The Official ZENITH/Heath Computer Users Magazine

REMark®

December 1989

Look for Santa Bill's Gift Ideas for this Year!

Page 7

The Evolution of DOS

Page 43


"How Can You Take Advantage of Me"
". . . If you don't call? I have everything you could possibly want! My software selection continues to grow, and remains my most popular feature. I'm fast, but, if you don't have the time to download, these software disks can now be purchased for a small copying charge! My message base has also become quite popular. Through it, HUGgies are exchanging more information than ever before. Finally, there's my legendary Bargain Centre. It alone, will make you come back for more! Did you know how inexpensive I am? Why pay $14 per hour connect time to someone else when your phone company charges less than $12 per hour (less on weekends) from anywhere within the continental U.S.! So, go ahead and take advantage of me. Just set your modem for 300, 1200, or 2400 baud (8N1) and call (616) 982-3956. You needn't type anything, I'll know you're there!"
McC

The Official ZENITH/Heath Computer Users Magazine

REMark®

Volume 10, Issue 12 • December 1989

PC Compatible

PC Compatible and H/Z-100

H/Z-100 (Only)

General

PC Compatibles

All models include the following series of computers: H/Z-130, 140, 150, 160, 170, 180, H/Z-200 and 300.

Resources

Why didn't somebody think of this sooner! Books and magazines are printed on both sides of the page, why not listings from your computer's printer? With HUG's ***"Both Sides"*** printer utility, now you can send your printer's output to both sides of the page; properly positioned, and page numbered! Cut your paper usage, and binder space usage in half! Order HUG's ***"Both Sides"*** printer utility today. ***Both Sides*** is available for MSDOS and is ***HUG P/N 885-3048***.

Managing Editor ..... Jim Buszkiewicz
(616) 982-3837

Software Engineer ..... Pat Swayne
(616) 982-3463

Production Coordinator ..... Lori Lerch
(616) 982-3794

Secretary ..... Margaret Bacon
(616) 982-3463

HUG Bulletin Board ..... (616) 982-3956
(Modem Only)

HUG Parts Ordering ..... (616) 982-3463

Contributing Editor ..... William M. Adney

Advertising .... Rupley's Advertising Service
Dept. REM, 240 Ward Avenue
P.O. Box 348
St. Joseph, MI 49085-0348
(616) 983-4550

Printer ..... Imperial Printing
St. Joseph, MI

	U.S. Domestic	APO/FPO & All Others
Initial	$22.95	$37.95*
Renewal	$19.95	$32.95*

*U.S. Funds

Limited back issues are available at $2.50, plus 10% shipping and handling — minimum $1.00 charge. Check HUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and appropriate added cost.

Send Payment to: Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022
(616) 982-3463

HUG

H8 - H/Z-89/90

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ACCOUNTING SYSTEM	885-8047-37	CPM	BUSINESS	20.00
ACTION GAMES	885-1220-[37]	CPM	GAME	20.00
ADVENTURE	885-1010	HDOS	GAME	10.00
ASCIRITY	885-1238-[37]	CPM	AMATEUR RADIO	20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PACKAGE	885-1119-[37]	HDOS	UTILITY	20.00
CASTLE	885-8032-[37]	HDOS	ENTERTAINMENT	20.00
CHEAPCALC	885-1131-[37]	HDOS	SPREADSHEET	20.00
CHECKOFF	885-8010	HDOS	CHECKBOOK SOFTWARE	25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY	20.00
DISK UTILITIES	885-1213-[37]	CPM	UTILITY	20.00
DUNGEONS & DRAGONS	885-1093-[37]	HDOS	GAME	20.00
FLOATING POINT PACKAGE	885-1063	HDOS	UTILITY	18.00
GALACTIC WARRIORS	885-8009-[37]	HDOS	GAME	20.00
GALACTIC WARRIORS	885-8009-[37]	CPM	GAME	20.00
GAMES 1	885-1029-[37]	HDOS	GAMES	18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY	30.00
HDOS PROGRAMMERS HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUPLICATION UTILITIES	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PRODUCTS THRU 1982	9.75
HUGMAN & MOVIE ANIMATION	885-1124	HDOS	ENTERTAINMENT	20.00
INFO. SYSTEM AND TEL. & MAIL SYSTEM	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DATABASE	25.00
MAPLE	885-8005	HDOS	COMMUNICATION	35.00
MAPLE	885-8012-[37]	CPM	COMMUNICATION	35.00
MICRONET CONNECTION	885-1122-[37]	HDOS	COMMUNICATION	16.00
MISCELLANEOUS UTILITIES	885-1089-[37]	HDOS	UTILITY	20.00
MORSE CODE TRANSCEIVER	885-8016	HDOS	AMATEUR RADIO	20.00
MORSE CODE TRANSCEIVER	885-8031-[37]	CPM	AMATEUR RADIO	20.00
PAGE EDITOR	885-1079-[37]	HDOS	UTILITY	25.00
PROGRAMS FOR PRINTERS	885-1082	HDOS	UTILITY	20.00
REMARK VOL 1 ISSUES 1-13	885-4001	N/A	1978 TO DECEMBER 1980	20.00
RUNOFF	885-1025	HDOS	TEXT PROCESSOR	35.00
SCICALC	885-8027	HDOS	UTILITY	20.00
SMALL BUSINESS PACKAGE	885-1071-[37]	HDOS	BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PACKAGE	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PACKAGE	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUNICATION & UTILITY	20.00
TINY BASIC COMPILER	885-1132-[37]	HDOS	LANGUAGE	25.00
TINY PASCAL	885-1086-[37]	HDOS	LANGUAGE	20.00
UDUMP	885-8004	HDOS	UTILITY	35.00
UTILITIES	885-1212-[37]	CPM	UTILITY	20.00
UTILITIES BY PS	885-1126	HDOS	UTILITY	20.00
VARIETY PACKAGE	885-1135-[37]	HDOS	UTILITY & GAMES	20.00
WHEW UTILITIES	885-1120-[37]	HDOS	UTILITY	20.00
XMET ROBOT X-ASSEMBLER	885-1229-[37]	CPM	UTILITY	20.00
Z80 ASSEMBLER	885-1078-[37]	HDOS	UTILITY	25.00
Z80 DEBUGGING TOOL (ALDT)	885-1116	HDOS	UTILITY	20.00

