fixes: 1) @LOC/@LOF — These actually return a 3-byte result (in A, BC). When the value returned (number of records) exceeds 65535, they return error number (# records MOD 65536). This is easily fixed by changing the instruction at 14DCH from OR A to CP A, so they will return the same value but with the Z flag set. Unfortunately, this instruction is the ORARET@ used by DEBUG (EXT), so the declaration of ORARET@ in SYS0 would need to be changed, along with one reference in each of SYS0 and SYS9 and two in SYS7 (DEBUG). 2) RESET file-spec — file access level error, "Source" p.339 lines 2471-2478, similar to FUNDATE. (But I'll bet you already got this one, right?) 3) @GTMOD — When low memory (driver region) is exactly filled, high memory isn't checked. Offhand, it looks like changing line 5210 (Source p.272) to CP @BYTEIO<-8+1 should probably do it.

Bug fixes or enhancements, depending on your viewpoint: 1) SYS1 — Move lines 1370-1380 (Source p.134) before the label

CMD3A (line 1320). This will keep SYS1 from displaying a CR for @CMNDI. (@EXIT would then display a CR every time, even when the line entered begins with ".") 2) BASIC translates some DOS errors inappropriately. Change X'1A' "No directory space available" from "Disk write protected" to "Too many files" (change X'52B8' or record, byte 23,44 from 44 to 43), change X'1E' "Directory full - can't extend file" from "Disk full" to "Too many files" (X'52BC' or 23,48 from 3D to 43), and X'21' "No device space available" from "Disk full" to "Too many files" (I couldn't find a more appropriate BASIC error, that's X'52C0' or 23,4C from 3D to 43). 3) BACKUP and COPY (except when CLONE=NO) — If the source file is **not** CREATED but the destination exists and is CREATED, the destination's size is unchanged but it loses its CREATE flag. It seems to me it would make most sense for the destination to end up matching the original, with the exact amount of space allocated.

Enhancements: 1) @SOUND — I never liked the pitches as supplied. Changing the note table at X'03D1' to 3D 5C 41 56 49 4D 52 45 5C 3E 62 3A 6E 34 7C 2E hex produces a nice E-flat major scale. 2) COM/DVR — The resident portion of this driver can easily be shortened by eleven bytes. If you'd like details or source code for this, just ask. 3) CREATE — I'd like to see an option NOT to set the CREATE flag on the file. (In other words, just initialize an "ordinary" file of the requested size.) This is very useful when I need to preallocate space for data that I'll only be writing once — my application can @INIT, @WRITE/@PUT, @CLOSE, and voila! my data file is exactly the size of my data. 4) SYS0 — Any chance of getting a more attractive startup logo than the current "Welcome to LS-DOS"?

Fm MISOSYS, Inc: Adam, The overlay length of SYS5 was not corrected when LSI introduced the patch to SYS5; that's taken care of by re-assembly. SYS2 and SYS12 overlay length words are now re-adjusted by moving LAST to precede the DS. That's not what I call a bug/enhancement, although it would save a few bytes

of memory if you used SYSTEM (SYSRES) - the whole reason for OVRLEN.

The EOF offset in MODEL A/III was already corrected.

CREATE (FILL) and PATCH (O=N) are documented only in the DOS/HLP file; I'll also note (FILL) in the 6.2 -> 6.3.1 upgrade docs; not 6.3.0 -> 6.3.1.

The CS parm of DATECONV is covered in Addendum 1A of 6.3.0 upgrade docs.

SYSTEM (CKDRV) is covered in the HELP; it's use is too technical to discuss it in the upgrade documentation. If MISOSYS ever puts out a DOS manual...

FORMAT/CMD actually grew a little bit; but I took a look at the code and was able to perform local optimization without message chopping. This dropped it to 18:206 saving a granule. I also optimized CONV/CMD to cut it back to 6:255; that you didn't mention. Another granule saved!

@LOC/@LOF: We don't see eye to eye. A record position has always been a 16-bit value; the functions were never documented to return a value in registers ABC. In fact, the actual result is perfect: NZ return implies that the result has overflowed a 16-bit value but the overflow byte is in a register - A! If the return flag state were always Z, then existing programs may not operate correctly; It works as *undocumented*...

RESET filespec was already fixed when I added the additional parameters.

@GTMOD: You're right about that one. I never had a report about it before. It's fixed in 6.3.1 per your suggestion.

I can't accept your suggestion for SYS1. Why LSI put in that CR @DSP is a mystery; but the code operates exactly as it is commented. The comments say "Do a line feed on CMNDI and @EXIT". The code specifically avoids the CR display on "." commands. Thus, changing it to CR would inhibit it for CMNDI but not for "."! Why change? Is that a preference? I'll leave it as is.

I am adverse to changing anything in

BASIC. First, it is not a MISOSYS program. Second, maybe programs exist which trap those errors. All changes to BASIC - other than the LSI enhancements - were requested by Tandy.

BACKUP & COPY vs CREATE: Other anomalies exist. If the source is created, the destination doesn't equal the size of the source. Why should a BACKUP copy use more space than necessary? If you want to restore the source of a "created" file from the BACKUP, the source is still correct. I think you are confusing the intent of a backup file; it is used to save a copy of a file's contents.

The pitch of all notes in @SOUND varies with the speed of the CPU; different versions of the Model 4 (i.e. 26-1069, 26-1069 rev C, 26-1069A, 26-1080, 26-1080A, etc.) have different effective CPU speeds. I have about 4 machines each producing a different set of tones because each runs at a different speed. Whose machine and whose ear should we use for one DOS? No need to change that.

COM/DVR: If you have some specifics, you have about one day to report; otherwise, the 6.3.1 is set in GOLD.

Although I differ with your suggested use for CREATE (i.e. you can easily code that in your assembly programs), and the documentation for CREATE has always underscored the aspect of NO DEALLOCATION, I went ahead and added a SHRINK parameter (no abbreviation - I couldn't find a better word).

Finally, I changed from the asterisks to a thin graphics box which works fine on the Model II as well (different character values). Thanks for the input.

Fm Adam Rubin: Thank you VERY much, Roy! I was very pleased just to be told that my list would be considered; I never expected a detailed reply to each suggestion. I realize 6.3.1 is almost finished, but there are three things I wanted to mention before it's finished.

1) Shortened COM/DVR: I have not only specifics, but the complete source code. If

you'll check your EasyPlex mailbox (or whatever they call it this week), you'll find a 7K /ASM file from me that will produce the 6.2 or 6.3 version of COM/DVR, standard or shortened. Comments from "The Source" are omitted, but all my changes and additions are commented. Set the EQUates for DOS63 and NEW as desired, assemble with PRO-MRAS's -GC switch, and you'll get either exactly what was released, or the 11-bytes-shorter version of it.

2) @LOC/@LOF: I never thought about it that way, but NZ to indicate an error makes sense. However, both the RS docs and "Programmer's Guide" state that if NZ is returned, then the error number is in A. This means that a file with 64K to 128K records returns error X'01', "Parity Error During Header Read"; a file with 128K to 192K records returns error X'02', "Seek Error During Read", and so on. Guess I'll have to live with this one, right?

3) I found one more spot where, if you want, you can free up a granule without any code changes. In BASIC/CMD, LSI's enhancements and initialization load right over the original MicroSoft initialization code when BASIC/CMD is loaded. (The LSI init code essentially duplicates the original, including the copyright notice.) Since MS's init code is now superfluous, changing record, byte 51,41 (hex) from 02 to 4A, and removing 51,8C through 54,4B (inclusive) will get rid of enough to free up a granule.

Again, thanks for looking at my suggestions and for your thoughtful reply.

Fm MISOSYS, Inc: Thanks for your input. I did look at your COM/DVR changes after our phone conversation and saw nothing that should prohibit their inclusion. They were integrated. I've always said that one person cannot deal with the entire DOS. That's why I'm extremely cautious about any change. There's just too many global aspects to consider. But as I said previously, I'm not touching BASIC.

Fm Pete Granzeau: Roy, I understand you are NOT fooling with BASIC (for obvious and good reason). But I always wished that the /BAS extension would automatically be added on the command line and in LOAD and RUN commands.

Fm MISOSYS, Inc: Don't you believe that I would prefer that too? I can't stand having to add an extension; and I refuse to use extensionless files. One day, in my spare time, I may look to work up a patch to add that. But it's also a nightmare with customer support. I know that from my LDOS 5.3 release where I dropped the BASIC/CMD program which just dealt with the extensionless programs. You can't begin to believe how many folks couldn't get their previously RUNnable programs to work because their BASIC programs had no extension of "/BAS".

Model 4 Programs

Key Stroke Memory

Fm Fred Sinder: Does anyone know of a fullblown keystroke memory program for LS-DOS 6.3, or earlier, that will record keystrokes for playback later and will work with most application programs, especially Multiplan? It may be something simple, such as being able to make type-ahead play from a diskfile. Can KSM be used to playback elaborate routines that include arrow key movements, etc? And, is it able to wait to deliver the next command until after the results from the preceding command are completed?

Fm MISOSYS, Inc: KISTORE is part of the GO:SYS package. It will store all keystrokes to a disk file to be played back at some future point through ZSHELL - the I/O redirection facility. Has been around for about five years or longer.

Fm David Huelsmann: Fred, What you may want to look into is a combination of the normal JCL function plus the utility program TYPEIN/CMD. I won't swear that TYPEIN will work with MULTIPLAN since I haven't tried it but it gives you the best chance. Nothing I know of will record all of your keystrokes while you are executing the program in question (LB - a database manager is an exception to this). So you will have to manually record the keystrokes and then feed them into TYPEIN (which will create a file) for later playback.

TYPEIN/CMD used to be available from MISOSYS (703)450-0239 as part of the LS-UTILITY package.

Fm MISOSYS, Inc: Dave, better rethink your position. KISTORE was part of the old ZSHELL package. KISTORE saved all keystrokes to a disk file. Then could be replayed using ZSHELL. Worked with just about anything.

Fm Fred Sinder: Thank you very much for recommending your GO:SYS disk to me. KISTORE exactly fills my needs. Interestingly, David Huelsmann's suggestion about using TYPEIN from the LS-UTILITY DISK, combined with KISTORE'S ability to record keystrokes, is what finally did it. Believe me, though, the rest of the programs on both the GO:SYS and LS-UTILITY DISK disks will get good use as well.

I, and I suspect others, would have purchased these two disks long ago, if we had known what was on them. Merely a suggestion, but if your ads could list the descriptive names of all the programs on with each package, you might get more sales and those of us who depend upon your continued existence will be better off.

Now, one last thing. Is there any way to easily edit the KISTORE keystroke files? For now, I deliberately record the keystrokes with extra characters, so that I can

go in later with a direct file editor, make desired changes and fill-in trailing blanks with X'00's nulls.

Fm MISOSYS, Inc: Probably the best editor to use to edit those "keystroke" files is SAID, the editor which comes with MRAS and EDAS assembler packages. SAID has a hex input mode so that any character value can be inserted; it also displays the hex value of the character at the cursor position so "unprintable" or strangely "displayable" characters can be discerned.

Everything that is in the GO:xxx packages are program products which have been sold for years. True, catalog space is at a premium. On the other hand, I am in the process of working up an entire new catalog. Since I took over PowerSoft products, I need a new catalog. I am toying with the idea of putting it on disk.

LS-DOS 6.3.1 is released

Ramdrive and/or Memdisk

Fm Bruce J Hutchison: I am having a problem using the extra memory in my Alpha-Tech board under LS-DOS 6.3.1. I have been using the Ramdrive driver uploaded by Michael Jacobs (74076,762). It has worked fine with LS-DOS 6.3.0 and the @BANK patches provided by Roy. My system is as follows: Model 4 Gate Array 5 Meg RS Hard Drive using RSHARD drivers from MISOSYS Anitek / Alpha Tech Memory Board with 512K Smartwatch.

I applied Roy's latest @BANK patches for 6.3.1 and then tried to install Ramdrive. The system was booted up in floppy mode with the hard drive powered off. I got an error message "Illegal access attempted to protected file" even though the protec-

tion level is "Full" and there is no owner password. I then tried installing the Memdisk that comes with 6.3.1. It worked perfectly, both on a disk with the @BANK patches and on a pure, unmodified 6.3.1 backup.

I then tried some experiments. I took a pristine Diskcopy-made backup of my master 6.3.1 disk. I purged everything from it except the System files and Memdisk/DCT. I then backed-up a copy of Memdisk from an unmodified 6.3.0 disk to the 6.3.1 disk. Upon trying to install 6.3.0 Memdisk while 6.3.1 was running, I got the message "Disk space full", and was returned to LS-DOS Ready. A quick list of the directory showed that SYS6/SYS was now 130.5K but the DATE, TIME & MOD FLAGS HAD NOT BEEN CHANGED! Yet when I listed the directory in hex mode all of the extent fields in the File Primary Directory Entry had been filled and an Extended Directory Entry had been created in an empty directory slot. I then made another disk purged the same way but this time write-protected it, both with software and on another test with a sticker over the notch. In each test the write protection worked but I received a new error message, "GAT Write Error".

Tonight I tried another test. I formatted a double-sided disk and copied all of the 6.3.1 system files to it starting on track 17. I placed a copy of Memdisk 6.3.0 on track 1. I again got the "Disk space full" message but the file sizes were not what I expected. SYS6/SYS was now 162K and the total size of all files on the disk only added up to 216K but at the top of the directory screen it said that 0K was available and a free space map shows the entire disk allocated. Another check of the directory in hex shows all extent fields filled and an extended entry being created for SYS6/SYS. I purged SYS6/SYS and then looked at a free space map. The last two grans of track 39 and all of tracks 1 through 16 are still shown as allocated, even though nothing had ever been put on them.

I am at a loss trying to figure out what is happening here. Any ideas? At this point I am sticking with 6.3.0, as I know that my memory board works with it, but I would like to use 6.3.1. I can understand Ramdrive

not working with 6.3.1, after all Roy is under no obligation to guarantee that all software from third parties will always work, but why doesn't Memdisk work? After all, going from 6.3.0 to 6.3.1 shouldn't change that much especially since so much effort was made to make the program interface in 6.3.0 the same as 6.2.1 and I consider that a much more major upgrade.

Fm MISOSYS, Inc: The clue is to check the loading location of any DCT program. The 6.x.x documentation provides for user programs which utilize any DOS library function to stay out of the DOS library overlay space; that means to stay away from 2400H-2FFFH. Unfortunately, some programmers did not heed that caution and originated a DCT at 2C00H - probably where it was safe from 6.0.0 up through 6.3.0. But in order to add additional capabilities in the SYSTEM command of 6.3.1, that library module takes up more space. Note that the SYSTEM command is used to install /DCT modules; thus, a /DCT module which origins its initialization code at 2C00H will overlay the SYSTEM command's /DCT loader during the load. What happens is that the loading crunches the SYSTEM code and all sorts of error messages can be the result. I am positive that the problem you are having is caused by that error. Even the RAMDISK/DCT done by HiTech which is on the XLR8er disk is at fault. I had to disassemble that to re-assemble it at a new origin. It's not the fault of the DOS. Folks must stay out of 2400H-2FFFH if the "program" is to utilize DOS library overlay facilities!

GRAFDISK Problems

Fm Pieter J. Plomp, Amsterdam, Holland: Roy, I am very much puzzled with trouble I ran into practically the first time I used LSDOS 6.3.1. After reading the last CN80 I decided to try the GRAFDISK program to create a GRAFDISK memory disk.

But I am getting a Directory Read Error when I use LSDOS 6.3.1 in drive 0. With

LSDOS 6.3.0 or TRSDOS 6.2.1 however everything works fine!

Of course I made backups of everything on new diskettes - but the same result. The directory sectors are without errors. Ergo the Read Error must be a software error due to LSDOS 6.3.1. Directory structure.

Before I shall publish this I should like to hear from you about it, as this could be a reason for a few not to update. But I told Stan Slater who sells Grafdisk on his CN80 diskette that it will not work with the new LSDOS. How can this be possible, and will it not occur with other programs?

To make it easy for you I shall send you a disk with my (mini) Dos 6.3.1, plus the Grafdisk files and a "auto" Do System.

Fm MISOSYS, Inc: Pieter, This is in response to your letter wherein you posed that attempting to use a shareware utility called GRAFDISK - which uses the hires graphics RAM as a RAMDISK - cannot be installed under LS-DOS 6.3.1. The reason is due to the GRAFDISK/DCT violating the documented program interface of the DOS. The DOS documents the user address range for programs utilizing DOS library functions as the range 3000H-HIGH\$. GRAFDISK/DCT loads starting from 2C00H. This causes it to overlay a portion of the SYSTEM library function module resulting in an error. Because of the additional functionality of the SYSTEM command in LS-DOS 6.3.1, the SYSTEM module is larger - but the SYSTEM module does NOT violate the documented range of the library overlay region - GRAFDISK does!

Another /DCT program which apparently violated the rule was RAMDISK/DCT supplied as part of the Model 4 XLR8er interface disk. This RAM drive program was developed for Hi-Tech and subsequently provided by MISOSYS. As I did not have the source code, I had to disassemble the program, re-ORG it to 3000H, then re-assemble it. The revised program is on recent XLR8er disks. If you need it, send in your XLR8er disk for a refresh (include mailer, label, and return postage

for "free-of-charge" refresh, otherwise it's \$10 + S&H). Note that the combination of PEXMEM and ERAMDISK with the Michel Houdé patches still works perfectly.

I caution anyone that using /DCT programs which violate the documented user range will cause you problems. I recommend you refer these problems to the source of the offending program.

Format patch

Fm Dave Krebs: Roy, A couple of questions and/or comments on the patches in TMQ IV.iii. First, is the DISKCOPY patch on page 10 for 6.3.0 Level L+? It looks like it is! Second, if it is, is the appropriate change made to 6.3.1?

Finally, the comment! The FIX631A/JCL worked fine on my backup of your master 6.3.1 disk, but not so good on my double sided boot disks that were updated with the procedure in the LS-DOS update! The patch aborts with the error "find line mismatch" 'cause my boot disk has the date in the area that you put "Level 1A" on your master disk, and the password of BOOT/SYS was still LSIDOS. I did get things going by ATTRIBing BOOT/SYS and changing the PW to "SYSTEM6" and used FED2 to change the DATE area to read "Level-1B". A suitable patch line would be:

```
d02,18=4c 65 76 65 6c 2d 31 42
```

I don't like to patch without a 'find' line. Would a find line of: f02,1a=2f work? Thanks for your time!

Fm MISOSYS, Inc: The DISKCOPY patch has nothing to do with the DOS Level; but the patch is only for the 6.3.0 release as it has been added to 6.3.1's DISKCOPY.

Anytime you are patching the BOOT/SYS file of a disk not directly made from 6.3.1, things may be different. For one thing, if only backups by class have been

done to it, the password of the /SYS files will not have been changed to "system6". That would cause problems with the password entry in the filespec of the patch line. Suggest you just do as you did, or patch a working floppy made from the 6.3.1 and back that up to your other system disks.

Incidentally, LS-DOS 6.3.1 doesn't replace any date string with the "level" nomenclature; that's identical to 6.3.0. But the last release of 6.3.1, for instance, had "Level-L+" in the position where 6.3.1 has "Level-1F" (currently). Your patch would only need to alter the last two characters of that string.

SPOOL Bug?

Fm William J. Newman: I have 6.3.1 purchased earlier this month. SPOOL gives me this with my Anitek upgrade to 512K. I have a 15 meg HD (no drivers, etc., being used)

```
SPOOL *PR (BANK=4,DISK=0)
```

Requested bank in use

When I do SPOOL *PR (BANK=2,DISK=0), the spooler is operational. How do I access other banks?

Fm MISOSYS, Inc: The DOS can only access memory banks known to the DOS. Adding memory boards beyond the 128K requires making modifications to the DOS so that the @BANK service call is "memory-aware". For the Alpha Tech memory board, use the AT patches we published in TMQ issue IV.iii. If MEMORY doesn't show extra memory, the DOS doesn't know of it!

Incidentally, the AT patches referenced herein are not for Anitek's latest hypermem board which uses a means of port access different from the Alpha Tech board.

Fm Fred Oberding: I have a problem reclaiming banks 8, 9, or 10; although banks 0-7 will re-claim okay. After doing a "SPOOL *PR (NO)" using either of the top 3 banks I get the following messages:

if using bank 8:
"Memory space reclaimed"

if using bank 9:
"Memory space reclaimed"
"Bank 1 released"

if using bank 10:
"Memory space reclaimed"
"Bank 2 released"

"MEMORY" shows these banks still in use and they can not be used for other programs. On the lower banks the message indicates that the correct bank has been released and it has been indeed released.

I haven't used SPOOL for sometime as I have a hardware spooler, but I had some spare time to kill this AM and tried to verify another problem. Looks like I found a bug; whether or not it started with 6.3.1, I can't say for sure.

Fm David Huelsmann: Interesting, Fred. I don't normally use SPOOL with a bank higher than two so I haven't run across that problem at all. Obviously, the logic utilized to reserve and free-up banks higher than 7 is a little more convoluted and something slipped in the implementation. Since the only difference between using up to bank two and up to bank seven is whatever check was placed in the code to ensure that only up to bank two was allowed, the bug you described would have to be caused by the 6.3.1 implementation.

Fm MISOSYS, Inc: Yes, there was a change to allow the bank parameter to accept up to 30; there may have been a "message" problem caused by the bank number which I didn't look at. What I overlooked was that SPOOL "improperly" masked off the figure to free up the used bank when SPOOL was removed. I

worked up a patch for that correction. See FIX631E/JCL.

6.3.1 JCL bug? Nope!

Fm Alan Varga: Hi, Roy, I just received my 6.3.1 disk and I really like the new features. I've only had one problem so far. I wrote a JCL to update my system disks in drive 0 while leaving the upgrade disk in drive 1 (check your mailbox for UPGRAD.JCL). When the backup command completes I am returned to DOS instead of continuing with the rest of the JCL. Am I unintentionally misusing JCL?

Aside from that I'm quite pleased. I read a couple of suggestions for a proposed 6.4 for whenever that comes out. My wish list is short and, I hope, simple. Could TED be modified so that <CTRL><F> defaults to the filespec most recently loaded with <CTRL><L>? Also, a MOVE command would sometimes be helpful. It would have the same syntax and work the same as COPY, except it would delete the source file when the copy was complete. Another way to do that might be a (D) parameter on the COPY command. Currently I am using a JCL with the COPY and REMOVE commands to accomplish the same thing, but I thought a library command would be faster.

Thanks for your continuing interest in the Model 4. I think this is a great machine and have never regretted buying this instead of a DOS-compatible jobby.

Fm MISOSYS, Inc: I would not recommend using JCL to perform the upgrade from :1 to :0 as you are mixing system files during the operation. On a 2-drive system, it would be useful to create a system Memdisk as drive :0, then upgrade from :2 to :1.

Have you ever looked at IFC in the GO:CMD package? That has a tagged MOVE command. IFC is a very powerful and flexible file utility. Tagged MOVE, COPY, DELETE, and RENAME. The tagged RENAME with wildcarding makes

it very easy to rename a mass of files using a template.

Fm Dick Hollenbeck, El Paso, TX: Gentlemen, Just received your new release and wanted to thank you for your efforts in keeping the Model 4 alive until 2012. It'll outlive me if the biblical "three score and ten" is valid.

I have encountered a problem: I'm sure it's another example of "forest for trees" that I'm frequently guilty of - but please help!

BACKGROUND: I'm running a 128k, 5 meg Model 4. The hard drive has been partitioned with three drives for LS-DOS. I've used the Radio Shack HD initializer (cat # 26-0316) for this partition. It was initially set up for TRSDOS 6.2.0, and has been successfully up graded through LS-DOS 6.3.0. The CONFIG/SYS file generated by the HD initializer, in addition to configuring the system for three (0, 1, and 2) hard drives and two (3 and 4) floppy drives, also reserves about 1.5k of high memory and sets both the click and forms filters. On boot-up, after the system has been configured, a STARTUP/JCL has been AUTO'ed on the boot disk. This JCL installs three (5, 6, and 7) diskDISK drives, write-protects everything but the two floppy drives, installs a "screen-saver" filter, and finally installs PRO-WAM. I've had no trouble until I tried to up-date everything to LS-DOS 6.3.1

As in the past, the up-date was accomplished primarily with the BACKUP :X :Y (S,I,OLD) command. I've also had to SYSGEN the new boot disk, and then copy the CONFIG/SYS file over from the old to the new disk. Everything's played like a symphony - never a problem!

PROBLEM: After preparing the new (LS-DOS 6.3.1) boot disk as described above, and testing for proper IPL, the system hangs up on about the second or third line of the STARTUP/JCL with the cursor "frozen" immediately after the last "oh."

WHAT I'VE TRIED: 1. Ensured the validity of the JCL file. 2. Tried switching out both SYS6/SYS and SYS11/SYS files,

separately and together with their LS-DOS 6.3.0 counterpart. 3. Made numerous LS-DOS 6.3.1 backups on several "good" disks of appropriate files.

DISCUSSION: Drive 3 is the logical drive containing the boot disk on boot-up after the system has been configured for hard drive, and refers to the lower, floppy drive. I conclude from this that my problem lies with the JCL file (or its handler). In as much as the JCL is not compiled, I do not understand the error message "Open FCB." Admittedly I have a bunch of stuff crammed in high memory, but I don't think any of it is overwriting the FCB? The line of "Oh's" is also confusing, as is the freeze-up. When a JCL aborts, it returns to DOS? I'd appreciate any guidance or suggestions you may feel appropriate to my situation. Right now, I'm frustrated.

