

# THE MISOSYS QUARTERLY

## Special for Spring

### In this issue:

- ☛ It's rude not to interrupt  
by J.F.R. "Frank" Slinkman
- ☛ A Model 4 mouse driver  
by Matthew Kent Reed
- ☛ We are on our own  
an article by Ken Strickler
- ☛ Profile 4+ to FilePro 16/dBASE III  
by David P. Krebs
- ☛ Plus a complete map to *Lair of the Dragon*



## PRICE LIST effective March 1, 1991

Prices subject to change without notice

### TRS-80 Software

Product Nomenclature	Mod III	Mod 4	Price S&H
AFM: Auto File Manager data base	P-50-310	n/a	\$49.95 D
BackRest for hard drives	P-12-244	P-12-244	\$34.95
BASIC/S Compiler System	P-20-010	n/a	\$29.95 B
BSORT / BSORT4	L-32-200	L-32-210	\$14.95
CON80Z / PRO-CON80Z.	M-30-033	M-31-033	\$19.95
diskDISK / LS-diskDISK	L-35-211	L-35-212	\$29.95
DISK NOTES from TMQ (per issue)			\$10.00
DoubleDuty		M-02-231	\$49.95
DSM51 / DSM4	L-35-204	L-35-205	\$49.95
DSMBLR / PRO-DUCE	M-30-053	M-31-053	\$29.95
EDAS / PRO-CREATE	M-20-082	M-21-082	\$44.95 D
EnhComp / PRO-EnhComp	M-20-072	M-21-072	\$59.95 D
Filters: Combined I & II	L-32-053	n/a	\$19.95
GO: Maintenance	n/a	M-33-100	\$49.95 B
GO: System Enhancement	n/a	M-33-200	\$49.95 B
GO: Utility	n/a	M-33-300	\$49.95 B
Hardware Interface Kit	n/a	M-12-110	\$24.95
HartFORTH/PRO-HartFORTH	M-20-071	M-21-071	\$49.95 B
LB Maintenance Utility-M4	n/a	L-50-515	\$19.95
LDOS 5.1.4 User Manual	L-40-020	n/a	\$15.00 D
LDOS 5.3 Mod3 Upgrade Kit	M-10-033	same	\$34.95
LED / LS-LED	L-30-020	L-30-021	\$19.95
Little Brother-M4 (Ver 1.0)	n/a	L-50-510	\$74.95 D
LS-DOS 6.3.1 Upgrade Kit - M4	n/a	M-11-043	\$39.95
LS-DOS 6.3.1 Diskette - M4	n/a	M-11-243	\$15.00
LS-DOS 6.3.1 Upgrade kit - M2/12/16		M-11-002	\$39.95 B
LS-Host/Term	n/a	L-35-281	\$39.95
LS-UTILITY	n/a	L-32-150	\$24.95
MC / PRO-MC	M-20-064	M-21-064	\$79.95 D
Mister ED	n/a	M-51-028	\$39.95 B
MRAS / PRO-MRAS	M-20-083	M-21-083	\$59.95 D
PowerDot (Epson or Tandy)	P-32-21?	n/a	\$19.95
PowerDraw	P-32-220	n/a	\$19.95
PowerDriver Plus (Epson).	P-50-200	P-50-200	\$17.95
PowerMail Plus	P-50-003	P-50-004	\$39.95 D
PowerMail Plus TextMerge	P-50-100	P-50-100	\$15.00
PowerScript	P-50-142	P-50-142	\$24.95
PRO-WAM	n/a	M-51-025	\$74.95 D
PRO-WAM Toolkit	n/a	M-51-225	\$29.95
Programmer's Guide DOS 6.	n/a	M-60-060	\$20.00 B
QuizMaster	L-51-500	n/a	\$19.95
RATFOR-M4		M-21-073	\$59.95 D
RSHARD - R/S HD driver	M-12-013	same	\$29.95
ST80-III	P-35-300	n/a	\$39.95
SuperUtilityPlus	P-32-132	P-32-104	\$44.95 D
SuperUtilityPlus CMD file diskette	P-32-832	P-32-804	\$20.00
Supreme HD Driver (PowerSoft-RS)	P-12-113	P-12-113	\$34.95
TBA / LS-TBA	L-21-010	L-21-011	\$19.95 D
THE SOURCE 3-Volume Set	n/a	L-60-020	\$40.00 D
Toolbox/Toolbelt	P-32-203	P-32-245	\$24.95
UNREL-T80	same	M-30-054	\$29.95
UTILITY-I	L-32-070	n/a	\$19.95

### TRS-80 Game Programs

Bouncezoids (M3)	M-55-GCB	\$14.95
Crazy Painter (M3)	M-55-GCP	\$14.95
Frogger (M3)	M-55-GCF	\$14.95
Kim Watt's Hits (M3)	P-55-GKW	\$9.95
Lair of the Dragon (M3/M4)	M-55-021	\$19.95
Lance Miklus' Hits (M3)	P-55-GLM	\$19.95
Leo Christopherson's (M3)	P-55-GLC	\$14.95
Scarfan (M3)	M-55-GCS	\$14.95
Space Castle (M3)	M-55-GCC	\$14.95

### Hardware

TeleTrends TT512P modem (M4P)	H-4P-512	\$74.95	E
XLR8er e/w 256K RAM (M4)	R-MB-004	\$182.00	F
Floppy drives (5.25" 360K 1/2 ht)	H-FD-360	\$75.00	D
Floppy drives (3.5" 720K 1/2 ht)	H-FD-720	\$85.00	B
Floppy Drive Case (2-1/2 ht drives)	H-FD-2SV	\$60.00	F
Hard drive kit e/w clock, 20Meg M3/M4	H-HD-020	\$475.00	?
Hard drive kit e/w clock, 40Meg M3/M4	H-HD-040	\$595.00	?
Hard drive joystick port option	H-HD-JSO	\$20.00	
Hard drive: Kalok KL320	R-HD-020	\$200.00	F
Hard drive: Seagate ST251-1	R-HD-040	\$320.00	F
Hard drive: Seagate ST157A (16B IDE)	R-HD-A40	\$320.00	F
Hard drive: Seagate ST-157N (SCSI)	R-HD-S40	\$350.00	F
Cable: dual floppy extender	H-FD-2EX	\$18.00	
Cable: 4Ft floppy (1 34EDC each end)	H-FD-C04	\$12.50	
Cable: 4Ft M3/M4 printer	H-RC-PM4	\$20.00	
Cable: 4Ft Radio Shack hard drive	H-HD-CT4	\$20.00	
Cable: 4Ft MISOSYS hard drive	H-HD-C04	\$22.50	
Cable: 26-1069 internal floppy	H-FD-2NG	\$20.00	
Cable: 26-1069A/26-1080 internal floppy	H-FD-2GA	\$20.00	
Cable: 26-1080/A internal floppy	H-FD-24P	\$20.00	
Cable: drive power Y	H-HD-CPY	\$5.00	
Cable: XT hard drive set	H-HD-CXT	\$5.00	
Cable: Custom IDC ribbon (M3/M4/M2)	??-??-???	varies	
Standby Power System: 200VA	R-PS-200	\$250.00	?
Standby Power System: 450VA	R-PS-450	\$399.00	?
HD Controller: Adaptec 4010A	H-HD-CA4	\$95.00	D
HD Controller: Xebec S1421A	H-HD-CX2	\$95.00	D
T80 to SCSI host adaptor	H-HD-MHA	\$75.00	D
Zoltrix ZOFAX 96/24 (PC XT/AT)	R-Z1-FAX	\$225.00	F

### The Fine Print

Freight codes: A = \$3.50; B = \$4.00; C = \$4.50; D = \$5.00; E = \$5.50; F = \$6.00; G = \$8.50; H = \$12.00; ? = varies; All unmarked are \$3.00 each; Canada/Mexico add \$1 per order; Foreign use US rates times 3 for air shipment. Virginia residents add 4.5% sales tax.

Virginia residents add 4.5% sales tax. We accept MasterCard and VISA; Checks must be drawn on a US bank. COD's are cash, money order, or certified check; add \$4 for COD.

**MISOSYS, Inc.**

**P.O. Box 239**

**Sterling, VA 22170-0239**

**703-450-4181; Orders only: 800-MISOSYS (800-647-6797)**



*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 9 Megabytes of RAM, Seagate ST4096 80-MB and ST251-1 40 MB hard drives, Expanzl card; Zofax 96/24 fax/modem board; 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 WIN-WORD; Submissions on paper and letters are scanned and converted to text using ReadRight optical character recognition software by OCR Systems. Final page composition is developed using PageMaker 3.1 by Aldus. Clip art comes from Dynamic Graphics' CLIPPER.

## Table of Contents

<b>The Blurb</b>	2
Points to Ponder	2
New Price List	2
Update on 1000 TL/2 and hard drives	3
TMQ Advertising	3
PD Software Librarian	3
DISK NOTES 5.3	4
Old TMQ's available	4
Ribbon cable assemblies	4
Zoltrix ZOFAX 96/24 FAX/modem	4
<b>Letters to the MISOSYS</b>	6
Corrections to last issue	6
Reusing MISOSYS hard drives for MS-DOS	10
SDFORM5 and early LDOS 5.3	10
Converting GOBBLER to TRSDOS 1.3	12
DoubleDuty and DiskDISK; C streams	12
ARCHIVE and RESTORE	14
<b>Inside TMQ</b>	17
It's rude <u>not</u> to interrupt, by J.F.R. Slinkman	17
A Model 4 Mouse Driver, by Matthew Kent Reed	23
We are on our own, by Ken Strickler	27
Profile 4 Plus to FilePro 16/dBASE III Plus, by David P. Krebs	34
Compleat map to <u>Lair of the Dragon</u>	38

### List of Advertisers

J.F.R. Slinkman	22
MISOSYS, Inc.	IFC,37,46-48,RC
Pacific Computer Exchange	IRC
TRSTimes magazine	33

### List of Patches in this Issue

Modifications to David Goblen's XLR8er patches	7
Fix to SDFORM5	11
Fix to RESTORE on MSCSI5 and MSCSI6	15
RSHARD fixes to RESTORE and ARCHIVE	16
VRHARD fixes to RESTORE and ARCHIVE	16



## Points to Ponder

Memory is on the increase. Texas Instruments has recently started to sample a 16 megabit DRAM chip. That makes but a handful of international semiconductor houses moving up to that chip density

And speaking of memory, a new company named Information Storage Devices has unveiled an analog memory chip, the ISD1016. In its simplest sense, connecting a battery, microphone, and speaker to the chip allows for a complete solid-state voice recorder. The chip device includes on chip pre-amplifiers, filters, addressing logic, and power amplifiers.

Not only is solid-state memory capacity being increased, disk drive capacity is increasing while the space occupied continues to shrink. Areal Technology's (wasn't that the Little Mermaid's name?) recently released its MD-2100 2.5" hard disk drive with a capacity of 100 megabytes on a single platter. It uses but 950 milliwatts of power from a single five volt supply.

Speaking of shrinking, Anico Research, Inc now has available an intelligent DB-25 connector, the PC-232, which provides a complete RS-232-C interface functionally equivalent to an RS-232-C serial computer board. By simply adding an address decoder and five volts of DC power, you have a working serial port. The connector includes circuitry incorporating a UART (universal asynchronous receiver transmitter), line drivers and receivers, and a dc/dc convertor (to provide 12V).

As the push towards laptops increases, so does the demand for increasing battery lifespan. Gates Energy Products Inc., of Gainesville, Florida has recently released the latest rechargeable battery, the nickel/metal-hydride combination in C, AA, and 7/5f-size cells. Nickel/hydride cells usually provide twice the energy storage as nickel cadmium batteries. Gates joins Toshiba and Sanyo in the production of the newer type rechargeable.

## The Blurb by Roy Soltoff

Finally, for those TRS-80 hardware tinkers out there, Standard Microsystems Corp., of Hauppauge, NY has released the COM20020 Arcnet IC which combines controller and transceiver functions in a single chip. The 24-pin DIP IC includes a 2Kx8-bit dual-port static RAM buffer, provides a glue-free interface to most microcontrollers, provides on-chip buffers to eliminate memory interface and control circuitry, and includes a variety of hardware diagnostic capabilities. The controller IC can be easily interfaced to a Z80.

Last but certainly not the least, try this out for size. If you have followed and kept up with all the new terminology associated with the computer industry, here's one to stump you. I came across Steve Gibson's Tech Talk column of December 10th, 1990 *Infoworld* which in my estimation, attempts to use every new-fangled term in one sentence. Talking about Geoworks new PC/GEOS, Steve wrote, "*Here we have a just-written, high-performance, multithreaded, pre-emptive multitasking, bit-mapped environment, sporting an outline font-based, intrinsically object-oriented, multisurfaced user interface, built upon a single imaging model, and supporting dynamic memory management and nested, overlapped, and arbitrarily shaped windows.*" Now it comes to light; new technology is an exercise in stringing together the most adjectives!

And a post script to one TMQ reader. Issue V.i of The MISOSYS Quarterly presented Gary Lee Phillips' article on adding a MIDI interface to your TRS-80. Somehow, the entire article was published without ever providing the expansion of the acronym, MIDI. Sorry, but I guess not all folks have followed the development of interfacing computers with musical instruments. MIDI stands for "Musical Instrument Digital Interface".

## New Price List

The previous changes to our price list were significant price reductions in October 1989. I can't remember when there were revisions to our shipping and handling charges. But over the years, MISOSYS has absorbed the increases charged by our freight carriers: USPS and UPS. This issue includes a new price list effective March 1, 1991. Most software and hardware prices remain unchanged; some increase while others decrease. There are changes to freight charges throughout.

Note that UPS now has separate charges for delivery to a residential address versus delivery to a commercial address.; residential deliveries carry a \$.30 premium. We use UPS in two cases: (1) when a shipment exceeds a threshold value, and (2) when a shipment exceeds a threshold weight. The weight threshold occurs at the crossover in freight costs between USPS and UPS; that's typically around eight ounces. USPS is usually used for shipments outside of the US; however, at some other threshold in valuation, UPS is used. Claims processing for insured foreign shipments via USPS is a rather drawn out affair. But MISOSYS will absorb the UPS residential differential. COD's will incur a \$4 additional charge.

## TMQ Schedule

I try to target mailing *THE MISOSYS QUARTERLY* the last week of the respec-



tive month as follows: **Spring issue in February**, Summer issue in May, Fall issue in August, and Winter issue in November (it didn't get out until December).

Note that your mailing label usually has the expiration date of your subscription. For instance, **those with "91/05" 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. You may extend your 91/05 subscription for one additional issue for one-fourth of the normal price: A=\$6.25; B=\$7.50; D=\$8.75; E=\$10. That will adjust your expiration date to 91/08, good through issue V.iv.

**Note to all readers:** As I have decided at this point to continue with Volume VI, you may at this time renew for the four issues of Volume VI and take 10% off of the subscription price: A=\$22.50; B=\$27.00; D=\$31.50; E=\$36. That will adjust your expiration date to 92/08, good through issue VI.iv. The 10% discount is valid only until the mailing of TMQ issue V.iv. With a 91/05 expiration, renewal for five more issues is as follows: A=\$28.75; B=\$39.50; D=\$40.25; E=\$46.

As I write this sentence, it's Monday the 11th of February. This issue will be off to the printers next week. It looks like this is on track

## Update on 1000 TL/2 hard drives

Here's an update on the 1000 TL/2's 8-bit IDE connector as reported on in last issue's Blurb. According to a question from Harry Hodges which appeared in the Answer Line column of *Infoworld* [February 4, 1991], the Miniscribe 8450XT is an XT-type IDE drive. Hodges wrote that they are rare; both Miniscribe and Western Digital made 8-bit IDE drives. There was an 8225XT and 8450XT. Hodges

noted that "you can use an AT-class IDE drive on an XT by using a card made by Silicon Valley Computer. You can use an XT IDE drive on an AT by declaring no drives in the setup and using an XT IDE paddle board [I give up; what's a paddle board?]. You can't hook an XT IDE drive to an AT card or to the IDE connector on an AT motherboard."

Incidentally, a while back, Maxtor bought out the bankrupt Miniscribe company. Maxtor is not without its troubles, though. A third quarter 1990 loss of \$3.96 million following a second quarter loss of \$1.6 million was enough to cause the resignations of Maxtor's president and CEO, George Scalise, as well as the company's sales and corporate affairs vice presidents. Maxtor discontinued production of the old Miniscribe 8051 40M drive.

Meanwhile over at Western Digital, it was announced that WD was ceasing production of its low-end XT and AT storage controllers and low-end video controllers, as well as closing down its facility in Puerto Rico. If you are looking for a low-end WDXT-GEN controller, I have a handful in stock that I'm willing to let go for \$25 each + \$5 S&H.

## 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. Current space rates are as follows (reduced from previous rates):

Full page	\$100
Half page	\$60
Quarter page	\$35
Ninth page	\$15

I compose the ninth-page ad layout so you have no artwork charge. Just submit your text. 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*, send it in.

## 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, unless you were the uploader. 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

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. 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 5.3

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 5.3 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.

## Old TMQ's available

For out of print issues (Volume I and Volume II), we are providing back issues of *THE MISOSYS QUARTERLY* via copier reprint at \$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. **Special price for either all four issues of Volume III, or all four issues of Volume IV: just \$24 per set + S&H \$5 (US), \$6 (CAN), \$14 ZoneD, \$20 (ZoneE).** Here's a synopsis of past issues:

**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; Using XLR8er RAM as graphics video RAM; Upgrade your 4P with external floppy drives; Doubling of files solved; SuperScript document file format; FELSWOOP PRO-WAM export utility.

**IV.iv** Five Twelve K: A better way; Multi-Command; Touch/ccc; Fixes for LS-DOS 6.3.1; DoubleDuty Version 2.6.0 released.

**V.i** 300 Dots on the TRS-80; Tandy 16/6000 Hard Disk Drives; NXWAM PRO-WAM application; A review of M.A.D.'s XROM; a MIDI interface for your TRS-80.

**V.ii** Image processing on the TRS-80

Model 4; A MAKE utility for MC; New XLR8er patches for LS-DOS 6.3.1; FORTH: A language for every application.

## Ribbon Cable Assemblies

MISOSYS uses a Cirris Systems cable tester for 100% testing of custom manufactured ribbon cables. My assortment of test assemblies supports the kinds of connectors typically associated with the TRS-80 microcomputer. 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. Note that we only support the manufacture of straight through pin-for-pin cables; i.e. we won't connect pin 1 to pin 11, etc...

I can provide cables using DB-25 M/F, 20-pin header; 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. This includes such things as replacement FDC-Floppy cables for all TRS-80 Model 4's (4, 4D, 4P); printer cables for Model III/4 or II/12/16/6000; RSHD primary-secondary interconnect cables; RSHD host cable.

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

## ZOLTRIX FAX/Modem

We're selling the Zoltrix ZOFAX 96/24 board for PC's/XT's/AT's/etc., which satisfies my technical requirements. This board is a Group III 9600 baud send/receive fax with 2400 baud modem. It comes with BITCOM and BITFAX software. The BITCOM modem software supports eight file transfer protocols: MODEM CheckSum, XMODEM CRC,



Relaxed XMODEM, YMODEM, YMODEM batch, Ymodem-G, Compuserve B+, and Kermit. The software appears flawless.

The BITFAX software is powerful, yet easy to use. BITFAX allows you to receive a fax in the background while you are running another program; RECV\_FAX is a removable TSR. You may also send a fax either in foreground from the command line, in a batch file, or as a hot-key pop-up, or as a scheduled task in background. The send side allows you to preview your fax before it is sent; allows you to optionally automatically include a cover page, allows you to merge graphics files (TIF, etc) with text, or combine multiple graphics or text files together on one page, or even merge the contents of your database with a text file for mail merge automated sending to multiple destinations. BITFAX supports printing a received fax directly to a 9-pin or 24-pin dot matrix printer or an HP LaserJet laser printer at either 150 DPI or 300 DPI. I print my faxes to my NEC Postscript printer using its HP LaserJet emulation at 150 DPI. I have included a chart, Figure I, which compares BitFax to other fax software packages.

The advantages of a fax board over a stand-alone fax machine are many. First, you have no paper waste; you can print the fax only when you want to. For another, the quality of a fax transmitted from a board - when the original was a text file - is far superior to that of a scanned and transmitted image. For another, the computing power of the host computer can provide the auxiliary fax utilities costing hundreds of dollars - if not thousands - more for equivalent power from the stand-alone unit. A fax board is not necessarily targeted for the user receiving a great deal of unscheduled faxes. And if you need to transmit non-original documents, unless you have a scanner attached to your computer, you won't be able to send them with a fax board. Evaluate your requirements for faxing; if you don't already have a 2400 baud modem, now is the time to plan for this implementation. I think you should consider the Zoltrix ZOFAX 96/24 fax/modem board. All you need is a free 8-bit slot and a telephone line.

Figure I - BitFax Comparison

	BitFax	Quicklink	JTFAX	Everex EFax	CPI Macronix
Background Send/Receive	Yes	No	Yes	No	Yes
Mail Merge	Yes	No	No	No	No
PhoneDatabase in dBASE-type Format	Yes	No	No	No	No
AutoPrint After Receive	Yes	No	Yes	No	No
PrintCapture	Yes	No	No	No	No
Video Capture	Yes	No	No	No	No
Built-inText Editor	Yes	Yes	No	No	No
CoverPageEditor	Yes	Yes	Yes	Yes	No
Run From DOS Batch Command	Yes	Yes	No	No	No
Schedule Multiple BatchTransmissions	Yes	No	No	Yes	No
Run Other DOS Programs after Scheduling	Yes	No	Yes	Yes	Yes
Pop-Up From Any Text Application	Yes	No	No	No	No
Color PCX Pictures Conveted to 16 Shades of Fax	Yes	No	No	No	No
Preview Fax image Before Sending	Yes	No	No	No	No
MergePictures with Text	Yes	No	Yes	Yes	No
Send Large PCX Files	Yes	No	No	No	No
File Formats Supported					
PCX, TXT, DOC, TIFF	Yes	Yes	Yes	Yes	Yes
TIFF-F	Yes	No	No	No	No
DCX	Yes	No	Yes	No	No
IMG	Yes	No	Yes	Yes	Yes
Read Word Processing Files Directly withoutFirstConverting	Yes	No	Yes	No	Yes
MemorySize Required					
Scheduler	128K	512K	180K	?	170K
Receive Fax	78K	N/A	?	?	?

## MISOSYS QUARTERLY SPECIALS

OFFERS EXPIRE MAY 31, 1991

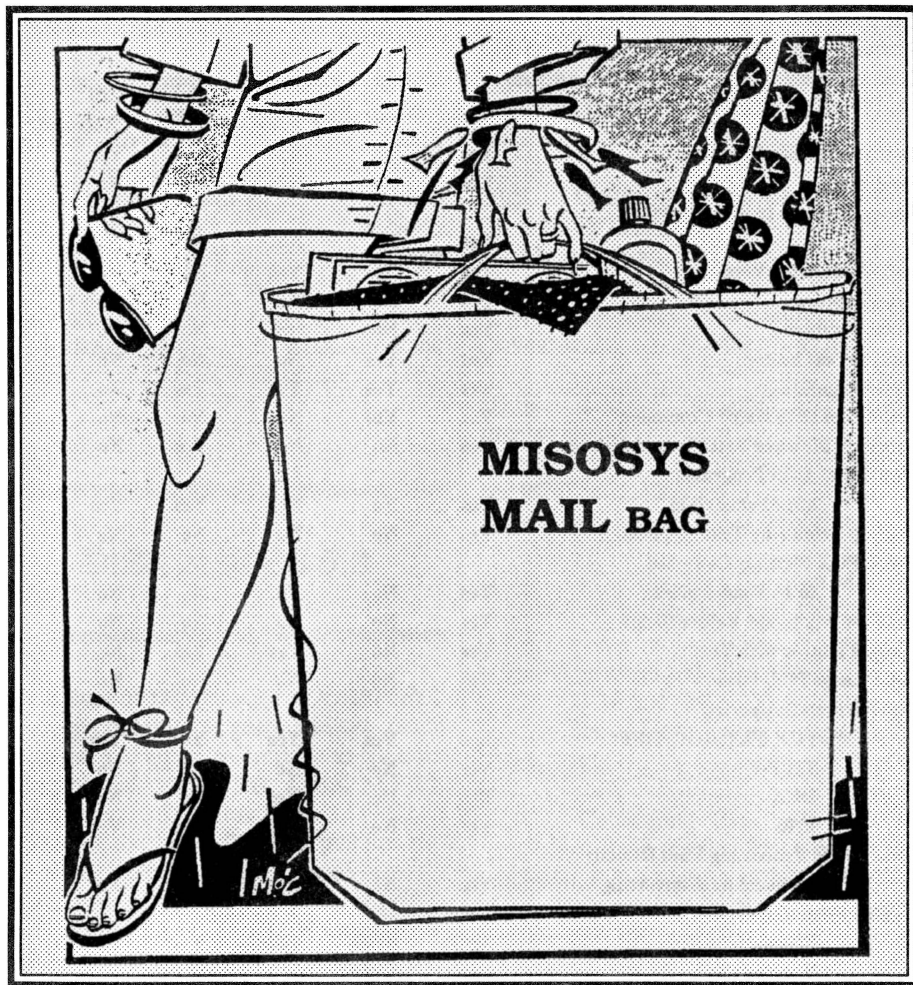
**DoubleDuty 30% off regular price (\$34.97+S&H)**

**TextMerge free with purchase of PowerMAIL Plus**

**\$25 off 20 Megabyte hard drive (\$450.00 + S&H)**

**\$20 off 40 Megabyte hard drive kit (\$570.00 + S&H)**

**The MISOSYS Quarterly, III.i-IV.iv: \$4 each + S&H**



## Corrections for Last Issue

**From Frank Slinkman, Richmond, VA:**  
There were two typos in my article, "Image Processing on the TRS-80 Model 4" — my fault, not yours.

First, in the third paragraph, MacPaint images have 1:1, not 1:2, aspect ratio pixels. Second, in the second paragraph on page 26, a reflection of the filter is used when dithering right-to-left, not left-to-right. I apologize for any confusion this may have caused your readers.

Changing the subject, I have installed David Goben's new @BANK routine (TMQ V.ii) and various fixes in my copy

of LS-DOS 6.3.1H. There is a MAJOR problem with Goben's concept. The problem is that he expropriates the system's BUR\$ and BAR\$ for use as an expanded BUR\$. Several popular programs (SuperScripsit and DeskMate, to name two) rely on BUR\$ and BAR\$, and will not run properly, if at all, if those fields are altered the way Goben alters them.

My solution (the same solution I used for the Richard King mod) — and Michel Houde's solution — is to use a separate XBUR\$. Fortunately, by putting the 16-bit XBUR\$ at the end of the XLR8 module, it exists at x'1000', which means only the MSB of its address is different than that of BUR\$ (x'0200'). This makes patching BOOT/SYS very easy.

This means that the VDCTL module must be moved up two bytes, which takes a

little more patching. Also, Goben's VDCTL code (as does the Houde' code) has an unnecessary OR A instruction after the IN0 A,(CBR) instruction. Unlike standard Z80 I/O instructions, the Z180/HD64180 IN0 instruction manages the Z flag. Thus the net cost of correcting the problem is the loss of only one byte of precious low memory.