H8 - H/Z-89/90 - H/Z-100 (Not PC) *

ADVENTURE	885-1222-[37]	CPM	GAME	10.00
BASIC-E	885-1215-[37]	CPM	LANGUAGE	20.00
CASSINO GAMES	885-1227-[37]	CPM	GAME	20.00
CHEAPCALC	885-1233-[37]	CPM	SPREADSHEET	20.00
CHECKOFF	885-8011-[37]	CPM	CHECKBOOK SOFTWARE	25.00
COPYDOS	885-1235-37	CPM	UTILITY	20.00
DISK DUMP & EDIT UTILITY	885-1225-[37]	CPM	UTILITY	30.00
DUNGEONS & DRAGONS	885-1209-[37]	CPM	GAMES	20.00
FAST ACTION GAMES	885-1228-[37]	CPM	GAME	20.00
FUN DISK I	885-1236-[37]	CPM	GAMES	20.00
FUN DISK II	885-1248-[37]	CPM	GAMES	35.00
GAMES DISK	885-1206-[37]	CPM	GAMES	20.00
GRADE	885-8036-[37]	CPM	GRADE BOOK	20.00
HRUN	885-1223-[37]	CPM	HDOS EMULATOR	40.00
HUG FILE MANAGER & UTILITIES	885-1246-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PRODUCTS 1983 THRU 1985	9.75
KEYMAP CPM-80	885-1230-[37]	CPM	UTILITY	20.00
MBASIC PAYROLL	885-1218-[37]	CPM	BUSINESS	60.00
MICRONET CONNECTION	885-1224-[37]	CPM	COMMUNICATION	16.00
NAVPROGSEVEN	885-1219-[37]	CPM	FLIGHT UTILITY	20.00
REMARK VOL 3 ISSUES 24-35	885-4003	N/A	1982	20.00
REMARK VOL 4 ISSUES 36-47	885-4004	N/A	1983	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00
SEA BATTLE	885-1211-[37]	CPM	GAME	20.00
UTILITIES BY PS	885-1226-[37]	CPM	UTILITY	20.00
UTILITIES	885-1237-[37]	CPM	UTILITY	20.00

Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
X-REFERENCE UTILITIES FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATION	20.00
H/Z-100 (Not PC) Only				
ACCOUNTING SYSTEM	885-8048-37	MSDOS	BUSINESS	20.00
CALC	885-8043-37	MSDOS	UTILITY	20.00
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	SPREADSHEET	20.00
CHECKBOOK MANAGER	885-3013-37	MSDOS	BUSINESS	20.00
CP/EMULATOR	885-3007-37	MSDOS	CPM EMULATOR	20.00
DBZ	885-8034-37	MSDOS	DBMS	25.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSDOS	PRINTER PLOTTING UTILITY	25.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHICS	885-3031-37	MSDOS	ENTERTAINMENT	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BACKGROUND PRINT SPOOLER	885-1247-37	CPM	UTILITY	20.00
KEYMAC	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CPM-85	885-1245-37	CPM	UTILITY	20.00
MAPLE	885-8023-37	CPM	COMMUNICATION	35.00
MATHFLASH	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PARTY	885-8042-37	MSDOS	ENTERTAINMENT	20.00
SCICALC	885-8028-37	MSDOS	UTILITY	20.00
SKYVIEWS	885-3015-37	MSDOS	ASTRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PACKAGE	885-3018-37	MSDOS	VARIOUS SPREADSHEETS	25.00
TREE-ID	885-3036-37	MSDOS	TREE IDENTIFIER	20.00
USEFUL PROGRAMS I	885-3022-37	MSDOS	UTILITIES	30.00
UTILITIES	885-3008-37	MSDOS	UTILITY	20.00
ZBASIC DUNGEONS & DRAGONS	885-3009-37	MSDOS	GAME	20.00
ZBASIC GRAPHIC GAMES	885-3004-37	MSDOS	GAMES	20.00
ZBASIC GAMES	885-3011-37	MSDOS	GAMES	20.00
ZPC II	885-3037-37	MSDOS	PC EMULATOR	60.00
ZPC UPGRADE DISK	885-3042-37	MSDOS	UTILITY	20.00
H/Z-100 and PC Compatibles				
ADVENTURE	885-3016	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048	MSDOS	UTILITY	20.00
CXREF	885-3051	MSDOS	UTILITY	17.00
DEBUG SUPPORT UTILITIES	885-3038	MSDOS	UTILITY	20.00
DPATH	885-8039	MSDOS	UTILITY	20.00
HADES	885-3040	MSDOS	UTILITY	40.00
HELP	885-8040	MSDOS	CAI	25.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
HUG BACKGROUND PRINT SPOOLER	885-3029	MSDOS	UTILITY	20.00
HUG EDITOR	885-3012	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THRU 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
HUGPBBS SOURCE LISTING	885-3028	MSDOS	COMMUNICATION	60.00
HUGPBBS	885-3027	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAGAZINE DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS's PC & Z100 UTILITIES	885-3052	MSDOS	UTILITY	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
SCREEN DUMP	885-3043	MSDOS	UTILITY	30.00
UTILITIES II	885-3014	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047	MSDOS	UTILITY	20.00
PC Compatibles				
ACCOUNTING SYSTEM	885-8049	MSDOS	BUSINESS	20.00
CARDCAT	885-6006	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOTTING UTILITY	25.00
GRADE	885-8037	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001	MSDOS	UTILITY	20.00
LAPTOP UTILITIES	885-6014	MSDOS	UTILITY	20.00
PS's PC UTILITIES	885-6011	MSDOS	UTILITIES	20.00
POWERING UP	885-4604	N/A	GUIDE TO USING PCS	12.00
SCREEN SAVER PLUS	885-6009	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00

The following HUG Price List contains a list of all products in the HUG Software Catalog and Software Catalog Update #1. For a detailed abstract of these products, refer to the HUG Software Catalog, Software Catalog Update #1, or previous issues of REMark.

Magazines everywhere, and no way to reference the wealth of information they hold? Not anymore! Now there's ***MAGBASE***; a database designed specifically for referencing magazine articles. Don't let those one-hundred-and-some back issues of REMark, or C Users Journal, or Veternary Medicine, (or any magazine) gather dust, use ***MAGBASE***, and *find* that article you read two years ago! ***MAGBASE*** is available for ***MSDOS HUG P/N 885-3050*** or ***CP/M (P/N 885-1249-[27]***.

LAPTOP OWNERS . . . don't feel left out! All of HUG's MSDOS software is available on 3-1/2" micro-floppies too! When ordering, just add a "-80" to the 7-digit HUG part number. For the standard 5-1/4" floppy, just add a "-37".

Make the no-hassle connection with your modem today! ***HUGMCP*** doesn't give you long menus to sift through like some modem packages do. With *HUGMCP*, YOU'RE always in control, not the software. Order ***HUG P/N 885-3033-37*** today, and see if it isn't the easiest-to-use modem software available. They say it's so easy to use, they didn't even need to look at the manual. "It's the only modem software that I use, and I'm in charge of the HUG bulletin board!" says Jim Buszkiewicz. *HUGMCP* runs on ANY Heath/Zenith computer that's capable of running MS-DOS!

ORDERING INFORMATION

For VISA and MasterCard phone orders, telephone the Heath Users' Group directly at (616) 982-3463. Have the part number(s), descriptions, and quantity ready for quick processing. By mail, send your order, plus 10% postage and handling ($1.00 minimum charge, up to a maximum of $5.00) to: Heath Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. VISA and MasterCard require minimum $10.00 order. No C.O.D.s accepted.

Questions regarding your subscription? Call Margaret Bacon at (616) 982-3463.