Fm MISOSYS, Inc: Dick, I think that you did not install 6.3.1 properly. To begin with, I checked all relevant SYS files on the 6.3.1 startup disk you supplied. BOOT, SYS0, SYS1, SYS2, and SYS11 all compare perfectly. Your JCL runs on my machine until it aborts when it cannot find the designated /DSK file.

You claim that you applied the upgrade as before with the command,

```
BACKUP :X :Y (S,I,OLD)
```

however, you neglected to mention what was drive :X and :Y! That's important. I have found some folks booting up 6.3.0, then backing up 6.3.1 from drive :1 to :0. That's not correct. That winds up mixing 6.3.1 system overlay files with the resident OS, which is 6.3.0. You also did that in testing, but you should never intermix DOS files from one release with another unless specifically told to do so. That's because system overlay files are implementation dependent on the resident SYS0/SYS code!

Something also peculiar about that startup disk was the configuration file. It has PRO-WAM in it. You can't SYSGEN properly with PRO-WAM. I wonder whether that CONFIG is in fact a legiti-

mate configuration or one SYSGEN'd after the JCL ran at one point. Is that a 6.3.1 generated file or the old 6.3.0 CONFIG/SYS which was copied over to a 6.3.1 disk? You really ought to rebuild your configuration from scratch. Start clean with no CONFIG, then install your HD driver and other filters of your choice. Then SYSGEN. Use the AUTO'd JCL to bring in PRO-WAM and DD, etc...

Backing up a new OS in :1 to an old OS in :0 is not the best procedure covered in the 6.3.0 Upgrade Documentation. It works **only** if your old CONFIG file has no DOS module (or fragment) configured which has changed address position between releases. Addendum 1A, page 6, covers the best procedure since it starts out the installation from scratch.

Another puzzling remark you made is "when the HD system is up and running - the JCL executes perfectly until it encounters a "device-in-use" situation; it then exits to DOS with an appropriate error message." Are you implying that the JCL runs if it is not invoked from an AUTO command? Or does that statement mean something else entirely?

I suggest you confirm just what specific commands you used to install 6.3.1; just where that CONFIG/SYS file came from; check to ensure that the hard drive system partition contains 6.3.1 /SYS files; and that the JCL does or does not work irrespective of being AUTO'd.

Fm Dick Hollenbeck: Roy, Thanks for your prompt reply to my "little" problem regarding updating my hard drive to LS-DOS 6.3.1. You suggested that the CONFIG/SYS file had become corrupted, and needless to say, you were right! The problem is now fixed. Thanks!

In the event some other folks have the same problem, I'd like to describe what I think happened. I have been using sloppy update procedures and getting by with them since TRSDOS 6.1.1 days.

When I received LS-DOS 6.3.1, I booted up the hard drive with the 6.3.0 boot disk to begin my update procedure. After the

configured system was established (with the 6.3.0 system files), the AUTO'd STARTUP/JCL ran, thereby installing diskDISK, a protective screen filter, and PRO-WAM. The important thing to remember here is that the system was running under 6.3.0.

I then placed a copy of my boot-disk in the upper floppy drive (logical drive #4 in my system), and a copy of LS-DOS 6.3.1 in the lower floppy drive (drive # 3), and entered the command:

```
BACKUP :3 :4 (S,I,OLD)
```

When the backup was complete, I SYSGEN'd the updated boot disk (but diskDISK, the screen filter, and PRO-WAM were all in the instant system). Consequently, the CONFIG/SYS file was bombed. I also suspect that there were some (major?) changes in the updated DO handler, which also contributed to the STARTUP/JCL's aborting.

The fix was easy (once I recognized the problem). Boot up in the floppy configuration with no special configuration present. Remove the AUTO'd STARTUP/JCL and then perform the update procedure - BACKUP :0 :1 (S,I,OLD) - on the boot disk. Reboot the hard drive with the updated boot disk, and do another backup (update) of all system files residing on the hard drive. Finally, reinstall the AUTO'd STARTUP/JCL, and reboot the system with the updated boot disk and updated hard drive resident system files. Worked like a champ!

OK, so I didn't rebuild the CONFIG/SYS file from scratch as you suggested. I'm not sure where I'd have to begin - suspect that I'd have to start completely over from scratch; ie: Reformat and partition the drive, then load it, etc.. When I first installed the hard drive with R/S's drivers, I sweat bullets getting it work like I wanted it to. I don't think I want to go through that mess again. However, if you've got a short cut to suggest, I'm more than willing to listen. I've nothing but respect for your suggestions, and am most appreciative of all the help you've given.

Fix for DIR - DIR1/FIX

Fm Fred Oberding: Roy, I don't know whether I fouled-up, or there is a problem with this patch. I couldn't quite figure out your JCL file, so I made two patch files. A 1-liner to boot/sys and the 4-liner to sys6/sys. The patch to sys6, record 10, byte 0C seems to be causing a problem with disk dates in the first half of the year. The month is changed to some special character, and the entry is wrapped around to a second line.

Fm MISOSYS, Inc: That's because I overlooked adding a patch to the routine which calculates the disk pack date. What was originally omitted from DIR1/FIX was

D07,D3=DC 04;F07,D3=76 2D

Seems like after I worked up the DIR1 patch, all my testing was on disks with a pack date from June through December which would, by coincidence, still display correctly. The patch listed in this issue is complete. Anyone still experiencing a problem with DIR incorrectly displaying disk pack dates from Jan-May, apply the above patch to SYS6/SYS.SYSTEM6.

FIX631D password

Fm Alan Varga: Roy, I am having trouble patching MEMDISK/DCT, and I'm wondering if you could offer any suggestions about what I'm doing wrong. I'm trying to patch a working copy of LS-DOS 6.3.1 but for some reason the file password isn't DRIVER, although my working copy should be an exact duplicate of my 6.3.1 master diskette. I've upgraded a couple of other disks from 6.3.0 to 6.3.1 and tried applying the patches to these as well, but with no better luck.

Fm MISOSYS, Inc: Seems like the password on MEMDISK/DCT is UTILITY,

not DRIVER - although it is supposed to be DRIVER. Adjust the PATCH command line accordingly. Now why didn't I pick that up? I suspect at the time that my system had global password matching disabled, although I usually do not disable that function.

I checked my procedure library and found that the JCL procedure which applies the passwords has "UTILITY" for both MEMDISK/DCT and FLOPPY/DCT. It's too late to change that now.

BOOT/SYS - What's it for?

Fm David J. Kelton, Richmond, VA: Roy, I just wanted to thank you for the best Christmas present I received this year. Your letter explaining how to fix BOOT/SYS file so the BACKUP command would work [see *TMQ IV.iii* pages 8-9] came on December 23 and I set about the task of fixing things up that very day. There was one small glitch. You indicated that I should fix the third byte of sector 1 of the BOOT/SYS file. When I examined it, it was correctly set to proper value. The third byte of sector zero, however, contained the value that BACKUP was putting in the DCT. When I changed that byte to X'4C' BACKUP worked correctly.

As I was backing up EVERYTHING, I started examining all the files on the disk to decide if I really wanted to keep it. At one point I SAVED a BASIC program to which I had added some comments. Lo and behold BOOT/SYS was again clobbered. This time one could see that the text comments I had added in the front of the program had been written on top of BOOT/SYS. (The first time I couldn't see anything that looked familiar.) I fixed BOOT/SYS again and continued on. All drives have now been reformatted and the system drive has been rebuilt with the contents of the original 5.1.3 and 5.3.0 disks. All seems to be OK now.

I was always under the impression that the BOOT/SYS file was really only important for disks used for booting. Now I know better! Would it be possible for you to expand on the importance of the BOOT/

SYS file as a TMQ article some time in the future? I don't like being at the mercy of my machine; I'd like to understand it as much as possible.

Again, thanks very much for your help and I hope you and the family had a happy holiday season.

Fm Carl Berger, Largo, FL: I have questions about BOOT/SYS: 1. What is the purpose of the BOOT/SYS file on a disk. Is it to help locate the directory? (especially a DATA disk). 2. Why is there no MOD date for BOOT/SYS in the directory? 3. Why does BACKUP/CMD abort when the BOOT/SYS file is corrupted? 4. What is causing my BOOT/SYS files to get corrupted with text, scripsit or BASIC program files? 5. How does text get into a password protected system file like BOOT/SYS? 6. Does LS-DOS put data at the end of files like BOOT/SYS when the disk is full? 7. Does this have anything to do with the fact that the BOOT/SYS file is larger on the system disk than on a data disk? 8. Will LS-DOS 6.3.1 correct this problem?

I thought this was only happening on my hard drive, but the other day my Mom's Scripsit system disk would not boot up her Model 4, so I tried to backup new system files to it and the backup aborted with the message: "Attempted to read system data record". After I looked at the BOOT/SYS file and found text in it I copied the BOOT/SYS from another LS-DOS 6.3.0 system disk and then backed up the scripsit files to a new copy of the system disk.

*One should always have
one's boots on, and be
ready to leave...*
Montaigne

Fm MISOSYS, Inc: Let's take a look at the history of BOOT/SYS in an attempt to understand it's uses, both past and present.

The word "BOOT" is derived from the common expression about "pulling one's self up by one's bootstraps", which essentially means self starting. Bringing up an operating system to its "Ready" state

Fixes to LS-DOS 6.3.1

Fm MISOSYS, Inc: Here's all of the fixes installed onto LS-DOS 6.3.1 disks to date. Note that the "level" letter advanced by one character for each fix installed. If, for instance, you see "Level 1D" when you boot your 6.3.1, you would need to apply patches starting from FIX631D/JCL. Note also that the level letter is stored in BOOT/SYS. If you are patching a disk (i.e. a hard drive system partition) formatted with 6.3.0 or earlier, the password on the BOOT/SYS file is "LSIDOS"; make any change to the fix as necessary.

```
. FIX631A/JCL - 03/08/90 - Cause FORMAT to
write all sectors of DIR cyl
. Apply via, DO FIX631A (D=d) where "d" is
drive to patch
//if -d
//. Must enter drive to patch!
//quit
//end
patch boot/sys.system6:#d#
(d02,1f=42:f02,1f=41)
patch format/cmd.utility:#d#
(d03,7f=21:f03,7f=32)
//exit
```

```
. FIX631B/JCL - 04/03/90 - Add missing code
to SETKI
. Apply via, DO FIX631B (D=d)
//if -d
//. Must enter drive to patch!
//quit
//end
patch boot/sys.system6:#d#
(d02,1f=43:f02,1f=42)
patch sys8/sys.system6:#d# setki1/fix
//exit
```

```
. SETKI1/FIX - Adds missing code to SETKI
. Use with FIX631B/JCL
LB3
X'2583'=96
X'258E'=3E 60 EF 50 59 79 B7 C9
. Eop
```

```
. FIX631C/JCL - 04/18/90 - Minor correction
to DIR
. Apply via, DO FIX631C (D=d)
//if -d
//. Must enter drive to patch!
//quit
//end
patch boot/sys.system6:#d#
(d02,1f=44:f02,1f=43)
patch sys6/sys.system6:#d# dir1/fix
//exit
```

```
. DIR1/FIX - 04/18/90 - Patch to LS-DOS
6.3.1 DIR command
. Corrects abort file with ext>4 and (O=N)
. Apply via DO FIX631C
D07,D3=DC 04;F07,D3=76 2D
D0B,64=DC 04;F0B,64=76 2D
D0B,9E=FD E5 CD 76 2D;F0B,9E=2B 3E 65 EF 7E
D0B,D1=C3 7C 2D 00;F0B,D1=FD CB 08 66
D10,0C=2B 3E 65 EF 7E C9 FD CB 08 66 FD E1
C3 4F 29
F10,0C="JanFebMarAprMay"
. Eop
```

```
. FIX631D/JCL - 04/30/90 - Minor correction
to DIR & Memdisk/DCT
. Corrects exit code for DIR; BOOT/SYS &
DIR/SYS passwords in Memdisk.
. Apply via, DO FIX631D (D=d)
//if -d
//. Must enter drive to patch!
//quit
//end
PATCH SYS6/SYS.SYSTEM6:#D# DIR2/FIX
PATCH MEMDISK/DCT.UTILITY:#D# (D04,40=F4
71:F04,40=F6 37)
PATCH MEMDISK/DCT.UTILITY:#D# (D04,60=F4
71:F04,60=F6 37)
PATCH BOOT/SYS.SYSTEM6:#D#
(D02,1F=45:F02,1F=44)
//exit
. DIR2/FIX - 04/27/90 - Corrects exit code
of DIR in 6.3.1 1D
. Apply via PATCH SYS6/SYS.SYSTEM6 DIR2
D08,B6=21 00 00 C8 3A 2F 26 0C B9 30 EB
F08,B6=67 6F C8 3A 2F 26 0C B9 D2 32 24
. Eop
```

```
. FIX631E/JCL - 04/30/90 - Corrects release
of banks > 7 in SPOOL
. Apply via, DO FIX631E (D=d)
//if -d
//. Must enter drive to patch!
//quit
//end
PATCH SYS8/SYS.SYSTEM6:#D# SPOOL1/FIX
PATCH BOOT/SYS.SYSTEM6:#D#
(D02,1F=46:F02,1F=45)
//exit
```

```
. SPOOL1/FIX - 04/30/90 - Corrects release
of banks > 7
. Apply to SYS8/SYS.SYSTEM6
D1F,A5=21;F1F,A5=CA
D20,1E=1F;F20,1E=07
D20,24=CD EF 29;F20,24=32 8C 2A
D21,82=69 26 00 11 8C 2A C5 06 02 3E 5F EF
C1 C9;F21,82="Spooler ahead"
D22,24="x freed ";F22,24=" released"
. Eop
```

requires a series of such steps. In the days of the very early microcomputers, there were no such things as inexpensive Read Only Memory devices (ROMs) or Programmable devices (PROMs). When you wanted to start up your computer, you generally used a series of front panel switches where you literally entered your code bit by bit.

Technology advanced to the point where the TRS-80 Model I Level II had three large ROMs totalling 12K of memory. Those ROMs contained primarily a BASIC interpreter, device drivers, and code to start up the machine - boot code.

Enter the floppy disk drive system. Since ROM memory was at a premium, it had little code to access the disk drive. In fact, it had the bare essentials to be able to read one sector of a floppy disk. The Model I started out as a single density system; tracks were numbered from 0-34 while sectors were numbered from 0-9 on a track. Typically the first sector on the first track is reserved for the "boot" sector; thus, the Model I ROM had code to read track 0 sector 0 into memory - specifically memory starting at 4200H. If it successfully read that sector, the ROM then continued CPU execution at 4200H. It was up to the 256 bytes contained in that sector to continue the process of bringing up the Disk Operating System. That code typically needed only a slightly more sophisticated floppy disk driver to continue reading the resident and startup part of the operating system which was stored in the SYS0/SYS file.

As the Model I TRSDOS blocked a disk into groups of sectors called "granules", five sectors were allocated to the Model I BOOT/SYS file - five being the size of the granule on that disk. So aside from the actual boot code, there were four additional sectors. They weren't used for too much. The second and third sectors had some hidden code which was used to originally display Randy Cook's copyright message given the simultaneous depression of three specific keys when RESET was pushed. The fourth and fifth sectors had identical copies of the first sector of the directory file, DIR/SYS, which is typically known as the Granule Allocation Table (GAT). These were never

used for anything; the hidden copyright sectors were likewise not needed for anything beyond their intended purpose. The only other significant piece of information contained in that BOOT/SYS file was a pointer to the track (cylinder) which contained the directory. Here's why.

Those were the early days of the 5.25" floppy disks; media was not guaranteed to be error-free. The DOS utilized a procedure during the format of a disk to verify the readability of each sector and to inhibit "bad" sectors from being used; this was known as the *lockout table* in the GAT. Since the directory had to be written to a track which contained no errors (i.e. no lockouts), its position was not fixed. The directory could be located on any track. It was necessary, therefore, for the DOS to know where the directory was located. Three schemes were used.

A little background is necessary. IBM designed the first floppy disk drive - the 8" floppy. Their scheme used a method of logically dividing the surface of a disk with concentric tracks and further subdividing each track into groups of bytes called sectors. The actual division was laid down during the formatting process. Each sector was prefaced with a header record, within which was information that identified the track, side, and sector numbers of the sector prefaced. There was also a code to designate the "type" of sector; this code was designated the "*data address mark*", or DAM. IBM format designated two types of sectors: one DAM indicated a *data* sector; the other a *deleted data* sector.

The floppy disk controller (FDC) chip used in the Model I machine supported four discrete data address marks, two of which were used by the DOS. One designated data sectors, while the other designated directory sectors. The DAM is encoded during the operation of writing a sector to the diskette; different FDC write commands tell the FDC to generate one of the four acceptable DAMs. This was one scheme used in the Model I DOS. The second was to maintain a data table in memory for each disk drive with one table entry being the track number of the directory. The other was to have a byte in the BOOT/SYS file contain the track number

of the track which stored the directory. Since the DIR/SYS file could be on any track, the DOS used the following method to read a directory sector: The table entry was used to identify the track number. If the FDC status following the sector read indicated the directory DAM with no read error, it was assumed to be correctly read. If it indicated a data DAM with no read error, the BOOT/SYS sector containing the directory pointer was read to update the in-memory disk table; then the sector read was re-attempted. This directory pointer was positioned as the third byte of the boot sector read by the ROM, which was sector 0 of track 0.

Technology advanced and Lobo Drives International began marketing a Model I expansion interface called the LX-80. This E/I had the unique ability of superimposing its own ROM during the boot process. Instead of the ROM's boot code reading sector 0, it read sector 1 of the boot track. This was probably to support CP/M which typically numbered sectors from 1-n on a track. But it also gave the ability for the same LDOS system disk to boot up on either a normal Model I with a Radio Shack E/I or one with a Lobo LX-80 E/I since the DOS used sector 0 for Radio Shack standards and sector 1 for the LX-80. To accomplish this, the hidden copyright code went. The bottom line is that as far as the Model I was concerned, BOOT/SYS contained just the startup code and the directory pointer. LDOS did use BOOT/SYS for additional data.

When Tandy introduced the Model III, they used a 1-n sector numbering scheme on a track; thus, the BOOT ROM read sector 1 of track 0. It read that sector into 4300H. The Model III also used a FDC chip which reduced the number of DAMs supported from four to two [see *Data Address Marks* by Roy Soltoff in *The LDOS Quarterly* July 1, 1981]. Unfortunately, the DAM used on the Model I for the directory sectors was NOT one of the two supported on the Model III. The LDOS implemented on the Model III was designed to be able to read both Model I and Model III LDOS disks. Logical Systems, Inc., bit the bullet when we decided to change the DAM convention used by our Model I LDOS to utilize the two DAMs supported commonly across both the Model

I FDC and Model III FDC chips. We took a lot of heat on that decision, but to this day, I firmly believe it was a correct decision. To be able to support both Model I and Model III diskettes, the directory pointer needed to be stored in both sector 0, for Model I, and sector 1, for Model III operation. As Model III LDOS was ported from the Model I version, many of the DOS functions and utilities which accessed the BOOT/SYS directory pointer actually still used the one in sector 0. Later, as new commands and functions were added in subsequent releases, the DOS shifted to use the pointer in sector 1; although for a time, both were used - which caused some headaches trying to read non-LDOS Model III disks.

LDOS is a very device independent DOS; it also supports a wide mixture of disk devices. You can freely intermix 40-track, 77-track (remember the 100 TPI drives?), 80-track, 5.25", 8", 3.5" drives in both single and dual headed configuration with the directory being anywhere in either single or double density configurations. LDOS has what is called "*automatic density recognition*" as well as automatic configuration of data disks. That's because certain configuration information is stored in the directory; disk's become *logged* when files are opened. In order to continue with this flexibility, it is important for certain DOS commands which do NOT open files to be able to log the disk - or get at the configuration information. For instance, when you issue a DEVICE command, you want to know the configuration of the current disk in a drive, not the last one logged. DEVICE has to log the current disk.

BACKUP also has to know the configuration of the current SOURCE disk, not the last one used in the drive, as well as the current DESTINATION disk to determine if a mirror-image backup can be performed or whether it has to perform a "reconstruct". BACKUP starts its logging operation by reading the BOOT sector to pick up the directory pointer; it doesn't rely on the in-memory disk table. So if the directory pointer is wrong, BACKUP will get wrong information. The problem issue is compounded when hard drives are introduced. A hard drive controller (HDC) supports only one data address mark; thus,

there are not two DAMs to differentiate the directory track sectors from normal data sectors. As the DOS has to utilize the dual convention, the hard disk driver emulates the dual DAM convention by returning the "directory read" status when the cylinder of a sector read matches the current value of the directory cylinder in the in-memory disk table! If the directory pointer in BOOT/SYS is incorrect, BACKUP will still think it is reading the directory because the in-memory table has been updated. But what is read is garbage! BACKUP actually doesn't update the system's table, but rather its own table; thus, only BACKUP is affected. Of course, it does impose a hardship if you cannot back up files from a hard disk.

LDOS also introduced additional uses for data storage in the BOOT/SYS file as previously mentioned. Sector 2 is used as a "system information sector". Data within that sector indicate, for instance, whether or not a CONFIG/SYS file has been SYSGENed; whether or not the date or time prompts should be issued on BOOT; and some default FORMAT parameters for floppy diskettes.

The architectural design of the Model 4 DOS incorporated more of an ability to be developed for multiple machines. The low-memory region was targeted specifically for I/O drivers which are inherently machine specific. All DOS implementation-specific functions not incorporated into the I/O driver zone were segregated to library C, stored in SYS8/SYS. The Model 4 BOOT/SYS file for a data diskette is virtually identical to a Model III data disk BOOT/SYS file. Differences exist for SYSTEM disks as the Model 4 system disk BOOT/SYS file contains not only the disk boot code but also the I/O drivers. The file is in "core-image" form, which means it is a pure binary file in contrast to standard /CMD files which are record-oriented load modules. The Model 4 BOOT/SYS file on a system disk takes up the entire track.

On a disk formatted by the Model 4 DOS, a data disk still has two boot code sectors; there are identical images in both sector 0 and 1 of track 0. True, all that code does when booted is to announce "DATA DISK". But they still contain two copies

of the directory pointer which should be of equal value. When a disk is converted to a system disk, by backing up an existing system disk to a data disk, the entire BOOT/SYS file is replaced except for the bits representing the "bootstrap step rate" (see the earlier discussion under LDOS topics), and the directory pointers if a reconstruct BACKUP is involved. The system disk boot code is also not identical across sectors 0 and 1, but the two sectors each still contain their own directory pointer - which should be identical.

So there, in more than a nutshell, is the evolution of BOOT/SYS and a summary of its utilization. Now let me respond specifically to the questions.

BOOT/SYS starts out without a date because it is typically not updated by users; it's alteration is strictly by operating system commands which must modify data contained within the BOOT/SYS file. The DOS accesses it by direct disk address, not as a file; thus no modification date is affected by the DOS as that's handled strictly by @CLOSE - a file facility. However, if you use PATCH to apply a fix to BOOT/SYS, it will have a modification date.

As far as understanding what causes BOOT/SYS files to be corrupted, it is almost beyond the scope of this publication since it is typically caused by programs gone haywire. But let me provide a little incite.

The DOS stands between a program accessing files by record and the disk which is accessed by drive, tracks, heads, and sectors. For every file access, the DOS has to translate a request to read or write a specific file record into the exact disk address. Before a program can access any record of a file, the file must be "opened". This is just a procedure utilized to have the DOS search available directories for a given file specification. Once the DOS finds the directory entry, it reads information on the physical disk storage occupied by the file and stores that information into a File Control Block (FCB) maintained by the program. The FCB can hold information on at most four extents [see the *Ext* column after invoking a DIR command]. For every read or write request, the DOS references the disk address in-

formation stored in that FCB. The only link between an actual file record and the actual disk address is that FCB; the DOS has to go back to the directory for more information only if it needs additional data on an extent not already maintained in the FCB. But information on what drive and what directory record is associated with the FCB is contained in the FCB. Alter its contents, and you will read or write a totally different sector. For instance, the difference between drive 1 and drive 0 is one bit. If the FCB entry indicating the drive had a one-bit change, instead of writing to drive 1, drive 0 would be written to. Alter the FCB information which indicates the granules allocated to a file, and you will read or write somewhere totally different from the expected.

Programs may also have bugs. DOS service functions are executed by a RST 28H Z80 instruction which has a machine code of 0EFH. If a program "crashed" and jumped to a memory location which contained that binary value, a DOS service would be requested. The service function for writing a physical disk sector is 53 (decimal). If the Z80 accumulator happened to contain that value at that point in time, then it is possible that a disk sector would be over-written if the values contained in the remaining Z80 registers were legal for a mounted disk drive (i.e. C contains a drive number; D contains a cylinder number; E contains a sector number). Assuming that happened, the 256 characters stored in memory pointed to by the contents of register pair HL would be written. If you were using a text editor, and registers D and C contained 0 with E containing 1, the BOOT/SYS directory pointer would get clobbered with, presumably, text! There are other possibilities of errors caused by inadvertent requests for DOS services stemming from crashed or buggy programs. The DOS doesn't automatically start overwriting BOOT/SYS.

Another possibility for crunching BOOT/SYS is from an error in the directory. I previously mentioned the Granule Allocation Table (GAT). This directory sector stores a bit field of what disk granules are in use, and consequently, what granules are free. If, by chance, the first byte of that

sector got set to zero (the first byte corresponds to the usage of cylinder 0), then the DOS would think the BOOT track was free to use. A request for disk storage space stemming from writing a new, or lengthened, disk file, could cause the BOOT/SYS file to be overwritten strictly because the GAT told the DOS that cylinder 0 was NOT allocated.