Also, in order to thwart non-standard access to the banks, the system's BUR\$ and BAR\$ must each be loaded with 0FEH upon initialization to make such software think it's dealing with only a 64K machine. Hopefully, this will keep it from doing things with the banks we don't want it to do.

I am including two patches, XLBOOTB3/FIX and XLSYS0B3/FIX (the "B3" patches), which make the necessary corrections.

I am also including two other patches, XLBOOTB4/FIX and XLSYS0B4/FIX, which incorporate my adaptation and expansion of an idea by Mel Patrick published in CN-80. The "B4" patches set two memory waits before any keyboard access, and remove them after the access. Even the "fussiest" keyboard should work fine at a setting of 2,1,80.

My "B4" patches do everything the "B3" patches do, but also allow the XLR8er to run at its fastest speed, namely 0,1,80 with 2T refresh cycles. This increases the effective clock rate by about 1.2% over Goben's settings, and allows from 7% to 14% faster RAM access, depending on the instruction used. The resulting speed difference is significant. The "B4" patches only take up an additional 32 bytes of low memory, from x'1016' through x'1035'.

The "B3" and "B4" patches are not compatible — the user must choose one set or the other. If the user wants speed, he should use the "B4" set. If saving 32 bytes of lowmem is more important than speed, he should use the "B3" set.

As a point of interest, I re-arranged Goben's code to make it more efficient (with a net savings of three bytes, enabling me to put XBUR\$ below x'08f0' and save two bytes of low memory). However, this isn't enough



```
.XLBOOTB3/FIX - create XBUR$ @ x'1000' (16-Jan-91 - J.F.R. Slinkman)
. apply to disk which has the David Goben patches via:
.   PATCH BOOT/SYS.SYSTEM6 [using] XLBOOTB3/FIX (O=N)
. also PATCH sys0.sys.system6 [using] xlsys0b3/fix
. fix lomem pointer @ x'0206'
d00,06=16;f00,06=15
. correct MODEND pointer in $FD header
d0c,3f=15;f0c,3f=14
. in @BANK, fix MSB to switch from BUR$ to new XBUR$
d06,8d=10;f06,8d=02
. @VDCTL - JP to fix
d09,9a=02;f09,9a=00
. add XBUR$ @ x'1000', move VDCTL to x'1002' & remove unnecessary
instruction
d0e,00=00 00 ED 38 38 CA 42 0D F5 AF ED 39 38 CD 42 0D E3
f0e,00=ED 38 38 B7 CA 42 0D F5 AF ED 39 38 CD 42 0D E3 ED
d0e,10=ED 21 38 E1 C9;f0e,10=23 38 E1 C9 B6
.eop
```

```
.XLSYS0B3/FIX (16-Jan-91) J.F.R. Slinkman
. change initialization to thwart non-standard bank manipulation -
. allows programs like SuperScripts and DeskMate to run.
. apply to disk which has the Goben patches via:
.   PATCH SYS0/SYS.SYSTEM6 [using] XLSYS0B3/FIX
. should also PATCH boot/sys.system6 [using] xlbootb3/fix (o=n)
d10,39=AF ED 39 38 ED 39 39 21 BF 88 ED 21 3A ED 29 36
f10,39=ED 38 36 E6 FC F6 03 ED 39 36 AF ED 39 38 ED 39
d10,49=21 00 02 3E FE 77 23 77 C9;f10,49=39 3E 88 ED 39 3A C3 CA 20
.eop
```

```
.XLBOOTB4/FIX - 16-Jan-91 J.F.R. Slinkman
. squeeze last cycle out of XLR8er, & make same fixes as xlbootb3/fix
. apply to disk which has the David Goben patches via:
.   PATCH BOOT/SYS.SYSTEM6 [using] XLBOOTB4/FIX (O=N)
. also PATCH sys0.sys.system6 [using] xlsys0b4/fix
. fix lomem pointer @ x'0206'
d00,06=36;f00,06=15
. correct MODEND pointer in $FD header
d0c,3f=35;f0c,3f=14
. in @BANK, fix MSB to switch from BUR$ to new XBUR$
d06,8d=10;f06,8d=02
. KISCAN @x'0983' - JP to trap
d07,83=C3 16 10;f07,83=DD 21 FC
. CTLFF @x'0aec' - JP to trap
d08,ec=C3 29 10;f08,ec=21 01 F4
. @VDCTL - JP to fix
d09,9a=02;f09,9a=00
. add XBUR$ @ x'1000', move VDCTL to x'1002' & remove unnecessary
instruction
d0e,00=00 00 ED 38 38 CA 42 0D F5 AF ED 39 38 CD 42 0D
f0e,00=ED 38 38 B7 CA 42 0D F5 AF ED 39 38 CD 42 0D E3
d0e,10=E3 ED 21 38 E1 C9;f0e,10=ED 23 38 E1 C9 B6
. SCAN @x'1016' & CTL255 @x'1029';put 2 mem waits before kbd access
d0e,16=DD 21 FC 08 3E 80 ED 39 32 CD 87 09 F5 AF ED 39
f0e,16=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
d03,26=32 F1 C9 21 01 F4 3E 80 ED 39 32 CD EF 0A 18 EC
f0e,26=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6
.eop
```

```
.XLSYS0B4/FIX (16-Jan-91) J.F.R. Slinkman
. squeeze last cycle out of XLR8er, and make same fixes as XLSYS0B3/
FIX
. apply to disk which has the Goben patches via:
.   PATCH SYS0/SYS.SYSTEM6 [using] XLSYS0B4/FIX
. should also PATCH boot/sys.system6 [using] xlbootb4/fix (o=n)
d10,34=AF ED 39 32 ED 39 38 ED 39 39 21 BF 88 ED 21 3A
f10,34=3E 40 ED 39 32 ED 38 36 E6 FC F6 03 ED 39 36 AF
d10,44=ED 29 36 21 00 02 3E FE 77 23 77 C9
f10,44=ED 39 38 ED 39 39 3E 88 ED 39 3A C3
.eop
```

difference to warrant yet another article on XLR8er bank handling. Instead, I'll probably include it with my new FEXMEM module, which uses DMA channel 0 transfers to replace LDIRs.

**Fm MISOSYS, Inc:** Frank, It's hard to believe the stupidity of some programs (i.e. programmers) who violate the DOS API (application program interface) by accessing DOS data directly when a service call exists to do the job!

## Corrections to 512K...

**Fm Richard R. King, Raleigh, NC:** Hello, Roy: I'd like to cover several items in this letter, a thanks, an observation regarding Double Duty, and an error in my 512K A Better Way article.

First and foremost, a big THANK YOU for LS-DOS release 6.3.1. I haven't had a lot of time to play around with it yet but those things I've tried appear to work as advertised. (Damn job has me learning my third operating system in a two year time span. I still prefer LS-DOS over all others.)

**512K A MESSED UP WAY:** I was able to check out my 512K patches relative to 6.3.1 and there appears to be no change to their applicability. That's good news. The bad news is that somehow, despite my assurances that "The corrected code is included with this article.", part of it wasn't. Somehow, some way I managed to send you the bad patches (again) for XLSYS0S/FIX. Peter Van Caseele brought that to my attention not long ago. I thought he was kidding but there it was. In an attempt to rectify this mess, the enclosed diskette contains the source code and / FIX code which, hopefully, are all the correct stuff. Maybe I should organize my hard disk better and destroy the bad stuff, if I can find it all.

**DOUBLE DUTY:** Have you ever paid attention to the TIME and DATE func-

tions in each of the three DD windows? Because of the revised code, you no doubt notice how the DOS is set up in the three partitions. Only recently I caught a real mess. After working in one window for several hours I jumped to window / partition 3 and for some reason requested the time. I couldn't believe my eyes - the response told me that the system clock was set to a value only seconds after I did the last boot. Playing around, I found that I could set a different date in each of the three partitions. Ditto the time. None knew what the other was doing, and each window's time would advance only while the window was current. I suppose for most applications this won't hurt a thing, but when writing files from two different partitions, the file date/timestamps don't tell the correct story. I realize you are not the author of the code, but in your revision of the package did you happen to catch this? Do you think there might be a cure for the problem?

**SYSGEN & SYSTEM (SYSRES):** Along with playing around and the receipt of 6.3.1 I ran into another question for which there appears to be no answer. To wit, I've SYSRES'd SYS1/SYS along with my hard disk driver. Now comes 6.3.1 and I install it but I realize that I have an old (6.3.0) version of SYS1 in memory. Try as I might I could not remove it. While I was able to patch DVRHI\$ to "ignore" SYS1, it was still found. I finally had to kill the sysgen function completely, install 6.3.1, re-define my VRDATA hard disk driver back into memory, SYSRES the new copy of SYS1 and then SYSGEN all over again. Lord, how I hated to have to do that! What pointer(s) did I miss? How does one remove a SYSRES'd module?

Along the same lines, how does one force a module such as FORMS/FLT into high memory if enough low memory is available to contain it? Do I need to temporarily patch DVRHI\$ to x'12FF', install the desired filter(s) and then restore DVRHI\$?

**EDAS MACROS:** I'm working on an article submission for TMQ (V.ii?) regarding the writing of SVC macros, expanded to do more function than what I've seen. Two example SVCxxxx/ASM files are included on the enclosed disk-

ette. You'll notice that each file starts out with a separate definition of the actual SVCx macro, followed by the remaining @name macros for that group. Each @name macro is designed to receive input parameters and set up the needed registers, then invoke the SVCx macro. In some cases an @name macro will also preserve registers altered by the DOS supervisor call, then restore them after the user's program regains control.

Here's my question. How can I define a common SVC macro in each group while avoiding the "duplicate macro definition" message during assembly? I tried the IF1, IF2 and IF3 pseudo-ops without success. (The intent is to let the first \*GET'n file define the SVC macro while the other files use a common SVC invocation name.) Is what I'm doing (appending a unique letter to the SVC definition name - ie., SVCx) the only way out, given that I want to avoid a separate 4K \*GET file for just that one macro? Do I make sense?

Also, what is the cost of updated documentation for EDAS? I have an older copy of 4.3A, where the 4.3 documentation was a 14 page addendum to the base EDAS IV code. This was part of the old LC package you once distributed.

I want to take you up on your HA/HDC combo offer. Will the Adaptec board occupy the same space as my current Xebec 1410 board in my VR Data box? (There's plenty of room; the mounting holes are my concern.)

**Fm MISOSYS, Inc:** Richard, What I'll probably do is put the patch revisions just on this issue's DISK NOTES. Folks are starting to get perplexed with all these bank patches.

As far as DoubleDuty is concerned, I've never paid attention to the date and time in each partition, but I can't see how they could not be identical. DoubleDuty doesn't store the system date and time with each partition; those data fields are common to all partitions. The only time you would see a difference, and only in the time, is if you have the time clock on the screen. When you switch partitions, the screen

image is swapped before the SYS1 overlay is reloaded. There would be a second or two time lag after the swap and the new screen appeared before the new partition was in execution. Thus, the time clock displayed on the new screen would momentarily be the last time which was written to that screen. As soon as execution continues into the next second, the screen's time would be updated. So if you have some concrete reproducible effects, please provide them.

I can provide you with no way to remove a SYSRES'd module which has been SYSGEN'd - or otherwise. The DOS provides no "module remove" facility. There is a public domain or shareware utility called "UNRES" which you should be able to find on the LDOS forum (CompuServe PCS-49). UNRES may be able to de-install the resident overlay if it was the last thing installed in memory. I know that UNRES was designed to remove normal filters and drivers; whether it can do it for SYSRES'd overlays is unknown to me.

FORMS/FLT provides no way to force it to install in high memory; it checks low memory for sufficient room first and will install there if sufficient free space exists. Your control over the installation would be the kludge to adjust the low-memory pointer to its maximum value, install the filter, then restore the pointer to its previous value. A simple "LOADHI" utility could be written to easily do that. A command such as LOADHI SET \*FF FORMS would save the pointer value (easily found as the two bytes immediately preceding the \*KI DCB; use @GTDCB to recover the address); issue the remaining portion of the command string via @CMNDR; then restore the pointer value. Where to load LOADHI? You would probably be safe above 7fffH. What I would suggest is to limit the utilization to just the SET command for loading filters. Then you can provide the "SET" requiring only SETHI \*FF FORMS. You could load SETHI above SET's library memory utilization, such as from 2E00H-2FFFFH; that's plenty of room for such a utility. I'll look for your contribution of LOADHI or SETHI; as many of my old college textbooks used to relate, it's left to the reader as an exercise!



You can easily avoid duplicating the definition of a macro by using the IFNDEF conditional. This is used quite frequently for just that purpose. Surround the macro definition with:

```
IFNDEF macroname
```

```
...
```

```
ENDIF
```

and it will work as you need.

Replacement user manuals may be purchased for 25% of a products price; thus with the price of EDAS at \$44.95, a replacement would be \$11.24.

The physical size of the Adaptec 4010A and Xebec S1421 controllers I use for my hard drive is identical to the S1410 Xebec used in your VR Data drive; that's a standard form factor. I'm not sure of the host adaptor since its been some time since I opened up my VR Data Hard Disk III, but my host adaptor is small enough (3.6" x 6.3").

## Odds and Ends

**Fm Ken Strickler, Stanwood, WA:** Dear Roy, I hope that you and yours had a restful and exciting vacation. You sure deserve one, whenever you can get away! I've seen from the last two TANDY catalogs that the MODEL IV has finally been retired! Well, it still had one of the longest runs of any of them.

Have you given any thought to the 1980-1990 COLLECTORS LDOS-LSDOS! You know, GOLD colored, small (5x8) format binder and disks to support model I/III/IV's. Extra binder for PROWAM and the GOLDEN OLDIES pages. Everything might be nice in that format, "C", EnhCOMP, et.al., but I suppose that would be a microscopic market!

Since I haven't heard anything else about

the TAPE BACKUP, I assume that it is still on the BACK BURNER, on SIMMER, and hasn't been knocked in the DIRT!

My cooling fan on the harddisk has developed a little growling noise on startup, but seems to settle down after a couple of minutes. I didn't see any maintenance instructions for the fan, so was just wondering —?

Another question, will the KRAFT TRACBALL work in place of the JOY-STICK on the PORT? If so, which model?

Have you seen the prices on the new systems lately? I can get an 80386/25, VGA monitor 1024x768, .28 pt crt, 80 meg HARDDISK, 1.2m + 1.44m floppy, for less than the original list price of a 4P! (landmark speed 28-29) I have even seen a 80486/25 for less than \$3000 (landmark speed - 114) What do you think the landmark speed of a model III or IV might be? (maybe 1 ?) Let's see, my percent of utilization would drop from 11.2% to .112%, and for only \$3000! (That is less than I had invested in my first Model I complex, and that is not counting inflation!) Not today, thanks anyway!

Here's hoping that this finds the Soltoff family in the best of health, and some of the kids will probably even be in school. More time for Brenda to regain her strength!

**Fm MISOSYS, Inc:** Ken, I never gave much thought to revised editions (i.e. gold editions) of the Model III/4 products. Mostly that's because I would rather spend my time on an endeavor which helps to earn revenue for some years down the road. At almost 47, with my oldest child nearing eight and my youngest nearing three, I have quite a few years left of work before I can retire. That means I have to ensure that my time is spent wisely (which means revenue earning). So don't look for any effort spent re-hashing existing products for the sake of appearance.

The tape backup software interface is on the back burner - I had to get a new stove with more back burners and fewer front ones - so no status is available.

As far as the cooling fan, I would not expect it to growl (has no mouth). The only times I found fans making noise is (1) when a wire is touching the blade, or (2) when the hub of the fan blade is cracked causing the blade to wobble. If the former, move the wire; if the latter, the blade may need to be replaced. The fan blade is just pressed on to the hub.

The joystick port interface on my host adaptor was designed to emulate the old Alpha Products joystick port; I believe the TRSSTICK also worked the same way. That means the joystick uses but five bits of a port: four for the directions and one for the fire button. The switches are binary; i.e. they are either open or closed. Modern joysticks use a variable potentiometer on cross directions to get a varying voltage potential to indicate a "degree of movement". In that way, the number of finite steps capable of being indicated rely on the precision of the A-D converter used to convert the analog voltage level to a digital value. You won't find trackballs using simple switches. If you are firmly interested in a trackball, get one with a serial interface and use either Mark Reed's mouse driver (in this issue), or David Goben's mouse driver which was printed in *Computer News* 80. Alternatively, I believe that MicroLabs sells a hardware interface to support a parallel mouse; Trackballs invariably emulate a mouse. In fact, I use a Chicony keyboard with imbedded trackball on my AST 386 machine because I wanted to move over to a trackball from a mouse. It took some getting used to, but I now feel that a trackball is superior to a mouse.

As far as prices of new hardware is concerned, the prices will invariably be reduced over time - with additional features included. Look at the 486/25 VGA machine you can get for \$2999 at The Price Club? I believe it includes a 110Meg drive or thereabouts. As far as inexpensive equipment goes, a shrewd buyer can save money. Someone who is not concerned if the manufacturer will be out of business in 6 months to a year is also able to save money. If you are concerned about support, stay with a major brand.

## Reusing MISOSYS hard drives for MS-DOS

**Fm Charles A. Ainsworth, Woodbridge, VA:** Dear Roy, Sometime late 1989 I purchased a couple of your drives, 40 MB, complete with all the necessary [hardware] for running on model 4; that included Xebec controllers.

At the time you launched these drives, I believe you stated that one of the ideas behind them was that they would, in course of time, be usable with other types of computers. Well, I have purchased a 286 AT compatible and would like to use one of those drives on it. I need some sort of guidance. Do I simply plug the Xebec disk controller into a free computer slot and connect the drive proper to it with the ribbon cables in the same way as it is connected on the model 4? What else has to be taken care of? Do you have a utility you could provide for low-level formatting? Any help in guiding me to installation would be greatly appreciated.

**Fm MISOSYS, Inc:** Dear Charles, This is in response to your letter concerning moving one of the 40 megabyte bubbles over to your 286 AT compatible. Originally, my attempt was to adapt a low-cost PCXT controller for use with the TRS-80; that proved unworkable. Even if it did, an XT controller can't work in an AT. That's because XT controllers are designed around the 8-bit bus whereas AT controllers are designed around the 16-bit bus. XT's also do not have the hard disk driver in the motherboard ROM; it is located in a ROM on the controller. The AT machines have the driver in ROM. So the two types of controllers are mutually exclusive and can work only in the machines they are designed for. The Xebec controller I use in the TRS-80 setup is not usable with a PC.

What you can re-use is the hard drive bubble itself and the pair of cables con-

necting the drive to the controller. You will need to add the drive internal to your PC-AT. What you need to acquire is an AT controller. Your AT may already have one; some AT clones provide a dual floppy/hard disk controller. Check your operators manual. If your AT has only a floppy controller, you will have to acquire an AT hard drive controller. I have a Western Digital WD-1003-WAH in stock which can do the trick. The price is \$99 + S&H. All the software for dealing with a hard drive should be part of your DOS (or the ROM). You need to use your setup software after installing a hard drive to tell the ROM the drive's characteristics. Depending on the manufacturer of your BIOS, the procedure varies. There should also be a low-level formatter on your DOS utilities disk.

When you first install a drive in an AT, you add its drive type to the setup table. The DOS generally takes some time as it attempts to read the unreadable hard drive. After it restores control to you, you then apply the low-level format. Then you use FDISK to partition the drive. Then you use FORMAT to apply DOS information. Finally, you move the DOS to the hard drive with the SYSTEM command, then copy the visible DOS files, usually to a subdirectory named DOS. Some manufacturers also provide a special install utility for hard drives which do all of the steps automatically; my AST machine provides such a utility. Read your manual for specifics.

## SDFORM5;

### Superscripts recovery

**Fm David J. Kelton, Richmond, Virginia:** Dear Roy: Received my 20 meg hard drive kit some weeks ago, and I must congratulate you on a fine job. You certainly couldn't have made the job any easier to switch from my RS 5MB unit to

your unit — unless of course you would've come with the unit and made the conversion yourself. Absolutely great job.

When I saw the SD and SDFORM files on the MSCSI disk, I really wondered if I needed them. I've been using diskDISK for some time and SubDisk appeared to be the same thing - maybe updated. The idea of it being updated pushed me to try it. I ran into the same problem that Henry Blumenthal did (TMQ V.ii, pg 15) in that SDFORM appeared to create a 5.1.x disk and not the 5.3 disk. Since I had already patched diskDISK's DDFORM to fix this problem (and others — TMQ II.i, pg 102 & IV.i, pg 57), I simply chose to ignore SubDisk and remain with diskDISK. I was still left wondering if I was missing out on something.

I tried the "work around" that you suggested, but I don't believe they work. See the attached JOBL0G1/DAT file in which I created a new SubDisk using SDFORM with all default settings. Then I used COPY, rather than BACKUP which has a known flaw, to copy a file from hard drive :0 to drive :4. The DIR listings show that CAL/CMD was properly dated on drive :0 but went to 5.1.x dating on the SubDisk. Next I CREATED a file on the SubDisk, but it also went into 5.1.x format. Finally, I DATECONV'd the disk and created a second file. This last file finally used the 5.3 dating scheme.

As a matter of fact, I even examined byte X'CB' of the GAT table of the SubDisk to see what version number SDFORM placed there. The value was X'51' which indicates its a 5.1.x disk. SDFORM is really behaving the same way DDFORM did before the patches were installed. Is diskDISK and SubDisk really the same thing just renamed or are there advantages to SubDisk?

Secondly, I would like to address the observations/questions from Harry Woodgear (TMQ V.ii, pg 17) regarding Superscripts under LDOS 5.1.x and 5.3. I have run Superscripts for several years under both DOSes and have had almost no problems — except for Electric Webster's dictionary and that problem is not, repeat not, with the DOS. I was using Superscripts Version 1.02.08 with Elec-



tric Webster Version 1.8 on my Model III with LDOS 5.1.3 and the RS 5MB hard drive. Then in September 1985, RS gave us registered owners Version 1.03, and (fortunately) allowed us to keep the V1.02 disks. I too started having the exact same problems as Harry Woodgear with my Electric Webster dictionary. When I re-installed V1.02.08, the problems went away! Cornucopia Software told me there was no fix to make EW work with V 1.03; so I've been running 1.02.08 since that time.

Regarding the "record length too long" message, you were right that Superscript has a maximum size of 174 "blocks" of text, each block holding about 985 bytes which includes the format control information. Based on my experience, Superscript can sometimes delete out a sector or block and apparently not realize it is available for use again. Thus, after repeated editing, I've seen a file with only the header through the "Dear Roy" on this letter occupy 56K on a disk! Superscript's compress utility will sometimes recover this lost space. The best technique for compressing is to convert to an ASCII file and then back, BUT you lose all formatting, underlining, etc.. David Goben's RECOVERY utility (Computer News 80, Feb., 1989) uses a technique to retrieve the underlines, bolds, etc, but formatting is still lost.

For example consider the directory information below. The file with this letter, MISOSYS/SCR (70 sectors), was "compressed" to MISOSYS2 (38 sectors). When it was "converted" to ASCII as MISOSYS3/TXT (26 sectors) then converted back to Superscript format as MISOSYS/SCR, the "minimum" file size was obtained. The added sectors are mostly needed for the header and index table.

I'm sending along a copy of my files regarding all the information I've collected from 80 Micro, CN80, and TMQ regarding the structure of Superscript files. Using this information, I was once able to recover a file in total that was "lost" when the power went out. Apparently Superscript does not update the file's ending record number (ERN) when it writes to the disk (shame!); only when it closes the file. If the file has been closed

Drive	:3	HARD20D	11/28/90	154H4	Free=	3804.0/	4928.0	Fi=199/256
Filespec	Attrib	LRL	#Recs	EOF	DE	File	Size	MOD Date Time
MISOSYS/SCR	—+ AL	256	70	255	1 s=	20.0	31-Dec-90	20:54
MISOSYS2/SCR	—+ AL	256	38	255	1 s=	20.0	31-Dec-90	20:53
MISOSYS3/SCR	—+ AL	256	34	255	1 s=	8.0	31-Dec-90	20:57
MISOSYS3/TXT	—+ AL	256	26	57	1 s=	8.0	31-Dec-90	20:54

once, then the index table is saved and updated as more text is added. If the user fails to close the file before power down (or a power failure occurs), the index table will point to a record beyond the file's last ERN and you will get a "Record Out of Range" error message. IF YOU HAVE NOT WRITTEN ANYTHING ELSE TO THAT DISK, use the information in the index table to determine the last record that Superscript thought it wrote; then use FED (or another file zipper) to modify the directory entry (DIR+20 & +21) to that value. You may be lucky enough to recover most of the "lost" text.

I now try to remember to open all Superscript files with the trailing exclamation point (i.e., filespec/SCR!) to force LDOS to update the ERN everytime the file is written to. This works well as Superscript must write to the file for every 985 bytes entered or more often when doing a lot of editing and anytime you do an @W command. Thus, LDOS 5.x.x provides a benefit that Superscript's author's should have included in the first place.

I am also sending a copy of this information to Harry in the hopes it will answer some of his questions before the next TMQ.

**Fm MISOSYS, Inc:** Dear David: This is in response to your letter concerning SDFORM5. I'm afraid to say that you have another quirk in your system. When I create a subdisk file using SDFORM5 under LDOS 5.3, it properly stores X'53' into the GAT+CB location and properly sets the extended year bit (bit-3 of GAT+CD). So why do you have a problem? Let's examine why SDFORM5 does what it does.

When you applied a patch to DDFORM, you changed it to always assume that it

was being used under LDOS 5.3; i.e., for proper use under LDOS 5.1, you would have to leave the file unpatched. SDFORM, on the other hand, was generated after the release of the X.3 DOS versions and is programmed to alter its GAT generation based on the version of the DOS it is running under. SDFORM5 retrieves the operating system version from the OSVER\$ storage byte; X'441F' on the MODEL III. It uses the retrieved value as the value to store at GAT+CB in the directory. It then examines the low-order nibble, and if a '3', sets the GAT+CD extended year bit. Thus, if SDFORM5 is invoked under LDOS 5.1, it establishes the subdisk file as a pre-extended year construct. Invoking it under LDOS 5.3, establishes the subdisk file as a post-extended year construct.

My hunch is that either (1) you were actually using LDOS 5.1, or (2) you have patched LDOS 5.3 to set up the OSVER\$ byte as X'51'; the latter was recommended as a last ditch effort to get around a RUNCOBOL problem. Could you have patched the DOS?

Now I shied away from saying that SDFORM5 properly constructed the GAT as I did uncover a problem. Another thing it was supposed to do when determining the DOS was X.3 vintage was to set the "DATA DISK" bit in GAT+CD (that's bit 7). Unfortunately, the code was SET 7,A when it should have been SET 7,B. Here's a patch to correct that.

```
PATCH SDFORM5
(D03,E1=F8:F03,E1=FF)
```

Incidentally, I am amazed at the frequency with which people contact me to learn how to recover data lost when the power went out. For years I have been professing the merits of an UPS or SBS for power protection in just that event. For a Model 4 - even with a hard disk, a small UPS should cost only about \$200-\$250. That's small insurance for the expense of data recovery - or worse, data loss!