Heath Computer Kits

Major Price Reductions!

For pricing call Heath Computer Sales TOLL FREE 1-800-253-0570

At Heath Company we're dedicated to making your computer purchase the easiest ever. That's why we've established a team of expert computer sales consultants to help you choose equipment to meet your particular needs — from education and business to computer-aided design and desktop publishing. And, if you decide you want a complete work station, we'll give you a special price that's sure to fit your budget.

Buying a computer has never been easier

We've made buying equipment a simple procedure. Just give our expert computer sales consultants a call at **1-800-253-0570**. They'll guide you through the process of choosing equipment that meets your exact needs. Or, if you have questions regarding other Heath computer products, give them a call. Our consultants are available to assist you Monday through Friday from 8:00 a.m. to 4:30 p.m. Eastern Standard Time.

Support, even after the purchase

We've been in business for over 40 years and we'll be here tomorrow to stand by our products. This means the help doesn't stop after you purchase your equipment. We continue to support you with our famous technical assistance and service. And, we include a 1-year limited warranty with all Heathkit products.

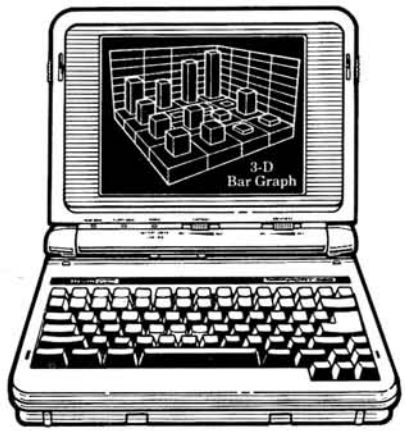
Compatibility that's unsurpassed

Not every PC is completely "IBM compatible," a discovery you could make when you're trying to run a favorite program. Heath computers are tested to ensure full compatibility with more than 150 widely used hardware and software products. Plus, with enough optional memory added, our computers can run under Xenix, Unix, or OS/2 in addition to MS-DOS. No one surpasses Heath's compatibility.

Torture-tested drives and controllers

When you purchase a drive with the Heath label, you know it has passed rigorous read/write and environmental tests that guarantee its reliability and data integrity. In fact, we tested and abused almost $2 million worth of drives and controllers last year. Why? Because our goal is to ensure that you, our customer, get only the best drives the market has to offer.

SPECIFICATIONS	***HS-2860***
Microprocessor	*80286*
Clock Speed	*12 MHz*
Memory	*Standard: 1 MB Optional: 2 MB*
EMS Memory	*Yes*
Floppy Disk Drive	*1.4 MB 3.5"*
Display	*Backlit Supertwist LCD, 640×400*
Standard Ports	*1 Parallel, 1 Serial, 640×400 CGA Video, External Drive*
Keyboard	*Full Size, 79 Keys*
Coprocessor Slot	*Yes*
Clock/Calendar	*Yes*
Operating System	*MS-DOS 3.3+*
Power Supply/Charger	*120/240 VAC, 50/60 Hz*
Dimensions	*3"H×12¼"W×12¼"D*



HS-2860

HS-3860

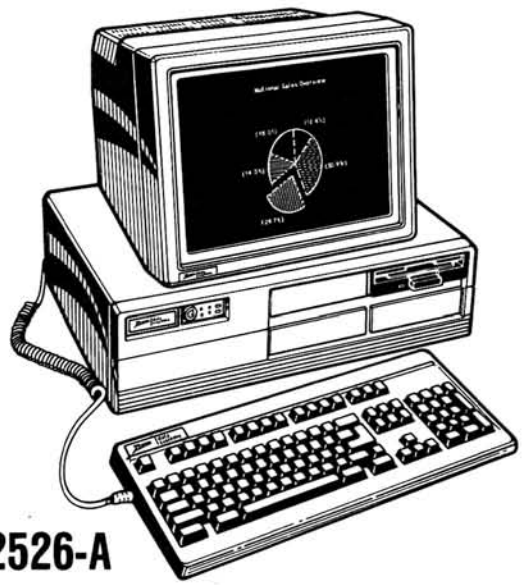
SPECIFICATIONS	***HS-3860***
Microprocessor	*80386*
Clock Speed	*12 MHz*
Memory	*Standard: 2 MB Optional: 3 MB*
EMS Memory	*Yes*
Floppy Disk Drive	*1.4 MB 3.5"*
Hard Disk Drive	*40 MB*
Display	*Page White, 640×400*
Standard Ports	*1 Parallel, 1 Serial, 640×400 CGA Video*
Keyboard	*Full Size, Detachable, 79 Keys*
Coprocessor Slot	*Yes*
Clock/Calendar	*Yes*
Operating System	*MS-DOS 3.3+*
Power Supply/Charger	*120 VAC, 60 Hz*
Dimensions	*4¾"H×13¼"W×14⅞"D*

Call 1-800-253-0570 Today!