Can the DOS be made smarter? I think not within the resources of the machine environment. Things happen; that's why we have backups.

Model 4 Programs

EnhComp BASIC compiler

Fm Ralf Folkerts: Roy, during writing the MCS-48 disassembler using Pro-EnhComp 2.6 I found a little 'bug' (it's not a real bug, but a little inconvenience).

When using BC with the PRT - option it will not react correctly when a printer timeout occurs. I've used the Syntax 'BC DIS48,,,PRT' to get a print-out with possible error codes. After the first page I recognized that the FORMS filter was set up incorrect. Since my printer has a large buffer I switched it off-line let Pro-EnhComp abort with a 'device times-out' error. I've pressed <BREAK>, too. What I got was a 'Compiler aborted' error. But in strange way: Each character was printed after some seconds pause. The pause must be the time between the time-outs.

As I already wrote: I can live with this without any problems; doesn't happen to often that I turn my prt offline while printing a Pro-EnhComp compilation. However: Maybe with a new release (will there ever be one ??) you may fix this!

Fm MISOSYS, Inc: It most likely does not test for errors coming back from the *PR device. You are a prime candidate

for the "SYSTEM (PRTIME=OFF)" command of 6.3.1.

Fm John Grindey, Derbyshire, ENGLAND: Roy, Small problem with PRO-EnhComp created programmes. I wrote a programme to split a long text file into smaller segments. The disk I used to write these to hadn't got enough free space to hold all the data. Instead of giving a Disk Full Error it carried on trying to write to the disk; the only indication of a problem was that the programme slowed down. I think the problem is in the code below which was disassembled from the programme, and looks as if it comes from SUPPORT/DAT record 0F.

(code deleted for publication)

I know from reading T.M.Q. that you are very busy so I will keep reading to see when it is worth sending my disk in for a refresh.

Fm MISOSYS, Inc: John, The problem you "discovered" in PRO-EnhComp originates in exactly the routine you reported. That's a routine common to all character (byte-at-a-time) device output. I suppose that in the Model III version, which is what the Model 4 version is based on, device output was primarily used for screen or printer which had no error return codes. Of course PRINT# output also goes through that routine!

In any event, its something you will have to live with, for now. It's impossible to come up with a patch; changing the routine to enable error code response detection could only be considered in a future revision of the compiler, if implemented. That's not on my near-time frame schedule.

Fm Carl Berger, Largo, FL: I downloaded a program named MOVE/CMD from the 8N1 BBS in Gainesville, Florida (904-477-1200) to use when I wanted to

move a file from one drive to another in one step: eg. MOVE FILENAME:O :1 causes the file to be copied from drive 0 to drive 1 then removed from drive 0. All went well until the destination drive got full and instead of the program terminating on the error, it just kept going and deleted the file that I was trying to move. Of course I restored the deleted file and moved it to a drive with free space and directory slots, but I decided to write my own move program in BASIC with an ON ERROR GOTO statement that would loop back to the beginning of the program after displaying the appropriate error message instead of deleting my file. I also wanted it to display helps such as DIR FILENAME or DEVICE to show enabled drives or FREE to show where the free space is located. I wrote a simple program in BASIC and it does what I want it to do when running from BASIC, but I also wanted it to run from DOS so I compiled it with Pro-EnhComp.

My problem is that Pro-EnhComp 2.6a 1 Mar 88 (Model4 mode, LS-DOS 6.3) does not generate an ERR code on a DOS error (at least not the way I'm using it). For example, if I compile the following program, I would expect the program to loop back to the beginning on ANY error:

```
10 'TEST/BAS
20 ON ERROR GOTO 900
30 C$="COPY TEMP/TXT:0 :9"
40 SYSTEM C$
50 R$="REMOVE TEMP/TXT:0"
60 SYSTEM R$
70 GOTO 20
900 PRINT"ERROR";ERR;ERL
910 RESUME 10
```

After the DOS Error message the program removes the file.

Is there a reason that ERR=0 after this type of error? Is it because I am using "SYSTEM" commands? BASIC generates an error code for DOS errors, why not the same in the compiled program?

Fm MISOSYS, Inc: EnhComp doesn't "trap" external errors (i.e. DOS error status from commands invoked via "SYSTEM").

DoubleDuty Partitions

Fm John G. Gelesh: I believe I have found a bug in the MEMORY command of LS-DOS 6.3.1. On page 2 of the memory listing, the heading prints a second time and the first module listed prints over it. This is a minor bug, and probably has a simple fix.

Also, how about a clarification on Double Duty. If DDUTY is invoked with the (M) parameter, Partition 3 IS locked out. Isn't it?

Fm MISOSYS, Inc: I'll look into the "MEMORY command module problem".

DDUTY will inhibit the use of partition 3 if invoked with the (M) parameter. It also will not reserve any memory for that partition. Also, if you are already using high memory, say for PRO-WAM with more than 128K, then you may as well not lock out partition 3 as it won't use any additional memory. But with PRO-WAM's LIB command, you won't need DDUTY's. But with other high-memory modules, you may want it.

Fm John G. Gelesh: Yes, I see what you mean by DDUTY not reserving high memory when PROWAM is installed. I am glad you pointed that out. I did not notice it. Before getting off the subject of the MEMORY command, did you notice the high-memory module (%ME) of DDUTY does not line up with the rest of the modules? It prints one space too short.

One other cosmetic item: DDUTY goes to LS-DOS Ready in Partitions 1 and 2. But goes to TRSDOS Ready in Partition 3. I have been trying different low-memory allocations with DDUTY and I think I have a bug in Partition 3. More on this later after I have done more research.

Fm MISOSYS, Inc: Methinks you are not using the MISOSYS-released version. The "Tandy" version did have a slight

cosmetic bug in the %MEM module header (not %ME) due to an extra NULL byte (hex 00) being in the header. I found that and fixed that. I am positive I changed all instances of TRSDOS to LS-DOS. Please ensure that you are using my version, 2.5.1. Each partition states LS-DOS Ready!

Fm John G. Gelesh: Roy, you're right about my having an old version of DDUTY I purchased it several years ago from a RS store in/near Pittsburgh, for a song. I hadn't realized you updated DDUTY when you acquired it.

Anyway here is the problem I have with my version (2.5). When I take a DIR of my largest hard drive partition (200+ files) using Partition 3 of DDUTY, I get about a page of UN-sorted directory and then a HIT error. Operating under Partitions 1 and 2 I get the full sorted directory with no errors. Try this with your version.

Do you plan to do any upgrades to DDUTY? How would I go about obtaining an upgrade?

Fm MISOSYS, Inc: I updated it briefly. There were a few small space reductions performed. I changed the TRSDOS Ready to LS-DOS Ready. I fixed the one bug which I found with the %MEM module name. I was unaware of any other bug.

I just tried what you reported - did a DIR from part 3 and I did get a directory read error. I'll investigate that soon. The "unsorted" directory is forced by DIR operating as a overlay requested through CMNDR. That forces it unsorted as no sort scratch space is available. Incidentally, I had 224 files on the drive.

I am contemplating upgrading DDUTY to allow selectable bank installation and a 2nd version using video line transfer for screen swapping so that it could work with AT-patched @BANK systems.

time passes...

Fm Roy Soltoff: The problem you had doing a DIR in partition 3 of DoubleDuty

had nothing to do with DDUTY; it was caused by a bug I introduced into 631. The problem reared if you invoked an unsorted directory and the drive had a file with more than 4 extents. See FIX631C/JCL.

DoubleDuty and Extra Memory

Fm Danny C. Mullen, Ft Polk, LA: Roy, You reference the Alpha Tech memory board patches in DISKNOTES 6. After installing the patches, then using PROWAM, I get garbage mixed in with the CAL window, but only with CAL - the others (ie. CARDX, ADDRESS, etc.) are as normal. This does not happen with unpatched DOS. I have PROWAM ver 2.01b.

Also, get garbage on the 3 screens of DOUBLE DUTY if using the patched version of DOS. Even after clearing each screen, the garbage will reappear on partitions 1 & 2. After clearing partition 3, it will stay normal.

I have tried the patches with LS-DOS 6.3 & 6.3.1 with the same results. Are there any later fixes which may not have been included with the disk? This is the first chance I've had to try the DISKNOTES 6 disk since purchase.

By the way, I am able to load DOUBLE DUTY and PROWAM using the patched DOS, but with the same results as described above.

Fm MISOSYS, Inc: The patches in DN6 were correct for 6.3.0. You might try altering the sequence of the PRO-WAM applications assigned to see if your problem follows CAL or its location in memory. Also, do you get the same result installing PRO-WAM into banks 1 or 2?

Fm Danny C. Mullen, Ft Polk, LA: Roy, I Am sending you this disk to see if you can look into the garbage problem that appears on Partition 1 & 2 & sometimes 3

when changing between screens under Double Duty. This is a virgin copy of LS-DOS 6.3 that was patched using the patches from DISKNOTES 6 for the Alpha Tech board. I only used the three patches that applied to 6.3 (BOOTAT/FIX, SYS0AT63/FIX, & SYS1AT/FIX). There are NO other patches installed in DOS. Excess files have been purged.

When garbage appears, you can clear the screen with the <SH><CLR><F> key combination, but if you leave that screen, then return, the garbage reappears. Also, you lose the cursor when in partition 3 - but garbage, if present on the first entry to part 3, when cleared, doesn't return; otherwise, all partitions appear to work as normal.

PROWAM appears to work OK on 6.3 with these patches, but with 6.3.1, did crazy things with CAL/APP as mentioned on LDOS Forum recently. Will there be any fix for this version?

Your BANKER/CMD from DISKNOTES 6 appears to show incorrect count of banks available on my machine (512K); when running it, it should show 16 banks, but instead shows: 32K Banks Available = 31/31, In use=<----->. It does show which banks are occupied if I put Double Duty or PROWAM in by using the '+' in the correct spots...but, if used after installing PROWAM, it may show 29/29 or 25/26. I don't really care about this, but do want to be able to run PROWAM & DOUBLE DUTY together without garbage on screen.

If I try to install PROWAM into any bank higher than 14, garbage and/or debug will come up and lock machine. Reset to get out. Seems that banks up and including 15 should be available, but, am I mis-counting the offset from 0?

Are you considering making source code for 6.3.1 available so as to update us owners of 'THE SOURCE' with latest changes? How much \$ if so?

Fm MISOSYS, Inc: Concerning the Alpha Tech memory board patches and DoubleDuty, I have both good news and bad

news. The good news is that I have found out why DoubleDuty does not work properly with DOS 6.x patched with the Alpha Tech memory board patches which I developed; I have one machine with that board in place and can duplicate the problem. Of course, the only problem I have been able to reproduce is that the screen image is not properly restored when DoubleDuty switches partitions. It's not the fault of DoubleDuty; it is the limitation of the bank switching hardware in the AlphaTech board and the manner in which the bank switching software was implemented to overcome the hardware limitation. Additional detail is needed to provide an understanding of the problem.

A stock Model 4 with 128K of memory can image any of three distinct 32K-banks of memory into the address space extending from 08000H-0FFFFH of the Z80. Coincident with this memory imaging model, the hardware can simultaneously alter the imaging of the highest 2K to that of the video RAM; this is regardless of which of the three 32K banks are currently imaged. The video RAM can be thought of as a shadow of RAM beneath the address range 0F800H-0FFFFH and can be brought to the surface whenever it is requested by means of a hardware port-controlled switching arrangement. Thus, one can transfer to or from video memory from the remaining 29K of bank-switched memory (another 1K of the address space from 0F700-0F7FF is used to image the memory-mapped keyboard switched in whenever video RAM is switched in).

DoubleDuty uses the @VDCTL supervisor call to move the video images from the selected partition to the video RAM. But the video images are saved in bank 2 (the 2.5 release uses origins of 854FH, 8F8AH, and 99C5H for the images of partitions 1, 2, and 3 respectively). This works properly in a stock Model 4 and is a correct programming method for the DOS service call. The memory management facility of the DOS provides for the correct behavior of device, disk, and interrupt handling irrespective of the memory bank imaged at the time of the DOS SVC request.

Let me now turn to the Alpha Tech memory board, referenced now as "AT". That AT

board was designed to use port 43H (RAMPORT) to control its 32K memory imaging. Recognize, for the moment, that the Alpha Tech modification replaces the existing 64K RAM chips with 256K RAM chips; thus, banks 0, 1, and 2 represent portions of memory in those 256K RAM chips. The AT board's port control used a decimal value of 0-30 to select any one of 31 distinct 32K memory blocks as imaged into the upper 32K address space. Furthermore, if a 0-value was selected, the memory management hardware of the Model 4 was effective for switching banks 0, 1, and 2 normally. This meant that any program which twiddled the Model 4 bank switching directly would continue to work even if an AT board was installed. On the other hand, if any other value was selected in port 43H, the Model 4's memory management hardware - **including the video RAM imaging** - would be ineffective!

What is happening with DoubleDuty is that the code to swap partitions is executing from bank 2 resident in the upper half address space. The partitions' "saved" video images are likewise in that bank as noted above. So next one has to examine how the DOS, patched by the AT patches, handles the enabling of video RAM.

Since video RAM cannot be imaged at 0F800H unless the Model 4 memory management hardware is effective, the code in the patched ENADISDORAM must output a zero value to the RAMPORT, then switch in the video RAM image. Once the transfer is made, the follow-up routine restores RAMPORT to its previous value. This action also will "temporarily switch from whatever bank is resident to bank 0 during the duration of the transfer to/from video RAM! Obviously what is happening is that the video screen RAM is not being updated from what is saved in bank 2 but from whatever is in bank 0. In fact, bank 0 will be overwritten when partitions are swapped by the action of saving the current screen.

You can observe this entire action by installing PRO-WAM then bringing up the MED application (part of MisterED). With that you can see the original video screen contents and see the screen being updated from the address ranges noted

above but from bank 0.

Knowing all this does you no good as the hardware memory management of the AT board prevents it from simultaneously displaying the video RAM while the resident bank is something other than bank 0. One possible solution would be to reprogram DoubleDuty to double buffer the video screen transfer (and other @VDCTL calls it does for cursor positioning, etc.). That would certainly slow down the swapping somewhat, and lengthen the code space needed. Unfortunately, no @VDCTL call can be performed when the AT board is switched to other than bank 0; thus, the %SWP driver in low memory would have to be longer to accommodate the code needed to handle all of the @VDCTL calls. I am not going to pursue that solution at this time.

time passes...

Fm MISOSYS, Inc: Danny, Here's some good news for you. Subsequent to my previous correspondence, I decided to rework DoubleDuty so that it could handle partitioning 128K even on a machine equipped with the Alpha Technology memory board and patched with my AT patches. It required more effort than I had expected because there was zero space available in the SWAPPER module of DDUTY. I had to work hard to compress some of the code so that I could squeeze in code to handle double buffering of the video screen swap using line transfer. I also had to rework the routine which snapshots the keyboard speed buffer stored in video RAM starting with DOS 6.2.0. While I was at it, I decided to also add a Bank parameter which accepts any number from 1-29 as the secondary 64K base bank; this version will use base plus base+1.

I have done some preliminary tests on both a 1 Meg AT Model 4 and a 384 K XLR8er'd Model 4. Tests were done with DDUTY installed in banks > 2 and even with PRO-WAM installed in a higher bank with the DOS running from a RAMDISK. Actually kind of interesting.

I do not experience any problem with CAL - or any other PRO-WAM application - under 6.3.1; this was even tried on the AT-patched machine using the new

DDUTY and PRO-WAM both in "extended" memory.

If BANKER is showing the wrong number of banks installed, it only means that the installation of SYS0 into memory - which has a routine patched by the SYS0AT/FIX which checks for RAM installed - detects 1 Meg of RAM. No one has previously reported that to me. I never tested the patches on a 512K Alpha Tech board; I have one working AT board and you are aware that 1 Meg is achieved by soldering two banks of memory chips atop each of the first 512K of chips - I will not alter that arrangement just for testing. My suspicion is that the board may just image the first physical 512K for both RAMPORT selections from 1-14 and for 15-30. If such is the case, software would be unable to detect whether 512K or 1024K of memory is installed. It would be wise to manually alter the Bank Available Ram (BAR) field to reflect only the 512K. The AT patches use four bytes from 414H to 417H; low order bit of low order byte is associated with bank 0 whereas the 2nd high order bit of the high order byte is associated with bank 30. Note that a one-meg machine uses the first 32K as non-switched RAM from 0-7FFFH; thus, there are only 31 switchable banks of 32K out of the remaining memory. These are designated 0-30. With a 512K machine, you have but 15 switchable banks designated 0-14. This means that the BAR values should be FF 7F 00 00, for you. You could change those with the MEMORY command.

Finally, I'm not making SOURCE to 6.3.1 available. I didn't do that with 6.3.0. I still have hundreds of *THE SOURCE* to TRSDOS 6.2, the three-volume set.

Fm Danny C. Mullen, Ft Polk, LA: Dear Roy, I wish to thank you for taking the time to do this. This will really allow me to 'max-out' while using ALDS, DDUTY, and now PRO-WAM at the same time. (Maybe I see TRIPLE/QUAD DUTY in the future?)

I was thinking about how to write some code to bypass @BANK when I finished a current project (a small thesaurus) for

PD, but really don't have the experience to tackle such. Have previously disassembled the SUPERDRIVE ramdisk emulator included with AT board to see how the author got around the switching problem you mentioned in last letter, but about the only thing I'm sure I discovered was an undocumented parameter to put its driver in high memory, and a spelling error in its installation message

GO:MTC and MAPPER

Fm Robert C. Scully, Oaklyn, NJ: Roy, Need some clarification on mapper as in "GO:MTC". As you can see from exhibit A, the program does not pickup all files as listed in the directory. However. PSS6/CMD and PMAP6/CMD both have no problem in displaying the files properly. Any chance of a fix?

Similarly, when using DIRCHECK/CMD, also in "GO:MTC", it indicates many GAT errors. Doing Dircheck :0(H) is when the false reporting of GAT errors occur. If, doing Dircheck :0(L), works fine as does pcheck6/cmd. Any answers?

Fm MISOSYS, Inc: MAPPER, as put on GO:SYS, was missing the PMAP01/FIX published in *NOTES FROM MISOSYS*, issue IV, page 4-56, December 1984. See the next letter for more info. Also, see TMQ IV.ii page 55 about DIRCHECK.

Fm Charles A. Ainsworth, Woodbridge, VA: Dear Roy, Now for a problem with MAPPER/CMD. I have used it several years together with ALLOC/CMD as I am very fond of centering my files around the directory of a logical drive to minimize head travel, especially in those so frequent searches for SYS/SYS files. Until I changed to my six-drive partitioning, that worked fine. But when the six-drive arrangement appeared on the scene and I used MAPPER, things seemed to go a bit wild. I am sending you several attachments to illustrate this: #1, a MAPPER run of mr drive zero. Note that things

are a bit crazy; some of the files are missing and the heading shows 150 cylinders. #2, the JCL I used to set up the drives (I have removed the "6" from your file names). #3 a printout from FREE; #4 a directory of drive zero; #5 a printout of FREE :0; #6, a printout of DEVICE; #7 a printout from DCT/CMD of drives 0/5; I haven't figured all these out and may not have occasion to do so and it doesn't bother me much as the system works well, but I can't figure out how or why the sheets for drives 2 and 5 show starting head as 9 (maybe due to something similar to floppies where logical number may be different from physical number?); #8, a printout, for comparison, from MAPPER run on a double-sided 80-track floppy; and #9, a printout from CALC/CMD. Nuff info? (grin).

One thing that puzzles me is that, although the default of FIXED has been used in the MSCSI setup in my JCL, both DEVICE and DCT show them as removable.

Fm MISOSYS, Inc: Charles, you were having a problem with MAPPER - the version which came from PRO-MACH2, I assume. You did mention ALLOC, etc., so I am suspecting that the version of MAPPER was from that package.

There was a patch published in *NOTES FROM MISOSYS*, Issue IV published in December 1984. That had a PMAP01/FIX which corrected the very problem you reported. Perhaps your version did not have that fix. I am enclosing a copy of that page.

On the other hand, I do recollect you purchased some of the Golden Oldies products. A version of MAPPER was also provided in GO:MTC. However, I just recently discovered - via another bug report similar to yours - that the version of MAPPER put onto the GO:MTC disk was not the patched version, but a version previous to the original MAPPER supplied on MACH2. Thus, the MAPPER used in GO:MTC re-introduced the very bug fixed back in 1984! That takes a slightly different patch. In the off chance that you were using this version, I'm also

sending a copy of that patch. As it turns out, applying the GO:MTC MAPPER patch makes the patched version byte-for-byte identical to the patched version used in MACH2 since 1984. Hope that wasn't too confusing.

```
. GMAP01/FIX - 03/22/90 -
Patch to GO:MTC MAPPER/CMD
. This fix corrects MAPPER
for hard drives configured
to multiple heads
. without using the DOUBLE
bit. Similar to PMAP01/FIX
from 12/84
. Apply via, PATCH MAPPER
GMAP01/FIX
D01,17=18;F01,17=24
D01,21=07 07 07
3C;F01,21=CB 3F CB 3F
D01,25=32 91 36 FD CB 04 6E
28 13 18 0A FD CB 04 6E 28
F01,25=CB 3F CB 3F CB 3F 32
91 36 FD CB 04 6E C8 3A 8B
D01,35=0B 21 91 36 34 3A 8D
36 87 32 8D 36 3A 8D
F01,35=36 CB 27 32 8B 36 C9
FD CB 04 6E C8 21 91
D01,44=4F;F01,44=34
D01,46=8E;F01,46=8D
D01,48=47 AF 80 0D 20 FC B7
20 01 3D
F01,48=CB 27 32 8D 36 3A 8C
36 CB 27
. end of patch
```

MS-DOS Topics

TRSCROSS for MS-DOS

Fm Peter P. Klemchuk, Yorktown Heights, NY: Gentlemen: I am writing to ask if you can advise me how to move data from time to time from Profile 4 Plus, an excellent data base system, on a Radio Shack Model 4P computer to a database on an IBM compatible computer operating under DOS 3.3. Prior to sending the data I set them up to print and direct the printed record to the RS232 port. The

signal is transmitted through a null modem to the RS232 port on the IBM compatible computer. As the data come to the IBM compatible a line may be received correctly but as the cursor moves to type the next line it erases the previous line as it goes along; and the cursor laying down the third line erases the second one. From time to time a line may remain as typed but only infrequently. As a result a fragmented record is received. It almost seems as if an end of line signal needs to be sent but isn't being transmitted.

Another problem is that the data file being transmitted is longer than 80 characters and Terminal operating under Microsoft Windows is limited to 80 characters. Do you have any suggestions as to how I might transmit a data file with as many as 130 characters per line when printed?

A third problem relates to transmitting documents prepared with Superscript under TRSDOS to the IBM compatible operating under DOS 3.3. I convert the documents to ASCII and they are fragmented when received at the DOS computer.

I would appreciate any suggestions you can provide to help me solve these problems. Thank you for your attention to this inquiry.

Fm MISOSYS, Inc: Peter, The problem you run into with the screen line overwrite is caused by the end-of-line convention employed on each machine. MS-DOS uses a CR-LF convention whereas TRS-80 uses just a CR. ASCII data which is transferred to MS-DOS from a TRS-80 needs to be massaged to translate CR to CR-LF. You may try to install the FORMS filter to the *CL device then specify FORMS (ADDLF). This will add a line feed on output following every RETURN.

I cannot suggest an easy way to handle lines longer than 80 characters using what you have at your disposal. You really need to employ a communications program which has a file transfer protocol.

Finally, I cannot speculate as to why your Superscript documents are fragmented.

Perhaps its because of the end-of-line convention. You will typically have a RETURN only at the end of a paragraph in a Superscript document - even when converted to ASCII; thus, you may be truncating paragraphs which exceed 80 characters.

A solution to all of these problems may be easily be obtained. By using our TRSCROSS product (\$89.95 + \$2 S&H), you can directly read your TRS-80 diskettes on your PC. TRSCROSS runs under MS-DOS and reads, writes, and formats TRS-80 media. It directly translates BASIC programs. It converts Superscript document files directly to ASCII or to Document Content Architecture: Revisable Form Text (DCA:RFT) which can be imported directly by some word processors thereby retaining most of the document formatting. TRSCROSS converts ASCII files and makes the end of line correction. Although it cannot directly convert Profile data files, you can generate a Profile report to a file then use TRSCROSS on that file.

Fm Ralph A. Marino, Pastor, Queensbury, NY: Dear Friends, The Queensbury United Methodist church has bought a copy of your TRSCROSS for use between a Model 4 in Model 3 mode and a 1000 TL/2 with hard drive.

I attempted to use your program today. Some files were successfully transferred but several disks which were working fine between two model 4's could not be read by TRSCROSS. When I tried to use the format option the computer hung up and could only be released with the reset button. It claimed to be formatting cylinder 0 and just sat there with the cursor blinking. <ESC> had no effect.

The problem of unreadable disks is not too serious but your technical support in overcoming the format problem is requested.

Fm MISOSYS, Inc: Pastor Marino, I would need a little more information before I can begin to think about why you are

experiencing a problem with TRSCROSS.

To begin with, the Tandy 1000 TL/2 is normally equipped with a single 3.5" floppy drive. Obviously you also have a 5.25" drive connected. What kind of drive is it, 360K or 1.2Meg? What drive letter is this? B:? D:? I have heard of problems where some Tandy machines are equipped with two 3.5" drives and a third 5.25" drive identified as D:. TRSCROSS does not seem to work in that environment when the 5.25" drive is neither the A: or B: drive. What is yours configured for? Since you probably boot off of the hard drive (most likely C:), you should be able to change your configuration so that the 5.25" floppy is the A: drive. Can you do this? If so, please try to utilize that arrangement; repeat your job tasks, then report any difficulty.