**Fm Dave Kelton:** Dear Roy: Thanks for your letter regarding SDFORM5; it took less than 5 minutes to isolate and fix the problem. Actually, I was not using LDOS 5.1 nor had I patched LDOS to set up OSVER\$ to X'51'. I had, in fact, not patched it to be X'53'!

When I purchased my system in January 1987, I dutifully followed your installation instructions and had absolutely no problems switching everything I had from LDOS 5.1 to 5.3; and, to the best of my recollection, never went back to 5.1 again. Somewhere along the line, I began installing all the patches that appeared in TMQ. One such patch SYS0A/FIX (TMQ I.iii, pg 113) didn't take (there was also no comment about what it was supposed to fix). The notes I inserted in the /FIX file indicated that there was no match for the "find" command; so, I assumed that the patch had already been installed by you prior to shipment (SN TC0A027I). I commented out the patch and proceeded on—all other patches installed correctly. When I booted my system from the original, unpatched version of LDOS 5.3, OSVER\$ was still at X'51'. So I started back through the patch list, found the skipped patch, recognized its probable purpose, installed it on a backup copy (no problem this time!?), rebooted, and voila! OSVER\$ was now correctly set. After also installing the patch for SDFORM5 which you sent, SDFORM5 behaved as documented and created a properly formatted 5.3 disk.

Maybe others with an early version of 5.3 have the same problem if they haven't installed the SYS0A/FIX patch. Thanks for pointing me in the right direction.

## Converting GOBBLER to TRSDOS 1.3

**Fm Dale W. Hill, Washita, OK:** Dear Sirs, I recently came up with a Model III which I put into one of our school's classroom. Then I realized that The GOBBLING BOX would be a good game to reward the kids with. Well, here is where I had a problem.

I pulled down my original copy of my Gobbler and tried to convert it to Model III DOS and both lights lit up on my drives. It was copyable directly from my LDOS disk to MODEL IV using the dos shell. It worked perfectly. But I cannot get it to do anything with a MODEL III DOS. It always gives me either "Radio Shack protected file" or error 04. Would you check this diskette and see if anything is wrong with it.

When I got the Gobbling Box, I just put it in with my master diskettes and never used it. Today, I tried the CONV command, Convert command, and directly copying it from :1 :0. I even tried taking off the WRite Tab, which may have ruined the disk. Can you help?

Thank you for your time, and I do believe the kids will enjoy the game if I can get up and running.

**Fm MISOSYS, Inc:** Dear Dale: This is in response to your letter of December 30th concerning The Gobbling Box. The problem with TRSDOS 1.3's CONVERT utility is that it is kind of brain damaged; it will not convert a file if it finds a *password* in the user password field, even though the OWNER PASSWORD FIELD shows total access. The files on the Gobbler disk have an extended date entry being updated by LDOS 5.3; thus, CONVERT thinks the files are protected.

I have altered the directory fields to reflect *blank* passwords. TRSDOS 1.3 should

now be able to convert the files. I would recommend you consider acquiring LDOS 5.3 which is superior to TRSDOS 1.3 in every way. Besides, it is very close to LS-DOS 6.3 in syntax of DOS commands so you would find very little change from Model 4 DOS operation.

As an aside, if you have LS-DOS 6.3.1, you could use its RESET command to *undate* the file using: RESET GOBBLER/CMD (DATE=OFF)

## DoubleDuty and DiskDISK; C Streams

**Fm Hans de Wolf, The Netherlands:** Dear Roy, Now a few (possible) bug reports. The first one is related to the combination of Double Duty and DiskDISK. I have reported this problem before, but I have never received an answer - maybe it is still somewhere in your mail stack, or maybe you just have not received it. This is the problem: I set up a DiskDISK configuration in one DoubleDuty memory partition. If I switch to another DoubleDuty partition, and change the DiskDISK configuration there (assigning other /DSK files to logical drive numbers), and switch back to the first DoubleDuty partition, then the DiskDISK assignments in this first memory partition will be changed by my actions in the second memory partition. It looks as if DoubleDuty does not include the drive assignment tables in the information that is saved when partitions are switched. I am afraid of what will happen to the directories when I open a file on a DiskDISK for output in the first DoubleDuty partition, switch to the second partition and change DiskDISK assignments, return to the first partition and continue to write to the opened file! I did not dare to try this out on my hard disk - restoring a corrupted configuration is definitely not my hobby. Can you confirm my diagnose that this could cause a problem - or is it safe to change DiskDISK



assignments in different DoubleDuty partitions?

The second problem may not be a bug, but can be caused by my misunderstanding of the C programming language, or a piece of missing information in the MC manual. I am working on a program that will print sample sheets of softfonts for the Hewlett-Packard LaserJet/DeskJet printers. To do this the program must first send an escape code to the printer, and then copy the soft font file to the printer. I tried to do this in MC by opening the printer and softfont file as streams, and copy the contents of the file with the code:

```
while ((c=getc(fontfile)) != EOF)
    putc(c,printer);
```

To my surprise, this code did not work and the printer started to print garbage halfway through the download process. After some debugging I found the cause: if the above statement is executed all ASCII TAB characters (ASCII value 9) are translated into multiple spaces. However, MC manual states explicitly for `getc` and `putc`: "Any of the 256 possible binary codes may be input using `getc()`." (page 4-94) and "Putc() is used to output single characters to the file pointer ... 'c' is any of the 256 possible character codes." (page 4-136). From this I conclude that no translation of TAB characters should take place. Can you explain to me what is wrong here: the MC package, the manual, or my understanding of the manual? I could not find a method in the MC manual to specify to the compiler/library that the fontfile and printer should be opened as 'binary' files. Do you have a solution? By the way: I found a temporary work-around solution for this problem by using the `system()` function to execute a DOS command "copy fontfile to \*PR", but this makes my code less portable.

**Fm MISOSYS, Inc: Dear Hans:** This is in response to your letter concerning using DiskDISK with DoubleDuty, and MC stream output. First, DiskDISK can be used with DoubleDuty, however, DoubleDuty assumes that all disk drives associated with the DOS are global to all partitions. This means that if you switch

the /DSK file associated with a drive slot then switch partitions, the new partition will be using the newly associated /DSK file. I would strongly recommend that you refrain from switching disks except to match the disk with the partition. Another way you could get into trouble is to switch floppy disks used by partitions then forget to re-insert the proper one. This would be more a problem for folks using large applications without a hard drive - applications which must access files on floppy data disks.

If there was more space available for partition data storage and DoubleDuty filter interface code, it could be enhanced to store the pack names of all mounted drives and check on matching names after a switch. That would protect against insertion of the wrong floppy, but you wouldn't have a way to correct the /DSK installation. Because I can't control which disk is inserted and guard against the wrong one, there is no reason to save the drive control table information. Actually, there's also no more room left to store the drive tables for each partition. The DoubleDuty manual should have been more explicit in this requirement, rather than just stating: "Before swapping partitions, always be sure that there is a program diskette in Drive 0. The diskette should be the diskette for the program you are moving to." That statement pertains to all disk drives, not just drive :0.

As far as MC stream operation goes, the tab expansion you experienced is easily explained; and it doesn't violate the statement of `putc()` supporting all 256 character values. But stream output - all character stream output - is governed by the `ioctl()` data structure. This structure controls certain aspects of stream output including tab expansion. Check out the `ioctl()` function on page 4-102. What you need to do is turn off tab expansion by setting the `IO_TABSTOP` to 0. Interestingly, this same question was raised by David Lamkins back in 1988. The same answer was given in *The MISOSYS Quarterly* issue II.iv, pages 74,75.

## MSCSI Hard Drive

**Fm L. Dean Dorsey, Little Neck, NY:** Dear Roy, When my hard disk was first received, 66 files were installed on logical drive 0. The directory heading showed 190 directory entries available out of the 256 total directory entries. When the hard disk was returned and re-installed, the same 66 files were placed on logical drive 0. The directory heading showed 189 directory entries available out of the 256 total. I don't know if this is a display error or a true indication of available directory entries. This is the only hard disk point on which I would like an answer.

**Concerning ARCHIVE and RESTORE:** The archiving sequence and the restoring sequence for the same floppy diskette are different. The result is that any archived file that was at the end of one floppy diskette and the beginning of the next floppy diskette ends up being restored in two extents. After multiple archive and restore operations there will be a number of fragmented files unless efforts are made to combine hard disk files into one extent before archiving.

This may be a rare and bizarre occurrence. One of my archive floppy diskettes was apparently exactly full at the end of archiving a file. The next file to be archived was entered into the directory before the archive program detected that the diskette was full. The result was a floppy directory with a filespec having 0 records and 0 k/bytes disk space. With the difference between archiving sequence and restoring sequence, the zero filespec was encountered before all the other files on the floppy were restored. The result was a message I have forgotten and exit to LS-DOS. If one re-started the restore operation with the next archive diskette, one might not be aware of the un-restored files without a verifying file count.

The restore program does not recognize the last archive diskette for a logical drive or sub-disk. After restoring all the files on the last archive diskette, the program is-

sues the message to enter the next archive diskette. The <BREAK> key must be used to return to LS-DOS.

I leave it to your judgement whether to patch the programs for these problems or issue addenda documentation for the MSCSI series of programs or publish the information in *The MISOSYS Quarterly*.

User recommendations: Immediately after archiving, obtain directory information on all the archive diskettes for: (1) identifying and deleting all filespecs with 0 record counts and 0 k/bytes diskette space; (2) Obtaining an archived file count - adjusted for the files archived on two diskettes. When restoring, verify the adjusted file counts obtained from the archiving operation and remember to use the <BREAK> key to exit the restore program when all the files in group have been restored.

**Fm MISOSYS, Inc:** Dear Dean, I can't guess at the discrepancy in the number of available files under your LDOS system partition with the scant data you supplied. Show me a hex listing of the directory and I will have data sufficient to reply. You can get that via LIST DIR/SYS:d (H,P).

Your troubles with ARCHIVE vs RESTORE sequence sound like you are using a set of archive diskettes more than once. I would not recommend that without either reformatting the disks or purging all files. ARCHIVE does not *clear* the disk before performing the archive; that can sometimes be a dangerous operation if you insert the wrong disk - safety first!

RESTORE will never recognize the end of the set of disks. When all have been restored, press BREAK. This was discussed on page 57 of *The MISOSYS Quarterly*, issue IV.iv.

It's true that ARCHIVE will initialize a file on an archive disk before writing to it and finding the disk full. Since the amount of free space available on an archive disk is captured first, the program could be changed to test for a zero amount then request a new disk as required. A patch for that is in this issue of TMQ.

## ARCHIVE and RESTORE

**Fm Richard R. King, Raleigh, NC:** Hello, Roy: This letter is a problem report for two products, one that you currently support and one that you probably don't.

RESTORE6/CMD: Enclosed is a diskette with a group of files selectively ARCHIVED from a much larger list. The source (hard) disk file list is list number "1". When the contents of the enclosed diskette are restored to a hard drive, certain files no longer have the same attributes shown on the (archive) source disk.

List number "2" was printed after doing a full restoration of the files on list "1". List "3" was printed after a partial restore from the master copy of the enclosed diskette.

List "4" contains three paired examples of RESTORE6d file entries in my hard disk's DIR/SYS, and the header record for said file in the ARCHIVE6d disk's file. The ARCHIVE header records properly reflect the source file's directory information, bytes '00' thru '15', proof not attached.

After spending some time looking over the files on the enclosed diskette, I believe it is safe to say that all failures occur when the file in question is 1 record in size and the true EOF is some value other than 255. When this combination occurs, the RESTORED file displays an EOF value of 255, the MOD flag is set on and the MOD DATE and MOD TIME reflect the restoring system's time and date values current during the RESTORation. This failure occurs using the RESTORE6 program from the VRHARD package and also from the MSCSI package.

This problem was brought to my attention by someone who has purchased more than one hard disk package from you. Said person has given me the software package from one of those purchases so that I might do the diagnostics and problem reporting to you.

RESTORE6/CMD: When invoking this program, one of the optional parameters is "Q=N". On both the VRHARD & MSCSI versions of the program, after entering

RESTORE :6 :3 (Q=N

I am still prompted with each source filename before the restore takes place. Trying variations of the parameter (changing case, full spelling, closing parentheses) has not alleviated the problem.

I'm returning your EDAS / PRO-CREATE manual in almost the same condition you sent it to me. The exception is a number of perceptible finger prints on the covers, which I'm ashamed to say are mine. (I tried to keep it clean; honest!) After looking through it and comparing it with my older version, I find that the newer documentation has dropped some tidbits of info found in the older version. While I appreciate your generosity, I've decided to just paste copies of the addendum information in their appropriate locations in my older manual and continue to use it. Many thanks for your generosity though. It is appreciated.

Out of all this, though, comes a possible small benefit to you. I had indexed the older manual version some time ago, so when your newer document came I decided to revise my index to coincide with its contents. It was while doing this index that I discovered the missing info tidbits. One missing tidbit is the deleted explanation of the plus sign "+" which is displayed for macro expansion lines. See the bottom of page 2-41 for an example.

Anyway, enclosed is a copy of that revised index. In addition, I've put ASCII copies of the two indexes (indices) on the enclosed diskette, so that you might make comparisons of them should you so desire. EDASINDX/S40 is my older documentation index, EDASINDX/S43 applies to your revised manual, enclosed.

My indexing scheme may not be the greatest, but here's an explanation of how to read it. A semicolon separates chapter information; a comma separates pages within a chapter. For example

ASTERISK "\*" 1-3; 2-8, 68

means that the Asterisk (also indexed as STAR) character is mentioned on pages 1-3, 2-8 and 2-68. A greater than sign ">" indicates that an entry spans a number of pages. For example,

SWITCHES, ASSEMBLER 2-52 > 57

says that the assembler switches are discussed on pages 2-52 thru 2-57.

Not all page numbers are shown for each index entry due to index space constraints. For example one "bottom of text" entry is

BOTTOM, TEXT "B" "\*" 1-3; 2-68

to show the explanation page (1-3) and a representative example of how it can be used (2-68). To be sure, pages 2-61 and 2-64 also show use of these symbols; no doubt there are others. Note also the inclusion of symbols such as the letter ("B") and asterisk ("\*") to further clarify some of the symbols used in the EDAS document.

Only the EDAS documentation is indexed. Other programs (such as SAID) are mentioned in this index but none of their specifics are indexed.

The diskette file was created using PSCRIPT/SCRIPTSIT. It's arranged in a single column format so that users can determine their own break points for a neat printout. The number of entries just happens to allow breaks at the end of a letter's list so that no letter spans columns.

DOUBLE DUTY 2.6: Works like a champ on my 128K machine, attached to a hard disk. I haven't tried it on my 512K machine because of the VRDATA controller addresses 0 & 1 getting in the way of the XLR8er. My recent query about system time and date problems is null and void. It was obviously fixed by you - they're accurate across all three windows now.

Another question if I may - how does one tell FORMS/FLT (via FORMS) to reset the line counter back to a new page?

By the way, IFNDEF fixed my assembler problem. You, Brenda and the children take care. Hope you've had a good summer.

**Fm MISOSYS, Inc:** Dear Rich: This is in response to your letter of October 8th, 1990. Thought I had forgotten about that one, did you? Well I did finally get around to checking into the RESTORE program and did indeed find the blemish. Your detailed report left no question as to what was happening, and you really helped me out in pinpointing the phenomenon. RESTORE did mess up the attributes and byte offset of a restored file when its size was less than a full sector. ARCHIVE/RESTORE was not necessarily intended to be used for such short files, but I never inhibited that capability. There is some code which tests for a null file, but uses an adjusted record length. Thus, for any file which has only one record, RESTORE would not recognize that the entire file had been restored and would not update the directory information with that recovered from the file's header saved in the archive set. I worked up a patch to correct for that case. Appropriate fix files are MSCRES52/FIX and MSCRES62/FIX for the MSCSI product, RSH54/FIX and RSH64/FIX for the RSHARD software product, and finally, VRH52/FIX and VRH62/FIX for the VRHARD software product.

While I was about it, you noted that the Q=N parameter doesn't work on both the VRHARD and MSCSI versions of RESTORE6. That was fixed up for MSCSI in June 1990 with MSCRES61/FIX; which also appeared in *The MISOSYS Quarterly*, issue IV.iv, page 58. To tell the truth, I never worked up patches for VRHARD. That's now been corrected. I have developed patches VRH51/FIX and VRH61/FIX which correct the parameters of RESTOREx. Also, VRH52/FIX and VRH62/FIX correct the one-record file problem in RESTOREx.

Lastly, as far as fixes go, I have worked up a patch for ARCHIVE stemming from a comment received from Dean Dorsey (see above). The fixes are MSCARC51/FIX and MSCARC61/FIX; to be applied to

both MSCSI and RSHARD. Apparently, as far as what is being patched, the code is identical in both those products. Not so for VRHARD. The patch for ARCHIVE in VRHARD is VRH53/FIX and VRH63/FIX. To make it easy for you to deal with, and since you sent me a disk with information on it, I am providing you a disk with patched files for VRHARD as well as all of the fixes mentioned above - and a few more old ones from TMQ.

As far as DoubleDuty is concerned, I did nothing to it to correct any problem associated with the system date/time displayed for each partition. If it is working now for you, something strange may have been in place before. I won't look for trouble unless I get another report.

Lastly, the FORMS will reset its line counter when it receives a character value of 6D. For instance, if the forms filter is active, a PRINT CHR\$(6); from BASIC will set the counter to 0. Gee, I thought you had a copy of *THE SOURCE*!

Incidentally, I'll mull over the index to the EDAS manual to see where best it could be disseminated. Most likely on DISK NOTES.

```
. MSCRES52/FIX - Patch to
MSCSI5 RESTORE5/CMD - 01/25/91
. Patch corrects error in
directory information of 1-
record files
. Apply via, PATCH RESTORE5
MSCRES52
D04,62=C3 28 5B;F04,62=2C 46 0B
D07,88=2C 46 0B 78 B1 C0 F1 C3
22 58
F07,88=00 00 00 00 00 00 00 00
00 00
. Eop
```

```
. MSCRES62/FIX - Patch to
MSCSI6 RESTORE6/CMD - 01/25/91
. Patch corrects error in
directory information of 1-
record files
. Apply via, PATCH RESTORE6
MSCRES62
D04,36=C3 9A 2E;F04,36=2C 46 0B
D06,F6=2C 46 0B 78 B1 C0 F1 C3
FA 2B
F06,F6=00 00 00 00 00 00 00 00
00 00
. Eop
```



. RSH54/FIX - Patch to RSHARD  
RESTORE5/CMD - 01/25/91  
. Patch corrects error in  
directory information of 1-  
record files  
. Apply via, PATCH RESTORE5  
RSH54  
D04,62=C3 28 5B  
F04,62=2C 46 0B  
D07,88=2C 46 0B 78 B1 C0 F1 C3  
22 58  
F07,88=00 00 00 00 00 00 00 00  
00 00  
. Eop

. RSH64/FIX - Patch to RSHARD  
RESTORE6/CMD - 01/25/91  
. Patch corrects error in  
directory information of 1-  
record files  
. Apply via, PATCH RESTORE6  
RSH64  
D04,36=C3 9A 38  
F04,36=2C 46 0B  
D06,F6=2C 46 0B 78 B1 C0 F1 C3  
FA 35  
F06,F6=00 00 00 00 00 00 00 00  
00 00  
. Eop

. MSCARC51/FIX - 01/28/91 -  
Patch to MSCSI's ARCHIVE5  
. Patch forces immediate  
request for new disk if 0 space  
left  
. Apply via, PATCH ARCHIVE5  
MSCARC51  
D03,3A=C3 77 5C  
F03,3A=21 00 53  
D07,D6=7C B5 CA CD 57 21 00 53  
C3 F2 57  
F07,D6=00 00 00 00 00 00 00 00  
00 00 00  
. Eop

. MSCARC61/FIX - 01/28/91 -  
Patch to MSCSI's ARCHIVE6  
. Patch forces immediate  
request for new disk if 0 space  
left  
. Apply via, PATCH ARCHIVE6  
MSCARC61  
D03,13=C3 E7 2F  
F03,13=21 00 27  
D07,42=7C B5 CA A6 2B 21 00 27  
C3 CB 2B  
F07,42=00 00 00 00 00 00 00 00  
00 00 00  
. Eop

. VRH51/FIX - Patch to VRHARD's  
RESTORE5 - 01/28/91  
. Corrects acceptance of  
parameters  
. Apply via, PATCH RESTORE5  
VRH51  
D00,96=C3 CD 5A:F00,96=32 9F 57  
D07,29=32 9F 57 E1 E5 C3 55 54  
F07,29=00 00 00 00 00 00 00 00  
. Eop

. VRH52/FIX - Patch to VRHARD  
RESTORE5/CMD - 01/28/91  
. Patch corrects error in  
directory information of 1-  
record files  
. Apply via, PATCH RESTORE5  
VRH52  
D04,13=C3 D5 5A:F04,13=2C 46 0B  
D07,31=2C 46 0B 78 B1 C0 F1 C3  
D7 57  
F07,31=00 00 00 00 00 00 00 00  
00 00  
. Eop

. VRH53/FIX - 01/28/91 - Patch  
to VRHARD's ARCHIVE5  
. Patch forces immediate  
request for new disk if 0 space  
left  
. Apply via, PATCH ARCHIVE5  
VRH53  
D02,F0=C3 50 5C:F02,F0=21 00 53  
D07,B0=7C B5 CA 82 57 21 00 53  
C3 A7 57  
F07,B0=00 00 00 00 00 00 00 00  
00 00 00  
. Eop

. VRH61/FIX - Patch to VRHARD's  
RESTORE6 - 01/25/91  
. Corrects acceptance of  
parameters  
. Apply via, PATCH RESTORE6  
VRH61  
D00,74=C3 1D 38:F00,74=32 6A 35  
D06,79=32 6A 35 E1 E5 C3 33 32  
F06,79=00 00 00 00 00 00 00 00  
. Eop

. VRH62/FIX - Patch to VRHARD  
RESTORE6/CMD - 01/28/91  
. Patch corrects error in  
directory information of 1-  
record files  
. Apply via, PATCH RESTORE6  
VRH62  
D03,DE=C3 25 38:F03,DE=2C 46 0B  
D06,81=2C 46 0B 78 B1 C0 F1 C3  
A2 35  
F06,81=00 00 00 00 00 00 00 00  
00 00  
. Eop

. VRH63/FIX - 01/28/91 - Patch  
to VRHARD's ARCHIVE6  
. Patch forces immediate  
request for new disk if 0 space  
left  
. Apply via, PATCH ARCHIVE6  
VRH63  
D02,BC=C3 72 39:F02,BC=21 00 31  
D06,CE=7C B5 CA 4E 35 21 00 31  
C3 73 35  
F06,CE=00 00 00 00 00 00 00 00  
00 00 00  
. Eop

## PRO-WAM Application Pack

Mister ED is loaded with editor applications. All are full screen which make your editing jobs easy. Best of all, these are all PRO-WAM applications so they can pop up even when you are using other programs and applications.

- DED edits disk sectors; • FED edits file records; and • MED edits memory pages (even alternate banks). All use a similar display screen and strikingly similar commands to enable you to edit anything. Get comfortable with one and you will know how to use all three of these editors.
- VED lets you edit the video screen with CARD-type editing. You get cut & paste; with this, you can easily use it as the clipboard facility found on more expensive systems.

- TED is just like the editor you get with LS-DOS 6.3; but ours works from PRO-WAM while you are using other programs! It's friendly, fast, and great for

writing notes when you are right in the middle of a program you can't interrupt.

Mister ED

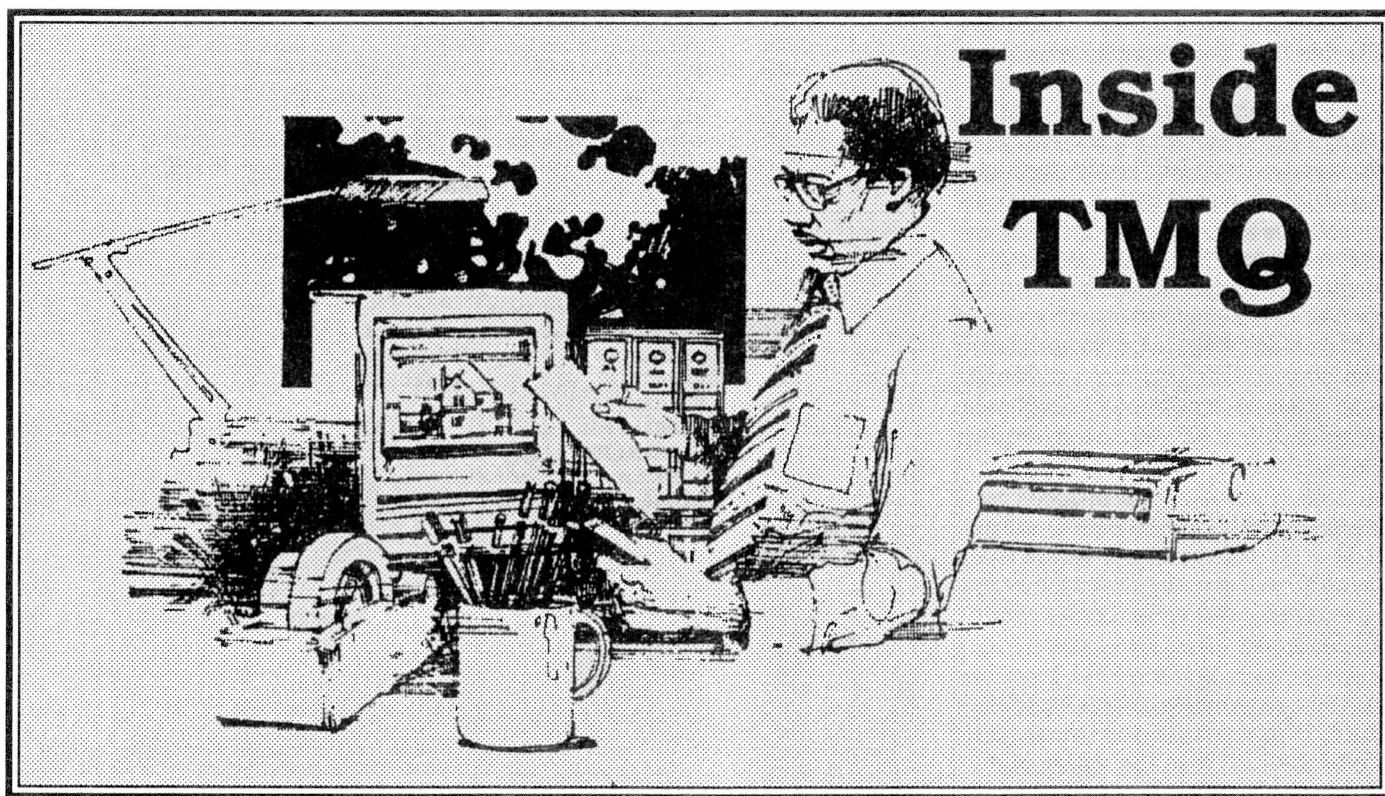
M-51-028

The Programmers' Toolkit provides all you need to know in order to write programs for the PRO-WAM environment. The package includes revised documentation which is in a convenient 5.5" by 8.5" format; the same as PRO-WAM release 2. The included WINLINK device driver supports programmed invocation of applications from any programming language environment which supports device I/O. Also included is a WINDOW/CCC function library for MC, and an assembler source code file for the PHRASE application to better illustrate how to write a PRO-WAM application.