Heath Computer Kits

HS-3629

SPECIFICATIONS	*HS-386-C/HS-3629*
Microprocessor	*80386*
Clock Speed	*HS-386-C: 16 MHz HS-3629: 25 MHz*
Memory	*HS-386-C: Standard: 1 MB Optional: 16 MB*
	HS-3629: Standard: 2 MB Optional: 64 MB
EMS Memory	*Yes*
Floppy Disk Drive	*1.4 MB 3.5"*
Expansion Slots	*4*
Video Card	*HS-386-C: Dual Port FTM/EGA*
	HS-3629: HVB-550
Standard Ports	*HS-386-C: 1 Parallel, 1 Serial*
	HS-3629: 1 Parallel, 2 Serial
Keyboard	*101 Keys*
Coprocessor Slot	*Yes*
Clock/Calendar	*Yes*
Operating System	*MS-DOS 3.3+*
Power Supply	*115/230 VAC, 50/60 Hz*
Dimensions	*6"H×21"W×17"D*



HS-2526-A

SPECIFICATIONS	*HS-2526-A*
Microprocessor	*80286*
Clock Speed	*12 MHz*
Memory	*Standard: 1 MB Optional: 6 MB*
EMS Memory	*Yes*
Floppy Disk Drive	*1.4 MB 3.5"*
Expansion Slots	*4*
Video Card	*31 kHz FTM/EGA, CGA, MDA, Hercules*
Standard Ports	*1 Parallel, 2 Serial*
Keyboard	*101 Keys*
Coprocessor Slot	*Yes*
Clock/Calendar	*Yes*
Operating System	*MS-DOS 3.3+*
Power Supply	*115/230 VAC, 50/60 Hz*
Dimensions	*6"H×21"W×17"D*

HS-42/HS-40-A

SPECIFICATIONS	*HS-42/HS-40-A*
Microprocessor	*80286*
Clock Speed	*HS-40-A: 8 MHz HS-42: 12 MHz*
Memory	*Standard: 1 MB Optional: 6 MB*
EMS Memory	*Yes*
Floppy Disk Drive	*Two 1.4 MB 3.5"*
Expansion Slots	*1.5 (after video installed)*
Video Card	*31 kHz FTM/EGA, CGA, MDA, Hercules*
Standard Ports	*1 Parallel, 2 Serial*
Keyboard	*101 Keys*
Coprocessor Slot	*Yes*
Clock/Calendar	*Yes*
Operating System	*MS-DOS 3.3+*
Power Supply	*115/230 VAC, 50/60 Hz*
Dimensions	*4"H×14"W×15"D*

Call 1-800-253-0570 Today!

*** ZENITH PC COMPUTER UPGRADES ***

SmartWatch from FBE Research Installs in ROM Socket on CPU Board in Zenith computer series Z-100/138/148/150/160. This clock/calendar contains a ten year battery and keeps your computer informed of both time and date at each boot-up. Instructions and software included. . . . . $ 38.00

Z-150 Series Hard Disk Drive Kit Includes new generation High Speed Seagate Drive with Auto Park heads. Each kit is complete with controller card, cables, hardware and instructions to mount the Hard Disk under your two floppy drives in the Z-150 series computers. 31 MEG ST-138/150 Kit $333.00

Z-148 Hard Disk Drive Kit Includes the Hard Disk Drive and controller in the kit above plus the Z-148 Expansion card described below. Each kit includes all cables, hardware and instructions to replace one floppy drive with a high speed low power Hard Disk Drive. 31 MEG ST-138/Z-148 Kit $413.00

Z-148/ST-138 Kit With SmartWatch $443.00

Z-148 Expansion Card adds 2 IBM expansion slots . . . . . $ 79.00
with SMARTWATCH clock/calander. $109.00

INTERNAL MODEM Fully Hayes compatible (software included)
1200/300 baud . . . . . $ 67.00
2400/1200/300 baud $109.00

EXTERNAL MODEM Fully Hayes compatible (software included)
1200/300 baud . . . . . $ 89.00
2400/1200/300 baud $128.00

VCE 150 Video Eliminator for Z-150
Allows use of EGA or any video card. Required memory chip included. . . . . $ 54.00

Z-150 MEMORY ZP640 PLUS Replacement PAL chip for your Z-150/Z160 computer. Allows the use of 256K RAM chips to incresed total memory to 704K. To complete this kit order 18 each 256x1 RAM chips. . . . . $ 18.00