You don't state what DOS was used on the TRS-80 diskettes; there were about a half a dozen different DOS vendors: TRSDOS 1.3, LDOS 5.1/5.3, DOSPLUS 3.x, MultiDOS, NEWDOS80, etc. You state that some disks were readable and some files were transferred successfully. What is different about the other disks? Were the disks readable but the files were not transferred properly? If readable, what types of files were they and what did you specify to be the mode in each case (ASCII, BAS, BIN, etc...) and what was wrong with the transferred file? If not readable, what did TRSCROSS report? You see, you provide little information to understand the nature of the problem.

Please provide a clear and complete description of the exact nature of your problem and surrounding circumstances. That is needed before any additional research can be performed.

Fm Andrew J. Borson, Broomall, PA: Dear folks: I have a Radio Shack Model III, and recently also purchased a Tandon AT-clone. I purchased your TRSCROSS to enable me to transfer ASCII files and some basic programs between the two computers.

The most important function, that of transferring from the Model III to the AT,

works fine. However, I have been completely unable to transfer from the AT to the Model III.

I have tried this using disks formatted by the Model III, and using disks formatted by TRSCROSS (and initialized as 360K disks). TRSCROSS seems to have put the file onto the disk, but attempts to read it from the disk consistently fail, and in fact, lead the disk drives (on both machines) to make some very unusual noises.

Is there anything I can do to enable this type of transfer to take place? It would be very helpful if there was some way that I could transfer back from the AT to the Model III.

Fm MISOSYS, Inc: Dr. Borson, Your letter alludes to a problem in transferring data back to your Model III. You don't state which Model III DOS you are using, but you do reference formatting a "TRS-80-type" disk on the PC as 360K. That implies 2-sided operation. Unless you have modified your Model III, you do not have 2-sided disk drives. If you are using TRSDOS 1.3 on the Model III, then you should specify format type 2 (see page 16 of the TRSCROSS documentation). If you are using LDOS and have single sided disks, specify format type 1. You should specify 40 cylinders and sides=1, unless you have 2-sided drives. I cannot understand, other than this specification error, why you would be having trouble.

You do specify in your letter that you are using an AT machine. That generally implies a 1.2Meg hi-capacity drive. Sometimes you may not be able to format a low-density disk on that kind of drive which would be readable on a low-density disk drive. If that is your problem, you may need to try to utilize a low-density (360K) disk drive on your AT. But without further details, I can only be vague.

If you continue to have problems, please write to MISOSYS with a more complete report.

MISOSYS, Inc.

MISOSYS sponsors a forum on CompuServe: PCS49



When you don't have to write in stone, don't let your editor weigh you down. You need SAID-86!
Editing was never so easy!

SAID-86 is a fast, flexible, full screen text editor for PC's. It is perfect for editing batch files, program listings, README files, CONFIG.SYS files, and anything you now do with EDLIN or the non-document mode of a word processor. Why struggle with huge editors; when all is said and done, SAID-86 will be your text editor of choice!

Check out this list of features

- ✓ WordStar-like editing commands are easy to use
- ✓ Pull-down menu system for commanding SAID-86
- ✓ Supports nine editing buffers with automatic swap to disk
- ✓ Supports up to 30 user-defined macros; 255 characters each
- ✓ Undelete the last nine deleted lines can save your bacon
- ✓ MOUSE support with automatic recognition
- ✓ HELP facility; shell to invoke DOS commands from SAID-86
- ✓ SAID-86 can expand or contract TABs

SAID-86 is reasonably priced at just \$29.95 + \$2S&H

Colorado Memory Systems 40 Megabyte tape drives

DJ10 Jumbo	\$295 +\$5S&H
AB10 Tape Adaptor	\$75 +\$3S&H
KE10 External Adaptor	\$110+\$5S&H
DC2000 tape	\$22.50

MISOSYS, Inc.
P.O. Box 239
Sterling, VA 22170-0239
800-MISOSYS
(US&Canada)
or 703-450-4181
M/C & VISA accepted.
S&H are U.S. only.

Now get the clearest view of your disk drive and memory contents with our DED86™



Powerful features in Version 2!

When you need to travel through your disk drive, why settle for a tool that isolates you from the road? DED86 gives you the direct controls you need to explore your disk. It's a full-screen sector-oriented disk/file editor and a page-oriented memory editor. When you want to "unerase" erased files, DED86's flexible KEEP facility does the job without you fussing over FATs.

Check out this list of features

- ✓ Look by cylinder/head/sector, sector or cluster; even reserved/hidden sectors
- ✓ Scan free clusters to search for erased data
- ✓ Jump about subdirectories
- ✓ Keep sectors & clusters for writing to a file; Great for partial file recovery
- ✓ Edit bytes in hexadecimal or ASCII, zap in 0s; Undo last edit!
- ✓ Search your disk drive, a file, or memory for ASCII or hexadecimal strings
- ✓ Obtain complete disk statistics in one screen; includes usage data
- ✓ Save/Restore sectors to/from auxiliary buffers; for moving data around
- ✓ Handles 5.25" & 3.5" drives & RAM disks; even large partition drives

DED86 is easy on your pocketbook; only \$29.95 + \$3S&H

Fujitsu Floppy	5.25" 360K 1/2 height	\$75+\$4S&H
Disk Drives:	3.5" 720K in 5.25" frame	\$85+\$3S&H
	3.5" 1.44M in 5.25" frame	\$90+\$3S&H

TRSCROSS: read/write/format TRS-80 media on your PC; converts SuperScript files and BASIC programs \$89.95+\$2S&H

Note: MISOSYS has a large selection of TRS-80 Model I/III/4 software (languages, utilities, games, applications) and hardware (RAM, hard drives, cables)

FIVE TWELVE K

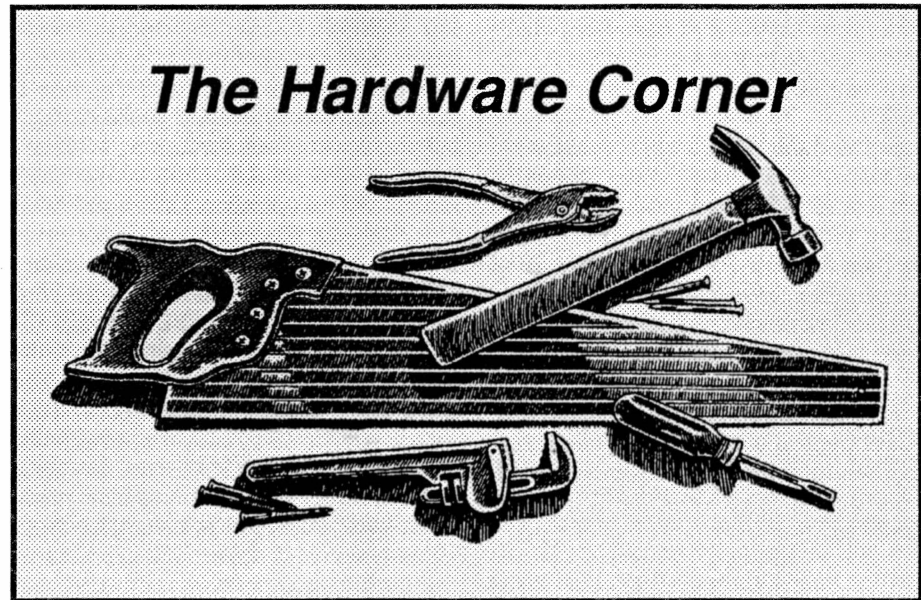
A Better way for most Model 4's

Richard King
808 Orleans Place
Raleigh, NC 27609
919-872-5846

THE MISOSYS QUARTERLY issue IV.ii offered a method of using the full 512K addressing capability of the XLR8er processor, Hitachi's HD64180. That article modified the early, non-gate array versions of Tandy's Model 4. Peter Van Caesele of Waterloo, Ontario installed it on his early Mod 4. He encountered and corrected some memory problems and says it now runs great. Several people have asked if it would be possible to install the changes on a Gate Array or 4P version of the Model 4. At that time I had to say, "Not yet". I recently purchased a no-Rev(ision) gate array machine for the paltry sum of \$35. The former owner got nosy and careless one day; he lifted the top off the machine successfully, but while re-installing it he broke the glass seal at the back of the video tube. One of my "spare parts" Model III's came to the rescue with a replacement.

The machine I modified for the TMQ IV.ii article was briefly tested in the "Port 84" mode (explained below) because I ran out of time meeting Roy's publication deadline and my job's traveling requirements. Writing some of my article during my trips caused several coding errors to slip through the cracks. Apologies for dropping a one byte instruction in XLBOOTK and several instructions in XLSYS0S. Both versions would boot alright then rear their ugly heads later. The corrected code is included with this article.

After changing my original test machine from Port 84 mode to XLR8er memory management mode and debugging the XLSYS0S problems, I turned to a marginal set of logics I have for the GA (Gate



Array) machines. Retrofitting my original design to this design proved difficult. Those of you who have waited for this update to my IV.ii article will find that a wider choice of 256K memory chips can be used. Early Model 4 owners can enjoy the benefits too, though at a slightly higher price than the previous design's costs.

Thanks to RTUG club member Richard Wright I was able to dismantle his Revision A, non-GA Model 4P to locate the attachment points needed for this article. At his request, the circuitry shown was not installed on his 4P. It is believed to be accurate but must emphasize that IT IS NOT TESTED. I've not yet run across a GA version of the 4P. Does anyone have the logics?

When Roy Soltoff and I talked about this article we concluded that I should treat it as a whole new subject with minimal references to IV.ii. For those who have read the other article, don't go away; much new text is entwined with the old. The primary slant of this article is toward the GA Model 4's but non-GA Model 4's and 4P's are included here too. Writing about multiple option and circuit board versions is a challenge; please bear with me.

WARNING - THESE CHANGES MAY CAUSE TANDY TO FROWN ON SERVING YOUR MACHINE, THOUGH THEY ARE REMOVEABLE SHOULD

THE NEED FOR SERVICE ARISE.

One of my primary goals was to develop these hardware mods in a way that would not require irrevocable changes to my (or your) machine. With the exception of eight capacitors on the non-GA Model 4, I've met that objective. Tandy socketed the right chips for this project. The changes can be removed in about 1/2 hour, restoring your machine to "pristine condition" should you ever have to take it in for service. It's even possible to remove and replace the warranty sticker undamaged if you have some 1,1,1-trichloroethane cleaner and a little patience.

DECISIONS:

One of the first things you must do when going to 512K of memory is to choose a (reversible if necessary) modification path. Either choice mandates that you alter the LS-DOS operating system and your Model 4 system board.

Choice 1: Continue using Tandy's hardware scheme for mapping the first three memory banks via hardware I/O Port 84. Advantage: All software should work as though you did not install these modifications. Drawbacks: "Permanent loss" of low memory addresses x'0FF4' through x'106E' which will be reserved for M. Houde's code; more wiring to the board than Choice 2; non-GA Model 4's must use a PAL/HAL chip in socket U72.

Choice 2: Ignore Tandy's Port 84 control of the memory banks and let the HD64180 manage all memory access. Advantages: Less board rewiring than Choice 1; moving M. Houde's code to lower memory (frees memory x'0FF4' et al. for other drivers); PAL/HAL chip U72 isn't needed. Drawbacks: None, unless an application program you're using ignores @BANK and does its own bank switching. Model III bank switchers and Model 4 LeScript may fall into the category.

WARNING - READ AND UNDERSTAND ALL THE INSTRUCTIONS BEFORE YOU DO ANY OF THEM. IF YOU HAVE ANY DOUBTS, CONTACT ME AND I'LL TRY TO HELP YOU.

PREPARATION:

To extend your XLR8ed GA machine to 512K you'll need: (2) 27, 51 or 56 ohm, 1/8 watt resistors (details below), (1) 40-pin and (2) 16-pin solder tail IC (Integrated Circuit) sockets; (2) 74LS157 multiplexer chips; wire wrapping wire; a 15 watt soldering iron and rosin core solder; misc. tools; (1) extra (old?) IC socket, any size, which we'll cut up for parts. All non-GA machines will need (1) 20-pin socket. Early Model 4's will need PAL chip U72 P/N 8075968 (see TMQ IV.iii, p.52).

If Port 84 will map memory banks 0-2 you'll then also need another 74LS157 and matching 16-pin solder tail socket.

NOTES:

1> You should purchase low profile sockets with the slim soldering tails. Low because it's tight where the XLR8er board is mounted; slim so the original system board sockets aren't damaged.

2> A second set of 256K memory chips is helpful but not necessary at the onset. Read the BUYING MEMORY section before procuring memory.

3> If you've installed the modifications in IV.ii and want to change over to this version, REMOVE ALL FORMER WIRING FROM BOTH SIDES OF THE SYSTEMBOARD before installing these circuits - This is the voice of experience talking.

4> The 74LS series of chips specified here can be replaced with faster chips such as the 74F or 74HCT series. If you do substitute, you must ground all unused input pins on the new chips to, as Signetics puts it, "prevent serious system problems". This grounding is not necessary on the 74LS family of logic.

SOFTWARE:

Create 2 new boot diskettes. Each should contain a fresh backup of LS-DOS 6.3.0 and utilities. Apply the appropriate /FIX patches: XLBOOTK & XLSYS0A if you've elected to retain Port 84; XLBOOTS & XLSYS0S otherwise. Everyone must apply XLR8S2, XLR8S12 and XLR8BU to complete the patch set. Put the remainder of M. Houde's programs and some of your own (I added PRO-WAM) on the diskettes, too. If your XLR8er is already installed and contains 256K memory chips, you may be able to test the patches to some extent by re-booting and then installing PRO-WAM or a Memdisk, as long as you use only banks 8 thru 14. Put a write protect tab on at least one diskette. You should also jot down any CONFIGurations on your various XLR8ed boot diskettes since they will have to be redefined after you've installed these modifications. CONFIG'd memory, starting at '0FF4', will change when the new XLBOOTx patch is applied.

GETTING STARTED:

To ensure that drive 0's heads are moved out from track 0, do a final DIRectory or CATalog listing of the diskette in drive 0, then shut off your machine and pull the power cord from the wall. Place your Model 4 on a towel on your workbench to protect the cabinet, then remove the screw from the rear of the machine, near the top. Roll the machine on its back, remove the anti tamper sticker if you can and the 10 screws holding the cover to the machine base. While you have it there, take note of the "26-10xx" model number on a tag on the bottom of the machine. 26-1067 & 1069 are non-GA Model 4's; 26-1069A & 1070 are GA units. Squeeze the top cover to the base while returning the unit to its normal upright position, then gently lift the top cover off. It's best to pull the top forward as far as possible and rotate it

counter clockwise (CCW) while slowly lifting it over any obstacles. Have a helper watch the back of the video tube through the slots above it, to ensure that you don't break the glass seal as you lift the tube's neck past some unforgiving hardware and wiring in its path. Lay the cover on its side; tag and then remove the ground wire and wide black wire connector from the video circuit board in the cover.

The Model 4P cover is fastened by two screws on each side of the case and two more under the carrying handle. The anti tamper and model number stickers are under the rear access door. 1080 is the non-GA Model 4P while 1080A is the GA version. (This article will not modify the 1080A units.) The cover lifts off the back of the machine; watch for a falling keyboard as you lift it. The system board is in a metal pan above the keyboard storage area. The pan is retained by screws at the back and sides of the chassis. Watch for several connectors located at the front end of the system board so as to not damage them; you will start to see the connectors as you pull the back of the pan away from the chassis.

CAUTION:

74F, 74HCT and CMOS chips are sensitive devices. They require special precautions to prevent damage from the static electricity that today's synthetic clothing can generate. When handling CMOS, I move to our stainless steel kitchen sink (which is connected to copper plumbing) and block the sink drain so chips don't fall into the food disposal. I rest a bare arm on the sink rim at all times and use its hand to remove and/or replace CMOS chips in a semi-awkward fashion while keeping body motion to a minimum. A second method is to lean against an unpainted metal appliance whose 3-pronged plug is inserted into a properly wired 3-holed outlet (no adapters, please). A third method is to use a grounded static draining wrist strap such as Radio Shack's 276-2399. All work well.

Carefully remove and set aside the (CMOS equipped) XLR8er board if it's installed. To remove the jumper cable from the XLR8er, I use one tine of a long thin pair of tweezers. I gently work the tine be-

tween the connector and one side of the XLR8er socket, then twist it slightly. I then move over to the other side of the socket and twist again, then push the tine deeper toward the other end of the connector and repeat the process several times until the connector and socket have parted company undamaged. Remove the radio frequency radiation shield covering the system board (6 screws). There's no need to remove the system board, all work can be done on the component side.

Using your tweezers, carefully remove the XLR8er cable from the system board socket. Use the proper "Verify Before Starting" list to ensure that the expected chips are at the locations shown in the left column of the list. With a battery powered checker, verify that there is continuity between the pairs of points shown in the right column of the list. On the 4P list you're told to verify that U133-1 thru U140-1 aren't interconnected. If you have the good fortune to find that they are you can ignore a later step that tells you to wire those pins #1 together.

SOLDERING HINT:

Just prior to heating a joint, wet the tip of your soldering iron with some fresh solder and shake the excess off onto a piece of newspaper. This wetted area will transfer heat more rapidly to the joint to be soldered.

If you are working on a non-GA Model 4, take a 40-pin socket from your inventory and solder a 6" wrap wire jumper to solder tail #28 as shown in the sketch. Use as little solder as possible to prevent system board socket damage or adjacent pin shorting. (The adapter is necessary because some revisions of the board do not use the *RFSH (Refresh) signal, making this tap wire the only way to get to it.) Route the wire to the left as shown while inserting this adapter socket into the system board's U57 socket (the Z-80A socket). Carefully ensure that all solder tails properly enter socket U57, then seat the adapter fully. Don't press TOO hard; you don't want to break the system board.

Much of the following text refers to the GA Model 4. If you have one of the other machine versions, cross-reference its

applicable sketches to those for the GA. Equivalent chips are shown in parenthesis in (4P; PAL chip) order.

SOCKETS & CHIPS:

In these next paragraphs you'll be told to LOOSELY insert IC's in their new sockets. This means "just enough to get all chip legs started in their respective socket positions". You'll need to pre-form the chips so that the legs align properly. You must support the socket while inserting its chip so you don't break the few (solder) tails holding the socket in place. Later you'll fully seat the chips.

Prepare a 16-pin socket by breaking off tails 2-7, 9-10 and 12-14. (Model 4P - also break off pin 11.) Tin the inside surfaces of the remaining tails, 1, 8, 11, 15 and 16. Tin the broad shoulders of the corresponding pins on U66 (4P = U112; PAL 4 = U63), which must be a '157 (in other words, a 74whatever157). Form (never BEND!) the tails together slightly so they fit U66 snugly, slide the socket over U66 and align the tails on the tinned shoulders of U66. Heat each of the (four or) five joints with your soldering iron to complete the bonds. LOOSELY insert a '157 chip in the socket. This is address multiplexer chip Z66 (Z112; Z63).

On GA and PAL Model 4's, prepare another 16-pin socket by breaking off tails 1-7 and 9-15. Tin tails 8 and 16 then prepare pins 8 and 16 of U63 (none; U65 - it must be a 16 pin chip) as before. Solder the socket to U63. Loosely insert a '157 chip in the socket. This is refresh chip Z63 (none; Z65).

4P's and PAL Model 4's in Port 84 Mode: 4P's remove PAL chip U110; Model 4's remove PAL chip U72. Prepare a 20-pin socket per the proper sketch. Jump over the next paragraph to continue.

All GA machines, carefully remove the 40 pin GA chip from socket U5 with your tweezers. U5 should be marked "4.2". Be doubly sure that the empty socket is connected as specified in the verify list. (Stop now if it isn't; call me if the hour is decent.) Prepare a 40-pin U5 socket adapter as shown in a sketch. If you'll retain Port 84, break off solder tails 32, 33 & 34 flush

with the plastic. Go to the next paragraph and wire your system board to address 128K. If you've chosen to ignore Port 84, break off solder tails 15 & 17 flush with the plastic. Strip a short length of wrap wire. Solder it across the broad shoulders of IC U5-15, -16 and -17. Wire your system board to address 64K.

Insert the modified adapter into socket U5 (U110, U72). Using the Verify List, ensure that the removed tail stubs do not make contact with the original socket (press the adapter lightly into the socket while testing), then loosely install the removed IC into the new adapter. (Non-GA and 4P owners go to the next paragraph.) If your GA machine has a no-Rev., Rev. A or Rev. B system board (marked on the back of the board, covered by the foil radiation shield - not to worry), you should see a wrap wire emerging from a hole in the board near U5-16 and going toward U33. For Port 84 use, solder the wire to U33-16 ('273) to address 128K. For non-Port 84 use, solder the wire to the bottom of the capacitor C39 (adjacent to U33) which grounds the wire to address 64K. The other end of this wire is connected to U5-16. On Rev. C or above GA machines the wire is replaced by jumper JPR13 (location unknown). The bottom two pins address 128K, the top two address 64K. Plug the jumper as specified above.

Port 84 users, prepare a 16-pin socket by removing tails 1-7 and 9-14. Prepare pins 8, 15 & 16 as noted above. Install the socket on U65 (U111; U76) which must be a '157. Loosely insert a '157 into the added socket. This is Port 84 chip Z65 (Z111; Z76).

WIRING:

With the hard part out of the way, it's time to add the necessary wiring to the chips you just installed. Do exercise care so as to not confuse the "Z" chip pins with the "U" chip pins. I specified LOOSE insertions of the IC chips so that you don't inadvertently solder a chip to its socket here. Except as noted, wires will be soldered to the shoulders of the named IC pins. Here's my method: Route the wire from pin to pin (neater is better than shorter) then add a little extra and cut it;

Strip 1/16" from each end of the wire; Tin the stripped ends and the wide shoulders of the IC's pins to which the wire ends will be attached. Hold the bare end of the wire against the IC shoulder and heat BOTH so that the solder flows between them. Remove the iron and hold the wire motionless while the joint cools. If you get too much solder on a joint, reheat the joint again and use a desoldering tool such as Tandy's PLASTIC (not aluminum) 64-2120 unit to clean the area, then start over. When soldering the wires to the chips, you must quench that burning desire to move your fingers because the wire they are holding is intolerably hot. Movement can cause cold solder joints which will render these modifications inoperative. After attaching both ends of a wire, use tweezers to support the wire at each solder joint while forming the jumper to lay near the board, out of harm's and XLR8er's connector pins' way. Use the drawings to find the wiring points. (Identified IC pins that show no "other connection" are already making contact through the socket installations done above.) One section of each page identifies the Port 84 circuitry; another section is for those who have chosen to ignore the port. The remaining sketches apply to both situations. Port 84 users may notice that the Multiplexer chip's pin #11 will be connected to two circuits. Due to circuit isolation done by an adapter socket, this will not cause problems. For Model 4 PAL machines, note also the jumper connecting U72-9 with U77-15 shown in the sketch.

Note the address input lines marked 'A16' and 'A17' in the sketches. Cut each of these wires about 12" long, then solder one end of each to its intended Z66 (Z112, Z63) multiplexer or Z65 (Z111, Z76) Port 84 pins. Wrap each wire around some component such as the body of the chip it's attached to (in between the pins) or a nearby resistor lead so as to form a strain relief. Put a tag on the loose "A16" wire end, then "weave" the two loose wires around each other (no more than 3 turns) to make them more manageable. The loose end of these address wires will later connect to the XLR8er board.

XLR8ER BOARD:

I've devised a better (compared to the

IV.ii article) connection technique for A16 and A17, shown in a sketch. It is delicate but takes up virtually no room in the machine. To implement the method, straighten two pins on an old IC chip. Grasp each pin as close to the chip body as possible with a pair of pliers, then gently rock the pliers up and down until the pin snaps off. (What's that? You don't have any old IC's? Use pin 1 from two removed 64K chips; it isn't connected to the innards.) Form each pin 60 to 80 degrees where it starts to narrow, then tin the broad part of its shoulder. Using CMOS handling cautions, tin the shoulders of pins 10 & 11 of the XLR8er board's U4 chip, a '157. Solder the snapped-off pins to U4-10 and -11 as shown in the sketch. Be sure they're properly spaced to easily enter an IC socket. Take an old IC socket and test its fit on the pins just installed. Adjust the pins if needed then cut off two sockets as shown in a sketch. An Xacto saw works great. Carefully form the solder tails as shown in the sketch. (You may want to delay this next step until you mount the XLR8er board so that you can trim the wires to a reasonable length.) Wrap each of the stripped (1/2") and tinned 12" wire ends around one of the tinned solder tails, heat the joint, then trim off any excess bare wire. Port 84 users MUST keep the A16 and A17 wires in proper order here. For the others, it makes no difference.

RESISTORS:

After wiring your system board, two resistors (one resistor on 4P's) must be put in the lines which drive pin #1 (MA8) and pin #9 (MA7) of the 256K chips. Your resistor value is determined by an "RP" resistor pack located adjacent to the memory chips; it looks like another IC chip. It's near U85 on GA boards, near U76 on older 4's and near U133 on 4P's. You should see a number like "270", "510" or "560" stamped on the resistor pack body. The number means "a value of 27 (or 5x) followed by zero 0's", ie., 27 (or 5x) ohms. Some units may be stamped "27", "51" or "56" directly. The new resistor for signals MA7 and MA8 should be the same value as that stamped on the resistor pack.