Programmers' Toolkit

M-51-225

MISOSYS, Inc  
P. O. Box 239, Sterling, VA 22170  
703-450-4181



## IT'S RUDE NOT TO INTERRUPT

by J.F.R. "Frank" Slinkman

J.F.R. "Frank" Slinkman  
1511 Old Compton Road,  
Richmond, VA 23233  
(804) 741-0205  
CompuServe 72411,650

When I first got my XLR8er, I phoned Hitachi and got all the manuals for the Z180/HD64180 microprocessor.

I then proceeded to experiment with some of its on-chip features, and put a couple of them to use in software. One use was using DMA transfers from RAM to graphics board ports in the XLR8er version of a hi-res game program I market, SLOTMOD4. I also used DMA transfers to create a simulated hi-res graphics video RAM (see TMQ IV.iii).

Another use was using a PRT (Program-

mable Reloadable Timer) channel to regulate the sampling rate for video data in software to convert an old Xerox Tele-Copier (Group I fax) to a scanner for the Model 4.

At that time I asked a usually knowledgeable friend, who I assumed would know, if the Model 4 could make use of the interrupts which can be generated by these features. I was told it couldn't. I think this person was assuming they were standard Mode 2 interrupts.

Of course, we all know the joke about the word "assume." Unfortunately, it applies to both of us in this case.

Recently, I've been working on project which requires extremely accurate timing at very short intervals. For this application, the "polling" method to determine the state of the TCR's (Timer Control

Register) TIF flag (which is set when the PRT has counted out), is simply not adequate. This "polling" method follows:

```
LD      C,TCR
LOOP   TSTIO    80H
;is TIF1 flag set? (use 40H for
TIF0)
JR      Z,LOOP
;poll again if not
IN0     A,TMDR1H ;else
reset TIF1 flag (the TSTIO
instruction
; counts as a read of TCR)
proceed...
```

The problem with this method is you can't achieve an exact interval between samples. The TSTIO instruction takes 12 T-states. The following JR instruction takes 8T to go to LOOP if the TIF flag is not set, or 6T to fall through to the IN0 instruction if the TIF flag is set.

On the surface, this looks fine:  $12T + 8T = 20T$ , and the PRT counts down at intervals of 20T. It looks like the loop should stay in sync with the PRT.

BUT, depending on the contents of MWI1



and MWIO in DCNTL, and REFW in RCR, this loop will actually take anywhere from a minimum average of 20.5T to a maximum average of 26T. At the "standard" setting of the Houdè XLR8er utilities (1,1,40 with REFW set), the loop will take 21.5T on average.

This is because Z180/HD64180 memory refresh occurs at intervals of either 10, 20, 40 or 80T, depending on the values of the MWI1 and MWI0 bits in DCNTL, and not at the start of every instruction like a normal Z80. The memory refresh can take either 2T or 3T, depending on the setting of the REFW bit in RCR.

Thus, at the standard Houdè settings, there will be a 3T refresh every 40T, which means 43T will actually elapse for every 40T worth of instructions, or an average of 21.5T for our 20T polling loop.

Regardless of the settings of MWI0, MWI1 and REFW, the polling method can (and periodically will) miss the setting of the interrupt flag by as much as 19T, and will seldom detect it immediately it is set. In other words, we will experience delays varying from 6T to 25T from the time the flag is set to the time we can react to it, with no way to know exactly how much time has passed.

This problem seemed insolvable as long as the Model 4 was in IM1 (interrupt mode 1). Because of the assumption that the Z180/HD64180 internal interrupts could only be used with mode 2 interrupts enabled, I then started to explore whether or not LS-DOS could be modified to permit mode 2 interrupts.

In the course of that research, while poring over the Hitachi HD64180 Hardware Manual to see the best way to use a PRT channel to create a simulated RTC interrupt, a sentence jumped off the page at me: "Internal interrupts...use the same vectored response mode as [IM2 interrupts]." Note it does not say they USE IM2 interrupts. It only says they are the SAME as IM2 interrupts!

It seemed too good to be true! If I was interpreting that correctly, it meant I could have my cake and eat it too — that I could use BOTH the native Model 4 mode 1

```

; TWOINTS/ASM 18-Dec-90
;
; By J.F.R. "Frank" Slinkman, 1511 Old Compton Road,
; Richmond, Va. 23233. (804) 741-0205. CIS ID# 72411,650.
;
; Routine to demonstrate use of Z180/HD64180 internal
; interrupts, and their simultaneous use with native Model 4
; mode 1 interrupts. Released to the public domain.
;
*GET EQUATES ;SVC & HD64180 defs & macros
DOSSEC EQU 2DH
OPREG$ EQU 78H ;storage for port 84H image
OPREG EQU 84H
CLKCTR EQU 712H ;counter for 30 times/sec interrupts
@@TIME EQU 078DH ;dos routine to display time
VIDRAM EQU 0F800H ;start of video ram
COL34 EQU VIDRAM+34
COL38 EQU VIDRAM+38
COL70 EQU VIDRAM+70
ROW11 EQU 11*80
ROW12 EQU 12*80
;
ORG 3000H
INTTABL
INT1VCT DW DOFKALL
INT2VCT DW DOFKALL
PRT0VCT DW DOFKALL
PRT1VCT DW DOFKALL
DMA0VCT DW DOFKALL
DMA1VCT DW DOFKALL
CSIOVCT DW DOFKALL
ASCOVCT DW DOFKALL
ASCLVCT DW DOFKALL
;
DOFKALL LD A,-1
OUT0 DCNTL,A ;set to 3 mem & 4 i/o wait states
OUT0 RCR,A ;set refresh interval to 10T
SVC @IPL ;reboot
;
TMR1DAT DW 2560 ;value for PRT1 data register
DW 5120-1 ;val for PRT1 reload register
TCRMASK DB 22H ;mask w/TIE1 and TDE1 bits
;
RTCMMSG DM 'RTCPRT'
TESTMSG DM 'syncing'
SYNMSG DM 'in sync'
;
OLDSEC DB 99
;
BEGIN LD A,86H
LD (OPREG$),A
OUT (OPREG),A ;bring up vidram
LD HL,VIDRAM
LD DE,VIDRAM+1
LD BC,1920-1
LD (HL),20H
LDIR ;clear it
LD HL,TESTMSG
LD DE,ROW11+COL38
LD C,7
LDIR ;write 'syncing' to screen
LD A,HIGH.INTTABL
LD I,A ;table address MSB
LD A,LOW.INTTABL
LD IL,A ;table address LSB to IL
OUT0 HL,PRTTASK ;table address LSB to IL
LD (PRT1VCT),HL ;PRT1 vector
;
BGN010 LD HL,OLDSEC ;-> stored last seconds value
BGN020 LD A,(DOSSEC) ;p/u dos seconds
CP (HL) ;== to last reading?
JR Z,BGN020 ;go if so

```



```

LD      (HL),A          ;if has changed, store value
LD      L,A
LD      H,0              ; and put seconds in HL
LD      DE,ROW11+COL34
LD      B,2              ;specify 2 decimal digits
SVC     @HEXD            ;display seconds on CRT
;
LD      A,(906H) ;p/u kbd row 7 data
CP      1                 ;is ENTER being pressed?
JR      Z,BGN050 ;go if so
CP      08H               ;is up-arrow being pressed?
JR      NZ,BGN040         ;go if not
LD      A,(CLKCTR)        ;here to slow system clock
INC     A                 ;delay sec change by 1/30th
BGN030 LD      (CLKCTR),A  ;modified value to CLKCTR
JR      BGN010            ;wait for further kbd input
;
BGN040 CP      10H         ;if down-arrow being pressed?
JR      NZ,BGN010         ;invalid or no input if not
LD      A,(CLKCTR)        ;here to speed up sys clock
OR      A
JR      Z,BGN010 ;can't if CLKCTR at bottom
DEC     A                 ;advance sec change by 1/30th
JR      BGN030
;
BGN050 LD      HL,SYNMSG   ;write 'in sync' to screen
LD      DE,ROW11+COL38
LD      C,7
LDIR
LD      DE,RTCTASK        ;start RTC task
LD      C,11
SVC     @ADTSK
;
BGN060 SVC     @CKBRKC     ;loop until BREAK key pressed
JR      Z,BGN060
;
XOR     A
OUT0    TCR,A             ;turn off PRT
LD      C,11
SVC     @RMTSK            ;turn off RTC task
LD      A,87H
LD      (OPREG$),A        ;bring up regular RAM
OUT     (OPREG),A
LD      HL,0              ;and exit to LS-DOS
SVC     @EXIT
;
RTCTASK DW     RTSK010
;
RTSK010 LD      A,86H
OUT     (OPREG),A         ;bring up vidram
LD      A,(DOSSEC)        ;p/u dos seconds
OR      A                 ;at even minute (ie, 0 secs)?
JR      Z,RTSK020         ;go if so
LD      L,A
LD      H,0               ;else put seconds in HL
LD      DE,ROW11+COL34
LD      B,2               ;specify 2 decimal digits
SVC     @HEXD            ;display seconds on CRT
RET     ;and exit
;
RTSK020 LD      HL,TMR1DAT ;-> data for PRT channel 1
LD      BC,4<8!TMDR1L    ;B=counter, C=int port number
OTIMR
LD      A,(HL)            ;p/u TCRMASK
OUT0    TCR,A             ;start PRT ch 1 w/interrupt
;
LD      HL,RTCMMSG
LD      DE,ROW11+COL34
LD      C,3
LDIR
LD      DE,ROW12+COL34
LD      C,3

```

interrupts AND the internal interrupts at the same time!

Not only that, they would be entirely independent of, and would not conflict with, each other. The two types could not interrupt each other because both manage the IEF1 flag, which enables and disables all maskable interrupts. If two interrupts occurred at the same time, the hardware interrupt would have priority, and the internal interrupt would be acknowledged when interrupts were re-enabled. Hardware interrupts could be disabled separately, by managing the bits in port x'e0'.

I set about to test this theory, and the program TWOINTS/ASM included with this article is that first test.

The results of the test? Yes, you CAN use both the Model 4's native IM1 interrupts and the Z180/HD64180 internal interrupts at the same time, and they do NOT conflict with each other.

To understand how use this information, we need to look at TWOINTS/ASM.

The function of the program is to use the Z180/HD64180's PRT Channel 1 to count out 60 times a second, with its interrupt enabled. The counter is started in such a way as its interrupts will be roughly 180 degrees out of phase with the native RTC interrupt. Both interrupts are used to calculate the time, and the two times (PRT-calculated and RTC-calculated) are displayed on the screen.

First are some EQUates, the purpose of most should be obvious. The only ones which require explanation are CLKCTR, which is a 30-step counter the system uses to determine whether or not it needs to add a second to the system clock, and @@TIME, which is the address of the LS-DOS time display routine. It performs the function of the @TIME SVC, moving the 8-byte (HH:MM:SS) ASCII system time string to the user buffer pointed to by HL.

Next, at INTTABL, we set up a table of vectors for use by the Z180/HD64180's internal interrupts. This table must be at the start of a 32-byte page, i.e., the lsb of its address must be 00H or a multiple of

20H. In actual practice, you may wish to put it at the top of lomem (12E0H-12FFH).

Initially, all vectors point to DOFKALL, a routine to put the Z180/HD64180 in its slowest operating mode (3,4,10) and then reboot via an @IPL SVC. This is done to reboot to LS-DOS off of disk instead of to Model 3 mode off of ROM, which is what happens if the processor attempts @IPL when it's running too fast.

The reboot is necessary because special code is needed in some cases to clear the flag which generated the interrupt. We can't simply jump to EI and RETI instructions, for example, because without code to reset the interrupt flag, the computer would constantly be in an interrupted state. Invoking @IPL simply automates the pressing of the reset button, which is what you'd have to do anyway when the computer locked up.

Next, at TMR1DAT, are five bytes of data for PRT channel 1. The first 16-bit word is for the PRT data register, and the second is for the reload register.

For those of you who are not familiar with how the PRT works, each channel has two 16-bit registers (each of which is loaded as though it was two 8-bit registers), one being the "data" register and one the "reload" register.

There is also an 8-bit Timer Control Register (TCR) which controls the operation, and shows the status, of both the channels. This is done with various bits. For each channel, there is a TDE (Timer Downcount Enable) bit, a TIE (Timer Interrupt Enable) bit, and a TIF (Timer Interrupt Flag) bit.

When a channel is started, it decrements the data register once every 20T. When it reaches zero, it sets the TIF flag bit for that channel. This is ANDed with the TIE flag; so if both are set, an interrupt is generated.

20T later, when it's time for the next decrement, instead of decrementing from 0 to -1, the PRT data register is loaded with the contents of the reload register. This is how the intervals between interrupts are determined.

```

        LDIR                                ;disp 'PRT' @ line 12, col 34
        LD      HL,ROW11+COL38
        SVC     @TIME                        ;disp time on line 11, col 38
        EX      DE,HL                        ;HL->TIME$ storage area
        LD      DE,PRTSEC
        LD      C,3
        LDIR                                ;copy dos time values to
                                           ; PRTTASK storage area
;
        SVC     @RPTSK                       ;now switch to task below
;
        LD      A,86H                        ;bring up vidram
        OUT     (OPREG),A
        LD      HL,ROW11+COL38
        SVC     @TIME                        ;disp time @ line 11, col 38
        LD      HL,ROW11+COL70               ;->1's digit of count display
;
; routine to display running decimal counter on CRT
; entry: HL -> 1's digit of running counter
INCNO LD      A,(HL)                        ;p/u value at char position
      OR      30H                          ;conv poss space to numeric
      INC     A                              ;bump it
      CP      '9'+1                         ;<= 9?
      JR      C,INCNO10                     ;OK if so
      LD      A,'0'                         ;else force to '0'
      INCNO10 LD      HL,A                  ;replace the character
      RET     C                             ;if was <= 9, finished
      DEC     HL                            ;else -> position to left
      JR      INCNO                         ; and perform carry
;
MAXMUMS DB      60,60,60,24                 ;# of ints/sec, sec/min,
                                           ; min/hr and hr/day
COUNTER DB      0                          ;count interrupts modulo 60
PRTSEC  DB      0                          ;store seconds
        DB      0                          ;store minutes
PRTHRS  DB      0                          ;store hours
;
PRTTASK EX     AF,AF'                      ;save AF reg
      INO      A,TCR                        ;reset TIF1, clear interrupt
      INO      A,TMDR1H
      EXX
      LD      A,86H                        ;save other regs
      OUT     (OPREG),A                    ;bring up vidram
      LD      HL,COUNTER                    ;-> PRT clock data regs
      LD      DE,MAXMUMS                    ;-> max values for the regs
      LD      B,4                          ;count # of data regs
      PRTSK010 INC     HL                    ;bump value
      LD      A,(DE)                        ;p/u max for this value
      SUB     HL                            ;
      JR      NZ,PRTSK020                  ;finished if value < its max
      LD      HL,A                          ; else reset value to 0
      INC     HL                            ; -> next value
      INC     DE                            ; -> next max
      DJNZ    PRTSK010                     ; and perform carry
                                           ; (falls thru if carry from
                                           ; hours, date not maintained)
      PRTSK020 LD      HL,ROW12+COL38        ;-> disp posn for PRT hours
      LD      DE,PRTHRS                     ;vals for dos @time routine
      CALL    @TIME+3 ;call dos @time routine
;
      LD      HL,ROW12+COL70                ;->1's digit of count display
      CALL    INCNO                         ;bump running counter
      EXX
      EX      AF,AF'                        ;restore regs
      EI
      RETI
;
      END      BEGIN

```

The values in TMR1DAT were calculated as follows: the XLR8er clock rate is 6.144 MHz. Since the PRT counts at intervals of 20T, its rate is 307.2 KHz. Since we want 60 interrupts per second, we must divide that by 60, to get 5120. Since the PRTs count through 0, we must change this to 5119. This is the value for the PRT reload register. We use about half that value to initiate the PRT data register, since the PRT will be started by an RTC interrupt, and we want the two interrupts to be 180 degrees out of phase.

The fifth byte is the value which must be written to the TCR to set the channel 1 downcount-enable (TDE1) and channel 1 interrupt-enable (TIE1) bits. (There's a joke or a pun in there somewhere — about "TIE1 on.")

At BEGIN, we bring up the video RAM and clear it (fill it with spaces), and put the message, "syncing," on the screen at row 11, column 38.

Next we must tell the Z180/HD64180 where our internal interrupt vector table is; so we load the msb of the address of our vector table into the I register and write the lsb of the address to internal register IL. Only the three most significant bits of IL matter because the lower 5 bits are supplied by the internal interrupt. (In this case, since the lsb is zero, the write to IL is unnecessary, as it was already set to zero the last RESET. It is done here only for demonstration purposes.)

Now we load the address of PRTTASK, our PRT channel 1 interrupt processing routine, into PRT1VCT, the vector table slot for PRT channel 1 interrupts.

The next section of code, from BGN010 to BGN050, makes it possible to synchronize the computer time with an external timing device. I used a digital wrist-watch.

Each time the system advances its seconds counter, this code displays the seconds on the screen and looks for certain input from the keyboard, by looking at the keyboard matrix starting at 900H. Specifically, it's looking at the part of the matrix which applies to the ENTER and up- and down-arrow keys.

If ENTER is being pressed (which means it must be HELD DOWN, as the scan is only being made once per second), the program exits this function via a JR to BGN050.

If ENTER is not pressed, then tests are made to see if either the up or down arrow keys are being pressed. If neither, then the code waits for another change in the seconds counter.

If you see the seconds display advance on the screen before it advances on your watch, hold down the up-arrow key until the two get into sync. This will occur because holding the key down will cause the system clock to run 1/30th slower, advancing the second counter once every 31/30ths of a second, by cancelling one CLKCTR decrement per second.

If you see the seconds display advance on your watch before it advances on the screen, hold down the down-arrow key. This causes the system clock to run 1/30th faster, advancing the second counter once every 29/30th of a second by adding one CLKCTR decrement per second.

Once the two are in sync, hold down the ENTER key until the word, "syncing" on the screen changes to "in sync." If you do not wish to sync the system clock to an external device, forget about the arrow keys and just hold down ENTER until "syncing" becomes "in sync."

Once ENTER is detected, the program goes to BGN050, which initiates the high-priority (60 times/second) task RTCTASK into task slot 11.

The rest of the code in the BEGIN section merely loops until the break key is pressed, since all the processing is interrupt driven. When BREAK is detected, it removes the task, stops the PRT, makes sure normal RAM configuration is restored, and returns to LS-DOS Ready.

The native RTC interrupt-handling routine, RTCTASK, has two stages. The first stage — the initiation stage — waits for the internal clock to cycle to and even minute (i.e., 0 seconds). While it's waiting, it puts the number of seconds on the screen so you'll have something to look at

for up to 59 seconds. Once it sees 0 seconds, it starts PRT channel 1, displays two messages and the current time from the system clock. Then it copies the system clock seconds, minutes and hours counters to an area used by the PRT interrupt processor, and uses the @RPTSK (replace task) SVC to tell the task processor to execute only the second stage on each subsequent RTC interrupt.

The second stage — which does the real "work" — brings up the video RAM, displays the system time, and then falls through to INCNO, which displays a running count of the number of RTC interrupts the task has received and processed.

The first thing the internal interrupt processor (PRTTASK) must do is clear the flag which generated the interrupt. To do this, we save the A register and read the TCR and one of the PRT channel 1 timer data registers. Then we save the other registers, and bring up the vidram.

(Note that the registers are saved via exchange instructions and not on the stack. This eliminates the need to check the value of the stack pointer. In real applications, the registers should be saved on the stack, or in RAM. If the code will be bringing up the video RAM, as TWOINTS does, it should not be enabled until the value of SP is verified to be below 0F400H. There is no way (other than the EX AF,AF instruction) to restore the AF register without using the stack, and you don't want to risk PUSHing things into the keyboard image or video RAM areas.)

Next, in the PTSK010 loop, we increment COUNTER and subtract it from its maximum, pointed to by DE. If 60 interrupts have not yet been received, we jump to PTSK020.

If this is the 60th interrupt, then we reset COUNTER to zero, and add one second to our PRT-maintained clock. This is done in the same fashion within the PTSK010 loop.

At PTSK020, we set up the HL and DE registers for the LS-DOS time display routine, and call it one instruction past its own LD DE,nn instruction (see *THE*



*SOURCE*, vol 1, p22). This displays the PRT-calculated time.

Then we point HL to the position of the ones digit of the running counter for PRT interrupts, and call INCNO to bump the decimal count before restoring the registers, re-enabling interrupts and returning.

On my Gate Array Model 4, I found that when the PRT reload register is loaded with 5119, the PRT time runs almost exactly 1/5120th slower than the RTC time. This difference, since it is virtually identical to the reload register count, led me to suspect at first that the PRT might be "skipping a beat" somewhere in its reload or interrupt-generating process.

When I used a reload value of 5118, the two times were in almost exact sync (1,614,380 RTC interrupts vs. 1,614,386 PRT interrupts, in one test). Also, using

5118, both times are in almost exact sync with my digital wristwatch (26,906 computer seconds vs. 26,905 wristwatch seconds, in the same test.)

So I modified the program to write 1280 to the PRT data register, 2559 to the reload register, run COUNTER modulo 120, and have the running PRT interrupt counter only count every other interrupt.

The result of that test confirmed that the PRT cycling is as described in the manual, and that the Z180 on my XLR8er board is not running at 6.144 MHz, but at approximately 6,143,124 Hz.

Timing oddities aside, the method used in TWOINTS/ASM can easily be adapted to make use of any of the internal interrupts generated by the Z180/HD64180's on-chip features, and increase their usefulness in certain applications.

The procedure is:

1. create a vector table at any address which is an even multiple of 20H (i.e., has its 5 least significant bits all 0's);
2. write the msb of the table address to the I register and the lsb to the IL internal register;
3. load the address of your interrupt processing routine into the proper slot in the vector table;
4. initiate the feature with the interrupt-enable flag bit in its control register set.

Here's hoping we'll be able to do the same thing with the Z280 on Anitek's proposed TRX-280 board!

# GIF4MOD4 AND HR2GIF

for the Model 4/4P/4D

They say a picture is worth 1,000 words. This picture was converted from GIF to TRS-80 format using GIF4MOD4. Until now, Model 4 users had no way to view GIF images or to send their own hi-res graphics creations to other types of computers. GIF4MOD4 will decode any GIF image up to 640 x 480 x 256 (VGA)



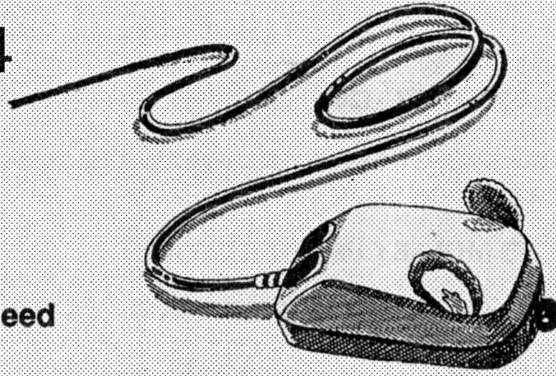
and put it on your hi-res screen. If you have no hi-res board, GIF4MOD4 puts it in an /HR disk file so you can dump it to your dot-matrix printer. HR2GIF converts /HR, /CHR and /BLK files to GIF format.

ORDER FORM -- Mail to  
J.F.R. Slinkman, 1511 Old Compton Rd., Richmond, VA 23233  
Please send me GIF4MOD4 & HR2GIF for my TRS-80 Model 4.  
I am enclosing \$37.95 + \$2.00 S&H (add \$2.00 outside N. America. Virginia residents please add 4-1/2% [\$1.71]).

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

# A Model 4 Mouse Driver

by Matthew Kent Reed



Matthew Kent Reed  
Box 368

West Lebanon, NH 03784-0368

The TRS-80 Model 4 is a fine computer, but because of its age, it is missing some features that have become common on more recent machines. Mice, for example, have become customary on Macintoshes and IBM compatibles, yet Model 4's have been unable to use them. But now, with my mouse driver and a Microsoft compatible serial mouse, the Model 4 can join the ranks of mouse-equipped computers.

I first became intrigued by the idea of using a mouse on a Model 4 after reading the August 1990 issue of *Computer News 80*. It included an article by David Goblen about Model 4 mouse use. According to the article, all you needed to do with a serial mouse was to plug it into the Model 4's RS-232C port. The mouse would then work with a mouse driver he had written and was selling through *Computer News 80*.

I was interested, so I ordered the mouse driver and started looking for a mouse to go with it. I decided to buy Mouse Systems' two-button Omnimouse II from Lyben Computer Systems. Its supplied connecting plug wouldn't fit in the Model 4, so I also needed to buy a DB9 female to DB25 male connector from Radio Shack. With the connector, the Omnimouse plugged into the Model 4 without difficulty. However, I had a problem: David

Goben's mouse driver didn't work with the Omnimouse. Since the Omnimouse is totally Microsoft compatible, I concluded that his driver was not written for Microsoft compatible mice.

Since I had a mouse, but no driver, I decided to write my own mouse driver for use with Microsoft compatible mice. The result is MMOUSE (see program listing 1). MMOUSE uses no stack space, so it should work in even the tightest systems. To install it, enter

MMOUSE

It will install itself in high memory and establish a new @MOUSE SVC. To remove MMOUSE and reclaim its memory space (if possible), enter

MMOUSE (REMOVE)

Keep in mind that the MMOUSE driver can't be SYSGENed. As an interesting note, the resident portion of Microsoft's latest MS-DOS mouse driver takes up 14K. My mouse driver occupies less than 400 bytes of high memory (admittedly with fewer features).

My driver adheres to the same software standard as David Goblen's driver, except for a few places where I thought it would be sensible to make some small changes. Any software written for one driver should work without modification with the other. Note that function 5 is supported only for compatibility with David Goblen's mouse driver; the Microsoft standard only supports a two-button mouse. Table I con-

tains the technical specifications for the new @MOUSE SVC:

## Programming for the Mouse

I wrote program listing 2, a simple screen editor, as a sample mouse program. After you run it, a mouse cursor will be displayed on the screen. Simply move the mouse and the cursor will follow. Click any button to move the text cursor to the location of the mouse cursor. Any text typed will be displayed at that location. Press <BREAK> at any time to exit. It is a simple program, but it illustrates some useful ideas.

If you are familiar with mouse programming on an IBM compatible, then writing Model 4 mouse programs shouldn't be too difficult. However, if you have no previous mouse programming experience, it might be helpful to know how a mouse works. A mouse doesn't keep track of absolute X and Y values. Instead, it provides measures of relative X and Y movement since the mouse last reported. The mouse driver converts these readings into the absolute X and Y coordinates of the mouse cursor. In this way, the mouse cursor moves in whichever direction the mouse moves, but it stays in a user specified range.

A program must first determine if the mouse driver has been installed. An example of how to do this is at the beginning of program listing 2.