*** IBM COMPATIBLE SOFTWARE ***

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-5063-30	Microsoft Windows	$ 99.00	$ 14.00
NU-413	Norton Utilities Adv.	$150.00	$ 99.00
WP-528	WORDPERFECT 5.0	$495.00	$269.00

*** Z-100 SERIES COMPUTER UPGRADES ***

High Density 1.2 Meg Drives. External floppy drive set-up complete with drive, power supply, case and cable. Ready to connect to your 8" floppy controller Single Drive Unit 217.00 . . . . . Dual Drive Unit $309.00
Bare Drive and Cable for internal mount $127.00

SmartWatch by FBE Research. If you don't have a clock for your Z- 100, get this one. More details under PC UPGRADE listings $38.00

ZMF100A by FBE Research. A modification package which allows 256K chips to be used on the old-style motherboard (part number85-2653-1) to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. Order 27 256x1 RAM chips to complete this kit. . . . $60.00

Gemini Emulator Board. Makes the Z-100 compatible with the IBM PC library of programs. $432.00

UCI EASY PC. IBM PC Emulator. Makes your Z-100 IBM Software Compatible. Full 8 MEG operation, color graphics and audio compatible. . . $477.00

UCI EASY-I/O S-100 board that provides IBM PC communications port compatibility with your EasyPC. EasyI/O-1, One Serial Port $91.00. EasyI/O-2,Two Serial Ports, One Game Port, Clock- Calendar $127.00

UCI Memory Upgrade Pal Chip Set For the Z-100's with the newer motherboard part number 181-4918 or greater. This chip set allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K RAM chips (sold separately) a total memory of 768K is obtained. PAL Chip Set . $64.00

UCI Memory Upgrade Card The board has sockets for up to 2 MEG of RAM (72 chips). Board with no RAM installed ...$288.00. Add $35.00 for EasyDrive RAM Drive Software if desired. Either 64K or 256K RAM chips may be used, see prices under MEMORY CHIPS listing.

UCI EasyWin Winchester Drive System at a reasonable price. Complete Hard Disk System for mounting inside your Z-100. System includes S-100 bus board, matched XT hard disk controller, EasyWin software,manual and Misc installation hardware. Order a hard disk (ST-125 or ST-138 recommended) under the SEAGATE HARD DISK DRIVE ONLY listing to complete the kit. $288.00

CDR Z-100 Speed Module Run your Z-100 Computer at 7.5MHz. Installs easily with no soldering. Externally switchable between Speed and Normal mode. Payload $44.00

*** Z-100 SERIES SOFTWARE ***

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-463-1	Z-Basic (16 bit)	$175.00	$12.00
MS-253-1	Basic-80 (8-bit)	$175.00	$12.00
CD-463-2	Condor File Manager	$299.00	$12.00
LT-Z100-3	All 3 Listed Above	$649.00	$32.00

NEW LOW PRICES
** SEAGATE HARD DISK DRIVES ***

All hard disks are shipped complete with installation instructions. The 100 series drives are 3.5" drives and come with a 5.25" frame and faceplate for standard mounting. RLL (run length limited) drives require a RLL controller. XT kits include controller card, cables, mounting hardware and instructions. Most AT computers contain a dual controller therefore only a $7.00cable set is required. If a dual AT controller is required add $124.00 for a 1:1interleave controller and cable set.

MODEL	CAPACITY/SPEED	DRIVE ONLY	XT KIT
ST-125	21 MEG, 28 MS	$ 227.00	$ 277.00
ST-225	21 MEG, 65 MS	$ 199.00	$ 249.00
ST-138	31 MEG, 28 MS	$ 279.00	$ 329.00
ST-238R	31 MEG, 65 MS RLL	$ 218.00	$ 273.00
ST-157R	49 MEG, 28 MS RLL	$ 306.00	$ 361.00
ST-251-1	42 MEG, 28 MS	$ 329.00	$ 379.00
ST-151	42 MEG, 24 MS	$ 379.00	$ 429.00
ST-4096	80 MEG, 28 MS	$ 554.00	$ 604.00
ST-4144R	122 MEG, 28 MS	$ 614.00	$ 673.00

*** FLOPPY DISK DRIVES ***

MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	$ 79.00
MITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	$ 92.00
MITSUBISHI M-353	3.5" in 5.25" frame 720K	$ 94.00
MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	$105.00

M-355 runs on AT compatible or special controller only.

* PAYLOAD CUSTOM ASSEMBLED COMPUTERS *

We assemble 8088 XT, 80286 AT or 80386 IBM compatible computers to your specifications. Please write or call for a work-up sheet showing items available and prices.