On GA machines, the MA8 resistor goes between Z66-12 and the common pin of

jumper JP1/2, the middle of (two or) three jumper pins located above the U72 memory chip (see sketch). To make the connection, remove the jumper block from JP1 and pry up one of its metal pin covers (seen when the jumper is on the pins) to reveal a pin hole from the top side. Form one resistor lead with a slight wriggle. (If you don't feel good about exposed resistor leads, put pieces of electrical spaghetti over them before you continue.) Push the bent end into the exposed jumper hole; it should bind slightly as it goes in. Make sure that it doesn't go all the way to the bottom of the jumper block so it can't touch any of the printed circuit land patterns on the board. When the end of the resistor lead is positioned properly in the jumper, roll the exposed lead over the top edge of the block to control how far it can be pushed into the block, as shown in a sketch. The other hole of the jumper block is placed on the pin common to JP1 and JP2; it connects to pins #1 of the lower set of memory chips. The loose end of the resistor is soldered to a piece of wrap wire which is soldered to Z66-12. In a similar fashion, the MA7 resistor goes between Z66-9 and the common pin of JP5/JP6, located to the right of U74. (A wide printed circuit "wire" can be seen going from the common pin to pins #9 of the lower set of memory chips.)

Non-GA Model 4's will attach both their resistors in a similar fashion. The MA8 resistor goes to E12 which connects to pins #1 of the left memory set. The MA7 resistor goes to E5 which connects to pins #9 of the left memory set. Eight capacitors must also be removed from the system board. These are to the left of the left memory set. A continuity tester will show that one end of each is common with pins #1 of the left memory sockets. The Multiplexer sketch shows the numbers found on my development machine. These capacitors are needed only if you have 16K memory chips in the left sockets. If your machine came from the factory with 64K, these capacitors and a small -5v. regulator labelled "Q5" (it looks like a black plastic transistor, above the memory sets) may already be removed.

The 4P board was designed in a more conventional manner eliminating the need for an add-on REFRESH circuit. How-

ever the board layout person failed to connect pin #1 of each memory chip to anything on my friend's machine. If the pins #1 of your lower memory sockets are not interconnected, you'll have to add those jumpers, something that might best be done on the back side of the board. If you do opt to use this method, route an insulated piece of wrap wire from U133-1 through one of the many circuit pass-thru holes in the board, then connect it to your MA8 resistor on the component side of the board. (The alternate method is to install the new memory chips and solder the jumper wires to the shoulder of pin #1 of each of them, a pain if you must ever change memory).

MEMORY:

(This dialogue assumes that you are working with memory chips which meet the specs set forth in BUYING MEMORY, below.) Using CMOS handling cautions, remove all 64K memory chips from the system board (tweezers work good here, too). On the GA and 4P machines they're the two horizontal rows of 8 chips near the bottom right of the board. The non-GA Model 4 has its chips stacked vertically in the same area. If you lack the proper static protective container, place the old chips on a piece of aluminum foil and later wrap them in it. If you possess two sets of 256K (CMOS) memory chips, install one set in the lower (GA machines), upper (4P machines) or leftmost (non-GA machines) set of memory sockets and the other set in the XLR8er's memory sockets. If you have only one 256K set, plug them into the system board. If 8 of the 64K chips that you removed from the system board are 150ns. or faster, install them in the XLR8er's memory sockets. (The latter combination lets you access banks 0 through 6 on the system board plus 7 & 8 on the XLR8er.)

TESTING:

DOUBLE CHECK ALL WIRING. Ask a family member or friend to verify it too. **WHY TAKE CHANCES?** Press the loosely socketed chips firmly into their sockets. Put the XLR8er cable in the system board's Z-80 socket (or adapter on the older model 4). Plug the other end of the cable into the (NOTE: CMOS-equipped) XLR8er board

and let the board hang from the cable. Trim the A16 and A17 jumper wires to length. Strip the end of each and solder it to the dual-socket address connector per the text several paragraphs above. (Leave yourself some working slack with these wires; their length is not super critical.) Attach the A16/A17 address connector to XLR8er board CMOS chip U4, pins 10 and 11. Ensure that the XLR8er board is not shorting to the system board by putting a piece of cardboard between them. With the computer top cover laying sideways on your work bench next to the computer base, connect the video cable (you did mark it, didn't you?) and ground wire, then power up the machine. Watch for smoke (heaven forbid). If none appears and disk drive 0 rattles its way to track 0, you should be off and flying on your added memory. Push a write protected test diskette into drive 0 and boot away. If drive 0 does not act in it's normal manner, **SHUT OFF THE MACHINE IMMEDIATELY** and investigate the cause of problem. If the machine starts to boot but fails to go to the ready prompt, memory speed may be too slow. To check memory, press and hold the <BREAK> key while pressing and releasing the <RESET> button to start the machine in Model III BASIC mode. Respond to the two prompts by pressing the <ENTER> key twice, then enter a short BASIC program such as filling the screen with one character. Let the program "cook" in memory for a minute or two, then RUN it. If it runs correctly then the memory on the system board is probably alright; go back and review your modifications. If "Strange things are happening" (shades of Red Buttons!), you may have a bad solder joint, one of the "Zxx" chips you've installed may be bad or you may need a set of faster memory chips.

I run the following tests on my machines on the kitchen table, to the frustration of my family. Install PEXMEM. Enter a "SETX (S)" to view the 64180 control registers. Increase the "SETX (R=" and then "SETX (M=" values until the keyboard becomes unusable, then reboot and "SETX (M=" one less than what caused keyboard loss. Set "SETX (R=80)" if at all possible - it's the maximum refresh stress test. I then install PRO-WAM in one of the newer system banks - 3, 4, 5 or 6 and

finally run MEMDIR for the heck of it. I have Mr. ED's MED (Memory EDitor) in PRO-WAM and I periodically invoke it to look around memory. The first time around I use it to write "BANK nn" at page '80' of each of the banks and fill the remainder of the page with x'00' or x'FF' (zeros in the first bank, foxes in the second, zeros in the third, etc.). Don't forget to write each page back to memory. If memory is not switching correctly or if you have 64K where 256K should be, you'll begin to see these labels pop up in the wrong place as you step through the banks. I let my machine cook for a minimum of 24 hours this way with the intensity turned down and the boot diskette removed, just in case refresh fails and the machine decides to throw stones at drive 0. For several hours the DOS ready prompt is displayed, for several hours MED will display some page of extended memory. If after 24 hours each page '80' is unchanged and the machine can then run a job or two (which may destroy page '80' of one or more banks - don't use LeScript to test here) then I feel comfortable with the modification. Note that our electric company does some unclean power switching on a random basis. If my test fails, they are my first suspect; I'll do another 24 hour run before condemning the changes. Another failure I've seen were phantom keystrokes on the screen after the machine sat a while at DOS ready. This turned out to be the (unmodified) keyboard circuitry goofing off after it warmed up. I cleared it by adding another refresh wait cycle via SETX.

CLOSING IT UP:

When you are satisfied that everything is working correctly remove the diskettes, power off the machine and remove the plug from the wall. Remove the (CMOS) XLR8er board from its connecting wiring (4P owners read the XLR8er manual here). Install the system board R.F. shield and (CMOS) XLR8er unit. (Non-GA Model 4 owners, did you move the XLR8er up on its copper clad mounting board per the sketch?) To avoid breaking those delicate tails on the XLR8er connector, use tweezers to install the A16/A17 jumper. Make a copy of this article, mark the wiring you've done on your machine; tuck it under the keyboard for future reference. When in-

stalling the top cover, rotate it counter-clockwise a tad to provide best clearances for the tube neck and XLR8er board. Get some helpful eyes; ease the cover down S-L-O-W-L-Y. It's a tight fit.

WOW! WHAT DO WE DO WITH THESE EXTRA 4 BANKS? Modify DoubleDuty to run two separate and complete 128K LS-DOS environments (nuts to that puny 64K stuff) plus a common Memdisk and/or PRO-WAM? My mouth waters at the prospects! You write it, I'll test it.

BUYING MEMORY

The designs presented here are capable of driving any 256K memory chips which require only A0-A7 refresh, unlike the special requirements in the TMQ IV.ii design. By the numbers the XLR8er should work fine with 150ns. memory, but in truth I'm not too sure. I've had weird experiences while developing the TMQ IV.ii article, and had a few more experiences on my way to completing this one. I have PRO-WAM 2.0 and the Mister ED application pack, an excellent enhancement. Mr. ED's MED (Memory EDitor) has been a real boon to debugging memory problems since it accesses, displays and modifies any memory bank. I've been using SAMSUNG memory chips (same as Intel) which appear to have a default "never used" memory bit pattern of x'00 00 FF FF 00 00 FF FF'.

ITEM: I was loading P-W into the XL's memory and things were working great for several months, then one day an exit from P-W returned me to a trashed screen. Calling P-W again displayed a trashed menu. I did a cold boot and recreated the environment. All was fine for a while but the problem came back. This time I did a warm boot, put P-W in the lower 256K and looked around the XL memory. Sure enough, garbage where once there was legibility. In the unused areas, I now saw '00 00 7F FF'. Swapping the bit 7 with the bit 6 chips, the pattern changed to '00 00 BF FF'. Replacement of the suspected chip (now bit 6) reinstated the familiar pattern and fixed my problems.

ITEM: I was testing the circuitry for this article, using my last set of SAMSungs on

the system board and some 64K chips on the XL board. The familiar pattern was not to be seen, replaced by something that indicated bits 0 & 7 were suspect. Swapped them with bits 1 & 6 and the pattern changed. Out of SAMs, I found two Hitachi's and plugged them in. The system board pattern became unpredictable. What's worse, the pattern I was getting from the XL board degraded, too. It is possible that the Hitachi's are bad, but I don't have enough of them for a valid comparison. I finally went to my first 512K machine (non-GA) and pulled some memory for use in the GA testbed. The display became familiar once again.

ITEM: OKI's 64K, 150ns memory performed flawlessly in my unXLR8ed machine at 4mhz. When the XL board was installed, the OKI's couldn't maintain a boot for more than a minute, sometimes dying before the drive 0 light shut off. Moving them to the XLR8er board didn't help (yes you can do it, but you get only 2 banks this way) - they still failed to refresh properly.

ITEM: As noted in this section of the TMQ IV.ii article, NEC 150ns chips displayed bad habits on both the system and XL boards. (Note that the XL board uses the "traditional" RAS refresh technique being implemented in this article.)

ITEM: Samsung 150ns chips give me solid, stable memory with 80 cycle refresh and 0 software wait states on my GA machine. They also worked reliably in my PAL machine when the IV.ii modification was installed.

ITEM: 150ns memory is good to 6.66mhz. The XLR8er runs at 6.1+ mhz. The system board runs at 4.02mhz. The video circuitry runs at 12.6mhz and 6.3 mhz. All of these timings are in contention for various memory cycles on the system board. Video memory access cycles (done by Motorola's 68045 video display controller) can lock out and delay main memory access cycles because the video controller is given priority by design.

ITEM: The HD64180 has a long span between refresh cycles compared to the Z-80 which does a refresh after EVERY instruction fetch cycle. This makes it more

imperative that every refresh cycle count if you want to run in XLR8ed mode. The 64180 does not attempt to makeup for lost time when it senses a WAIT condition as a result of a video lockout.

ITEM: My TMQ IV.ii circuit would not work in Peter VanCaeseele's pre-Revision A machine until he installed 100ns. memory. It worked perfectly in my Revision C machine with 150ns. at 0,1,80. Note that Peter's keyboard can't handle 0 memory waits - he gets too much key bounce.

ITEM: The Ignore Port 84 circuit in this article would not run in my Rev. C PAL machine until I put 100ns. memory on the system board. The same 120ns. and 150ns. memory chips that fail in my Rev. C PAL machine work perfectly in my GA no Revision machine at 0,1,80 with these "Ignore Port 84" modifications installed. Putting much faster 74F157 multiplexer chips (to lower signal propagation delay times) in my Rev. C machine did not significantly improve performance - 120ns. memory held a boot for less than half a minute.

ITEM: This article's "Use Port 84" circuit was not reliable in my GA machine with 150ns. memory installed. It ran perfectly with 120ns. memory. However the "Ignore Port 84" circuit runs fine with 150ns chips.

CONCLUSION: Shooting from the hip, I'd recommend 120ns or faster memory on your system board to provide an extra safety margin when using an XLR8er. It is also very obvious that the Gate Array boards are much more reliable in the memory management area than the older units when adding the extra 128K of memory. I picked up some 120ns. Hitachi and Samsung memory at the Charlotte hamfest and ran them at 80 refresh cycles, 0 software wait states. Not a hint of a problem except as noted above or when we get one of our infamous power glitches here. While I'm sure that there are many good brands of memory, make mine Intel & Samsung; I feel comfortable with them. Oh, by the way, those 100ns. chips came out of an IBM PS/2 whose owner was upgrading to 1024K chips. They're a brand I've not seen before, NMBS. Additional

markings are "AAA2801P-10". (See the IV.ii article for a list of other memory chips that may work.)

TECHNICAL

Here's the rundown on how it all works, for all you technical types. Let me begin by first telling you a little about memory chips. 64K memory chips require 16 bits of address information to produce one bit of data. In addition to the address lines, the chip requires a read/write control line and two data lines, one to input a data bit and another to output a data bit. Then there are the power lines. The original 16K chips required three voltages and a ground to operate. 64K chips were improved to the point where they could operate on only one voltage level plus ground. Now when you stop to count them all up, that's a lot of lines, folks.

Someone obviously got a brainstorm and deduced that if the CPU (Z-80) could hold the address up long enough, then the address could be split in half with half being fed to the memory for a while (where the memory would lock in on it) and then the other half could be fed in. After the second half was digested by the memory controls, the data bit would be pumped out on the output line. The new design would require another control line or two but that's 2 lines added and 5 (for 16K), 6 (for 64K), 7 (for 256K) or more lines saved, and each line represents a chip pin.

"Since the internals of a memory chip are laid out like a grid or a two dimensional array, why not refer to the first half of the address sequence as the ROW address and the second half as the COLUMN address?" "Sure, why not? Now what shall we call the address control lines - how about Row Address Select and Column Address Select?" "Nah, too long, kind of like International Business Machines! If they could reduce that name to three letters, why can't we? How about trimming them to RAS and CAS?" "That's great, but we still need a way to control the refresh cycle, and that sounds like another line." "Hey, I have an idea. Do you think it would work if we did an automatic refresh cycle during each RAS period. We wouldn't need a separate control

```
. XLBOOTK/FIX - To use XLR8er special features (27-Dec-89)
. Written by Richard R. King, 808 Orleans Place, Raleigh, NC
27609
. PATCH boot/sys.lsidos USING xlbootk/fix
. Must also PATCH sys0/sys.lsidos USING xlsys0a/fix
. Must also apply remaining patches from Michel Houde
. Must also apply 512K hardware mods. See Misosys Quarterly
vol IV.ii, .iv
. fix lomem pointer @x'0206'
D00,06=6F 10;F00,06=F4 0F
. new @MUL16 svc @x'06c9'
D04,C9=57 5D 6F ED 6C ED 5C 7B 5A 16 00 19 C9
F04,C9=C5 EB 4F 21 00 00 7D 06 08 29 17 CB 01
. TASKER patch1. end of old @mul16 @x'06d6'
D04,D6=ED 28 38 E3 E5 AF ED 39 38 21 78 00 C9
F04,D6=30 03 19 CE 00 10 F5 4F 7D 6C 61 C1 C9
. @BANK beginning @x'0877'. XBUR$ @x'087b'
D06,77=79 C3 FD 0F 00 00 00;F06,77=E6 7F FE 03 D2 ED 0D
. @VDCTL @x'0b9a' JP to fix
D09,9A=5A 10;F09,9A=42 0D
. ROWCOL 2 ADDR @X'0ddf' multiply
D0b,DF=2E 50 ED 6C 00 00 00 00;F0b,DF=21 50 00 CD C9 06 65 6F
. change delay in FDC driver @x'0f26'
D0D,26=24;F0D,26=12
. add XLR8 module @x'0ff4'. @BANK and @VDCTL
. Byte 13 of next line is bank count. 0F = 512K, 07 = 256K,
17 = 768K.
D0D,F4=18 FE 6E 10 04 58 4C 52 38 E6 7F FE 0F 30 54 D6
F0D,F4=00 00 00 00 00 00 00 00 00 00 00 00 6d b6 6d b6
D0E,04=03 30 0A AF ED 39 38 79 E6 7F C3 7E 08 E5 04 10
F0E,04=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
D0E,14=12 21 05 80 39 38 3A 07 07 07 C6 18 ED 39 38 E1
F0E,14=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
D0E,24=C3 DB 08 6F 2C 26 01 3E 7A CB 0C CE 00 2D 20 F9
F0E,24=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
D0E,34=6F 7C 26 08 10 04 2F A6 18 0E 10 05 A6 3E 08 E1
F0E,34=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
D0E,44=C9 10 09 ED 34 20 F6 B6 77 AF E1 C9 10 03 3A 02
F0E,44=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
D0E,54=02 E1 C8 C3 ED 0D ED 38 38 B7 CA 42 0D F5 AF ED
F0E,54=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
D0E,64=39 38 CD 42 0D E3 ED 21 38 E1 C9
F0E,64=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D
.eop
```

line then." "Yeah! We can do that! If a RAS comes up and goes down without a CAS, then a refresh cycle would occur but data would never be output to the data buss." "By Jove, I think we've got it!"

Well, it does work, but there remained the little problem of a program in a tight loop not giving the remaining rows a chance to get refresh cycles periodically, and they are important since DRAM, Dynamic Random Access Memory, is little more than 16,000+ or 64,000+ or 256,000+ little capacitors (times 2) that can hold a charge for only so long (about 4+ milliseconds) before they are either refreshed or lose that charge and the information it

represents. The Z-80 took on the responsibility of managing refresh cycles and addressing along with its other assigned data processing duties. Hitachi's HD64180 continues that tradition (as do many other CPU's).

Along came the newer 256K memories and another brainstorm. This time someone decided that since not all CPU's directly manage refresh (some types require external refresh management circuits), they would design a refresh counter right on the chip and let the computer designer make use of it by accessing the chip backwards. The system designer could hold CAS active then activate RAS for

```
. XLBOOTS/FIX - To address 512K and use XLR8er special features. 05-JAN-90
. Written by Frank Slinkman, 4108-C Fairlake Lane, Glen Allen, VA 23060
. From dos READY, enter the command PATCH BOOT/SYS.LSIDOS USING XLBOOTS/FIX
. Must also patch sys0/sys.lsidos using XLSYS0S/FIX
. Must also apply 512K hardware mods. See Misosys Quarterly vol IV.ii, .iv
. This code replaces M. Houde's XLBOOTA/FIX patch code (won't overlay it).
. New @MUL16 svc @ x'06C9'
d04,c9=57 5D 6F ED 6C ED 5C 7B 5A 16 00 19 C9
f04,c9=C5 EB 4F 21 00 00 7D 06 08 29 17 CB 01
. TASKER patch1. End of old @MUL16 @ x'06d6'
d04,d6=ED 28 38 E3 E5 AF ED 39 38 21 78 00 C9
f04,d6=30 03 19 CE 00 10 F5 4F 7D 6C 61 C1 C9
. New @BANK routine begins @ x'0877'. 32 bit XBUR$ now @ x'08d7'
. Byte 5, next line = max # banks +1. 0F=512K, 07=256K on motherboard only.
d06,77=79 E6 7F FE 0F 30 55 E5 04
f06,77=E6 7F FE 03 D2 ED 0D 05 FA
d06,80=10 20 21 05 80 39 38 4A 07 07 07 ED 39 38 21 02
f06,80=B3 08 0E 86 28 19 0E 46 05 28 14 05 28 09 05 C2
d06,90=02 46 79 E6 7F 77 E1 A9 B0 4F CB 79 06 00 C8 E3
f06,90=ED 0D 3A 02 02 BF C9 47 CD 9F 08 C0 78 0E C6 E6
d06,a0=BF C9 6F 2C 26 01 3E D6 CB 0C CE 00 2D 20 F9 6F
f06,a0=07 07 07 07 B1 32 B0 08 AF 3E 08 E5 21 00 02 CB
d06,b0=7C 26 08 10 06 2F A6 77 AF E1 C9 10 05 A6 3E 08
f06,b0=46 E1 C9 E5 21 05 80 39 E1 DA ED 0D FE 01 17 47
d06,c0=E1 C9 10 07 ED 34 20 F6 B6 18 EC 10 05 3A 02 02
f06,c0=3A 01 02 A0 20 C9 78 1F 3F CE 00 07 07 07 07 47
d06,d0=E1 C9 E1 3E 2B B7 C9 00 00 00 00 ED 38 38 CA 42
f06,d0=3A 78 00 E6 8F B0 32 78 00 D3 84 3A 02 02 47 79
d06,e0=0D F5 AF ED 39 38 CD 42 0D E3 ED 21 38 E1 C9 00
f06,e0=E6 7F 32 02 02 A9 B0 4F CB 79 06 00 C8 E3 BF C9
. @VDCTL @ x'0b9a' JP address to fix.
d09,9a=DB 08;f09,9a=42 0D
. ROWCOL 2 addr @ x'0ddf' multiply.
d0b,df=2E 50 ED 6C 00 00 00 00
f0b,df=21 50 00 CD C9 06 65 6F
. Change delay in FDC @ x'0f26'.
d0d,26=24;f0d,26=12
. eop
```

one or more cycles as time permitted. Each RAS cycle would refresh another row of memory. Any information present on the address pins is ignored when CAS goes active before RAS. (Just how the memory chip can tell which came first, the chicken or the egg, is beyond me.) This then was the premise for my construction article in TMQ IV.ii. That circuit didn't care if the Z-80 or other CPU couldn't handle large memory chip refreshes. Refresh addressing was automatically maintained and optimized. While the 64180 can handle most 256K chips, the Model 4's system board thwarted the capability as you shall see later. Not all

chip manufacturers jumped on the CAS BEFORE RAS refresh design, perhaps because there is a royalty payment involved. Some of the memories that have CAS before RAS refresh apparently have some sticky timings which caused problems in the IV.ii design when those chips were tried.

Now that you hopefully know a little more about DRAMs, RAS, CAS and memory refresh, let's apply that knowledge to the subject at hand.

Tandy made some significant changes to the Model 4's circuitry over its life cycle.

Retrofitting my "early model 4" modifications to the newer GA machine circuits proved difficult. Let me explain. In the first Model 4 versions, the designers controlled the addressing of a memory set (the term "set" in this article refers to a hardware bank of 8 memory chips - not related to software switching of 32K memory banks) by sending RAS to both memory sets at all times while selectively sending the CAS signal through PAL (Programmable Array Logic - another story) chip U72 to only one or the other memory sets. Refreshing two memory sets was not a problem.

GA (Gate Array - a super-PAL chip) Model 4's on the other hand, have the RAS signal routed to only one memory set or the other and always send the CAS signal to all memory chips when needed. This created a problem because memory refresh is activated by the RAS signal and both memory sets need it. Since RAS control is managed by U5, Tandy engineers brought the REFRESH signal to it, and when active, it causes RAS to be present at both memory sets concurrently.

The design of the first Model 4P's is somewhere in the middle of it all. Like the GA machines it too manipulates the RAS signal while making the CAS signal common. Like the early Model 4's it uses PAL chips. Unlike either, it was apparently never designed to work with 16K memory chips. In essence it appears to be the design test bed for the GA Model 4. Then of course there is the GA version of the 4P which I haven't seen yet. That the XLR8er's designers were able to make their board work across this gamut at all is a credit to their ingenuity, or maybe it was just blind luck. Getting another 128K squeezed into the overall Model 4 picture has proven to be a real challenge. In the end, abandoning the CAS-before-RAS memory refresh technique used in the IV.ii article became a real plus. This updated version of my IV.ii modification gives the user a wider choice of 256K memory chips than was formerly possible. Non-GA owners can enjoy the benefits too, at a slightly higher price.

The XLR8er board uses much of Steve Ciarcia's SB180 computer design featured in BYTE magazine, 9/85, with minor

modifications. The XLR8er has a hard wired one cycle instruction fetch wait generator (in case you were wondering why your timing tests don't match your calculations; see BYTE p.94, IC15) and the same addressing and refresh circuits (p.95, IC14). The XLR8er's designers replaced the ROM in Steve's design with the RAM on the Mod 4's system board. In other words, the first 256K of memory addressing is sent to the system board memory. The trouble is, there aren't any spare pins in that Z80 socket to transport memory address lines A16 and A17 across the gap. The problem is easily resolved with our jumpers to XLR8er chip U4, but we must control how they manipulate memory and interface with Port 84's memory management. That's where the rest of the circuitry and some of the software modifications come in. One design continues to use Port 84, but controls its ability to address the system board chips. Another, simpler design (2 wires) ignores Port 84 (ONLY AS IT RELATES TO MEMORY MANAGEMENT - all other features of the port are functional) and probably improves machine reliability in the process. The following discussions talk to the 1069A & 1070 (Gate Array) boards. Having a set of Model 4 logics is helpful to follow this dialogue.