If the mouse driver has been installed, the program must set the maximum X and Y values for the mouse cursor. This range will be different for different types of programs. For instance, in a text program, XSIZE should be set to 80 and YSIZE to 24. After this, the mouse cursor will be returned as X and Y values from 0 to 79 and 0 to 23, respectively. These X and Y values can be used as is, with no range checking or division. For low-resolution graphics programs, XSIZE and YSIZE should be set to 160 and 72. XSIZE and YSIZE should never be set higher than 640 and 240, which is the maximum resolution of the high resolution graphics screen.

After it sets XSIZE and YSIZE, the pro-

gram should set the mouse sensitivity. Possible sensitivity values range from zero to three, with three the most sensitive. I find that three is a good value for my 200 dots-per-inch (dpi) mouse, but if your mouse has a higher dpi you might want a lower value. It is a good idea to save the original mouse sensitivity upon entering a program so that it can be restored upon exit.

After it sets mouse sensitivity, the program should initialize the mouse cursor to some position on the screen (usually the center). Then the program can use the @MOUSE SVC whenever it needs mouse input. One important thing to remember is that, like keys on a keyboard, mouse buttons "bounce". If your program is waiting for a single button press, it might be a good idea for the program to wait until the button is released before acting upon it.

Many MS-DOS programs with mouse support use a free-floating mouse cursor. The cursor, which is a function of the MS-DOS mouse driver, moves on the screen as the mouse moves on the desk. I wrote MSKEY, a routine in program listing 2, to emulate this feature. It behaves similarly to the @KEY SVC. After setting XSIZE and YSIZE to 80 and 24, a program can call MSKEY whenever it needs keyboard input. MSKEY will wait for a key to be pressed, just like the @SVC, but MSKEY will display a mouse cursor while it waits. After a key or a mouse button is pressed, the routine will erase the mouse cursor and return to the calling program. If register A is non-zero, that key was pressed. If A contains zero, a mouse button was pressed. Use function 1 of the @MOUSE SVC to determine which one.

## Technical Details

If you don't care about the inner workings of the mouse driver, you can skip this section. You already have enough information to use or program for the mouse. However, if you like to know exactly how things work, this section might interest you.

The mouse driver, after installing itself in high memory, initializes the RS-232 port.

### Table I - Mouse SVC

#### @MOUSE .... SVC-120

This SVC performs various mouse functions depending on the function code passed in register B.

#### GET MOUSE CURSOR POSITION

Registers Affected: AF, BC, DE, HL

B => 1; Gets mouse cursor and button status  
 HL <= Contains mouse cursor X value (0 - XSIZE)  
 DE <= Contains mouse cursor Y value (0 - YSIZE)  
 A <= Contains button status  
 Bit 0: Reset if right button pressed  
 Bit 1: Reset if middle button pressed (both buttons)  
 Bit 2: Reset if left button pressed

#### SET MOUSE CURSOR POSITION

Registers Affected: AF, BC, DE, HL

B => 2; Sets mouse cursor position  
 HL => Contains X value (0 - XSIZE)  
 DE => Contains Y value (0 - YSIZE)  
 Z <= Set if operation was successful

#### GET SENSITIVITY AND MAXIMUM VALUES

Registers Affected: AF, BC, DE, HL

B => 3; Gets sensitivity and X and Y maximum values  
 HL <= Contains current X maximum (XSIZE)  
 DE <= Contains current Y maximum (YSIZE)  
 A <= Contains current sensitivity (0 - 3)

#### SET SENSITIVITY AND MAXIMUM VALUES

Registers Affected: AF, BC, DE, HL

B => 4; Sets sensitivity and X and Y maximum values  
 HL => Contains new X maximum  
 DE => Contains new Y maximum  
 C => Contains sensitivity (0 - 3); 3 is most sensitive  
 Z <= Set if operation is successful

#### GET MOUSE TYPE

Registers Affected: AF, BC

B => 5; Gets mouse type  
 A <= Contains mouse type; 0 if 2 button, 1 if 3 button



```

; MMOUSE - Model 4 mouse driver for a
; Microsoft compatible mouse, version 1.0
; copyright (c) 1990, by Matthew Reed
;
@DSPLY EQU 10
@LOGOT EQU 12
@PARAM EQU 17
@GTMOD EQU 83
@MUL16 EQU 91
@DIV16 EQU 94
@HIGH$ EQU 100
@FLAGS$ EQU 101
@CKBRKC EQU 106
@MOUSE EQU 120
STR EQU 20H
ABR EQU 10H
SVC MACRO #NUM
LD A, #NUM
RST 28H
ENDM

; Start of installation
;
BEGIN ORG 3000H
LD (STACK), SP ; Save stack
LD SP, BEGIN ; Make new stack
SVC @CKBRKC ; Abort if break pressed
JP NZ, ABORT
PUSH HL ; Save command line
LD HL, TITLE$ ; Display title
SVC @DSPLY
POP HL ; Restore command line

; Test for parameters
;
SVC @FLAGS$ ; Set IY
LD A, (IY+26) ; SVC table MSB
LD (SVCA1), A ; Stuff in program
LD (SVCA2), A
LD DE, PARM$ ; Parameter table
SVC @PARAM ; Parse parameters
JP NZ, PARME ; Abort if error
LD BC, 0 ; REMOVE = OFF
REMOVE EQU $-2
LD A, B ; Is it ON?
OR C
JP NZ, RMV ; REMOVE if ON

; Install $MOUSE
;
LD DE, MOUSE$ ; "$MOUSE"
SVC @GTMOD ; Is it in high memory?
JR Z, LINK ; Install if so

; Get old HIGH$
;
LD HL, 0 ; Get current HIGH$
LD B, L
SVC @HIGH$
JP NZ, NOMEM$ ; No high memory
LD (OLDHI), HL ; Put in module

; Relocate references
;
LD IX, RELTAB ; Relocation table
LD DE, MODEND ; Module end
OR A
SBC HL, DE ; Difference to add
LD B, H ; BC = difference
LD C, L
RLOOP LD L, (IX+0) ; Get address to change

```

First it writes to port 0E8H to reset the Model 4 UART. Then it configures the UART to 8 data bits, 1 stop bit, and no parity by a write to 0EAH. Finally, it sets the baud rate to 1200 baud through a write to port 0E9H.

The Model 4 supports eight different interrupts; one of them is undefined. The three interrupts that are currently supported by LS-DOS are the RS-232 receive, I/O bus, and real time clock interrupts. Every time one of these interrupts is generated, LS-DOS reads port 0E0H, the interrupt latch. This latch contains one bit for each possible interrupt. If a bit in the interrupt latch is set, then LS-DOS knows it should act upon the corresponding interrupt. It does this by calling the appropriate interrupt routine, whose address is stored in a table at INTVC\$, X'003E'.

The mouse driver hooks into the RS-232 receive interrupt. To do this, it sets bit 5 of port 0E0H and of WFLAG\$ in the FLAGS\$ table, which holds a copy of the port. It also puts its address in the INTVC\$ table. Now, every time the mouse sends a control byte (one in a series of three), the UART generates an RS-232 receive interrupt. Since the interrupt is enabled, LS-DOS calls the appropriate address in INTVC\$, that of the mouse driver. Because it takes a short while for the byte to become available after the interrupt, the mouse driver tests bit 7 of port 0EAH. If the bit is reset, the driver returns, but if it is set, the byte is ready. When the byte is available, the driver reads it from port 0EBH.

Microsoft compatible mice send their control information using a three byte data format:

```

Byte 1: U 1 L R Y7 Y6 X7 X6
Byte 2: U 0 X5 X4 X3 X2 X1 X0
Byte 3: U 0 Y5 Y4 Y3 Y2 Y1 Y0

```

The U stands for unused, and L and R stand for the left and right mouse buttons. If a button is pressed, then the appropriate bit is set. X and Y are signed bytes indicating how far the mouse has moved since information was last sent.

The mouse driver determines whether it has received the first control byte by test-

ing if bit 6 is set. If it is, the driver jumps to its first byte routine. If it isn't, the driver checks a counter to see whether byte 2 or 3 is next. It calls the appropriate routine on the basis of that counter. After each routine finishes updating information, it returns to the task processor, which returns to the program that is running.

Since the mouse driver is an interrupt, it runs in the background behind any program (provided the program doesn't disable interrupts). For that reason, the @MOUSE SVC never even needs to read the mouse. The information it uses is constantly updated by the interrupt portion of the driver.

## Conclusions

With my mouse driver and a Microsoft compatible mouse, you now have everything you need to begin writing mouse compatible software.

## References

Flynn, Mary Kathleen. "Mice and Trackballs: Choices for the New Generation of Applications." *PC Magazine*, August 1990, pp. 211-277.

Goben, David. "Using a Mouse on the Model 4 without an Interface (Part 1)." *Computer News 80*, August 1990, pp. 4-6

*Model 4/4P Technical Reference Manual*. Fort Worth: Tandy Corporation, 1985.

*Model III Technical Reference Manual*. Fort Worth: Tandy Corporation, 1981.

*Mouse Systems Users Manual*. Fremont, California: Mouse Systems Corporation, 1988.

Soltoff, Roy. *The Programmers Guide to LDOS/TRSDOS Version 6*. Sterling, Virginia: MISOSYS, 1984.

Stanton, Tom. "Tracking the Mouse's Progress." *PC Magazine*, February 14, 1989, pp. 249-269.

*The Source. Vols. I-III*. Milwaukee, Wisconsin: Logical Systems, 1984.

```

LD      H, (IX+1)
LD      A, H                ; Is it end?
OR      L
JR      Z, MOVH             ; If so, move to high
LD      E, (HL)             ; Get address
INC     HL
LD      D, (HL)
EX      DE, HL
ADD     HL, BC               ; Add difference
EX      DE, HL
LD      (HL), D             ; Put it back
DEC     HL
LD      (HL), E
INC     IX                  ; Advance to next entry
INC     IX
JR      RLOOP               ; Loop until finished

; -----
; Move into high memory
; -----
MOVH    LD      DE, (OLDHI)   ; Destination
LD      HL, MODEND          ; Last byte of module
LD      BC, LENGTH          ; Length of module
LDDR    LD      DE, HL       ; Move into high memory
EX      DE, HL              ; HL = HIGH$
SVC     @HIGH$              ; Set it
JP      NZ, NOMEM           ; Abort if error
INC     HL                  ; Ahead one

; -----
; Install SVC 120
; -----
LINK    LD      BC, SVC120-MOUSE ; Find SVC address
ADD     HL, BC
LD      (01F0H), HL         ; Put in SVC table
SVCA1   EQU     $-1

; -----
; Initialize RS-232 port
; -----
OUT     (0E8H), A           ; Reset UART
LD      A, 77H              ; 1200 baud
OUT     (0E9H), A           ; Set it
LD      A, 01101100B        ; UART settings
OUT     (0EAH), A           ; Set it

; -----
; Install RS-232 receive interrupt
; -----
DI
LD      BC, INTTASK-SVC120  ; Find interrupt address
ADD     HL, BC
LD      (0048H), HL         ; Put in interrupt table
SET     5, (IY+'W'-'A')     ; Enable receive interrupt
LD      A, (IY+'W'-'A')     ; Get byte
OUT     (0E0H), A           ; Out to port
EI
LD      HL, 80               ; Set maximum values
LD      DE, 24
LD      BC, 0403H           ; Sensitivity
SVC     @MOUSE

; -----
; Success, exit
; -----
LD      HL, SCC$             ; Installation success
DB      ODDH
RSCC    LD      HL, RSCC$    ; Remove success
SVC     @DSPLY
EXIT    LD      HL, 0        ; No error
DB      ODDH
ABORT   LD      HL, -1       ; Error
LD      SP, $-$             ; Restore stack
STACK   EQU     $-2
RET

```



## ; Remove mouse driver

```

RMV    LD      DE,MOUSE$      ; Module name
      SVC      @GTMOD         ; Look in high memory
      JP       NZ,NTINST      ; Abort if not found
      PUSH     HL              ; Save address
      LD       HL,1AF4H        ; SVC error
      LD       (01F0H),HL      ; Put in table
SVCA2  EQU     $-1
      DI
      LD       HL,1C48H        ; No interrupt
      LD       (0048H),HL      ; Put in table
      RES      5,(IY+'W'-'A')  ; Remove interrupt
      LD       A,(IY+'W'-'A')  ; Get byte
      OUT      (0E0H),A        ; Out to port
      EI
      LD       HL,0            ; Get HIGH$
      LD       B,H
      SVC      @HIGH$
      JP       NZ,NOMEM       ; Abort if error
      EX       DE,HL          ; DE = HIGH$
      POP      HL              ; HL = module address
      DEC      HL              ; Back one
      OR       A              ; Compare
      SBC      HL,DE
      JP       NZ,CNTRMV      ; Abort if error
      EX       DE,HL          ; Switch to HL
      INC      HL              ; Go to OLDHI
      INC      HL
      INC      HL
      LD       A,(HL)
      INC      HL
      LD       H,(HL)
      LD       L,A
      LD       B,0            ; HL = (OLDHI)
      SVC      @HIGH$         ; Set new HIGH$
      JR       Z,RSCC         ; Go if success

```

## ; Error routines

```

NOMEM  LD      HL,NOMEM$      ; No memory
      DB      ODDH
PARME  LD      HL,PARME$      ; Parameter error
      DB      ODDH
NTINST LD      HL,NTINST$     ; Driver not installed
      DB      ODDH
CNTRMV LD      HL,CNTRMV$     ; Can't remove
      SVC      @LOGOT
      JR       ABORT

```

## ; Messages

```

TITLE$ DB      'MMOUSE - Model 4 mouse driver for a'
      DB      'Microsoft compatible mouse,' ,0AH
      DB      ' version 1.0,'
      DB      ' copyright (c) 1990 by Matthew Reed',0AH
      DB      ' all rights reserved',0AH,0DH
SCC$   DB      'Mouse driver successfully installed',0DH
RSCC$  DB      'Mouse driver turned off and memory reclaimed',0DH
NOMEM$ DB      'High memory address can't be changed!',0DH
PARME$ DB      'Parameter error!',0DH
NTINST$DB      'Mouse driver is not installed!',0DH
CNTRMV$DB      'Mouse driver turned off but memory can't be
reclaimed',0DH
MOUSE$ DB      '$MOUSE',03H
;
; Parameter table
;
PARM$  DB      80H
      DB      STR.OR.ABR.OR.6
      DB      'REMOVE'

```

## WE ARE ON OUR OWN: TANDY DROPPED US!

**Ken Strickler**  
P.O. Box 773  
Stanwood, WA 99292-0773

Well, TANDY has finally dropped the MODEL IV from its catalog, and the only software in their NEW SOFTWARE catalog for the model III or model IV are conversion programs to convert data files or basic programs to the platforms of other computers! Does that mean WE ARE FINISHED - HARDLY!

While we have been dropped as a current production computer, we have been left with a rich legacy of both software and hardware, to say nothing of the number of people who have learned to program on the Model I/III/IV and are more than just content to stay with a computer system that has reached maturity. While we don't have the intense graphics that are required to play the games which seem to take up so much time on the modern day, high speed demons, the keyboard input that we generate hardly presses the computer to its maximum limits. I suppose that a case could be made for more speed if your application included highly intensive mathematical formulas requiring much longer times for resolution. It is interesting, however, that when I was discussing the use of a computer to solve complex formulas and a recursive type loop (repeated entry) with a friend of mine who has a degree in higher math, he pointed out that the computer doesn't really do a very good job as far as accuracy on those repetitive entry type of complex equations, due to the cumulative error of the varies entries. As an example, he wrote about a 5 line program included below:

```

10 DEFDBL A,B:FOR X=1 TO 100
20 A=1/X:A=SQR(A):B=1/A:B=B*B
30 PRINT X,A,B
40 NEXT X
50 END

```



This program demonstrates that even defining a number as double precision still leaves errors which can be additive! This even happens in the BIGGIES! So that means that computers are real good for simple math. I would be the first to admit that were it not for computers, some of the math that is currently done, would not be done at all! So I guess that means that we accept a little error for some kind of results, but the error still exists! Of course, doing the math by hand is inaccurate also, since many reverse operations won't work out to the exact correct answer. If you are using a program like ZBASIC from ZEDCOR, the length of the Double-Precision field can be defined as large as 54 positions, and while it increases the amount of time that the computer requires to resolve an equation, the accuracy is increased.

It is true that the early Model I/III/IV suffered from small storage devices, but that problem has been eliminated by the after market groups such as MISOSYS and AEROCOMP who provide storage capacities for the MEGA-APPETITES. Double sided 80 track floppy drives, in 5.25 or 3.5 inch format provide up to 720K of storage and HARD DISK storage is available up to 80 megabytes which should satisfy even the hungriest programmer! Just recently ANITEK has released MEMORY EXPANSIONS going BEYOND 8meg, if desired! That might be a little overkill in most cases, but if you have the need, they have the memory! For those interested in collecting data from the "outside world", ALPHA PRODUCTS provides interface modules for receiving data, and controlling devices.

As you can see, the platform (computer hardware) which the TRS80 is built on will easily fill the "HOME USERS" requirements. The only area that the TRS80 can be faulted for any deficiency would be in the graphic display area, as we remain in monochrome with a maximum resolution of 640 x 240 if the hi-res graphic card is installed. By not having that capability, the temptation to use the computer as a "GAME" playing device is diminished. That is not to say that there were not some excellent programs written for entertainment on the model I/III/IV computers! A program like Lair of the Dragon

```

RRSP    DB      0
        DW      REMOVE
        DB      0
;
; Resident portion of mouse driver
;
MOUSE   JR      SVC120
OLDHI   DW      $-0 ; Old HIGH$
        DB      MODDCB-MOUSE-5
        DB      '$MOUSE'
MODDCB  DW      $-0
        DW      0
;
; SVC 120 handler
;
SVC120  DEC     B ; Go if function 1
        JR      Z,FUNCT1
        DEC     B ; Go if function 2
        JR      Z,FUNCT2
        DEC     B ; Go if function 3
        JR      Z,FUNCT3
        DEC     B ; Go if function 4
        JR      Z,FUNCT4
        DEC     B ; Return if not function 5
        LD      A,43 ; SVC error
        RET     NZ
FUNCT5  XOR     A ; 2 button mouse
        RET
;
; Return X, Y, and buttons
; A = buttons, HL = X, DE = Y
;
FUNCT1  LD      HL,$-0 ; Y value
YVAL    EQU     $-2
        LD      C,21 ; Amount to divide
YDIV    EQU     $-1
        SVC     @DIV16 ; Divide
        EX      DE,HL ; Put Y in DE
        LD      HL,$-0 ; X value
XVAL    EQU     $-2
        LD      C,9 ; Amount to divide
XDIV    EQU     $-1
        SVC     @DIV16 ; Divide
        LD      A,07H ; Get button value
BUTTONS EQU     $-1
        LD      B,00000111B ; No buttons pressed
        AND     00110000B ; Mask out rest
        JR      Z,F3 ; If no buttons, ahead
        CP      00110000B ; If both buttons,
        JR      NZ,F1
        LD      B,00000101B ; signal middle
F1       CP      00100000B ; If left button,
        JR      NZ,F2
        LD      B,00000011B ; signal left
F2       CP      00010000B ; If right button,
        JR      NZ,F3
        LD      B,00000110B ; signal right
F3       LD      A,B ; Put in A
        RET
;
; Set X, Y value
; Entry: HL = X, DE = Y
; Success, Z flag set
;
FUNCT2  PUSH    DE ; Save Y
        PUSH    HL ; Save X
        LD      BC,80 ; X maximum
XMAX    EQU     $-2
        OR      A ; Reset carry
        SBC     HL,BC ; Is it over maximum?
        JR      NC,FERR ; Go if error
        POP     HL ; Restore X

```

```

RX01  LD      A, (XDIV)      ; Get X divisor
      EQU     $-2
      CALL    MULT          ; Multiply
RX02  EQU     $-2
      LD      (XVAL), HL    ; Put in XVAL
RX03  EQU     $-2
      POP     HL            ; Restore Y
      PUSH    HL            ; Save Y
      LD      BC, 24        ; Y maximum
YMAX  EQU     $-2
      OR      A             ; Reset carry
      SBC     HL, BC        ; Is it over maximum?
      JR      NC, FERR1     ; Go if error
      POP     HL            ; Restore Y
      LD      A, (YDIV)     ; Get Y divisor
RX04  EQU     $-2
      CALL    MULT          ; Multiply
RX05  EQU     $-2
      LD      (YVAL), HL    ; Put in YVAL
RX06  EQU     $-2
      XOR     A             ; Zero for no error
      RET
MULT  LD      C, A          ; Put in C
      SVC     @MULT16      ; Multiply
      LD      H, L          ; Correct byte
      LD      L, A
      RET
FERR  POP     DE            ; Clear stack
FERR1 POP     DE
      INC     A             ; Signal error
      RET

;-----
; Get sensitivity, X and Y maximums
; Exit: A = sensitivity, HL = X maximum
;       DE = Y maximum
;-----
FUNCT3 LD      A, 3         ; Sensitivity
SENS  EQU     $-1
      LD      HL, (YMAX)    ; Y maximum
RX07  EQU     $-2
      EX      DE, HL
      LD      HL, (XMAX)    ; X maximum
RX08  EQU     $-2
      RET

;-----
; Set sensitivity, X and Y maximums
; Entry: A = sensitivity, HL = X maximum
;       DE = Y maximum
; Success, Z flag set
;-----
FUNCT4 CP      4           ; If too high, return
      RET      NC
      LD      (SENS), A     ; Store sensitivity
RX09  EQU     $-2
      LD      B, C          ; Put in counter
      INC     B             ; Make non-zero
      LD      A, 32        ; X tick maximum
      LD      C, 16        ; Y tick maximum
FNCL  SRL      A           ; Divide by 2
      SRL     C            ; Divide by 2
      DJNZ    FNCL         ; Loop until done
      LD      (TXMAX+1), A  ; Store X tick maximum
RX10  EQU     $-2
      LD      A, C
      LD      (TYMAX+1), A  ; Store Y tick maximum
RX11  EQU     $-2
      PUSH    HL           ; Save X maximum
      PUSH    DE           ; Save Y maximum
      CALL    FDIV         ; Divide Y maximum by A
RX12  EQU     $-2
      JR      Z, FERR      ; If zero, division error
      LD      (YDIV), A    ; Store divisor

```

from MISOSYS or ZORK on the MODEL III uses English words and phrases to move through an adventure. A program like that helps to learn the keyboard layout, but it is sure helpful to have taken typing to get some idea where the keys are!

If there is a member, of your family or someone that you know who would like to try out a computer, but doesn't think that they need one of the current offerings, like a cohort of mine at work who bought a couple of Model III computers from a school district at an auction for \$11.00 each! Of course there was no software, but a \$49.95 + shipping LDOS 5.3 from MISOSYS quickly solved that problem. We are now trying to find a source for the BASIC MANUAL, but until then, LDOS comes with ON-LINE HELP for BASIC, which helps at least for syntax. He has a MAC that has basic, but the syntax is sometimes a little different! Now his granddaughter (age 3) now has a computer to PLINK on. He has written a couple of little programs, simple math and such, which entertain and educate at the same time!

With so many school districts upgrading to other systems, our computer type may even be easier to get. PACIFIC COMPUTER EXCHANGE always has ads for MODEL's III and IV for sale. In addition to the hardware, they have used software also. We can't forget the HUGE PUBLIC DOMAIN software libraries that are available both through VIC McCLUNG listed in MISOSYS, and CN80 in Casper, WY.

## LEARNING TO EVALUATE SYSTEM REQUIREMENTS

Becoming your own SYSTEM ANALYST sounds like it might be a difficult task, but since we are out here on our own, NOW IS THE TIME! Everyone who owns their own computer is already performing that task, by evaluating the software or hardware that they need to do the task at hand! (You never thought of that like that, did you!) You have already probably selected a Word-Processor (most popular application) and maybe a Spread Sheet, Data Base or Home Budget program. For those who "HAVE TO KNOW", the authors of



TRS80 software have provided us with the "TOOLS" for ROLLING OUR OWN, in "C", PASCAL, FORTRAN, BASIC (either compiled or interpreted) and ASSEMBLER, just to name a few. MISOSYS provided the SOURCE CODE listing for the operating system to allow those new to computers an insight as to how tasks are performed, and where software routines are located, just in case you THINK YOU CAN DO IT BETTER! With the new and vastly larger systems, not only the software source code is not available (maybe Donald Trump could BUY IT), but the complexity of some of the schemes that have been devised to increase speed and "THROUGHPUT" (the total amount of work that a computer can do) boggle the minds of most, except the programmer who wrote it, or those inhabitants of PLUTO! Mere mortals for the most part are left with applying programs written by others to problems we are trying to solve. (Sounds like ANALYZING a SYSTEM application to me!)