*** VIDEO MONITORS ***

ZCM-1490	ZENITH Color Flat Screen VGA	$679.00
MA2565	SAMSUNG Amber TTL 720x350	$ 89.00
CW4644	SAMSUNG Color RGB 640x200	$274.00
CM4531	SAMSUNG Color EGA 640 x350	$389.00
CN4551	SAMSUNG Multi-sync VGA 800x560	$489.00

*** VIDEO CARDS ***

EGA350 PARADISE AUTOSWITCH 640x350	$ 164.00
EGA440 PARADISE AUTOSWITCH 640x480	$ 184.00
VGAPLUS PARADISE 800x600	$ 254.00
VGAPLUS16 PARADISE 16 BIT 800x600	$ 314.00
AUTOEGA STB SYSTEMS 640x350	$ 149.00
AUTO VGA STB SYSTEMS 800x600	$ 199.00

*** MURATA FAX MACHINES ***

M-1200 FAX, PHONE & COPIER	$597.00
M-1600 SAME AS M-1200 + 30 MEMORY DIALER	$697.00
M-900 20 MEMORY DIALER + SHEET FEEDER	$734.00

*** MEMORY CHIPS, ECT. ***

Memory chips are once again at reasonable prices. The market prices have been changing daily, therefore we are only able to list estimated prices. Please call for the current price before placing your order. We buy in large quantities and work on the smallest of margins in order to bring you great values.

41256	256x1	100 ns	$ 4.30
41256	256x1	120 ns	$ 4.05
41256	256x1	150 ns	$ 3.95
1Mbit	1Mx1	80 ns	$13.50
1Mbit	1Mx1	100 ns	$12.25

DISCOVER
MasterCard
VISA

ORDERS: 1-800-444-3538
FAX: (713) 486-8994

15718 SYLVAN LAKE
HOUSTON, TEXAS
77062

PAYLOAD

Your satisfaction is guaranteed. All hardware carries a 90 day Payload warranty. VISA and MASTERCARD orders welcome with no surcharges. Add $5.00 to all prepaid orders for handling and shipping in Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders to approved accounts. We accept purchase orders from schools, government and approved accounts. Mail or Phone your order for prompt service. Texas residents please add 8.0% state sales tax.

Christmas Gifts, FORMAT, Mace Utilities, HADES II, HUG Laptop Utilities, *Powering Up*, FlipFast Guides, MS-DOS Course, IMAGER, Covox Voice Master Key System

William M. Adney
P.O. Box 531655
Grand Prairie, TX 75053-1655

For those of you who have joined HUG in the last year, it has been my custom for the last several years to offer a few suggestions for gifts that you might what to consider for your favorite computer user or even for yourself. And even though it may be a little early, let me be one of the first to wish you and yours a Merry Christmas or Happy Holidays. For the most part, I have tried to keep these suggestions limited to items under $100, but a few exceed that because of increasing prices. If you need to drop a subtle hint to Santa Claus, you can do so by circling the appropriate item on the list at the end of the article.

Since one of the key objectives of this column is to discuss various technical issues that you need to know about, we'll begin with some basics on disks and bad sectors. Then you can use a couple of the Christmas gift suggestions to "recover" disk space that has been lost because of bad sectors.

Recovering Disk Space

I originally began fooling around with the idea of recovering disk space that was marked as "bad" (i.e., bad sectors) when I got my first hard disk for my Z-100 some years ago. When I wrote the second MS-DOS FlipFast book, I discovered some interesting features that are part of the ROM BIOS on a PC compatible, and I began experimenting with some ideas on my Z-248. Even though I expected a few bad sectors on a brand new hard disk — I forget the exact number — I have a vague recollection that it was on the order of 80,000 or 90,000 "bytes unavailable" as reported during the original FORMAT and CHKDSK. I recently recovered about 60,000 bytes that were marked "bad" during the FORMAT of a cheap 3.5" DS/DD floppy disk (720 KB), and it occurred to me that this would be a good topic to discuss because the procedure is not obvious unless you know something about how FORMAT works.

The FORMAT Program

Everyone knows (I hope) that you must FORMAT a new disk — any disk, floppy or hard — before you can use it. This discussion applies specifically to Zenith MS-DOS 3.3 Plus because other DOS versions (e.g., IBM's PC-DOS) are slightly different. In particular, the Zenith FORMAT program performs different functions, depending on whether you are formatting a floppy disk or a hard disk.

When you FORMAT a floppy disk, the program writes a "skeleton" on the disk that includes the Boot Sector, two File Allocation Tables (FATs), the Disk Directory, and the File or Data Area. According to Microsoft, there are two FATs on any disk because the second is a duplicate (for backup) of the first. In addition to creating this skeleton, FORMAT tests the floppy disk to make sure that each sector is "good." A "bad sector" on a new disk — any disk — is usually the result of some kind of imperfection in the media, and of course, you don't want to even attempt to record data on that particular spot.