PORT 84: Most 256K memories are really four sets of 64K in one package. One of the memory address lines is used to select which quadrant of memory will be accessed while the other 8 address lines are used to point to the row and later the column in that quadrant. In a normal bank switching scheme, address line A16 is used to access the 2nd and 4th quadrants when on, the 1st and 3rd quadrants when off. (A17 works with A16 to access all four quadrants.) If the Model 4's Port 84 is retained to access the first three banks, A16 must only be allowed access to the 4th quadrant of the system board memory. The "Use Port 84" circuit, Z65, monitors A17. As long as address line A17 is not active, the lower 128K of the system board's memory is in use and Port 84 manages it. Gate array U5 (on non-GA machines it's PAL U110 or U72) uses bits 4, 5 and 6 of Port 84 to control memory partitioning via the *RASEN0, *RASEN1 and MPA15 signals. *RASEN0 and *RASEN1 normally select the set of 64K

```
. XLSYS0S/FIX - To address 512K and use XLR8er special features. 05-JAN-90
. Written by Frank Slinkman, 4108-C Fairlake Lane, Glen Allen, VA 23060
. From dos READY, enter the command PATCH SYS0/SYS.LSIDOS USING XLSYS0S/FIX
. Must also patch boot/sys.lsidos using XLBOOTS/FIX
. Must also apply 512K hardware mods. See Micosys Quarterly vol IV.ii, .iv
. This code replaces M. Houde's XLSYS0A/FIX patch code (won't overlay it).
. New @MUL8 svc @ x'190A'
d06,f0=57 ED 5C 7B C9;f06,f0=C5 57 AF 06 08
. TASKER patch2. End of old @MUL8 @ x'190F'
d06,f5=ED 29 38 7C 32 02 02 C9;f06,f5=87 CB 23 30 01 82 10 F8
. Fix call to modified @BANK instead of svc @ x'1A16' (disk I/O)
d09,74=CD 77 08;f09,74=3E 66 EF
. Tasker, begins @ x'1C11'
d0b,97=67 CD D6 06;f0b,97=F5 21 78 00
. Tasker, ends @ x'1C3C'
d0b,c2=E1 CD 0F 19;f0b,c2=F1 32 02 02
. Special registers set up in initialization (SYSINIT4/ASM)
. Byte 9, next line = default mem waits: 40=1, 00=0, 80=2, C0=3 M-waits
d0c,d6=D3 84 3A 6F 00 D3 FE 3E 40 ED 39 32 ED 38 36 E6
f0c,d6=47 3E A7 0E 84 ED 41 21 FF FF 22 0E 04 22 1C 00
. Byte 3, next line = default refresh: 02=40, 00=10, 01=20, 03=80 T-states
d0c,e6=FC F6 02 ED 39 36 AF ED 39 38 ED 39 39 3E 88 ED
f0c,e6=56 36 55 ED 79 5E 77 ED 41 BE 72 ED 79 73 ED 41
. Bytes 3 through 9 inserted to mark BUR$ & BAR$ as "in use" - RRR
d0c,f6=39 3A 3D 32 00 02 32 01 02 18 09 00 00 00 00 00
f0c,f6=3E FE 28 02 3E F8 32 01 02 32 00 02 3A 6F 00 D3
d0d,06=00;f0d,06=FE
. eop
```

memory chips to be accessed for the current read or write cycle while MPA15 selects which half of the chip set will be active.

Our revised Port 84 circuitry lets *RASEN0 and MPA15 map the first two 64K segments of the motherboard's new memory. However, whenever address line A17 goes active, Z65 takes mapping away from Gate Array U5. A16 takes over the duties of *RASEN0 and A15 replaces MPA15. (In a similar manner, when the HD64180 activates its address line A18, reads and writes to the system board memory are blocked and RAM access is directed to the XLR8er board's memory.) XLBOOTK's software settings of the 64180's address translator will not allow the 64180 to activate address line A16 unless A17 and/or A18 are active too, ensuring system board memory access integrity.

In order to ensure that *RAS0 is always active during memory access, we add an adapter to U5 (or U110) to isolate U5-32, -33 and -34 from the original socket and circuits. We then ground the original destinations of these signals at chip pins U30-1 and -12. The *RASGATE line coming into U30 becomes the sole control of the *RAS0 signal to memory. (Note that the non-GA Model 4 doesn't need this modification. It uses a different bank switching scheme.)

A BETTER WAY:

The 64180 has an internal address translator which is software controllable and which has an addressing resolution of 4K. (In other words, changing the address translator register by a value of "1" causes the outputted address to shift 4K.) Since software can manipulate the 64180's address translator, we really don't need

Port 84 control and are probably better off without it. Also, Houde's fixes WON'T MANAGE MORE THAN 11 BANKS OF MEMORY correctly, though he provides several utilities which work with up to 31 banks. I guess he felt no one would utilize every last bit of the available elbow room.

Frank Slinkman to the rescue. Frank wrote a new(er) @BANK control routine which can manage up to 31 banks, a total of 1 meg of memory. Frank's code eliminates the need for Port 84 and simplifies even further the bank management process. The simpler "Ignore Port 84" hardware circuit works in conjunction with Frank's code to completely free low memory starting at x'0FF4' for more important system drivers. (F.Y.I., two of the three HD64180 versions have an A19 address line to access a full meg of memory, and Frank's code can properly manage it all.) Frank's XLBOOTS and XLSYS0S patches also turn on the original memory manager's bits (at BAR\$ & BUR\$) in an effort to trap anyone not using LS-DOS's @BANK SVC. Be aware that no one's XLR8er support software tests each bank for correct operation during the boot process. If, for example, you loaded PRO-WAM in a phantom bank 9 after building a system MEMDISK in phantom banks 10 thru 14, you could wind up with a vegetable. Will programs fail to run if you opt to not use Port 84? A few may not, but only time and testing will tell.

Frank's code was my model for modifying Michel Houde's XLBOOTA to support Port 84 and 512K. Frank's coding techniques are beautiful. I was able to keep the expansion of Houde's code down to only two more bytes while managing up to 768K of memory. (Yes, the system board will accept another set of 256K chips in those empty sockets, but the XLR8er won't address more than 512K. Too, some changes that were neatly done to the first memory set won't be as neat for the second memory set.) My modifications are in XLBOOTK/FIX.

MEMORY REFRESH AND IGNORE PORT 84:

A revised memory refresh circuit is necessary due to the way Tandy segments memory into 32K banks. Normal ROW/COLUMN address pairs to 64K chips are A0/A8, A1/A9, A2/A10, A3/A11, A4/A12, A5/A13, A6/A14 and A7/A15. (For 256K chips, we add A16/A17 to the list.) This is how the Model 4P is designed. On the non-GA Model 4's, A6/A7 & A14/A15 are paired to control banking. On GA units, A0/A7 & A14/A15 are paired. (Both machines have other pairing changes but these are the important ones for the discussion.) Our refresh problem is in the wiring of the A7 line; it should be strobed during the ROW portion of the memory address cycle, not the COLUMN portion. Since modifying the board to rectify the situation is unwise, Z63 (none, Z65) does it for us electrically. During each refresh cycle, Z63 disconnects A14 from the memory address multiplexer and connects A7 to that position. (A7 still drives its normal address line during refresh. This has no effect; we're not in a read or write cycle.) Z63 lets us properly refresh most brands of 256K chips using the standard RAS refresh mode. A normal Model 4 can't use Z63 and 256K memory since the Z-80 manages only A0 through A6 during refresh.

ADDRESSING MEMORY:

256K memory chips have an address line which is not present on 64K chips, "MA8". We need to present Row and Column addresses to it at the proper times. Z66 is assigned to that chore. It is piggybacked on one of the two memory address multiplexers already present on the system board and provides addressing signals to memory when its host does so. The added resistor dampens signal oscillations which occur in the memory address lines. Without the resistor, the address might not be stable when the time comes to strobe it into memory. The resistor's value is matched to similar resistors already used to access memory. Its output is fed to Pin #1 of each memory chip via a Tandy-supplied jumper pin on the system board. (This handy pin doesn't exist on the 4P boards and memory pins #1 are not wired

together by Tandy so wrapping wire jumpers are used to connect pin 1 of each of the memory chips to each other and to Z112.)

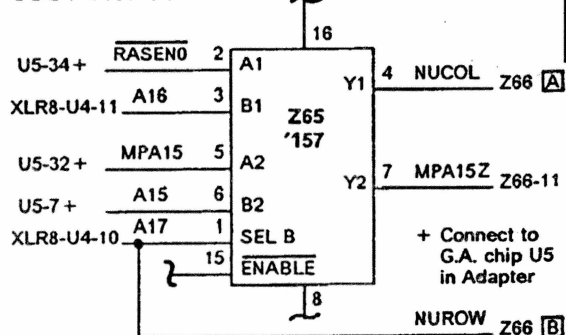
Those of you who are still awake are probably wondering about the second resistor connected to Z66 (or Z63). What we've done is route the refresh address line A7 through the same multiplexer and the second resistor to memory. If we were to route the modified A7 signal through the normal system board channels that particular signal would arrive at the memory chips later than it's kinfolk due to timing delays introduced by each new circuit the signal is passing through. As you will learn in a moment, timing is already critical on the system board and additional delays can render it useless.

HD64180 MANUALS

You've looked for them; I found a place to get them. I'd sure like to stagger the requests to the supplier so they don't get swamped as soon as this issue comes out. Give me a call before 11 pm., Eastern (daylight) time; I'll gladly tell you what's available and how to order them. Publications include a 64180 Product Brief, Hardware Manual, Programmer's Manual, Technical Q&A and reprints of Steve Ciarcia's BYTE articles "The Smart Spooler", "The BCC180 Multitasking Controller" and "The SB180 Single Board Computer" which I referred to earlier in my text.

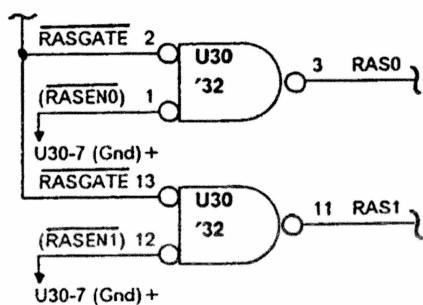
Hitachi is now making two versions of the 64180. One of them has a "Z" suffix to indicate that it is totally compatible with Z-80 timings. The first versions (including the one used in the XLR8er) had some minor timing differences which should not affect the XLR8er. They could give problems to those trying to interface the processor with Z-80 family of peripherals such as the PIO, DMA, CTC, SIO or DART chips, should the experimenter fail to add several timing compensation circuits recommended by Hitachi.

USE PORT 84

MODEL 4
Gate Array
26-1069A 26-1070

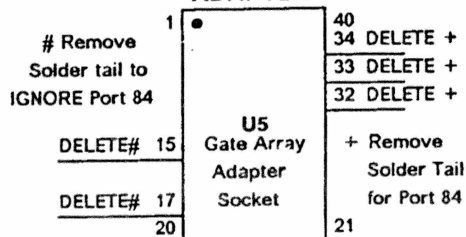
VERIFY BEFORE STARTING:

U1 = Z80	U1-4 ↔ U3-17
U2 = (LS)244	U1-5 < > U3-15
U3 = (LS)244	U1-37 ↔ U2-6
U5 = "4.2"	U3-5 < > U5-7
U30 = (LS)32	U5-32 ↔ U66-11
U63 = 16 pins	U5-33 < > U30-12
U65 = (LS)157	U5-34 ↔ U30-1
U66 = (LS)157	



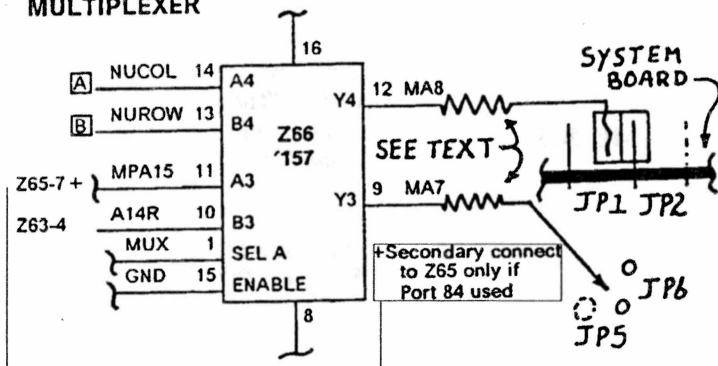
+ NOTE: U30-1 & U30-12 must be isolated from U5-33 & U5-34 by adapter installed in U5.

ADAPTER



NOTE: For IGNORE Port 84, must also solder a jumper across chip pins U5-15, -16, -17

MULTIPLEXER

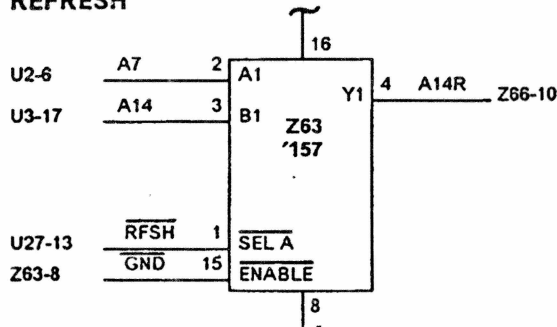


IGNORE PORT 84

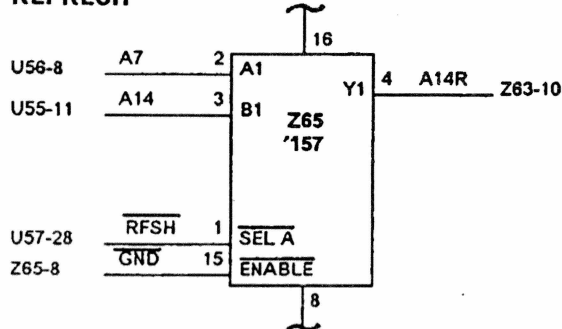
XLR8-U4-11 A16 / NUCOL Z66 A
XLR8-U4-10 A17 / NUROW Z66 B

NOTE: Solder jumper to shoulders of U5-15, U5-16 & U5-17.

REFRESH



REFRESH

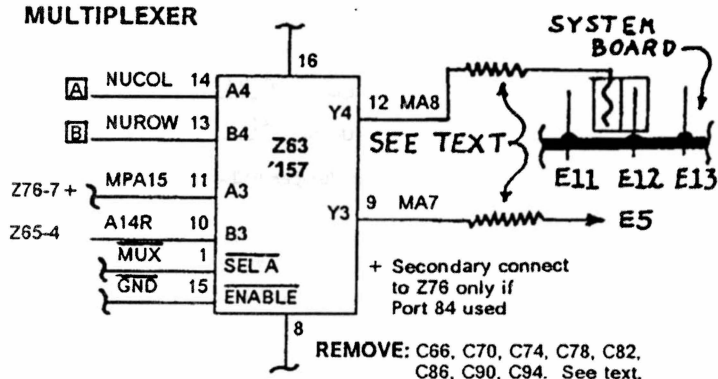


VERIFY BEFORE STARTING:

U55 = (LS)244 | U55-2 ↔ U57-5
 U56 = (LS)244 | U55-9 < > U63-10
 U57 = 280 | U55-11 ↔ U57-4
 U63 = (LS)157 | U55-18 < > U72-8
 U65 = 16 pins | U56-8 ↔ U57-37
 | U63-11 < > U72-13

NOTE: Remove all TMQ IV.ii wiring before installing these changes

MULTIPLEXER



IGNORE PORT 84

XLR8-U4-11 A16 / NUCOL Z63 [A]

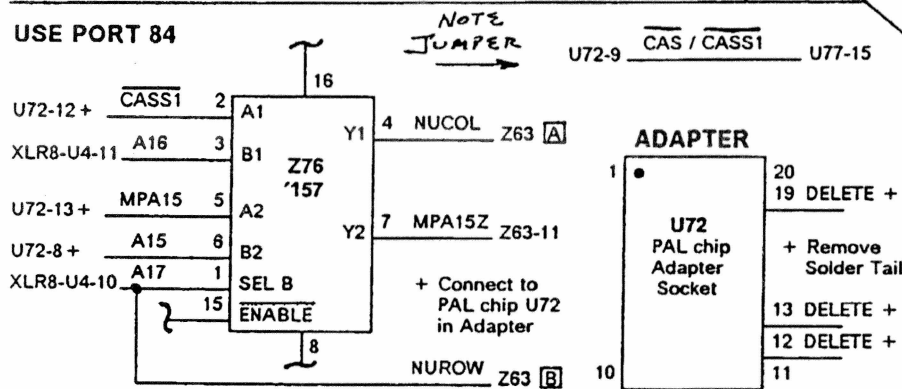
XLR8-U4-10 A17 / NUROW Z63 [B]

NOTE: Remove PAL chip U72. Install these jumpers in U72 socket.

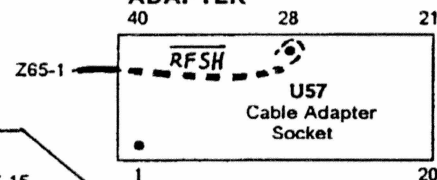
U72-8 A15 / MPA15 U72-13

U72-9 CAS / CASS1 U72-12

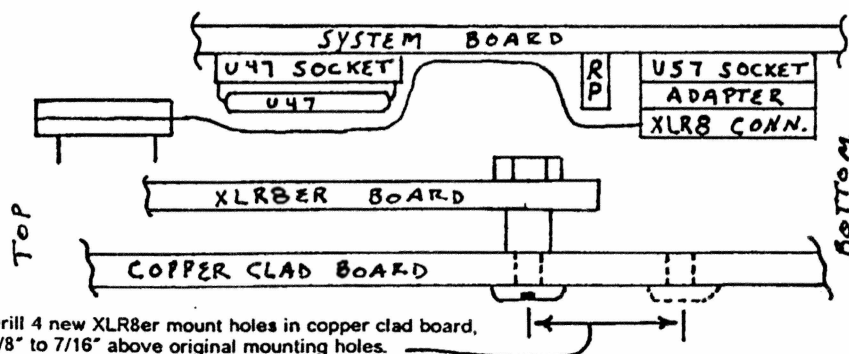
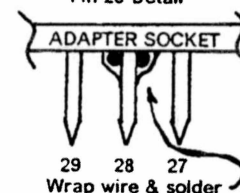
USE PORT 84



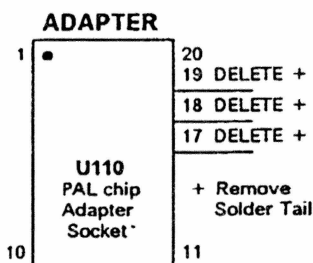
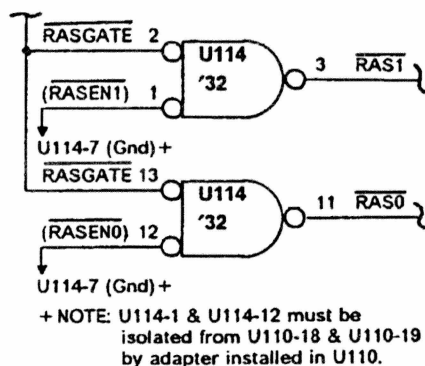
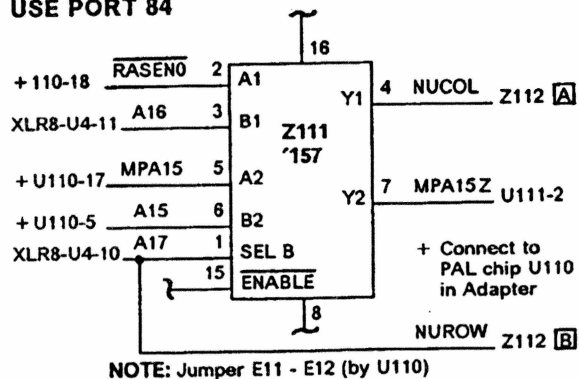
ADAPTER



Pin 28 Detail



USE PORT 84

MODEL 4P
PAL Chip
28-1080

VERIFY BEFORE STARTING:

U133-1 thru U140-1 not interconnected.
U108, U109, U110, U126, U127 = 20 pin PAL or HAL chips

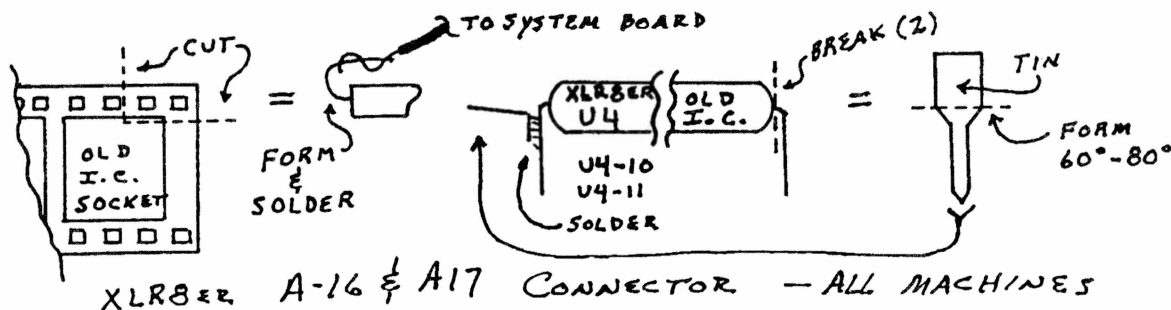
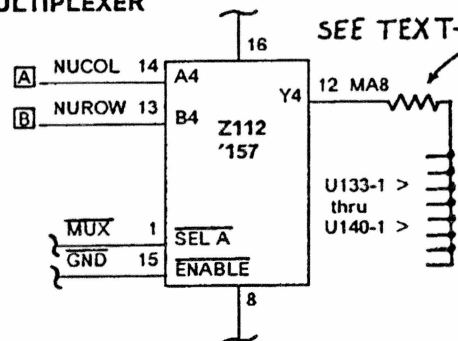
U26 = (LS)244	U28-14 < > U110-5
U68 = (LS)244	U47-37 < > U68-2
U71 = (LS)245	U68-18 < > U111-3
U89 = (LS)273	U110-17 < > U111-2
U111 = (LS)157	U110-18 < > U114-1
U112 = (LS)157	U110-19 < > U114-12
	U111-4 < > U133-9

IGNORE PORT 84

XLR8-U4-11 A16 / NUCOL Z112 [A]
XLR8-U4-10 A17 / NUROW Z112 [B]

NOTE: Jumper E12 - E13 (by U110)

MULTIPLEXER



Floppy Drives

Model 4 Floppy Drives

Fm George L. Aplin, Blackmans Bay, Tasmania: Dear Sir, I am considering the possibility of external drives being used with my Tandy Model 4P which has two double sided 80 track (720K each side) drives, and I would like your advice whether it is possible to use as external drives a 3.5" 720K in 5.25" frame with 3.5" 1.44M in 5.25" frame (PC-AT). If this is possible would you be able to provide any necessary instructions that may be required to make the drives compatible with my Model 4P.

I should say at this stage that I am having my old s/s 40 track single sided drive and a 40 track double sided drive being prepared for use as external drives and intend to add the above drives if this is possible. The project which I am working on will require the preparation of discs to enable me to transfer data to and from other computers, that have different sized disk drives. The external drives will be switched to use only the drives required. If the above is possible please advise me of the cost of the drives mentioned.

Would you also kindly advise me of the cost of the latest Trscross program to be used on my Tandy 4P for use in transferring data and programs to and from MS-DOS and if you see any problem transferring from my D/S 80 track drives.

In the event of a satisfactory answer and ordering any or all of the above please inform me if your firm will accept 'Diners Club International' Credit Card as payment.

Fm MISOSYS, Inc: George, You can add 3.5" drives; they are treated as 80-track 2-sided drives - just like their 5.25" counterparts. You cannot gain any additional usable capacity with the 1.44M drives as they require a different floppy disk controller. The additional capacity comes from a faster data transfer rate and additional sectors per cylinder. The faster transfer rate requires the FDC clock rate

to be 2MHz rather than the 1MHz which is standard on the TRS-80 for 5.25" disk drives.

On the other hand, your Model 4P does not have any capability for external floppy disk drives. There was an article in an old *80 Microcomputing* which I recently reprinted in *THE MISOSYS QUARTERLY* issue IV.iii which shows you how you can add an external floppy disk drive connector. We sell the two connectors needed for that modification, but it requires soldering and modifying your motherboard.

I sell 3.5" 720K drives in a 5.25" frame for \$85. I also have available a case for \$60 which houses two half-height drives. If you need two, then I recommend the floppy extender cable which extends the two edgecard connectors of the drive to a single connector external to the case. That's \$15. Two drives in a case weigh approximately 11 pounds. That's about \$20 surface mail and \$65 air parcel post. With just one drive, the S&H would drop to about \$18.50 and \$60 respectively.

TRSCROSS does not run on the TRS-80; it runs under MS-DOS. So if you need something to run on your 4P and read/write MS-DOS media, you may need something like Hypersoft's Supercross package.

MISOSYS accepts VISA and MasterCard charges. I am enclosing a brochure of our product line.

Hard Drives

Using Four Hard Drives

Fm Ron Wick: Does anyone know why four hard drives won't work on a Model 4? I can get three drives to work in any of the four positions, but when a fourth drive is added the Model 4 continually reboots. Since any of the four positions on the WD controller work for three drives and the controller has recently been checked out, I don't think the problem is in the controller. An upgraded version of the ROMs on the

WD controller were suggested and tried, but I still can't get four drives on line at once. It seems to be a LDOS or TRSDOS software or a problem with the Model 4 itself. I wrote to Roy about this problem, with my last order, and got my order, but no answer about my problem.

Fm Fred Oberding: Ron, "TRSHD6/DCT" will allow 4 physical hard drives, however Roy's "RSHARD6/DCT" will only allow 2. Are you sure you are in the limits of only 8 logical drives assigned, i.e.; 2 floppies & 6 hard drives or whatever combo of 8 logical drives?

I have seen four physical HD's on a Mod 4 using the older 8x300 controller board, but not the smaller & newer WD1010, that's not to say it shouldn't work, it should. There are two mods for the WD1010 board to alleviate problems with allowing multiple secondary HD's to power up.