## PLACING THE "SYSTEM" IN ITS ENVIRONMENT

Now that the task to be performed has been identified, where this task is to be performed must be evaluated. You know computers won't work UNDER WATER or ON THE SURFACE OF THE SUN, or WITHOUT POWER! If the intended information is to be gathered from a "HOSTILE" environment (defined as one that the computer can't operate in), provisions to protect the computer must be made! Think of a computer as sort of a "SEMI-HUMAN" device, which enjoys the climate that we do MOST OF THE TIME, tending toward 68-70 degree, 50% humidity, comfortable, short shirt sleeve weather. Bright sun gives me a "SUN BURN", and really HEATS me up, and it does it to the computer too! To keep the computer at the right temperature, and operating at its best, nothing finest "FOOD" should be used. While coffee, and coke are OK for us, the computer will PUKE on that. The computer likes a steady diet of 120vac, 60Hz electricity! If its diet is suddenly interrupted during operation, BAD THINGS happen! His BIG EYE (CRT) goes off, he loses his mind, and with that goes your valued current infor-

```

RX13 EQU $-2
LD A, OFFH ; Get remainder
SUB L
LD (TYMAX), A ; Store it
RX14 EQU $-2
POP HL ; Restore Y maximum
LD (YMAX), HL ; Store
RX15 EQU $-2
POP DE ; Restore X maximum
PUSH DE ; Save X maximum
LD A, (TXMAX+1) ; Tick maximum
RX16 EQU $-2
CALL FDIV ; Divide the two
RX17 EQU $-2
JR Z, FERR1 ; If zero, division error
LD (XDIV), A ; Store divisor
RX18 EQU $-2
LD A, OFFH ; Get remainder
SUB L
LD (TXMAX), A ; Store it
RX19 EQU $-2
POP HL ; Restore X maximum
LD (XMAX), HL ; Store
RX20 EQU $-2
LD HL, 0 ; Zero X and Y
PUSH HL
POP DE
JP FUNCT2 ; Exit through function 2
RX21 EQU $-2
LD H, A ; MSB
LD A, D ; Is DE zero?
OR E
RET Z ; Return if error
XOR A ; Zero A
LD L, A
DEC L ; Back one
FDIV1 SBC HL, DE ; Subtract
INC A ; Add one
JR NC, FDIV1 ; Loop if more
ADD HL, DE ; Find remainder
DEC A ; -1
RET

; -----
; Interrupt task to read mouse
; -----
INTTASK IN A, (0EAH) ; Is byte ready?
AND 80H
RET Z ; Return if not
LD HL, INTC ; Interrupt counter
RX22 EQU $-2
IN A, (0EBH) ; Read byte
BIT 6, A ; Go if first byte
JR NZ, FBYTE
AND 00111111B ; Mask out 7 and 6
RLC (HL) ; See which routine
JR C, SBYTE ; Go if second
TBYTE OR 00H ; Y increment
YINC EQU $-1
LD HL, YVAL ; Y value
RX23 EQU $-2
LD BC, 01F8H ; Maximum count
TYMAX EQU $-2
JR TBYTE1 ; Jump past
SBYTE OR 00H ; X increment
XINC EQU $-1
LD HL, XVAL ; X value
RX24 EQU $-2
LD BC, 02D0H ; Maximum count
TXMAX EQU $-2
TBYTE1 LD E, (HL) ; Put (HL) in DE
INC HL
LD D, (HL)

```



```

RX25  JP      M,ADD2      ; If minus, subtract
      EQU     $-2
      ADD     A,E         ; Add LSB
      LD      E,A         ; Put in E
      JR      NC,ADD1      ; If no overflow, ahead
      INC     D           ; Increment MSB if so
ADD1   EX     DE,HL
      LD      (ADHL),HL   ; Save HL
RX26   EQU     $-2
      OR      A
      SBC     HL,BC       ; Is it too high?
      LD      HL,$-$      ; Restore HL
ADHL   EQU     $-2
      EX     DE,HL
      JR      C,ADD4      ; If not, ahead
ADDERR LD      D,B        ; Transfer BC to DE
      LD      E,C
      DEC     DE          ; Back one
      JR      ADD4        ; Skip subtract
ADD2   ADD     A,E         ; Add LSB
      LD      E,A         ; Put in E
      JR      C,ADD4      ; If no underflow, ahead
      XOR     A           ; Zero A
      OR      D           ; Is D zero?
      JR      NZ,ADD3     ; If not, skip
      LD      DE,0100H    ; Allow for subtract
ADD3   DEC     D          ; Decrement one
ADD4   LD      (HL),D      ; Put DE at (HL)
      DEC     HL
      LD      (HL),E
      RET
FBYTE  LD      (HL),10101010B ; Reprime counter
      LD      C,A         ; Store A
      LD      (BUTTONS),A ; Store buttons
RX27   EQU     $-2
      RLCA              ; Rotate Y7 and Y6
      RLCA              ; into position
      RLCA
      RLCA
      AND     11000000B    ; Mask out rest
      LD      (YINC),A ; Put in memory
RX28   EQU     $-2
      LD      A,C         ; Restore first byte
      RRCA              ; Rotate X7 and X6
      RRCA              ; into position
      AND     11000000B    ; Mask out rest
      LD      (XINC),A    ; Put in memory
RX29   EQU     $-2
      RET
INTC   DB      10101010B
MODEND EQU     $-1
LENGTH EQU     $-MOUSE
RELTAB DW      RX01,RX02,RX03,RX04,RX05,RX06,RX07,RX08
      DW      RX09,RX10,RX11,RX12,RX13,RX14,RX15,RX16
      DW      RX17,RX18,RX19,RX20,RX21,RX22,RX23,RX24
      DW      RX25,RX26,RX27,RX28,RX29
      DW      0
      END      BEGIN

```

mation! Of course, for a price, a power "filtering/replacement" source can be provided. (Read that UPS or SBS) This little protection will cost as much as a harddisk, but if you live in, or operate in an area where the power isn't too steady or reliable, it can sure save you a bunch of time. (Ya'll did buy an SBS from Roy, now didn't you!) If the application is a critical one, is a "BACKUP" computer

required in the event of the inevitable "failure"? (Who can afford "extra" computers? If you aren't running on a \$10,000 system, and have to have to "latest, greatest, and fastest", used computers for backups are available. I have 7 model 4P's in various configurations which are used in various locations!)

Now that we have decided what and where

we are going to do "OUR THING", looking at the OUTPUT and where it will be used should be done. If the information is for our own use only, it doesn't matter if we use some obscure disk (say 2.8") and format (say 61.5 tracks, with 21.37 sectors each) since only this particular system will be using the programs and information. But should you happen to scribe a particularly ingenious piece of SOFTWARE and wish to share it with the world, somewhere in your system there should be an "OUTSIDE WORLD" port, probably a 5.25 inch FLOPPY (in our case) disk drive which can record the software code, to be sent to the "waiting world". (In my case, while adding the MISOSYS 40 megger, XLR8er with 256K, and a 720K 5.25", DS disk drive in one of the 2 Half-Height (HH) drive slots, the old 40 track SS drive remains as drive 0. A possible upgrade to a double sided drive, but remaining 5.25" so that disks written, "OTHERS CAN READ". ( 40 track disks can be read on an 80 track drive thanks to the RD40 utility from MISOSYS, part of the old MARK IV collection.) Since the programs "purchased" are provided on 40 track single sided disks, and since magnetic media loses its level of magnetism, occasionally it is necessary to "rejuvenate" the disk by backing it up to a new disk, and then re-backing it up to itself, or use a program called PREFORM6 (available through MISOSYS from POWERSOFT) to re-magnetize the media. Also, with drive 0 as a 40 track drive, the system harddisk can be shut down, and just run from drive 0, a new operating system, or other "UNKNOWN" (read that untested) disk is tried. As a system grows in complexity, the ability to maintain and protect valued data is important. (Feel SORRY for all of the IBM platform computer users who inadvertently caught a "VIRUS" and lost, or are still losing valuable data and time! We are most fortunate as a small group of computer users not to have been so infected, Hopefully those remaining with this computer system are interested in ways to increase its usefulness, not destroy it!)

## ARE WE REALLY ALONE ?

The answer, TODAY, IN A WORD, "NO"! Those subscribers who read and support the various magazines dedicated to the

MODEL I/III/IV market still have superb hardware and software help from some of the original developers and recent additions. Roy Soltoff (MISOSYS, HARDWARE-SOFTWARE), Peter Ray (ANITEK, HARDWARE-SOFTWARE), Chris Fara (MICRODEX, HARDWARE-SOFTWARE), David Goben (SOFTWARE), David Miller (SOFTWARE), just to name a few, are providing high quality hardware and software, tutoring and "Hand-Holding" for the TRS80 computers. It is through their "Dedication" alone that we can continue to survive and thrive with our discontinued computers. Most of the early GURUS for the TRS80 market (Kim Watt, Hardin Brothers, a couple of names that come to mind) have exited, either for the MS-DOS market, or "PRIVATE ENTERPRISE". MISOSYS still sells the Super Utility by Kim Watt and Powersoft, a fine collection of what I call "DYNAMITE" which is occasionally used to recover a disk error on a disk which hadn't backed up often enough! Interestingly you can not only "REVIVE" a disk, but "TOTAL DESTRUCTION" is also possible! As for Hardin Brothers, his excellent articles live in the dusty old pages of 80 MICRO (remember that mag?).

Even our OPERATING SYSTEM (if you are using LS-DOS) is being upgraded as new and better ideas come along. Currently the latest version is LS-DOS 6.3.1-Level H, which we have all upgraded to, RIGHT! If that isn't enough, David Goben always has a couple of new patches, to make it do something else, published in CN80. Roy has been applying the "midnight oil" to make the system run better and better!

In addition to *TMQ*, *CN80* and *TRSTIMES* are "GETTING THE WORD" out to the subscribers! What we need are more subscribers! It isn't enough to subscribe, and let your "FRIENDS" read the magazine, a couple of times may be OK, but nothing replaces the "GREEN" as an expression of our gratitude!

```

;-----
; MPNT - a mouse demonstration program
; copyright (c) 1990 by Matthew Reed
;-----
@DSP EQU 2
@KBD EQU 8
@DSPLY EQU 10
@VDCTL EQU 15
@GTMOD EQU 83
@FLAGS$ EQU 101
@MOUSE EQU 120
SVC MACRO #NUM
    LD A, #NUM
    RST 28H
ENDM
ORG 3000H
START LD DE, MOUSE$ ; Is driver in memory?
      SVC @GTMOD
      JR NZ, NO ; If not, not installed
      SVC @FLAGS$ ; Find SVC table entry 120
      LD H, (IY+26)
      LD L, 0F0H
      LD A, (HL) ; Get address pointed to
      INC L ; at table
      LD H, (HL)
      LD L, A
      LD A, (HL) ; Is first byte "LD A,n",
      CP 3EH ; start of SVC error?
      JR NZ, YES ; If not, installed
      LD HL, NINST$ ; Not installed
      SVC @DSPLY ; Display
      RET
YES LD C, 10H ; Enable reverse video
    SVC @DSP
    LD C, 11H ; but turn it off
    SVC @DSP
    LD B, 3 ; Get XSIZE, YSIZE,
    SVC @MOUSE ; sensitivity
    LD (FXSIZE), HL ; Save values
    LD (FYSIZE), DE
    LD (FSENS), A
    LD HL, 80 ; Set driver to 80
    LD DE, 24 ; by 24 with
    LD BC, 0403H ; sensitivity of 3
    SVC @MOUSE
LOOP CALL MSKEY ; Wait for key
      OR A ; Ahead if button pressed
      JR Z, MOUSE
      CP 80H ; Return if BREAK
      JR NZ, L1
      LD HL, 0 ; Old values
FXSIZE EQU $-2
      LD DE, 0
FYSIZE EQU $-2
      LD BC, 0400H
FSENS EQU $-2
      SVC @MOUSE ; Set them
      RET ; Return
L1 LD C, A
    SVC @DSP ; Display byte
    JR LOOP ; Loop
MOUSE LD B, 1 ; Get mouse position
      SVC @MOUSE
      LD H, E ; Set text cursor to
      LD B, 3 ; mouse position
      SVC @VDCTL
      JR LOOP ; Loop
;-----
; Recognize keyboard and mouse input
;-----
MSKEY LD B, 1 ; Get mouse X, Y
      SVC @MOUSE

```



```

LD      H,E          ; Convert to 8 bit X, Y
LD      (PXY),HL ; Stuff in routine
LD      B,1          ; Get byte at X, Y
SVC     @VDCTL
LD      (PCHAR),A    ; Stuff in routine
XOR     128          ; Make reverse video
LD      C,A
LD      B,2          ; Put cursor at X, Y
SVC     @VDCTL
KM1     LD      B,1    ; Get mouse X, Y, buttons
SVC     @MOUSE
AND     00000111B    ; Are buttons pressed?
CP      00000111B
JR      Z,KM2        ; Ahead if not
XOR     A            ; Make A zero and return
JR      PTROFF
KM2     LD      D,E    ; Convert to 8 bit X, Y
LD      E,L          ; in DE
LD      HL,(PXY) ; Old pointer coordinates
OR      A
SBC     HL,DE        ; Are they different?
JR      Z,KBD        ; If no change, keyboard
CALL    PTROFF       ; Remove pointer
JR      MSKEY        ; Loop
KBD     SVC     @KBD   ; Is there key?
JR      NZ,KM1       ; If not, loop
PTROFF  PUSH    AF    ; Get byte
LD      HL,$-$       ; Screen address
PXY     EQU     $-2
LD      C,0          ; Character at pointer
PCHAR   EQU     $-1
LD      B,2
SVC     @VDCTL       ; Put on screen
POP     AF           ; Restore byte
OR      A            ; Set to NZ
RET
MOUSE$  DB      '$MOUSE',03H
NINST$  DB      'Mouse driver not installed!',0DH
END      START

```

## ARE WE GOING TO BE ALONE ?

Well, in the future, near or far, the answer is probably "YES". The MS-DOS world is just too big to be ignored! If you had a choice to write programs for say a 10,000 user group, of which maybe 50% needed your program, or a user group of say 50,000,000, of which 1% needed your program, at \$10 a pop profit you are going for \$50,000 or \$5,000,000! It doesn't take a "ROCKET SCIENTIST" to see where the "MONEY" is! The TRS80 has plenty of power to be used as a "TOOL", and since most of my work is not graphic intensive, I'll stay with the Model 4. Sometimes, the program doesn't lend itself to the Model 4, and then it is totally written for MS-DOS.

With that said, ALL READERS ARE HEREBY ELEVATED TO THE TITLE OF "SYSTEM ANALYST, PROTEMP" (at least!), and Until Next Time———

## TRSTimes magazine

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

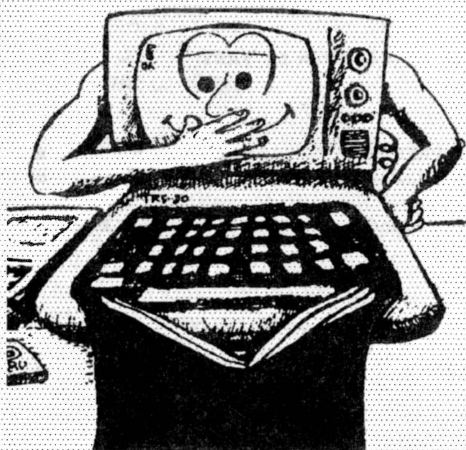
We are in our fourth year of publication and each issue typically features: 'Type-in' programs in Basic and Assembly language, Hands-on tutorials, Hints & Tips, Reviews, Questions & Answers, Letters, Nationwide ads, Humor and more.

1991 calendar year subscription rates (6 issues):

U.S & Canada: \$18.00

Europe & South America: \$23.00 surface or \$29.00 air mail

Asia, Australia & New Zealand: \$25.00 surface or \$32.00 air mail  
(all payments in U.S currency, please)



TRSTimes magazine

5721 Topanga Canyon Blvd. #4  
Woodland Hills, CA. 91367



## Profile 4 Plus to FilePro 16/dBASE III Plus

David P. Krebs  
124 Woodhill Drive  
Amherst, OH 44001-1614  
CIS: 73125,105

Profile 4 Plus is not a BAD database for the model 4. If it is combined with user menus and TYPEIN/CMD (from MISOSYS), can be very pleasant to work. Printing can be enhanced with the use of PROSE, a shareware enhanced printing module. It is available in the File Cabinet (CN80). Profile surely has the capacity, given the disk space to handle VERY large files. I have one file with over 1000 records and another with 650 and they run very well. Both of my DB's are fully automated with user menus, indexes built with the help of TYPEIN and enhanced printing from PROSE. All this is a key-stroke away with my multiple user menus.

There comes a time when we all probably will have to say goodbye to our Model 4 and start talking MS-DOS. When I first did this at work, I wanted a data base that I wouldn't have to learn, so I ordered filePro 16 from the Small Computer Company. It's the same program (Profile 16) that ran on the Tandy 6000 under XENIX.

Some time ago, I decided to try to convert one of my Model 4 DB's to filePro 16. I called Small Computer because they sold a program called TRANSFER. When I found out that TRANSFER was going to cost \$150 (1984 price), I asked, "isn't there another way?" The answer was, "yes, I'll send you a data sheet detailing the KEY record header and all you have to do is write a BASIC program, read the TRSDOS file and write an MS-DOS file with the header in front of each record". Getting the file(s) to my MS-DOS machine was up to me.

They sent the specs, I wrote the program, and it worked fine. Not too long ago, Luis M. Garcia-Barrio wrote a generic program that did basically the same as mine

### Table II

#### Deleted Record ...

Offset	Length	Description
00	1	x'00' for a deleted record
01	1	x'00' Reserved
02	4	Forward freechain pointer
06	4	Backward freechain pointer
10	10	x'00' Reserved

#### Used Record ...

00	1	x'01' Marks as a used record
01	1	x'00' Reserved
02	2	Creation Date - see below
04	2	Created by - see below
06	2	Update Date
08	2	Updated by
10	2	Batch update date
12	8	Reserved

```

1 ' FP16.BAS
2 ' Converts Profile 4+ DB to filePRO 16
3 ' Oct 15, 1990 (Days = 2844)
10 CLEAR
15 CLS
20 DEFINT A-Z
60 DT$ = MKI$(2844) ' Days from 1-1-83 to 10-15-90
70 ONE$=CHR$(&H01)
80 ZERO$=CHR$(&H00)
85 SPAC$=" "
90 HEAD$=ONE$+ZERO$+DT$+ZERO$+ZERO$+DT$+ZERO$+ZERO$+DT$+STRING$(8,
SPAC$)
100 OPEN "R",1,"LW000000.KEY",64
110 OPEN "R",2,"LW000000.DAT",100
120 OPEN "R",3,"KEY",80
125 OPEN "R",4,"DATA",96
130 FIELD 1, 60 AS KEYIN$ ' I had some extra in the /KEY
140 FIELD 2, 96 AS DATIN$ ' I don't want all the /DAT segment
150 FIELD 3, 80 AS KEYOUT$ ' Don't forget 20 bytes for the header!
155 FIELD 4, 96 AS DATOUT$ ' This one goes as-is
156 '
157 ' write the first record for filePro housekeeping
158 '
160 LSET KEYOUT$ = STRING$(20,ZERO$)+STRING$(60,SPAC$)
164 LSET DATOUT$ = STRING$(96,SPAC$)
166 PUT 3, 1: PUT 4, 1
168 '
170 LAST = LOF(1)/64 ' Calculate # of records in /KEY
175 FOR I = 1 TO LAST
180 GET 1, I
190 GET 2, I
195 J = J + 1: LOCATE 10, 30, 0: PRINT J; ' Count records read
200 IF LEFT$(KEYIN$,1) = ZERO$ THEN 250 ' Skip Deleted records
202 '
203 ' Lets change some of the data around ... move a sorting key to
204 ' the front of KEY
205 '
210 ID$ = MID$(KEYIN$,41,8)
212 KEY1$ = LEFT$(KEYIN$,40)
214 KEY2$ = MID$(KEYIN$,49,12)
210 LSET KEYOUT$ = HEAD$ + ID$ + KEY1$ + KEY2$ 'Add header to old /
KEY
220 LSET DATOUT$ = DATIN$ ' Transfer revised /DAT
230 PUT 3, I + 1: PUT 4, I + 1 ' Write both records
240 K=K+1:LOCATE 10,30,0:PRINT K; ' And count 'em
250 NEXT I
300 CLOSE
350 LOCATE 12, 30, 0: PRINT "Conversion Complete"

```

```

1 ' BD3.BAS
2 ' Converts Profile 4+ DB to dBASE 3+ DB
3 '
10 CLEAR
15 CLS
20 DEFINT A-Z
30 CR$ = CHR$(13)
40 LF$ = CHR$(10)
50 NULL$ = CHR$(0)
100 OPEN "R",1,"LW000000.KEY",64
110 OPEN "R",2,"LW000000.DAT",100
120 OPEN "R",3,"BIKE.DBF",158
130 FIELD 1, 60 AS KEYIN$
140 FIELD 2, 96 AS DATIN$
150 FIELD 3, 158 AS DB3OUT$
160 LAST = LOF(1)/64
170 FOR I = 1 TO LAST
180   GET 1,I
190   GET 2,I
195   J = J + 1: LOCATE 10,30,0: PRINT J; ' Count the records read
200   IF LEFT$(KEYIN$,1) = NULL$ THEN 250 ' Don't write the deleted
   records
201 ' Make some changes to the DB structure here
202   ID$=MID$(KEYIN$,41,8) ' Moves a sort key to the front of the
   file
204   KEY1$=LEFT$(KEYIN$,40) ' fix the rest of the KEY file
206   KEY2$=MID$(KEYIN$,49,12)
210   LSET DB3OUT$ = ID$ + KEY1$ + KEY2$ + DATIN$ + CR$ + LF$
220   PUT 3,I ' Write the new DB3 record
230   K=K+1: LOCATE 10,50,0: PRINT K; ' and count 'em
250 NEXT I
300 CLOSE
310 END

```

did for ANY PROFILE 4 Plus data base file. It's in his TRS-Link #27. We each did the same job with a little different twist. He elected to convert all records, /KEY to KEY and /DAT, /DA2, /DA3 to DATA, deleted records and all. This meant that one would have to go through the data base, record at a time and delete the 'deleted' records. I elected to eliminate the deleted records at transfer time. This was an easy task because Profile 4 Plus flags deleted records with 'x'00' in the first byte of the /KEY record. At the same time I could do some modification of the structure of the database.

Before I go any further, I should say a word about the files in the two programs. They are both flat files. A PROFILE 4 Plus record can be 1,024 long, 256 bytes in each of a /KEY, /DAT, /DA2 and /DA3 file. The idea is to put the critical data in the /KEY file and the non-critical stuff in the /DA\* files. A filePro 16 record can be 4,608 bytes long, a KEY file of 512 bytes and a DATA file of 4,096 bytes. The same rules apply to where to put what kind of data. This means some combining will have to be done during conversion.

In my conversions, I usually write a custom program based on a prototype that allows me to restructure the data base, eliminate the deleted records and adds the necessary header to the KEY file of filePro 16. Actually, the program is written to run in GW-BASIC because, it allows a larger record size than Model 4 BASIC.

The FIRST thing to do is to obtain a copy of your Profile 4 Plus /MAP. If you don't have one, use Profile's creation program (Define Files) to print one. Use this as a guide in defining the data base with filePro 16's Define Files. Sound familiar? You could convert the data base as-is or change some field lengths, etc. A couple of things you might consider is splitting up the city, state and zip codes or expanding the ZIP code to ZIP +4. filePro has really neat edits for State and ZIP codes. The options are endless. One of the silent gotchas in filePro 16 is that once you use a field number and go on to the next, you CANNOT insert a field into an existing data base without an add-on program (cost more than \$200). The secret is to save some empty field numbers in the KEY file when you design the DB. You don't have to reserve any file space, just the

field numbers. An example would be if you used 15 fields in your KEY file, add an extra 10 with a description of 'EXTRA' and a field length of 0. The DATA file isn't a problem because you can add fields at the end.

The filePro 16 KEY header is constructed as per Table II. Notes: Dates are stored as the number of days since January 1, 1983. User ID's are stored as the user ID number from the \etc\passwd file (Unix/Xenix) or as 'x'00' for MS-DOS.

FP16 is one of my sample transfer programs to be run in GW-BASIC.

A lot has been said about the actual transfer of data from the Model 4 to the MS-DOS format, so I won't go into much detail in that regard. I will only mention two methods that I think are the BEST! The first is to ARC the files needed with ARC4, transfer the files and PKXARC them on the MS-DOS side. The second is to ATTRIB the files on the Model 4, changing the LRL to 256 and transfer them individually. [Note: a third method is to use TRSCROSS to directly copy the files from a TRS-80 disk to an MS-DOS disk.]

So you say, you want to transfer your Profile 4 Plus DB to dBASE III Plus? It's just as easy! Do the same thing! You won't need the 20 byte header, but you will have to combine the /KEY and /DA\* files into one file. Here's a sample of the above program that does the job and does some manipulation of the data at the same time.

The BD3 program will create a flat file that can be imported into dBASE III Plus after a DB is created according to the portions of the Profile 4 Plus data you have extracted.

## Recovering Superscript Documents

David J. Kelton  
1000 Boroughbridge Road  
Richmond, VA 23225-4440

Superscript file crashes happen in a couple of ways: bad disk media can cause a sector to become unreadable, or you can exit the program without closing the file (using the RESET key, power failure, removing the disk and turning off the computer, etc.), causing an end-of-file error when you try to re-open it. You can recover these files by using the program RECOVERY/BAS. You may also need to know something about the file structure to allow RECOVERY to do its best job. With this knowledge, it may also be possible to use a direct file patch utility (like FED) to fix defective sector or directory information.

The root cause of the "record out of range" error is that Superscript does not update the ending record number (ERN) in the directory entry for the file until the file is closed with a "Quit" command, <@><Q>. One way to help minimize the need to recover files is to use frequent Quit commands followed by the "Return to current document" command, <R>, at the main menu. This works under all systems. With LDOS [or LS-DOS] use the trailing exclamation point (!) on the file name when opening the file. This trailing exclamation point will cause LDOS to update the ending record number (ERN) every time the file is written to.

### Program Operation

SRCBLOCK (by David Kelton), which operates under Model III LDOS will layout the header information which Superscript uses to access the text. This program also displays the maximum number of sectors (records) that should

appear in a directory listing. If the directory listing shows fewer records, then Recovery will not be able to recover all text. IF NOTHING HAS BEEN WRITTEN TO THE DISK SINCE THE LAST TIME THE FILE WAS PROPERLY CLOSED, it may be possible to recover the text by "zapping" the ending record number (ERN) in the DIR/SYS file to value displayed by SRCBLOCK. Further text recovery can then be done by RECOVERY if needed.

RECOVERY (by David Goben), which operates under Model I/III/4 Disk Basic, reconstructs Superscript file using whichever method yields the best results (see File Specs below). It asks for the file spec of the defective file and then asks for the output filespec. The output file will contain an ASCII conversion of the file.

Once RECOVERY receives this data, it tries to build a reconstruction table from the information in the index area of the defective file. If it accomplishes this, RECOVERY rebuilds the file in its proper order. If any of the used index sectors are defective, RECOVERY tries to reconstruct the file sequentially.

If the index sectors are unreadable, you can still use RECOVERY to gather enough information about each block to reconstruct the file. Success depends on your knowing the following facts:

1. Each block is four sectors long.
2. By adding seven to the first byte of each block you can tell where, in the last used sector of the block, the text data ends.
3. The values of the first and second bytes of each block are the same as those of the second and third bytes in the index cluster that references the block.
4. Text does not begin until after the seventh byte of the first sector of a block.
5. If a block begins with a byte value of zero (does not give an offset value for the end of data in its final occupied sector), it is a header or footer block. You should ignore such blocks dur-

ing recovery.

6. If blocks are not stored in sequential order, you must move the text into proper sequence after recovery.
7. If a sector is unreadable, RECOVERY tries to read it four times before reporting the problem and proceeding to the next sector.

When the program recovers a file, it stores the data in the destination file in ASCII format. Convert the data back to Superscript format as follows:

1. Create a new Superscript file with a new name.
2. Arrange the new file's framework such as line spacing and margins to match those used in the defective file.
3. Exit the new file (via @Q).
4. Select the ASCII conversion option at Superscript's main menu.
5. Answer the prompt asking "FROM which format do you wish to convert (Script/ASCII)?" with an "A" for ASCII.
6. Answer the "Name of Script file" prompt with the name of the new Superscript file created in 1-3 above.
7. Answer the "Name of ASCII file" prompt with the name of the "destination" file entered during Recovery's operation.
8. When the conversion is complete, edit the new file and insert any special features, such as centering, underlining, headers, or footnotes, that Recovery deleted. Recovery replaces the "copyright" symbol which precedes all the special features with #&#. Use a GLOBAL search-and-replace command to search for #&# and replace with the clear key <CLEAR>. This will restore the "copyright" symbol.



## File Structure

Unlike most word processors, SuperScript save files as blocks of text. It uses an elaborate system of pointers to locate the blocks in the proper order, to insert new blocks, etc.. A SuperScript file consists of four fundamental sections:

Sector 0 contains document header and other vital information.

Sectors 1-4 contain disk block indexing information.

Sector 5 contains list of new page markers.

Sectors 6-EOF contain the blocks of text information.

Each text block consists of four disk sectors (256 bytes/sector). Each block contains a seven byte header, up to 985 bytes of text, paragraph and control information, and 32 bytes of overflow space to accommodate minor changes without starting a new block. Each of the four possible header/footer pages will occupy its own block, if present. A document may not contain more than 174 blocks, numbered 0-173, with block zero starting at document record number six. (Note that the sector/record numbers are base zero. A directory listing will show one additional record since it is base 1.) A SuperScript file, therefore, has a maximum length of 702 sectors (6+4\*174) which will require about 175.5K of disk space and hold approximately 685,560 bytes of text including all headers and footers, if present.

Note that, if used, each header, footer and tab block will occupy one entire block (985 bytes). Thus, over 4900 bytes can be "lost" to text if all five headers, footers and tab blocks are present.

**[Editor's Note:** For detailed information on the document file structure, refer to SuperScript Document File Format, *The MISOSYS Quarterly*, Spring 1990 (Volume IV.iii, pages 21-23.)

If you are interested in this complex jumble of information, you can refer to the Ap-

pendix of David A. Kater's *TRS-80 Word Processing with SuperScript* (\$19.95, Byte Books, ISBN 0-07-033360-2) gives a detailed description of this storage format. This book is also INVALUABLE to serious SuperScript users in that it provides fabulous tips for streamlining your word processing time and efforts. (The book is not longer in print nor available from the publisher. Have to find on book shelves or in a used book store.)

Armed with the above information, a SuperScript user with a clobbered file may be able to "zap" it to a point where it will load properly and allow final repair with the normal SuperScript editing functions. The information in the disk block index MUST agree with what is actually contained in the disk blocks themselves. For example, when the FF byte is encountered in the index, it is a sign that the previous block should contain the FF end-of-text code somewhere among the valid bytes of that block.

You can recover a document by using the index information to rebuild the file in order (blocks are not always grouped sequentially, which is the reason for the index); or if the index sectors are defective, by creating a sequential listing of the file and then correcting the discrepancies in the text. Lost text may be recoverable by zapping the disk directory based on the information contained in the index table and header/footer/tab location bytes.

## References:

*80 Micro Article:* In Search of Lost SuperScript Files, April 1987, p. 114 (by David Goblen)

*Computer New 80:* Recovering SuperScript Text Files, February 1989, p. 15 (by David Goblen)

*The MISOSYS Quarterly* SuperScript Document File Format, Vol IV.iii Spring 1990, p21 (by Tom Price)

Program Name: RECOVERY/BAS and SCRBLOCK/BAS

Requirements: 32K, Model I/III/4, Disk Basic, SuperScript

## WANTED Script or SuperScript Users ...

Now you can use that fabulous word processor from Radio Shack with your non-Tandy printer! No point in missing out! Your printer CAN work great with SuperScript+!

## PowerDRIVER

These printers are fully supported

EPSON MX-80/100 Series  
EPSON FX/RX 80/100 Series  
Most all EPSON "compatibles."  
C.Itoh 8510 Prowriter  
C.Itoh Starwriter Daisy Wheel  
C.Itoh A10-20 Daisy Wheel  
Okidata 92/93 Dot Matrix

Model 4 Now Supported!

All features of SuperScript are supported to the fullest capabilities of the printer involved. Drivers are easy to install, easy to use, and no extra commands to learn!

**PowerDRIVER** Only \$17.95 each  
PLEASE SPECIFY PRINTER TYPE

## PowerSCRIPT

A HIGH-POWER modification for Script from Radio Shack.

Give Script Full Power! PowerSCRIPT is a modification for the original Script/LC which allows it to work on Model I or III and gives it the power of many of the newer expensive word processors. Define printer codes for ANY printer. Embed printer codes in the middle of a line! Alphabetized directories! User definable printer filters! User-definable HELP-FILE! Chain files together at print time from any part of the text. Logical EXIT to DOS, optional automatic linefeed after C/R, FETCH, CHAIN, and KILL command and more! Compatible with most major DOS's Simply use the enclosed INSTALL program, and you will be up and running in NO TIME! Includes full documentation. Get lots of new features for a very small price!

**PowerSCRIPT** Only \$24.95 each

US/Canada Please add \$3 Shipping/Handling Foreign \$6

**POWERJOFF**

PRODUCTS FROM MISOSYS, INC.

MISOSYS, Inc

P. O. Box 239

Sterling, VA 22170-0239

703-450-4181 or 800-MISOSYS

## Lair of the Dragon

To my knowledge, David Goben's "Lair of the Dragon", published by MISOSYS, was the last commercially produced adventure game for the TRS-80. Lair is also one of the most complex adventure games produced for the TRS-80 environment. To those who have acquired Lair from us, and for those who may be interested in delving into an adventure, this issue makes that quest just a little bit easier.

On the next seven pages, what unfolds is a map depicting the entire world of Lair of the Dragon. So at Lair's new price of \$19.95 (plus S&H), and armed with this map which makes your challenge less severe, what further excuse do you have for missing out on one of the TRS-80's last true challenges? Whet your appetite for Lair's story as you read the following introduction...

Imagine yourself sitting down behind your computer, inserting a program disk, and as your fingers kiss the keys, a blast from the beyond suddenly rips you right out of your seat and transports you into a mythical land filled with mystery and magic? This is the stuff that interactive fiction is made of: for in place of just sitting back safely and comfortably in your easy chair as you read about the daring and exciting exploits of someone else, you instead read about the events that YOU YOURSELF create for the main character, who ALSO happens to be YOU! And instead of reading from start to finish a story that never varies in its content no matter how many times

that you may read it, you can change the plot and vary the action within the story each time that you choose to go through it. YOU will decide how the main character will approach and overcome a problem. YOU will be the one who will have to defend yourself against an offending foe. You alone will decide when, how and where the daring deeds are to be done. Instead of reading about how the main character cleverly solved a puzzle or ingeniously navigated a mind-twisting maze, it shall be for YOU to test your skills at deducting and reasoning these courses. And if you fail to solve a puzzle or else get hopelessly lost, you will have only yourself to blame.

Here the hero or heroine is not safely shielded from death by being locked into a singular, never-changing storyline. Here they can get killed for a foolish misstep, an undiplomatic gesture, or by failing to watch out for an approaching danger. It is you who is responsible for exploring the lands and seeking behind the hidden corners. You are the one who will be solely responsible for the ultimate outcome of the adventure.

Let me tell you, fighting off an old geezer of a dragon is no easy task. Especially if it is a MEAN old geezer of a dragon. And ESPECIALLY if it happens that the battle is waged on the fields of a MegAdventure. What would YOU do if you had to pit your wits against an ill-tempered dragon who has been terrorizing your village, raping your fields and starving your children? And how would you go about tracking it down in the mysterious northern mountains, the likes of which only heart-quickenning tales of terror and death are told?

MegAdventure is unlike virtually any other interactive fiction adventure that you have ever played, for it will more than just

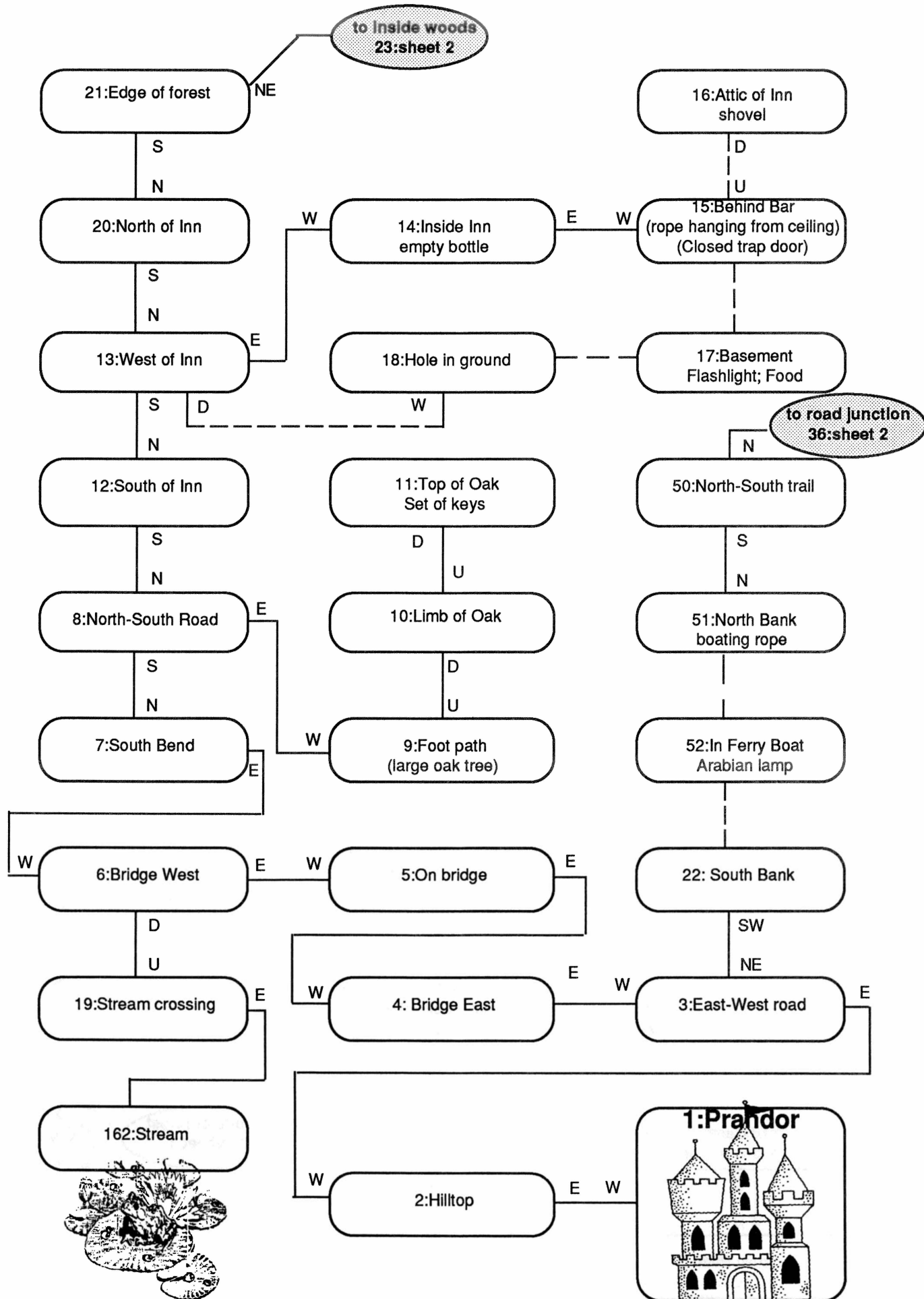
paint its pictures upon the canvas of your imagination - it will slap the sweat right onto your forehead!

If you truly believe that discovery is one of the finest points in life, if you would like to test your ability to think logically to the fullest extent of your ability, if you would like to take on the largest adventure ever written in the genre of interactive fiction, and if you have the guts to face that which would make any other mortal elf cringe in fear, then Lair of the Dragon, the first of the MegAdventure series, is your cup of poison; for reward is a hard-earned commodity here, not given easily to the timid and the faint-hearted.

So grab your flashlight, sack a lunch, and don't forget to pack your sneakers; for you are about to trek into an adventure whose sheer realism and vastitude will overwhelm you. Come on board to an exciting way to read a story; one that stars YOU as the daring deed-doer. And if this is your first journey into interactive fiction adventures, then WELCOME! You are in for a glorious challenge. And if you are an old hand at adventuring, then be prepared for a worthy opponent.

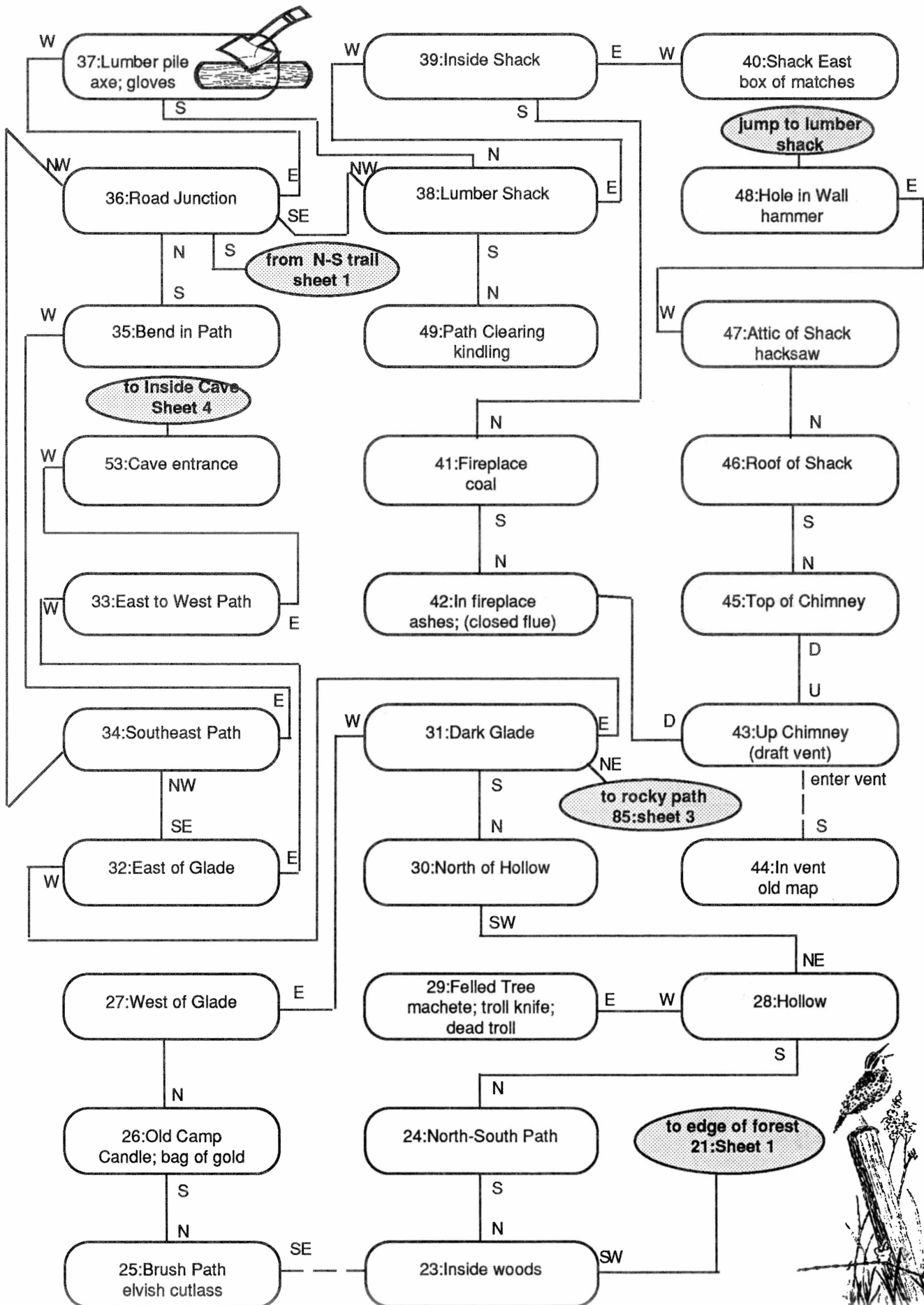
MegAdventure rips the door to adventure right off its hinges! Lair of the Dragon is available for TRS-80 Model III/4 computers or MS-DOS systems. Lair of the Dragon [TRS-80 M-55-021; MS-DOS M-86-021], just \$19.95 plus \$3 S&H (\$4 Canada, \$5 foreign).