HD:35 - Insure that R-6 on WD1010 board is a 10 ohm 1/2 watt resistor.

HD:36 - PAL at U-21 on WD1010 must be a 16L2 device and not a 16L8ACJ.

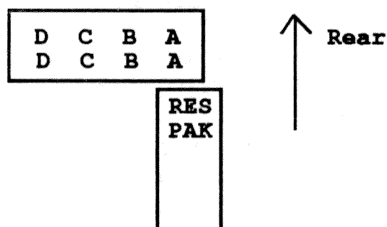
These are both mandatory mods., that are suppose to be done whenever a unit is brought in for service. However, newer techs might not have ever read these older tech bulletins. Hope this helps.

Fm Mark P. Fishman: Ron, Do all four hard drives, by any chance, still have the cable termination resistor pack installed? You are only supposed to have **one** of those on the daisy-chain, in the last drive. Odds are that two will work, maybe even three, but four and you are trying to sink (or source) too much current. Could be related to your reboot problem.

I am assuming that you have tried all four drives individually to make sure that it isn't a particular drive causing the trouble.

Fm Bob Haynes: The resistor pack Mark discusses can be found by removing the round access plate on the underside of each drive unit and is often blue in color

and looking like an IC chip. As he says, it should be ABSENT on all but the LAST drive. While you're in there, also recheck that each drive is uniquely addressed via the "DSx" jumpers. With the drive upside down and front towards you, they are found near the res pack as pictured below. There should be one and ONLY one jumper on each drive, corresponding to the physical drive assignment number. The letters "a" thru "d" represent the positions for physical HD bubbles 1 thru 4 respectively. Hope this helps!



MSCSI6 Software queries

Fm Frank Slinkman: Roy, Some suggestions, comments, problems re the software package that came with my hard disk:

RESTORE6: when all the files fit on one floppy (in this case, a bunch of small files ARCHIVED with S=0), some strange things happen. First, it doesn't seem to know when it's done. When all files are restored, it asks for the next disk.

Second, a file named NEGATE/BAS automatically restores, even though the file before it and the file after it gave the Y/N/C prompt. Third (a suggestion), for the sake of uniformity across the system, make that prompt behave like the Y/N prompt when you BACKUP or PURGE with Q=Y — i.e., make the default NO. As it is now, if you hit <ENTER> three or four times, it defaults to YES.

SD: It would be *real* helpful if the program could set the MOD flag in the /DSK file's directory entry. As it is now, the only way you can tell if a sub-disk has been modified is bring it up and do a DIR of it's directory.

ARCHIVE6: add a parameter which lets you ARCHIVE only MODified files.

Fm MISOSYS, Inc: Frank, Archiving many small files which fit on one disk is not what ARCHIVE was designed for; it would be faster and take less space to use BACKUP. ARCHIVE's primary purpose is to permit you to archive large files which fit onto multiple disks; BACKUP is perfectly suitable to handle hundreds of files onto multiple disks whenever any does not exceed the size of a destination disk.

The parameter handling in RESTORE had a bug; see the MSCRES61/FIX which follows my response to Ralf Folkerts.

The MOD flag is managed by the DOS. SD can not keep the file open in a state such that the MOD flag - or even the date - ever gets updated, as the file never gets "closed". MOD flag and date updating is only done by the @CLOSE routine when the "update permission" is set. This also keeps the "file open bit" set in the directory. Since a user can clobber a diskDISK DCT (by SYSTEMing a FLOPPY driver, for instance), there is no guarantee that DiskDISK (or SD, one and the same) can ever ensure that it controls the disengagement of the "mounted" diskDISK; thus, in order to ensure that the user does not get into a problem downstream with the "File already open" error, it is best to just not attempt to monkey with the mod flag in the "host" file. A small price to pay for less user problems. You have to consider all of the ramifications before you jump in and assume that a given change is beneficial.

In my opinion, archive/restore is not the answer for backup of today's large capacity hard drives. That's why the next project here after release of LS-DOS 6.3.1 is to write a backup/restore utility for the CMS 40M tape cartridge drive I sell for MS-DOS. That's the answer! No changes to ARCHIVE/RESTORE will be considered.

Fm Frank Slinkman: I still think that, whatever mechanical means is used to backup the disk, it ought to manage the

MOD flag. Perhaps something to consider with your disk-to-tape program.

MSCSI's SDFORM6

Fm Ralf Folkerts: Roy, I have a few problems with SDFORM6 and SD6 of your Hard-Drive. When creating a Type '2' SubDisk it will not have bit '3' of Byte X'CD' of the GAT set. That results in DIR - outputs being in the 'old' style without time. The date is wrong, too. I haven't tried it with type '1' SubDisks; type 5 is OK (i.e. has bit '3' SET).

When I dug into that problem using LSFEDII I recognized that SubDisk still uses a password of 'LSIDOS' for the SubDisk's DIR/SYS!

I won't disassemble SDFORM6 to try to find out where the error lies; I'm sure it's a 'take a look and change one minute job' for you! It's not very important since I have LSFEDII and can change that bit quick. Have read the back issues of TMQ but haven't found a patch there. Haven't found any in either LIB 0 and LIB 2 on your CompuServe Forum.

Hope to get the patches ... since then I will still let the hard-disks speed impress me and make use of your great SubDisk - utility!

Fm MISOSYS, Inc: Subdisk, and likewise Diskdisk, can't be shipped to generate the "SYSTEM6" password since they are not 6.3.1 relevant but only 6.3.x relevant. Sometimes I think I gave myself more problems than it was worth when I changed the password of the /SYS files to "SYSTEM6".

The problem with the extended year bit for TYPE=2 subdisks was caused by overlooking code in DiskDISK which specifically changed the configuration byte (GAT+X'CD') to a 47H on a type 2 diskdisk. The following patch command will correct this:

```
PATCH SDFORM6
(D02, CA=4F: F02, CA=47)
```

RESTORE on MSCSI

Fm Ralf Folkerts: Roy, I've restored my Hard-Drive 0 because I wanted to place the /sys and other often-used files around the directory. To do that I've first ARCHIVE6ed all my /DSK files from drive 0, then FMed all other files (including /sys). After I restored the 'normal' files I started to RESTORE6 the /DSK-files. Then I got an error: RESTORE6 won't let me RESTORE to drive 0. It accepts any drive except 0!

RESTORE won't accept the q=n parameter. I've tried it but it always asked for the files to be restored. I've tried (q=n) both with drives specified on the command line and without. I haven't tried answering the RESTORE question with C instead of Y!

When RESTORE6 prompts to insert the source disk and the wrong disk is inserted into that drive it aborts with 'Unknown error code'. That only happens when you specify a source FILE that doesn't exist on the specified source disk. Example: RESTORE6 DFUE/DSK:6 :4. When you insert a disk (by error) that doesn't contain the specified file it will abort with Unknown error code. It works fine if you specify a source-drive and enter an empty (only formatted) disk. It then asks for a new source disk.

It won't find the end of the archive! I had 5 disks of archived files (I've archived all files > 720 sec. from my drive 0). Then I restored using the form RESTORE6 :6 :4 (q=n) to restore all files (note: I've swapped drives 4 and 0 to gain access to the HARD20A partition). It then was queried for all files (see 1.). When it came to the last file of the archive (ASSEMBLY/DSK) it restored the entire file (Sectors to restore: 1441, Restoring sector: 1441) and then asked to *Enter a new source disk and <ENTER>*. Since that was my last disk I pressed enter. Then I was asked again if I want to restore ASSEMBLY/DSK. When I answered "n" it asked me to insert a new source disk.

Fm MISOSYS, Inc: Here's some answers to your queries and/or problems. I checked the code and found that drive :0 was specifically inhibited from selection

as the destination drive for a RESTORE. I can't understand why the test is there because the message prompt, "*Enter DESTINATION drive to restore <0-7>?*" certainly implies that :0 should be acceptable. MSCRESx1/FIX shown below corrects this.

The "Q=N" parameter does not work because of a coding error. This is not necessarily a new report. It's a case of a program getting fixed without the "change" ever making it into the source code. RESTORE comes from RSHARD which comes from VRHARD. Back in 1987, someone reported the problem of RESTORE not using parameters it accepts. That was fixed in RSHARD; the fix was published in TMQ II.ii; but the source code was not corrected. I did not pick up on this when RESTORE was assembled to the MSCSI disk. MSCRESx1/FIX shown below corrects this.

An erroneous error code gets generated when an I/O error is detected because the program has a missing LD C, A instruction. I use macros to generate code streams for invoking DOS service calls. Some years ago, my ERROR macro had that instruction in it so the error code was transferred to the C register as required for DOS 6. At some future point in time, I took over LSI's products; their source code also used a similar macro file, but without the LD C, A instruction in the ERROR macro. I believe that I had then switched to a common macro file without the instruction for future products. But RESTORE was an old product which was bundled with my old hard disk driver for the VR Data drive. It's subsequent reassembly for use with the MSCSI driver package was performed without recognizing the instruction omitted from the macro expansion. It's a case where there was no bug, a bug was introduced simply by a reassembly with an altered macro. The following fix also corrects that.

As far as finding the end of an archive, the documentation is not clear on this. You and others assume that it should be the statement, "*RESTORE keeps track of which diskettes have been completed and will automatically terminate the restoral process once it recognizes that the reconstruction is complete*". I am taking myself off

the hook by keying in on that phrase "*once it recognizes*". RESTORE will ALWAYS recognize a completed job when you are restoring a single file spread across one or more archive disks. RESTORE will NEVER recognize a completed job when multiple files are being restored; simply said, it cannot! The reason is simply explained. When I wrote RESTORE, I had previously dealt with backup/restore programs which required you to always re-insert the backup diskettes in a sequence identical to that in which they were made. That is bad for the user. For one thing, you have to make sure you marked each diskette with a sequence number. Second, in the event that one of the disks becomes lost or damaged, any disk remaining in the backup set could not be restored. You would be out of luck. Thus, I designed my RESTORE so that the set of disks could be freely "shuffled" into any sequence; RESTORE doesn't care in what order the disks are restored since it keeps track of what file segments on each disk have been restored. ARCHIVE doesn't write anything special to the header of the last file segment archived on the last disk, in fact, it wouldn't even know what was the last file to archive until after the fact; thus, there is no way to recover the count of disks used in the archive set by RESTORE. RESTORE doesn't "know" when you have restored the last disk in the set - because "last" is not specific to a particular diskette.

```
. MSCRES51/FIX - Patch to
RESTORE5/CMD on MSCSI5 disk
- 06/12/90
. Apply via PATCH RESTORE5
MSCRES51
. Correct handling of
parameters
D00,90=C3 20 5B
F00,90=32 EA 57
. Permit restore to drive
:0
D01,5C=00 00:F01,5C=28 F3
. Patch for parameters
D07,80=32 EA 57 E1 E5 C3 4F
54
F07,80=00 00 00 00 00 00 00
00
. Eop
```

```

. MSCRES61/FIX - Patch to
RESTORE6/CMD on MSCSI6 disk
- 06/12/90
. Apply via PATCH RESTORE6
MSCRES61
. Correct handling of
parameters
D00,7B=C3 92 2E;F00,7B=32
C2 2B
. Permit restore to drive
:0
D01,43=00;F01,43=28
D01,48=00;F01,48=F3
. Correct error code re-
porting
D06,C2=F6 C0 4F;F06,C2=6F
26 00
. Patch for parameters
D06,EE=32 C2 2B E1 E5 C3 3A
28;F06,EE=00 00 00 00 00 00
00 00
. Eop

```

Initialization JCL

Fm Jeff Joseph, Wheeling IL: I just got a 40 meg hard drive unit and am having some difficulty partitioning it the way I want it. My aim is to use all the space available on the drive while using a minimum number of logical drive slots. With the DOS limit of 13.3 meg in a single volume, I want three partitions of 13.3 megabytes each.

My first attempt to do this, INIT401/JCL, eliminates the 4th drive and divides its 203 cylinders among the other three, resulting in 270 cylinders (I presume MSCSI6/DCT is accepting number of physical cylinders; the Guide indicates a max of 203 logical cylinders are addressable in a single volume).

In my second attempt, INIT402/JCL, I took a different tack and tried to partition by head rather than by cylinder. Three drives are initialized, each getting two heads and all 406 (812?) cylinders. MSCSI6/DCT, when initializing the last two drives, does NOT indicate the previous heads are in use (perhaps because of the 7 unused cylinders?)

INIT403/JCL, a desperation attempt, tries two drives using all 6 heads and 406

cylinders. Interestingly, MSCSI6 doesn't produce any error message at this.

These three JCL files seem to function, but when I look at the free space on the hard disk partitions it doesn't jive with the capacity of the drive.

The JCL file provided with the drive works OK, and I think I understand how it works, but I can't understand why my JCL files won't allow me to get three larger partitions out of the hard disk.

The last README entry is dated 10/25. My computer is a 26-1070 Model 4D with XLR8er and hi-res running LS-DOS 6.3 and Michel Houdé's PEXMEM interface. During the installation there were no other drivers or filters in memory.

I am very pleased with the product, and with the fact that the driver uses only 327 bytes. The clock and joystick options are particularly thoughtful of you.

Fm MISOSYS, Inc: Jeff, Here's the problems associated with your attempts to set up the 40 Megabyte drive using maximum-sized partitions.

The DOS does indeed support a maximum of about 13 Megabytes per logical drive, but that figure is derived from an ability to utilize the maximum of a number of parameters. Here's the parameter limits: a maximum of 256 sectors per cylinder; a maximum of eight granules per cylinder; a maximum of 32 sectors per track; a maximum of 203 logical cylinders; a maximum of eight heads per cylinder. If one multiplies 256 bytes per sector times 256 sector per cylinder times 203 cylinders, the result is 13,303,808 bytes. Dividing that by 1.024 bytes per "K", you get 12992K.

In the case of the Seagate 40Meg drive, its physical parameters are 820 cylinders with six heads. It is left as an exercise to determine that the formatted capacity is 40,304,640 bytes. One would at first glance think that it could be divided into three large partitions. That would be true if the disk driver ignored physical parameters and treated the device as having logical

blocks numbered 0 to 157439 and translated arbitrary logical parameters to physical blocks. This is actually how some systems do treat disk storage devices. CP/M has historically treated disks in that manner. In fact, the SCSI controller addressing of a disk device is by a logical block address - not by an absolute cylinder, sector, and head number as is the case with the Western Digital WD1000-type controllers. The only parameter translation employed, to my knowledge, under a LDOS-type OS is that of associating two physical cylinders to equal one logical cylinder. That's the concept employed in the "double bit" of the Drive Control Table.

Now how can we divide up that 40 megabyte drive using physical parameters, as is the case with our DOS. If you employ a partition with six heads, you can have at most 203 cylinders. Why? Because six heads equals 192 sectors per cylinder. Attempting to use the DBLBIT to include more than 203 cylinders would imply a capacity of two physical cylinders per logical cylinder indicating 384 sectors per cylinder - clearly exceeding the maximum permissible by the DOS. If you want the maximum number of physical cylinders, 406 using the DBLBIT, you have to reduce the number of heads for the partition to four. It could be possible to specify a partition of 268 cylinders and six heads provided the driver reflected those 1608 "logical sectors" to the DOS as 201 cylinders and eight heads and translated sector I/O parameters of cylinder (0-200) and sector (0-255 implying heads 0-7) as physical parameters of cylinder 0-267 and sector 0-191 (implying heads 0-5). Such a translation scheme would require more resident memory space. It would also be more difficult to impose a standard partition separation across competing drivers of alternate DOS products as no standard was instituted to begin with as was the case with MS-DOS, for instance. With great hindsight it would be perfect if a disk-stored disk partition map was uniform and standard in the TRS-80 environment. But we all know that hindsight is 20-20.

Incidentally, in MISOSYS' hard disk drivers, a request for number of cylinders for a partition is neither asking for the

physical or logical number of cylinders; it is asking just for a number. The default value has already been calculated prior to the prompt from the maximum usable based on the number of heads requested. It decides whether it has to use the DBLBIT based on whether or not the entry exceeds the DOS' limit of 203. Thus, your illustration of using two heads and 406 cylinders correctly produces a partition of only 6.65 Megabytes.

Another quirk of my hard disk drivers is that the "heads already in use" display will not show a head-in-use **unless all cylinders of that head have been assigned**. To do otherwise would not be correct as you still could select additional cylinders on that head for a drive assignment.

There's also a reason why MSCSI6 did not reject your entry of six heads and 406 cylinders in spite of its illegality. The driver calculates the default displayed in the message as either 203 or 406 based on the quantity of heads requested (203 for 5-8, or 406 for 1-4). It correctly uses the default if you respond to the number-of-cylinders query with just [ENTER]. However, it does not compare your entry, if made, against that limit but rather against the limit of contiguous cylinders assignable on the heads requested for the partition. What the driver should do is to calculate the default value taking both figures into consideration using a formula of the form:

$\min(\text{contiguous, heads} > 4 ? 203 : 406)$

illustrated with C. But it currently doesn't. True, you can wind up entering a value which exceeds correct maximums and will not work properly. I would rather not work this up as a patch. I'll probably put it on the back burner for another release of the driver.

The length of this response was justified by my consideration of it for TMQ. Thanks for keeping me honest.

MSCSI with two heads/ partition

Fm Ralf Folkerts: Roy, I got the H/A, controller, ... today! Great stuff! I'm really impressed by the speed of the hard-disk (although I use a 'slow' Kalok Octagon Drive!

May there be an error in the MSCSI6/DCT code? I tried to partition my Kalok drive (615 cyl, 4 heads) by both head and cylinder. I planned two partitions with 308 cylinders and two heads and two with two heads and 307 cyl. That confused the MSCSI6! After I had completed the setup of the first two partitions he reported only 6 free cyls left!

I've tried to set up: Drive 2, 2 heads starting at head 1, 308 cyls. Drive 3, 2 heads starting at head 1, 307 cyls. Drive 4 (I planned 2 heads starting at 3, 308 cyls) - but got 6 cyls. left!

Fm MISOSYS, Inc: After testing out the installation of MSCSI6 using a 20 Megabyte drive (a Kalok, as well) partitioned according to your scheme, I rooted out the problem you were having. Seems that during the port of the driver initializer code stream from RSHARD to MSCSI, that portion which dealt with the automatic selection of starting head, given that the requested heads equaled the remaining heads, incorporated a change which introduced the bug. The code was expecting an entry in a table pointed to by the IY index register to contain the maximum number of heads on the drive. What the entry contained was the maximum number of heads remaining on the drive with unallocated cylinders. Given the correct entry, the code would have suppressed the "starting head" prompt and chosen it for you. One piece of code first matches if your selection is equal to the total number of heads on the drive; if so, it will always assume a starting head of zero. That's where the bug got into. In your case, it assumed a starting head of zero which had no available capacity. The bug propagated to a search routine which calculates the total number of contiguous cylinders available using that

head; the "006" result came from "running off the end of the table". It turns out to be an easy fix. The total number of heads existing on the drive is stored in another table pointed to by index register IX. So the fix is MSCSIx2/FIX.

Incidentally, the Kalok is not slow (I know, you quoted that). Compared to 5.25" 20 Megabyte drives, the 40 ms average seek rate for the Kalok is pretty good. That's one reason I switched over to using Kalok 20 Megabyte drives from the Seagate ST-225's.

```
. MSCSI52/FIX - Patch to
MSCSI5/DCT - 06/12/90
. Corrects installation
when less than maximum
heads are used
. Apply via PATCH MSCSI5/
DCT MSCSI52
D01,26=DD BE 02
F01,26=FD BE 00
. Eop
```

```
. MSCSI62/FIX - Patch to
MSCSI6/DCT - 06/12/90
. Corrects installation
when less than maximum
heads are used
. Apply via PATCH MSCSI6/
DCT MSCSI62
D01,04=DD BE 02
F01,04=FD BE 00
. Eop
```

More MSCSI hard-disk queries

Fm Ralf Folkerts: Roy, Now that I'm getting familiar with the Hard-Disk and have almost completed the disk-to-HD conversion I have a few comments, suggestions, and questions:

On my installation disk there where two fix files doubled: SYS0CLK & SYS0CLK6 and SYS3CLK & SYS3CLK6. I've checked the contents of the /FIX files and found out that they where identical (I've installed the /FIXes from TMQ).

The /FIX files where for LS-DOS 6.3.0; I've typed the 6.3.1 /FIXes from TMQ. They are no big problems; just a reminder that you should put the newest /FIXes on the disk and remove the double-versions!

During installation there was a feature I wish the MSCSI SW had; maybe with a new compilation you may include them (provided other users request that feature too): Could you 'generate' *JL output from the Installation Parameters? (i.e. send the parameters entered into FSCSI and, if the drive is already formatted, the current settings to the *JL device?) That would be useful (I think) during MSCSI partitioning, too. Or the HDCHECK to send a list of locked-out areas.

Again, this is not a problem but a feature I missed during installation (because I ROUTED *JL TO *PR to get a list of what I've done)

I don't know the sales-figures of that product but have you thought about re-assembly of FM/CMD? It was the most useful tool I had during disk to HD conversion. I had all my programs on system-disks and planned installed them in separate SD - Files (assigned as drive 1 & 2). A BACKUP was no solution since many of the system-files (not the /SYS but a batch of other files I had on the system-disk (i.e. CLK4,...) would have been copied, too. The solution was FM! I've Set-up drive 1 as the SD-Disk and them used

FM :6 :1 :0 (new,inv,vis,move)

That copied all files from drive 6 (floppy) to drive 1 (SD) provided they didn't exist on drive 0 (the system HD)! I think a version of FM is a MUST for every HD user. If you consider to produce a 'new' version of FM for the 6.3.n DOS you can be sure I would buy one (fair price provided)!

The new format of the documentation is fine. They may be laid next to the computer and don't waste too much place on the desk. The only thing I you could improve is the binding. The spiral-binding you use is good for seldom-used manuals. When you use them much (for reference) the pages will soon begin to tear at the binding (although I think I use

them carefully even the 'new' MSCSI manual started to make problems. Therefore I use to punch holes into them and then put them into a binder (like you did with your RATFOR manual and obviously planned to do with the ProWAM & Mister ED manuals (since they have the holes for three-ring binders punched into)). That's the point when the holes from the ring-binder start generating trouble. I have to affix reinforcement rings on both sides. Since every user should have a 'normal' binder why don't you only punch in holes for three-ring binders (or don't punch anything at all into the docs)?

In the MHD-T34 manual there is a little error on page 10. You write '... is a 2K 2764 device ...'. That should either be 2716 - 2K or 2764 - 8K Byte.

I've low-level formatted my Kalok drive using 3 as the step rate. When I re-invoked the low-level formatter later he reported a step-rate of '2' as current parameter! Should have been '3'?

In the MSCSI manual you give false names for the volumes the HDINIT Jobs initializes. You write that they will be initialized as MSCHARD1, ... They will be initialized as HARD20A, HARD20B, ... That's on page 22.

The section on the 40 MB drive in the README file implies a problem I suspected - but have not found in the manual. Does the XEBEC controller use 1 cyl. for saving the drive's characteristics (leaves only 614 cyl. on a 615 cyl. drive for the use)? Does XEBEC have info on that controller? Do you know what it will cost? Can you give me their address? Same for Dallas Semiconductor!

I think one step SDFORM should perform is to password-protect the /DSK files so that they can't be removed (i.e. allow all access except remove without password) to prevent inadvertent deletion of /dsk - files!

One last point: I will try if a 'correct' layout of the /SYS files will speed up a hard-disk system, too. If so, would you change the HDnnINIT jobs ??

Fm MISOSYS, Inc: I have now corrected the master disk to delete the extraneous (i.e. duplicate) files. I also added copies of the 6.3.1-relevant clock patches to the MSCSI disk so that folks getting a new drive have no need for DiskNotes 4.3 as far as the clock patches are concerned.

Have you ever looked at IFC? That's on GO:CMD. It allows controlled copying of files from one drive to another. It operates from a menu rather than from the command line like FM did. When I took over the LSI products many years ago, I found too much duplication between FM and IFC. After the 6.3 release in 1987, I had to decide what products to modify with extended date support. That decision was based on the sales units of each product. Since FM and IFC were similar products, and since inexperienced users were more likely to prefer IFC's menu-driven commands, I chose to modify IFC and not FM. Some decisions here at MISOSYS have to be based on economics and not just emotion.

One idea I am toying with is a small utility to display information on the parameters of a drive and attempt to determine how it is (was) partitioned. That may prove useful in a few cases. Actually, such a utility for a Radio Shack drive would be quite useful as many folks do not know how their drive was partitioned. There should be information available by interrogating a drive sufficient to regenerate its partitioning information. For instance, after your one and only startup disk is clobbered, a utility could read the drive's track zero sectors to begin determining the location of the first directory (pointer in BOOT/SYS). It could then assume different quantities of heads to locate the directory. Once it found that, all partition information for the first drive would be at hand. Then one would continue to search for the next BOOT/SYS, and so forth. Given sufficient intelligence, such a program could even generate the JCL to re-install the drive. I just haven't had the time for that.

You did not get the single sheet of additional info which I provide when I ship a Kalok drive. It states that the recovered step rate is not correct and should always be entered manually if re-formatting. I'll

note your other corrections.

As far as the 3-hole punching is concerned, some of our manuals (like PRO-WAM) have 3-holes punched because they still remain from when we were able to supply binders. But since I gave up on binders, I felt it important to provide a binding - thus the GBC plastic bindings. I find no justification to concurrently punch 3-ring holes as well. Over the years, I have found too many problems with binders. Plastic binders don't ship well - especially in cold weather. If I ship my manuals 3-hole punched without any kind of binding, I feel it is less than professional. Also, other European customers have complained about the limited availability of the 11" binders because their A4 and B5 standards are not only different lengths but different hole positions. I just feel that if I were to continue to provide 3-hole punched manuals, I would have to provide bindings. After I use up pre-printed pre-punched manuals for a product, you will find no 3-hole punched manuals from MISOSYS. For instance, I duplicate MRAS manuals in-house; those are just spiral bound.

The Xebec drive does indeed reserve one cylinder for the setup information. Xebec is no longer in the business of manufacturing controllers; manuals are hard to come by. I have one, but I have no authority to copy it; it is copyrighted. A Dallas Semiconductor Handbook should provide the info needed. But why do you need that? If there is sufficient need, I could probably provide the information contained in those manuals. Give me a use and a need.

If SDFORM password-protected the /DSK file, then you would have to enter the password each time you invoked SD, or it would have to do some sneaky password workaround. Certainly, you could apply a password and permit read/write access. But each and every user can do that. I'll give it some thought for the next go-around - if there is one. For now, one could use the following:

```
ATTRIB filespec/DSK
(O="SUBDISK", PR=REMOVE)
```

Down The Road With The MISOSYS Hard Disk

Ken Strickler
PO Box 773
Stanwood, WA 98292-0773

Well, it's been a couple of months now since I got the 40 MEGGER up and running, and so far, I couldn't be happier! There are several providers of HARD disks still available for the TRS-80 market, but since I needed the Real Time Clock, and Joystick interface, my choices were limited to just MISOSYS. The choice is still the best, in my case, because the HARDWARE and SOFTWARE were designed by MISOSYS, a company which has supported the TRS-80 since the beginning. Since Roy had already written several packages to support other hard disks, he is well aware of the interfacing problems, and took the time to design the hardware and software to eliminate the problems at the users end.

Those who have purchased the harddisks, probably have also purchased the ZCAT program which helps to keep track of the vast amount of data that can now be stored. The ZCAT program is available in the Golden Oldies Series, specifically the GO:CMD package. ZCAT will catalog some 2100+ files! The PARMDIR command in the GO:SYS package can be used to provide a detailed directory of the disks that you have catalogued.

A typical procedure to utilize your new harddisk could be to partition the disk into volumes, using the SDxFORM command provided with the harddisk, the size required for storing the data on your disks. As we all know, many disks purchased are not full of data, and SDxFORM will allow building disks of just the proper size. Just take a DIR of the disk to be copied, and use SDxFORM to build a one slightly larger on the harddisk. Remember you will have to have a BOOT and DIR record. I usually make it about 9K larger. If I have several disks which make up the purchased program as in TMAKER which had 4 disks, I just add all of the required spaces together, and make 1 large disk, say 720K. When you use the volumes, the SDx command is used to assign vacant

device numbers to the volumes, and then the device looks just like a disk drive! Additionally, I try to use a MEMDISK or ERAMDISK driver for the extended memory so that I can keep my harddisk WRITE-PROTECTED, and keep from accidentally erasing it, or writing to it. Also, if I have a program which makes repeated disk accesses to a particular file, I move that file into the ramdisk. If you have the xlr8er card, a file of up to 320K may be placed in the ramdisk. The anitek modification will allow about 960K if you have the 1 meg option. (64K used as 64K mod 4).

If you haven't decided to get a harddisk for your system yet, and you are still using the 'ole reliable, now might be the time!

XLR8er Board

XLR8er incompatibilities

Fm Peter Van Caesele: Roy, Are you (or someone) keeping track of software that is not compatible with the XLR8er board? If no one is currently doing it, I would do it. I have found that the RATIO instruction in Zbasic 3.0 (M3 version) causes a crash. Programs already compiled with this command work fine. Must be an illegal instruction in the compiler. Peter.

Fm MISOSYS, Inc: I am not keeping such a list; I have received little input on this.

Solution to my XLR8er problem

Fm Frank Slinkman: As some of you may recall, I've been wrestling with problems with my rev A gate array Model 4, with XLR8er and graphics board, for some time now. Among the symptoms were the refusal to run in Mod 3 mode, appearance of characters as though generated by the

keyboard, sudden and mysterious lock-ups and reboots.

Because I have the graphics board, the XLR8er had to be installed in back of the motherboard, with the connecting cable folded 90 degrees, and run over the top of the board.

Last night, I shielded this cable with aluminum foil, covering the foil with tape. The computer has now run nearly 24 hours (at 0,1,80 and 1,1,80, by the way) with absolutely no hint of problem. It runs perfectly in M3 mode, and will now boot up with an M3 DOS (both TRSDOS 1.3 and LDOS) with no problems whatsoever.

This M4, which I was about to junk, is now in perfect order, thanks to a little foil and tape to prevent (presumed) cross-talk between the XLR8er cable and God knows what.

Fm John G. Gelesh: Frank, I laughed when I first heard of the aluminum foil bit. But when I tried installing an XLR8er last summer, I learned my lesson. The XLR8er would work perfectly with a 4.5-inch cable. Not at all with a six-inch cable. Of course one cannot mount an XLR8er with a 4.5-inch cable, so I wrapped the six-inch cable with wax paper and aluminum foil. Result: NO PROBLEMS! I have used this set-up for almost a year now with excellent results. Anyone trying a XLR8er installation must try this mod.

Fm Mark Mueller: John, Out of curiosity, and since I now have a Revision A machine to fiddle with, I set up my RFI testing equipment and diddled with the cable from the XLR8er to the Z-80 socket, looking for potential sources of the problem. Guess what? The longer XLR8er cable RADIATES INTO the Revision A board, causing missed memory accesses, etc., and is a problem on the motherboard. There are some long traces near where the XLR8er has to go and they are efficient little antennas around 6 MHz.

Just for laughs, I set up a test transmitter,

and with as little as 15 mv/m field intensity near the left side of the motherboard (at 6 MHz) the motherboard went haywire. Even with a Z80 in the socket. So..... ya wanna use an XLR8er with a Revision A board? Shield the cable so it won't radiate into the bus and control lines. I didn't pursue it further, but I suspect an unloaded/floating line in there somewhere. And then there is the dissimilar metal RF detector built into some boards due to some kind of formulation problem with the solder (it absorbs moisture over time).

Fm LDOS Support: Ah, that makes a lot more sense! I could never come up with a explanation for the long cable problems that had the problem in the cable itself, and would be fixed by shielding. Note that pre-shielded ribbon cable is available, but it is not very flexible. "Real" shielding can also be commercially purchased to retrofit existing ribbon cable, but I'm sure the cost is considerably higher than aluminum foil and some insulating material.

Fm Mark Mueller: Joe, I don't know what prompted me to do the test, but, "since I had it apart anyway", it seemed like the thing to do. I had heard something similar as a rumor once, with no data to back it up, but my experience working with TTL and CMOS in high RF environments led me to suspect that the motherboard was the culprit and not the XLR8er/cable per se. As for shielding, some 2" clear packing tape and a little foil is probably just as good as the other stuff for shielding the cable. I have some of the "ribbon shield", and it is a pain to work with even for 15-line-wide cable.

XLRHR/CMD and DMA

Fm Jon Roberts to Mike Harrow: Mike, I recently downloaded XLRHR/ARC to try out. My system config is 4P w/xlr8er and 15 meg RS HD.all I get when I run is screen clears and "XLR8er Graphics" is displayed. then the machine sits there and

looks at me. Depressing any key returns me to original screen (dos command lines). Could this archive be corrupt? I have downloaded it four times with fasterm II (used x and y modem) Or am I not doing this right ... xlrhr filename/HR:d? Any help would be appreciated.

Fm Mike Harrow: Jon, Sorry to hear about the problem you're having with XLRHR/ARC. I must apologize to you and others who encounter this problem. I should have included with the documentation additional info to explain why this might occur.

The program is executing properly but DMAC1 of your XLR8er board needs to be enabled via hardware. XLRHR/CMD uses OTIR to display the graphics logo then sets up DMAC1 for a transfer from memory to I/O (graphics board). The DMA won't start the transfer until DREQ1 is pulled low - at pin 54 of the HD64180 chip.

Apparently some XLR8er boards already have this pin low. Unfortunately yours isn't one of them. The result - only the graphics logo will be displayed giving you the impression the program is at fault and doesn't work.

I've modified my XLR8er board to permanently enable DMAC1 so that programs like XLRHR/CMD will work. If you decide to modify your XLR8er please be careful. The mod requires a resistor (150 ohms) be soldered from pin 54 of the HD64180 to ground. Pin 1 of the HD64180 is a convenient ground point. This mod will allow XLRHR/CMD and Frank Slinkman's GRAM/CMD (see TMQ IV.iii) to work.

TRSCROSS™

(Pronounced TRISS-CROSS)

TRSCROSS runs on your PC or compatible, yet reads your TRS-80 diskettes! Copy files in either direction!

The FASTEST and EASIEST file transfer and conversion program for moving files off the TRS-80™ and over to MS-DOS (or PC-DOS) or back

TRSCROSS™
Copyright 1986 by Breeze/QSD, Inc.
All rights reserved

- 1 - Copy from TRS-80 diskette
- 2 - Copy to TRS-80 diskette
- 3 - Format TRS-80 diskette
- 4 - Purge TRS-80 diskette
- 5 - Display directory (PC or TRS-80)
- 6 - Exit

Shown above is the Main Menu displayed when running TRSCROSS on your PC or compatible.

TRSCROSS is as easy to use as it looks to be! The program is very straight forward, well thought out, and simple to operate. TRSCROSS has several "help" features built into the program to keep operation as easy as possible. Just pop in your TRS-80 disk to your PC and copy the files right to your PC data disk or hard disk. *It couldn't be any faster or easier!* Packed in the PowerSoft binder is a typeset instruction manual with Index. All steps are detailed. Advanced features, for those that desire to use them include executing menu options right from DOS or from a .BAT file or macro. This can really speed up transfers when similar operations are performed frequently.

TRSCROSS allows you to "TAG" all files to be moved in ONE pass!

TRSCROSS converts TRS-80 BASIC programs and SuperSCRIPSIT files in ONE PASS while COPYING to MS-DOS!

No need to save your programs or files in ASCII or run a separate conversion program first before transferring. TRSCROSS reads your tokenized BASIC program or SuperSCRIPSIT file directly off your TRS-80 disk and performs the conversion all in ONE pass while being transferred directly to your PC or compatible computer. **Automatically converts most BASIC syntax**, and lines that need special attention can be listed to a printer. (Does not convert PEEKs, POKEs, graphics, machine language calls or sub-routines.)

TRSCROSS will even FORMAT a TRS-80 disk right on your PC! (Handy for those who use both machines!) Former TRS-80 users who no longer have their TRS-80, but still have diskettes with valuable data... this is exactly what you've been waiting for!

TRSCROSS will READ FROM and COPY to the following

TRS-80 double-density formats:

TRSDOS 1.2/1.3, TRSDOS 6.2*, LDOS 5.3*, DOSPLUS, NEWDOS/80*, & MultiDOS.

DOS formats listed above flagged with * signify that earlier versions of these DOS's are readable as well, but one or more sectors may be skipped due to a format problem in that version of the DOS. (Disks that were formatted with SUPER UTILITY™ or SUW4/4P™ do not have this problem.) TRSDOS 6.02.01, or higher should not have this problem. Disks formatted in any 80 track format, or single density are not supported.

TRSCROSS Requires: PC or compatible computer, 128K and a normal 360KB (40 track) PC or 1.2MB (80 track) AT drive. Double-sided operation is fully supported. If you have more than one disk drive, fixed drive, or RAM disk, operation will be much smoother. TANDY 1000 requires more than 128KB memory (DMA). TANDY 2000 is not supported at this time due to a difference in disk controller and floppy drives. "Special" data files (like PROFILE+) would need to be converted to ASCII on a TRS-80 first before they would be of use on a PC or compatible.

If you use both types of computers, or you plan to retire your TRS-80, this is for you. TRSCROSS will allow access to your TRS-80 diskettes for years to come. Copy your TRS-80 word processor data files as well as your Visicalc data files over to MS-DOS and continue using them with your new application.

Only \$89.95

Plus \$3 S&H (U.S.) or \$5 Canadian or \$12 Foreign
Virginia Residents must add appropriate sales tax.

MISOSYS, Inc.
P.O. Box 239
Sterling, VA 22170
Phone: 703-450-4181 (Orders only: 800-MISOSYS)

PowerSoft Products from MISOSYS, Inc

SUPER UTILITY PLUS: *The greatest utility ever written for the TRS-80. Every TRS-80 magazine has said so!*

Five-Star Excellent Reviews in 80-MICRO, 80-US, INFOWORLD, POPULAR ELECTRONICS, FAMILY COMPUTING, Creative Computing & more!

If you use a TRS-80 with disk drives, then this is a must-have program that you will wonder how you did without for so long! Super Utility has won numerous awards, has received many 5-Star reviews and this could be your last chance to purchase a copy at this unheard of price. Super Utility does so many things, you will never use its full potential, but it isn't that hard to use since it is completely menu-driven with the most common defaults built right in. It is configurable for all the popular TRS-80 operating systems and will even allow you to set one drive for one system and another drive for a different operating system and copy files easily between the two. Even between Model I and III or 4, regardless of density, track number, number of sides, or system used. We have thousands of letters in our files over the years about how Super Utility has saved the user from various problems. Super Utility removes or decodes passwords (strips them right off a disk in one pass), reformats a disk without erasing the data, fixes problems, backs up most protected disks, etc. This was the very best utility ever written for the TRS-80 and now is the time to get your own copy. Super Utility has over 65 functions and features. Too many to describe! A fantastic buy. Does not work on hard disks. Our ToolBox or ToolBelt has similar features for hard drive use, as well as floppy. SU+ does not support Newdos/80 double-sided disks.

Specify Model I/III or Model 4/4D/4P: \$4 S&H

\$34.95

LDOS ToolBox (Hard Disk Check, Repair, Modify, much more! Like a "SU+" for hard disk) \$24.95

If you own a hard disk and use LDOS, this is the perfect insurance policy for your data. The LDOS TOOLBOX is like a Super Utility+ for hard disks. Features Disk Check and Disk Repair, Sector Modification, plus many, many other useful utilities that makes using a hard drive even easier. Each program contains a builtin help command, so many times you don't even need to look things up in the manual - just press <Enter> for help! A very wise buy for hard disk users.

Model 4 ToolBelt(same for Model 4 DOS 6 use. OK for 6.3. Like a "SU+" for hard disk) \$24.95

This is similar to the LDOS TOOLBOX, except it is for the Model 4 TRSDOS 6 operating system (all versions).

Back/Rest - Super Fast Hard Disk Backup and Restore. Saves hours of time! For I, III or 4. \$34.95

BACK/REST has proven to be a great time-saver for thousands of TRS-80 hard drive users. When reviewed by 80-MICRO, they gave it FIVE STARS - perfect! It saves hours of time and is very easy to use. BACK/REST can back up 10 megabytes in about 10 minutes and 20 meg in about 30-40 minutes. It also tells you how many disks to have ready. Works under LDOS or TRSDOS 6 (both versions on same disk). Great utility for hard disk users!

Superior Hard Disk Drivers for Tandy disk systems. Mix Model III and 4 easily. \$49.95

These hard disk drivers out-perform the Tandy drivers in many ways. Our drivers allow you to combine LDOS and TRSDOS 6 on the same drive and boot from either system (with floppy disk). They run faster and take much less memory from the system. Only for use with Tandy Hard Drives. Available also for Percom drives.

PowerMail Plus (Please specify Model 4, III/I. 5 Star mailing list-data system! \$4S&H \$34.95

This program was written because all the other mailing list/data base systems couldn't keep track of all the types of data most folks wanted to keep track of. You needed speed, you needed hard drive support, and you needed a crash-proof data structure. PowerMail+ was top-rated (5 stars) in several publications and has never been topped. Works on floppies or hard disk under all popular TRS-80 operating systems. Allows importing of data from several other once popular mailing systems to avoid re-typing. Written in machine language by the author of Super Utility, this program is FAST and sorts up to 10 levels very quickly. If you keep track of names and addresses along with associated data for any situation, this is the one to use. Many churches, organizations and businesses use PowerMail+ for all the different kinds of lists they need to pull from. Each record has 24 user-definable "flags" to allow total customization for your exact needs.

Text-Merge Form Letter Module - Create customized "form letters" and Labels with PowerMAIL+! \$15.00

This optional module for PowerMail allows you to create customized "form letters" or custom labels, lists, etc. with PowerMail Plus and any word processor that saves text in ASCII format. Very easy to use and really gets the effect you want. Allows completely definable report generating from your PowerMail+ data.

PowerDraw (animated TRS-80 screen graphics! Easy to use. Great for kids or adults!) \$19.95

INFOWORLD, 80-MICRO and 80-US magazines really loved this program when they reviewed it. It does many things and is fun to use as well. First, PowerDRAW allows you to create graphics (mixed with text if desired) and save them to disk. It also allows you to create up to 33 "frames" of animation and "play" them like a movie. It also allows you to save the graphics in several modes, including BASIC listings, CMD file format, and others. These can then be merged into your own programs, etc., either in BASIC or machine language! Many of PowerSoft's opening screens were created with PowerDraw. In fact, it even creates animated opening screens (like we use in Super Utility, PowerTool, etc.) to really pep up the program. It also allows you to print the screens on Epson-type and several other type of printers. Lastly, PowerDraw has the ability to load in many types of TRS-80 graphic's and convert them to BASIC listings like a BASIC program generator!

PowerDOT 2.0 for printers. Mix text with graphics - no problem. Build new fonts. \$3S&H \$19.95

This program is similar to PowerDraw, but quite different. It allows you to create "hi-res" type screen graphics combined with text, and allows you to create drawings much larger than your screen. The screen is a "window" to a much larger drawing area and you use the arrow keys to move about the drawing. In a way, it is similar to Macpaint for the Macintosh computer. It also allows you to create custom fonts for ads, etc. Many of our early ads were created with PowerDot. It creates the hi-res effect due to each TRS-80 block pixel being printed as a single dot. Please specify if EPSON, Okidata, Prowriter, or Radio Shack printer.

Note: Technical support of these products is not included at these prices.

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239

703-450-4181 (Orders only to 800-MISOSYS [647-6797])

Unless otherwise noted, include \$2S&H per program package in US; Canada is US+\$1; Foreign is US*3. COD is \$3.50 additional.

Virginia residents add 4.5% sales tax, VISA/MC accepted.

You'll be smooth sailing this summer with a 20 or 40 MB MISOSYS Hard Drive connected to your TRS-80 Model 4.



MISOSYS has been shipping complete drive kit packages since September 1989 which plug into Model 4/4P/4D and Model III computers; let us build one up for you! Our 15.5" x 7" x 5.25" (LWH) beige drive case has space for two half-height drives, 115V/230V 60 watt power supply and fan, hard disk controller (HDC), host adaptor, and a 50-pin SCSI female connector for the host interface.

Our host adaptor, which interfaces the 50-pin expansion port of the TRS-80 (host) to the 50-pin SCSI port of the HDC, sports an optional hardware real time clock using a DS1287 clock module. With its internal battery lifetime in excess of 10 years, never enter date and time again. It even adjusts for daylight saving time! Another available option is a joystick port and Kraft MAZEMASTER joystick with a port interface identical to the old Alpha Products joystick; thus, any software which operated from that joystick will operate from this one.

Software provided with the host adaptor supporting the S1421 and 4010A controllers includes: A low level formatter; An installation utility and driver; A high level formatter used to add DOS directory information; A sub-disk partitioning utility; Utilities to archive/restore the hard disk files onto/from floppy diskettes; A utility to park the drive's read/write head; A utility to set or read the optional hardware clock; A keyboard filter which allows the optional joystick to generate five keycodes; and a utility to change the joystick filter's generated "keystroke" values after installation. Optional LDOS 5.3 software is available.

Twenty megabyte drive packages are currently built with Kalok 3.5" hard drives; Forty megabyte packages use a Seagate ST251 28 millisecond drive. Drive packages are offered as 'pre-assembled kits'. Your 'kit' will be assembled to order and fully tested; all you will need to do is plug it in and install the software. Drive kits require a 50-pin SCSI to 50-pin edgecard host connecting cable (see price schedule).

• Prices are currently in effect and subject to change:

• Complete Drive Kits:

• 20 Megabyte kit:	\$495
• 40 Megabyte kit:	\$645
• Hardware clock option	\$30
• Joystick option	\$20
• Host interface cable	\$20
• Second software interface	\$30

• Piece Parts:

• 20 Meg drive	\$200
• 40 Meg drive	\$345
• Case & Power Supply	\$125
• H/A with software	\$75
• Xebec 1421 HDC	\$95
• Adaptec 4010 HDC	\$95
• Drive power Y cable	\$5
• XT drive cable set	\$5

• Note: freight charges are additional. Allow two weeks for personal checks to clear your bank.

Hard Drive Specials until September 30th

Purchase a drive kit and we'll include the host interface cable and give you your choice of the hardware clock or joystick port options free - up to a \$50 savings. Purchase a H/A and HDC combo and you choose a clock or joystick port option free.

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239
703-450-4181
orders to 800-MISOSYS (647-6797)

VISA and MasterCARD accepted
Virginia residents please include 4.5% State sales tax

Note: Our 800 ORDER LINE now can accept calls from Virginia and CANADA.

Floppy Drives and Accessories

5.25" 360K 1/2-height	\$75+\$4 S&H
3.5" 720K in 5.25" frame	\$85+\$3 S&H
2SV5 drive case & P/S	\$60+\$5S&H
Single drive host cable *	\$10
Dual floppy extender cable **	\$15

Our Model 2SV5 dual vertical external floppy disk drive case will hold two 5.25" half-height disk drives or one standard height drive. * needed for one or two drives; ** needed for two drives. S&H prices are UPS ground to continental U.S.

Our LS-DOS 6.3.1 release has a little something for everyone



- ☆ The DATE command, "Date?" prompt on boot, and the @DATE SVC now support a date range of 32 years; from **January 1, 1980 through December 31, 2011**.
- ☆ **Enable or disable the printer time-out** and error generation with SYSTEM (PRTIME=ON|OFF)
- ☆ Customize the display of the time field in the DIR command to display **12-hr or 24-hr clock time** with SYSTEM (AMPM=ON|OFF).
- ☆ Both ASCII and hexadecimal display output from the LIST command is **paged a screen at a time**. Or run it non-stop under your control.
- ☆ MEMORY displays (or prints) the status of switchable memory banks known to the DOS, as well as a **map of modules** resident in I/O driver system memory and high memory.
- ☆ Specify SYSTEM (DRIVE=d1,SWAP=d2) to **switch drive d1 for d2**. Either may be the system drive, and a Job Control Language file may be active on either of the swapped drives.
- ☆ The TED text editor now has commands to **print the entire text buffer**, or the contents of the first block encountered. Obtain directories from TED, too!
- ☆ Have extended memory **known to the DOS**? The SPOOL command now permits the BANK parameter entry to range from 0-30 instead of 0-7.
- ☆ **Alter the logical record length** of a file with "RESET filespec (LRL=n)"
- ☆ Specify "RESET filespec (DATE=OFF)" to restore a file's directory entry to the old-style dating of pre-6.3 release. Specify "RESET filespec (DATE=ON)" to establish a file's directory date as that of the **current system date and time**.
- ☆ Felt uncomfortable with the *alleged* protection scheme of 6.3? **LS-DOS 6.3.1 has no anti-piracy protection!** MISOSYS trusts its customers to honor our copyrights.
- ☆ Best of all, a **6.3.1 diskette is available as a replacement for your 6.3.0 diskette for \$15** (plus \$2 S&H in US). There's no need to return your current master.
- ☆ The 6.3.1 diskette comes with a 30-day warranty; written customer support is available for 30 days from the purchase date. Versions for the Model 4 and Model II/12 are available. **If you do not already have an LS-DOS 6.3.0, order the 6.3.1 Upgrade Kit with 90 days of customer support for \$39.95 (+\$2 S&H).**

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239
703-450-4181 [orders to 800-MISOSYS (647-6797)]

Our 800 **ORDER**
LINE now can accept calls from
Virginia and
CANADA

PowerSCRIPT

MISOSYS SPECIALS OF THE MONTH

GO:MTC

If you're still using Tandy's Scripsit, then give it full power with **PowerSCRIPT!** Define printer codes for any printer. Embed printer codes in the middle of a line! Alphabetized directories! User-definable print filters! User-definable HELP file! Chain files together at print time from any part of the text. Extended cursor control; Logical exit to DOS; optional automatic linefeed after CR; fetch, chain, and Kill command, and more. Compatible with most Model III DOSs and with Model 4 SCRIPSIT on TRSDOS/LS-DOS. Simply use the enclosed INSTALL program and you will be up and running in no time! Get lots of new features for a very small price. Comes with documentation in a convenient 5.25 by 8.5" booklet; at \$17.47 + \$2 S&H, it's 30% off until 9/30/90!

Don't wait for an accident to happen; have these maintenance tools handy as insurance. GO:MTC programs are designed to provide support services for your computer operation. All programs are 6.3 compatible. **DIRCHECK** performs an integrity check of your disk's directory and repair certain kinds of errors. **FIXGAT** reconstructs a corrupted Granule Allocation Table. **IOMON** traps disk I/O errors giving you hands-on control. **MAPPER** shows just where files are stored on your disk. **RAMTEST** performs an exhaustive test of all RAM in your computer. **UNREMOVE** restores a file inadvertently deleted. Documentation is printed in a convenient 5.5" by 8.5" booklet. 50% off until 9/30/90; Just \$29.98 (+\$2S&H)!