# Sheet 1

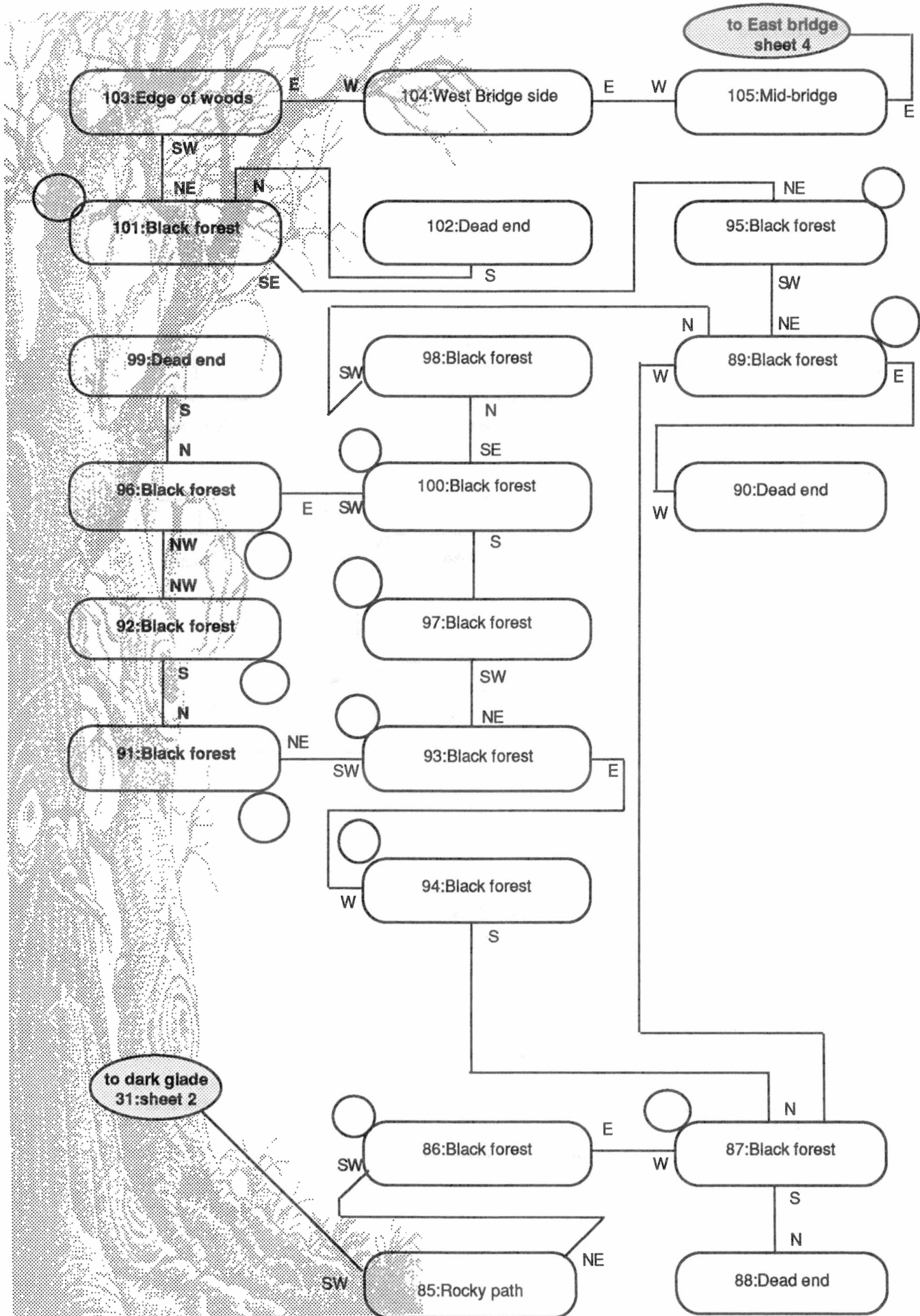




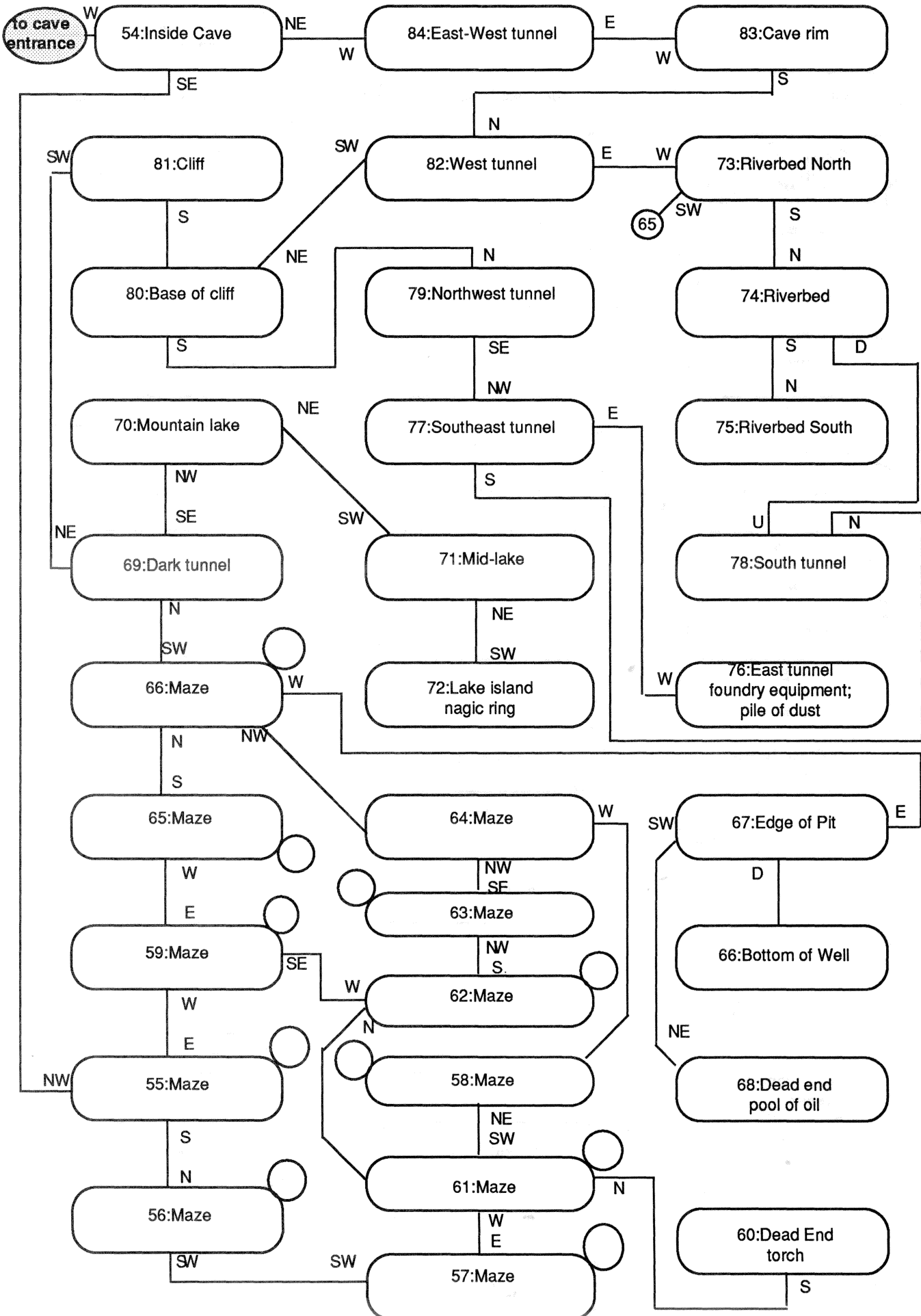
# Sheet 2



# Sheet 3

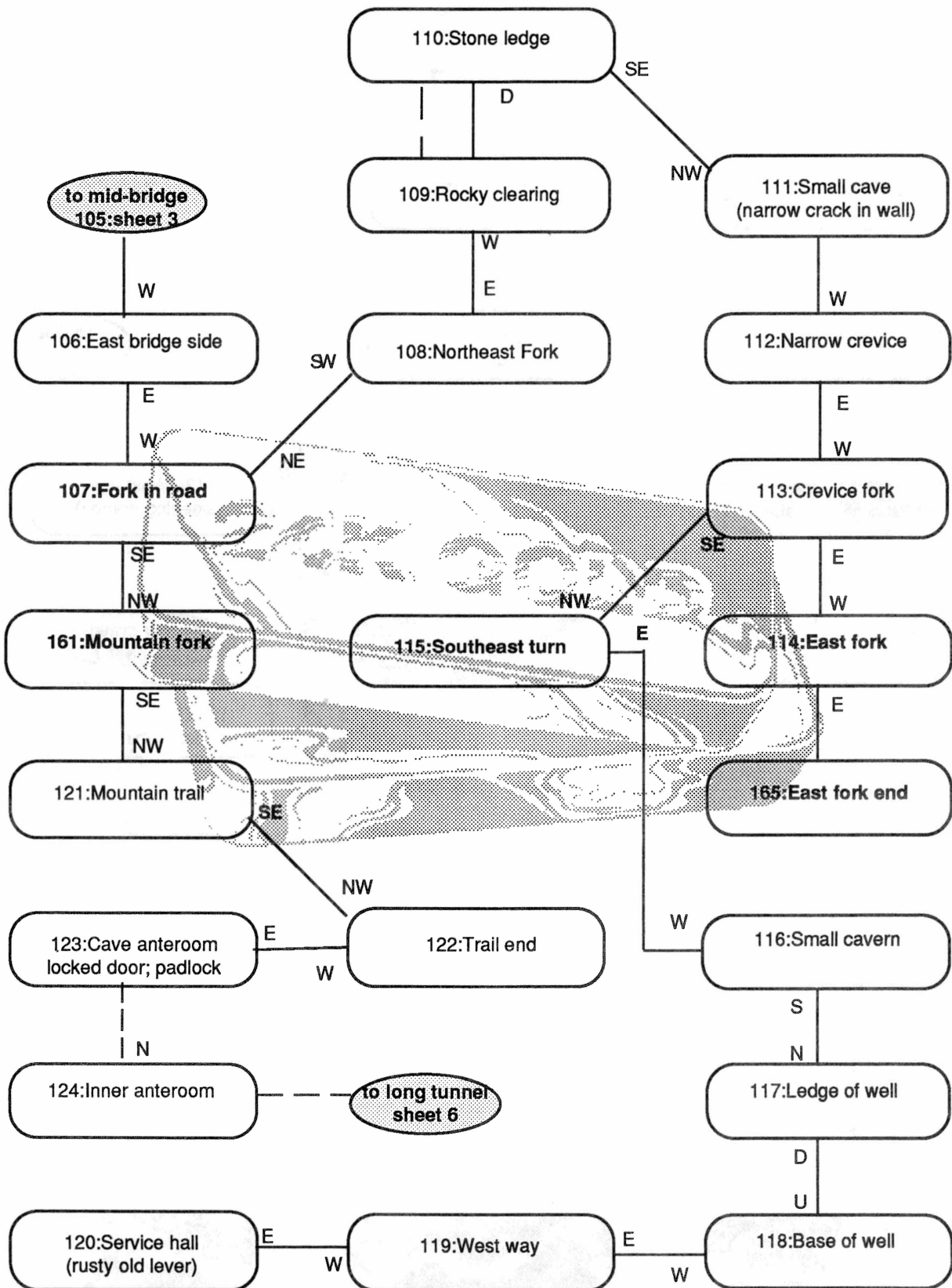


# Sheet 4

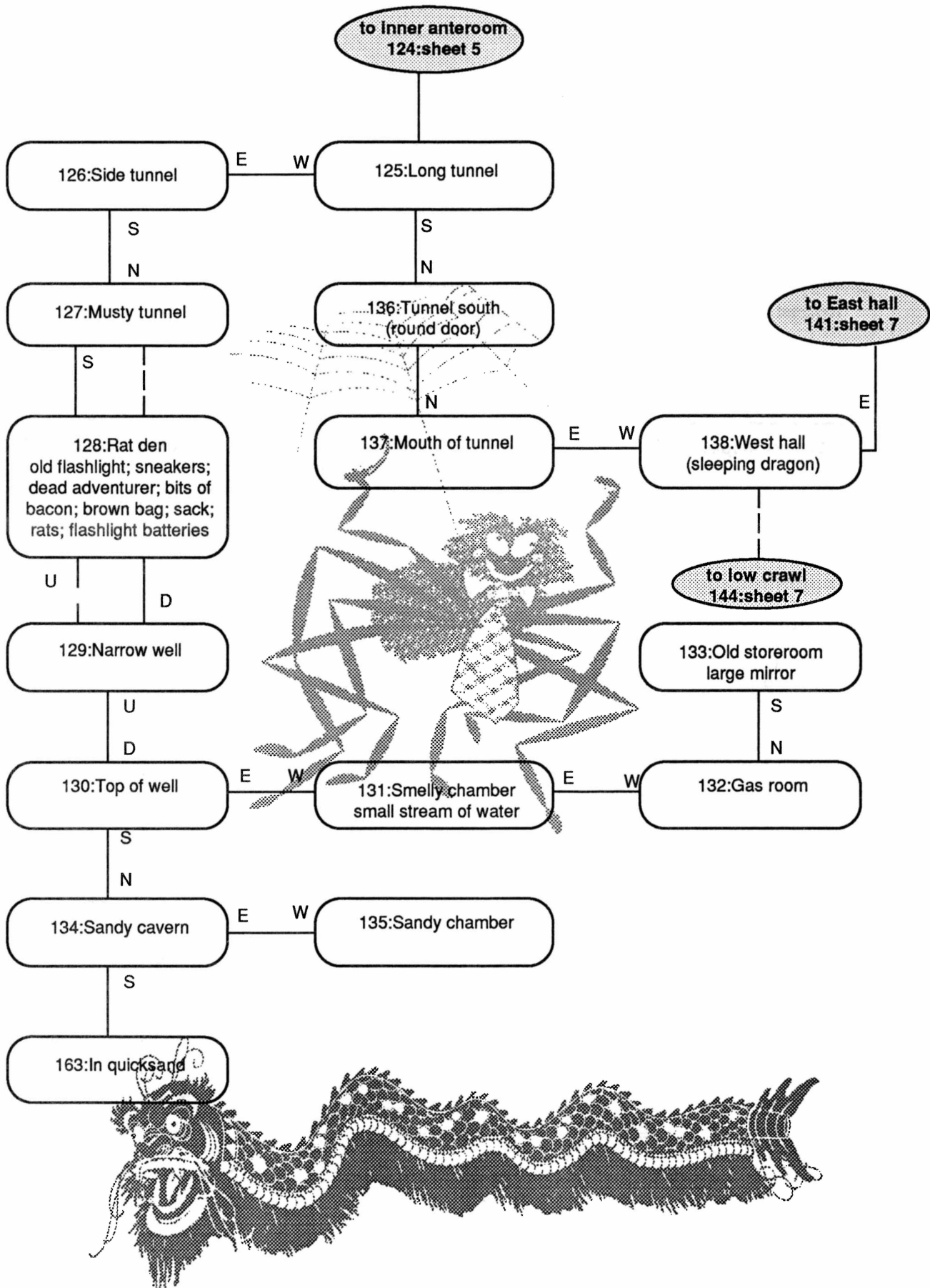




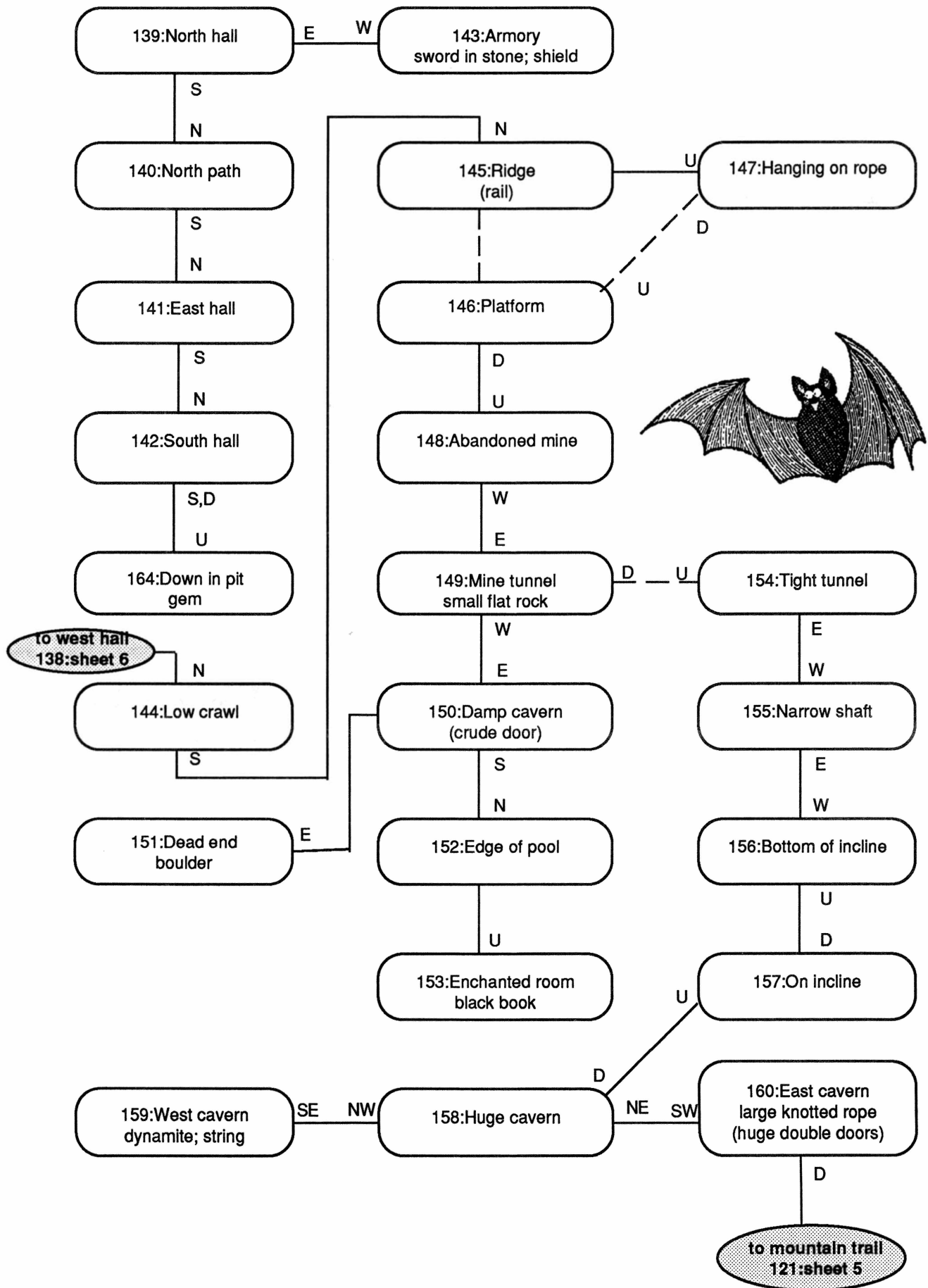
# Sheet 5



# Sheet 6



# Sheet 7







# PowerMail Plus

"This is one of the BEST mail-list programs we've seen for the TRS-80, and for its price, it's a down-right bargain."

INFOWORLD

Rated FOUR STARS in the May 1984 80-Micro!

Also selected as a 1st Place Winner in 80-Micro's Readers Choice

## PowerMail Plus



MISOSYS, Inc  
PO Box 239  
Sterling, VA 22170  
Model 4  
(Model I/III and II/12/16  
versions also available)  
\$150

Easy to use?	★★★★☆
Good docs?	★★★★☆
Bug free?	★★★★★
Does the job?	★★★★★

### ● INFOWORLD's Essential Guide To The TRS-80 said...

"If you need to maintain large mailing data-bases, you should definitely consider PowerMAIL Plus... It is exceptionally fast The program is intrinsically efficient... overall concept and design are excellent. PowerMail Plus offers the kind of features that businesses, in particular, need."

### ● 80-MICRO said in their May '84 issue,

"If you're looking for a mailing-list system, I highly recommend PowerSoft's PowerMail Plus. (The) manual is high in quality, content, and workmanship. The commands were easy to use and remember."

### ● OMNI's Complete Catalog of Computer Software wrote...

"This is a very potent mailing-list program with many versatile features. ...It lives up to (PowerSoft's) normal high-standards... it's an excellent mailing list program."

PowerMAIL Plus is a highly acclaimed, deluxe mailing/data storage system written entirely in machine language for maximum operating speed. Besides being a mailing system, it is also capable of generating custom personalized "form letters" to whatever groupings of labels you wish using the TEXT-MERGE module, available separately. Many of its features, such as extensive data "flags" enable you to use it as a limited data-base manager, too. PowerMAIL+ is a "DISK-BASED" system, rather than "RAM-BASED", therefore it can keep track of over 520,000 individual entries by combining up to eight different mailing lists (which, of course would require a hard drive). There are no "slow" periods when PowerMAIL+ is running. Features have been added to this program that others have always lacked. You now have the ability to keep track of mailings using 24 user defined "flags" that are incorporated into the PowerMAIL+ program. Separate any category in any manner you wish.

PowerMail+ can use multiple floppies by logging them in and out of the system. The program will run in as little as a 32K one drive environment, but dual drives or 80 track or double sided drives, or a hard drive are recommended for more serious use. The more available freespace, the more names you can store. A Model III or 4 data disk will hold about 1150 names. Double that for 80 track or 40 track double-sided drives (4D). Works great on hard drives as well as floppies! Handles drives 0-7. PowerSoft used it on their hard drive system for over four years with excellent results! Does not pre-allocate the entire drive, but allows you to define how many records you need at initialization time. If you need more room later, there is a way get your names onto a LARGER file easily. Also you may have a file on each disk or platter, if you have multiple drives, or a hard drive. This further increases overall storage, as PowerMAIL+ will treat the files individually, or as an overall total. TWELVE LEVEL SORT! The program will sort up to TWELVE levels that you wish in any order. Other features include the ability to separate your flags and put them onto another file, merge files together, and then separate whatever you want, improved "key" search, improved field lengths, improved disk I/O, and much improved print routines. The main thing here is that PowerMAIL+ was designed to be easy to use, easy to start up, and easy to train people to enter your names.

With PowerMAIL+, you have many print options based on Labels or Lists. If you choose labels, you have a choice of two modes and are asked how many labels across and the spacing required. The default settings are for the standard one up label commonly available (you may choose up to 4 across). If you choose listings, you have four choices to choose from there. If printing a LIST rather than LABELS, a user-definable "header" will be printed, along with the page number. This really makes an impressive printout, as well as adds to the usefulness.

PowerMAIL Plus' print system can be controlled from flag settings, allowing you to print only a part of the file. You may keep lists within lists. No need to keep separate files for different classifications. Use the flag or DATA fields for separating them for printout. There is a lot of flexibility here! In addition, you can tell it to SET FLAGS after printing to designate WHO got

a label, etc. This will contribute tremendously to the efficiency of your mass mailings, because for the first time you can keep track of who has been sent a particular mailing. Avoid duplicate mailings with ease! Let PowerMAIL Plus do it for you!

PowerMAIL Plus also incorporates a very fast search mode for locating a particular entry in the shortest possible time (supports wild cards too), as well as a sort-merge routine which allows you to sort files larger than can fit into memory. Best of all, PowerMAIL Plus includes routines which will convert files created by some of the other popular mailing list systems to its own internal format, so you don't waste any time retyping.

### Converts data files from these mail systems:

Radio Shack MailList Expanded  
Radio Shack MailList Compressed  
Special Delivery  
Extra Special Delivery

POSTMAN  
PowerMail 1.0  
Galactic Mailfile

## Text-Merge

TEXT-MERGE is a stand-alone form letter utility for PowerMAIL Plus. It will take a form letter prepared in ASCII format by any word processor or text editor (i.e. TED), and merge the contents of a PowerMAIL Plus ADDER file into the letter. The form letter may be as long as available memory (usually about 32K) or as short as a few lines. TEXT-MERGE is ideal for creating personalized mailings or specially-formatted mailing labels. You use PowerMAIL Plus' flag system to separate those records for which you want form letters printed into an ADDER file, then simply run TEXT-MERGE.

Prompts allow you to specify such parameters as page size, number of printed lines per page, line length, left margin, whether or not linefeeds follow carriage returns, and whether or not you want to pause after each page. During printing a count of letters printed is displayed on the screen.

Since TEXT-MERGE accepts all ASCII codes from 0 to 255 as valid data, you can create special print effects by embedding printer control codes in your form letter text. If you have a word processor which will permit this.

Fields from the PowerMAIL Plus record are inserted into the form letter at places marked by @n, where n is a digit from 0 to 9, which corresponds to the record fields. Field data may be inserted in any order, and each field may appear as many times as necessary.

Here is how you might start out a form letter;

Mr. @2 @1

@3

@4 @5

@6, @7 @8

Dear Mr. @1 ,

Here is what the resulting printout would look like;

Mr. Robert Jones  
ABC Paper Co.  
1632 North St. Suite 101  
Ft. Worth, TX 76751

Dear Mr. Jones,

Model I/III/MAX-80 or Model 4/4P. Please Specify which computer!

# BACK/REST 1.3

with **RESTORE/DCT** now included!

A **FAST** Hard Drive "Image" Backup and Restore Utility.  
Backups or restores an entire logical drive, or file-by-file.  
*Files larger than a floppy are **NO PROBLEM** any longer!*

This is a **MUST-HAVE** program if you use a **hard drive** with LDOS or TRSDOS 6".

Awarded \*5 **STARS**\* in the **October '85** Issue of 80-MICRO! To quote from the review...

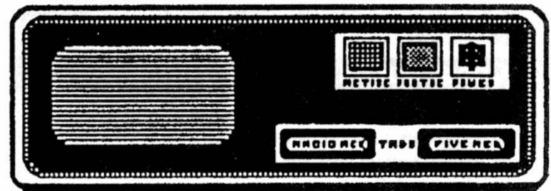
*"**BACK/REST** is strongly recommended as insurance against **possible** digital disaster."*

★ ★ ★ ★ ★  
80 Micro, October 1985  
**Backrest** runs on the Models I, III (48K), and 4 (64K) and requires one disk drive and a hard drive. Powersoft Products, 17060 Dallas Parkway, Suite 114, Dallas, TX 75248, 214-733-4475. \$99.95.  
Easy to use: ★ ★ ★ ★ ★  
Good docs: ★ ★ ★ ★ ★  
Bugs: ★ ★ ★ ★ ★  
Does the job: ★ ★ ★ ★ ★

- **1991 Price:** \$34.95 + S&H
- For Model I/III or Mod 4 TRS-80 (or MAX-80).
- *All versions included on same disk.*
- Works under LDOS 5 or TRSDOS/LS-DOS 6.x.
- TRSDOS 6.2/LS-DOS 6.3 is fully supported.
- Program is not protected.
- **BACK/REST** will work with all TANDY hard drives and most others that use LDOS or TRSDOS 6.

How many times have you hard drive users put off making that all-important backup because of the length of time it takes? You know you should backup everyday, right? Let's face it, the normal file-by-file backup method employed by today's operating systems takes so long that it actually discourages people from doing backups. So you only do it every now and then... When you think of it... Maybe once a week. *Right?* You're ok until the day that disaster strikes.

Then there's ALWAYS the problem of files **too large to fit** on a floppy! Like PROFILE or other types of data files, **BACK/REST** solves all that! It segments those and can restore them back without you having to worry about it and figure it all out. **BACK/REST** is a **FAST** backup and restore utility for hard drives which can create a mirror-image of your hard drive partition on floppy disks in 20% of the time a file-by-file backup would normally take; a 5 megabyte drive should take about 10-15 minutes. There is **NO EXCUSE** for not backing up your hard drive now... do yourself a favor. **BACK/REST** is simple to use and will even tell you how



many floppies you will need to have formatted in advance for backup. We've tried to think of everything! It's a terrible feeling when you are doing a backup and are requested to insert another disk, yet you don't have any more formatted! Now **BACK/REST** will tell you how many disks you need to have on hand **BEFORE** you start. Furthermore, only allocated cylinders of the hard drive are copied to speed things up substantially.

Through the use of a special DRIVER program (**RESTORE/DCT**) supplied with **BACK/REST 1.3**, you can **READ** or **COPY** individual files from your **BACK/REST** backup floppies back to the hard drive. **RESTORE/DCT** is installed with the **SYSTEM (DRIVE=x, DRIVER)** command just like any other disk driver and will prompt you when to swap disks in order to extract the file you want. You now have the best of all worlds: a **FAST, COMPLETE** backup/restore utility for a hard drive that gives you the option of restoring either an entire hard drive partition **OR** a single file! When restoring to hard drive using the "image" mode, the floppies can be inserted and read in any order. When restoring "file-by-file", you are prompted for the proper disk #. You may restore a single file or any set of files that you want to. Never be at the mercy of a crashed hard drive again. **BACK/REST** is the answer to every hard drive user's prayers!

If you've invested good money into a hard drive system, it doesn't make any sense not to have a **BACKUP** routine. Especially if you're running your business on it! **BACK/REST** makes that job easier and saves much time. And **TIME IS MONEY. Don't delay! Save HOURS!** **BACK/REST** will work with all TANDY hard drives and most others that use LDOS or TRSDOS- 6, including the Tandy 15 and 35 Meg drives.

**MISOSYS, Inc.**  
**PO Box 239**  
**Sterling, VA 22170-0239**  
**703-450-4181 (orders to 800-MISOSYS)**

# 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 + \$3S&H

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.

### TRSCROSS

Now you can transfer TRS-80 Model III/4 files directly to your MS-DOS disks right on your PC. Convert BASIC programs; Convert SuperScript document files to DCA-RFT. Only \$89.95 + \$4S&H

Why buy just a FAX board, when the ZOFAX 96/24 from MISOSYS includes a 2400 baud modem for a few bucks more?

Turn your PC into a FAX machine!

- ✓ Send and receive FAX from any CCITT Group III Fax Machine or PC Fax
  - ✓ Auto receive and print incoming Fax messages
  - ✓ Background receiving
  - ✓ Distribut Fax messages to multiple destinations
  - ✓ Fax mail merge
  - ✓ Time schedule transmission to take advantage of low nighttime rates
  - ✓ 2400 bps Fully Hayes Compatible Modem
  - ✓ Includes powerful but easy to use BITCOM and BITFAX software
- Introductory price: \$225 + \$6 S&H

### Speed up your program's computational execution with an IIT Math Coprocessor

IIT's CMOS coprocessors use less power, execute faster, support 4 x 4 matrix transformation, have thirty-two 80-bit numeric registers, come with a factory 5-year warranty, and cost less! If you use your machine for spreadsheets, desktop publishing, CAD, etc., consider an IIT coprocessor. For XT's or AT's with a math coprocessor socket. Installation instructions included. S&H is \$5.

IT-2C87-100	\$237.00
IT-2C87-125	\$289.00
IT-2C87-200	\$307.00
IT-3C87-20	\$349.00
IT-3C87-25	\$447.00
IT-3C87-33	\$545.00
IT-3C87SX-16	\$354.00
IT-3C87SX-20	\$385.00

### EXPANZI!™ Disk Expander Card

With the new EXPANZI! data compression card, you can boost hard disk capacity up to three times. EXPANZI! plugs into any open slot and intercepts calls to and from the disk controllers. Compresses and decompresses in real time. Requires PC/XT/AT or compatible running DOS 3.x. Just \$179.10 + \$6S&H.

Are you still fussing with floppies for BACKUP? CMS' QIC40 DJ10 tape drive from MISOSYS is your solution!



The Colorado Memory Systems' JUMBO tape drive is one drive that fits all computers. It comes ready for internal use in AT's, XT's, and PC's, and connects to your floppy disk controller. Our adaptor board can be used to connect JUMBO when all FDC connectors are in use. Kits are available for XT/AT or PS/2 machines which convert Jumbo to external use.

- In about 5.5 minutes, a DJ10 backs up 10MB's file-by-file - the fastest in the industry! 40MB's gets backed up in about 18 to 20 minutes. Uses industry-standard DC2000 or DC2010 tape cartridges.
- The DJ10 plugs into your floppy disk controller to save cost, power, and a slot. Needs 5-1/4" (or 3.5" with faceplate) mounting slot.
- Optional adaptor board mounts in your host computer to provide an additional tape port — especially useful in 286 and 386 computers. When used with the external DJ10, it lets you share your drive between computers. Note: external adaptor includes "Tape Adaptor"
- The DJ10 has custom chips, high speed brushless motors, automatic circuits, no pots to vibrate out of calibration, and fewest parts to assure long life and technical leadership for years to come.

DJ10 Jumbo	\$275 (\$5S&H)
Tape Adaptor	\$75 (\$3S&H)
External Adaptor	\$110 (\$5S&H)
DC2000 tape (40M)	\$20.00
DC2120 tape (60M)	\$25.00

Note: DJ10 price includes one DC2000 tape.

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



### Powerful features in Version 3!

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. Works with DOS 4.

### 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
- ✓ Touch a directory file entry with your date and/or time
- ✓ Obtain complete disk statistics in one screen; includes usage data
- ✓ Alter file attributes: archive, system, hidden, read
- ✓ Save/Restore sectors to/from auxiliary buffers; for moving data around
- ✓ DOS subshell available while using DED86 lets you shell to other programs
- ✓ Handles 5.25" & 3.5" drives & RAM disks; even large partition drives

DED86 is easy on your pocketbook; only \$29.95 + \$5S&H

MISOSYS, Inc.  
P.O. Box 239  
Sterling, VA 22170-0239

800-MISOSYS  
or 703-450-4181



# ATTENTION!

**Radio Shack Tandy Owners ...**  
**Find the computer equipment that Tandy no longer sells!**  
**Original Tandy Equipment!**

## COMPUTERS

Model 3, 2 Dr.	\$255
Model 4, 2 Dr.	\$345
Model 4, 2 Dr. 128K	\$365
Model 4P, 2 Dr.	\$345
Model II	\$295
T-6000, 15 HD.	\$745
Tandy 1000's, 1200's, & 2000's	Call

## LAPTOPS

Model 100, 24K	\$235
Model 100, 32K	\$255
Model 200	\$335
Model 600	\$325
Tandy-1400	\$665
Tandy-1400 HD	\$945
Tandy-1100 FD	\$535

## HARD DRIVES

for model 3's and 4's

Tandy 5 Meg HD.	\$275
Tandy 12 Meg HD.	\$345
Tandy 15 Meg HD.	\$425

\*All hard drives include cable and software.

## PRINTERS

DMP 105	\$ 75
DMP 120	\$145
DMP 130	\$175
DMP 200	\$155
DMP 420	\$295
DMP 430	\$365
DMP 2100	\$395
DMP 2100P	\$445
DMP 2110	\$350
Line printer V	\$195
DWP 210	\$135
DWP 410	\$245
DWP 510	\$345

## ACCESSORIES

Tractor feed for 2100	\$115
Tractor feed for DW II	\$115
Tractor feed for 410 (new)	\$ 75
DW II sheet feeder (new)	\$195

## SOFTWARE

Allwrite Model 4	\$55	PFS Report Model 4	\$40
CPM Plus Model 4	\$55	Profile 4 Plus	\$75
DOS Plus Model 4	\$40	Scipsit Pro Model 4	\$75
Deskmate Model 4D	\$69	Superscripts Model 4	\$75
Deskmate Model 4	\$69	TRS DOS 6.2 Utilities Model 4	\$29
Formation Model 4	\$40	TK Solver Model 4	\$75
General Ledger Model 3/4	\$75	Target Planner Calc Model 4	\$40
Micro Term Model 4	\$35	T/Maker Model 4	\$55
Model 4/4P TRS DOS Version 6	\$29	Assembly Language Tutor Disk Model 3	\$30
Model 4D TRS DOS Version 6	\$29	Bulletin Board Model 1-2-3	\$45
Money Decisions III Business Statistics and Forecasting Model 4	\$25	Inventory Control Model 3	\$30
Multipan Model	\$95	Power Tool Model 3	\$29
PFS File Model 4	\$40	Profile 3 Plus HD Model 3	\$40

\*All equipment is guaranteed to be in good working order.

\*Equipment is cleaned and tested.

\*Drives are cleaned and timed as needed.

We accept VISA & MasterCard or C.O.D. The above prices do not reflect shipping cost. Inventory changes daily; please call for availability. If you don't see what you need, please call and we will do our best to locate it for you.

## PACIFIC COMPUTER EXCHANGE

The One Source For Used Tandy Equipment!  
 1031 S.E. Mill, Suite B • Portland Oregon 97214  
**(503) 236-2949**



# MISOSYS 20 or 40 MB Hard Drives for your Model III or 4

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 sports a hardware real time clock. With its internal battery lifetime in excess of 10 years, never enter date and time again. A joystick port option adds a 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 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 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. 20MB drive packages are currently built with a Kalok 3.5" drive; 40MB packages use a Seagate ST251-1 28 millisecond drive. Drive packages are offered as 'pre-assembled kits', assembled to order and fully tested; all you will need to do is plug it in and install the software. 50-pin host interface cable included.

1991 Prices currently in effect:

## Complete Drive Kits:

20 Megabyte kit:	\$475
40 Megabyte kit:	\$595
Joystick option	\$20
LDOS software interface	\$30

## Piece Parts:

20 Meg drive (KL320)	\$200
40 Meg drive (ST-251-1)	\$320
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.

Prices subject to change without notice

## PRO-WAM™ Version 2 Window & Application Manager



## LB Data Manager A flexible data manager

Our applications turn your 128K Model 4 into a sophisticated business or personal machine because easily installed PRO-WAM comes with many useful and powerful menu-driven time savers and work organizers. PRO-WAM is accessed with a single keystroke; its export and import functions allow you to move data across windows between programs.

- Address CARDS, LABELS, and HEAD display & export
- BRINGUP tickler file; new PRINTING and sorting
- CALENDAR flags BRINGUP items visually on screen
- Ten 3 x 5 CARD files with FORMS and FIELDS
- Virtual PHRASE access for export
- New TODO list manager with "who does it"
- Plus many other vital applications!

PRO-WAM [M-51-025]

\$74.95 + \$5S&H

LB is easily used by anyone for managing their data. It's menu driven for ease of use; absolutely no programming needed. Requires a Model 4 with 128K or a hard drive. LB86, an MS-DOS version is also available. Now activate PRO-WAM from newly compiled LB beta release (hardware restrictions apply)

- Store up to 65534 records per data base
- Up to 1024 characters (64 fields) per record
- Nine field types for flexibility
- Select and sort on up to 8 fields (multiple indexes)
- 10 input/update screens per data base
- 10 printout formats per data base
- Extensive on-line help available

LB [L-50-510]

\$74.95 + \$5S&H

## DoubleDuty 30% off until May 31st

DoubleDuty 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 multi-tasking, 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.

DoubleDuty works with all expanded memory known to the DOS. A BANK parameter lets you set DoubleDuty to 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 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, 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). If you don't already own DoubleDuty, now is the time to acquire it. The regular price of DoubleDuty is \$49.95. On sale now until May 31st for \$34.97 + \$3S&H (U.S.).



MISOSYS, Inc.  
P. O. Box 239  
Sterling, VA 22170-0239  
U.S.A

Contents: Printed Matter

**BULK RATE**  
**U. S. POSTAGE**  
**PAID**  
**Sterling, VA**  
**PERMIT NO. 74**

Attention Postmaster: Address Correction Requested  
Forwarding and return postage guaranteed