For most disk sizes and types, DOS does not usually work with individual sectors; rather, it works with *groups* of sectors called *CLUSTERS*. For example, the 3.5" disk I mentioned earlier has 2 sectors per cluster. Since the sector size is the standard 512 bytes, there are 1,024 bytes in each cluster. The number of sectors per cluster on any disk is called the *CLUSTER FACTOR*. By definition, the cluster factor can be 1, 2, 4, 8, 16, and higher, in some cases, for a hard disk. The cluster factor is important because it defines the actual minimum space used to store a file on a disk, regardless of the size of the file. For example, let's say you have a small AUTOEXEC.BAT file that shows up in the DIR command as 33 bytes. Because the cluster factor on this 720 KB disk is 2, the file actually uses 2 sectors (1,024 bytes) of physical storage space on the disk, even though the file is only 33 bytes. The situation becomes more pronounced as the cluster factor increases. For a hard drive partition with a cluster factor of 8 (a common one), it would take 8 sectors (4,096 bytes) of physical disk space to store this 33-byte AUTOEXEC.BAT file. And yes, this technique is somewhat clumsy and wasteful of disk space, but this is a common space allocation methodology used by most operating systems, including mainframes, for most magnetic media.

When FORMAT creates the skeleton on a disk, it also creates the File Allocation Tables (FATs) which are nothing more than a map of the possible locations where data may be stored on a disk. The possible locations for data are mapped out in units of clusters beginning with CLUSTER NUMBER 2 (by definition, clusters 0 and 1 are reserved). And when FORMAT finds a bad sector during formatting, it actually marks the entire cluster bad in the FAT. Even if this cheap disk I played with only had one bad sector, the entire cluster would be marked as bad, and I would actually "lose" 2 sectors (cluster factor 2) or 1,024 bytes of disk space reported by FORMAT or CHKDSK. Again, the situation becomes worse as the cluster factor increases.

The most important point to understand is that somehow FORMAT can only mark bad clusters, not sectors, in the FATs. And that happens no matter what kind of disk type or size you are formatting. Because of some of the unique features of Zenith MS-DOS, the FORMAT program specifically discriminates between floppy disks and hard disks.

When FORMAT is run on a floppy disk, it tests the disk, and if a bad sector is found, it marks it in both FATs with an FF7

hex. For example, this el-cheapo floppy disk had 86,016 bytes marked as bad by the FORMAT program. Since I know that this disk format has a cluster factor of 2, I can divide that number by 1,024 (512 x 2) to find that 84 clusters are marked bad in the FATs.

There is a little more to it than that however. For those interested in the hairy technical details, the standard disk I/O interrupt (INT 13H) is used to format a disk. For a floppy disk, Function 5 is used to format a track. For a hard drive, Function 7 is used to format a track. The interesting point here is that Function 6 is used to Flag a bad TRACK, not just a cluster. That is, *ALL* clusters on a track are flagged as bad which means that you will not lose just a cluster, but an entire track's worth of clusters.

By now you may be wondering what all this has to do with recovering the 60,000 bytes on the floppy I mentioned at the beginning. First, I ran the usual FORMAT program to initialize the disk. Then, I used HUG's HADES II to physically edit both FATs and reset all of the FF7 hex values to zeros. Then I used the Mace Utilities REMEDY program (10 passes) to test and mark the "real" bad clusters on the floppy. I did the same thing with each of the partitions on my Seagate ST-4096 80 MB hard drive and was able to recover some previously marked bad clusters on it too.

I have purposely not gone into a lot of the details on exactly what I did because if you don't understand how to do it from the description above, you should not do it anyway. By the way, the Mace REMEDY program performs essentially the same function as the Zenith MS-DOS DETECT program on a hard drive. Both perform a non-destructive test on a hard drive that checks for bad sectors. In both programs, a bad sector is defined as any sector that DOS cannot read the *FIRST* time (i.e., one try). The DETECT program only records the bad sector information in the Bad Sector Table, and the problem cluster is not actually indicated in the FATs until FORMAT is run on that partition again. The REMEDY program actually marks the bad cluster in the FATs during the process AND attempts to move the file to a good cluster. In other words, REMEDY updates the FATs so that DOS will not attempt to record any data in that bad cluster after the program is run.

For my money, I think this is a better approach because the FATs are updated as "new" bad sectors are found. Bad sectors can occur from time to time after a lot of use because of heat or slight media imperfections that were not detected during the original low-level format with the Zenith PREP program. Because many bad sectors are usually found after the system has been operating for some time, I recommend that you always run any diagnostic program, like DETECT or REMEDY, after the system has been turned on for at least three hours. That should allow plenty of time for the entire system, including the hard drive, to warm up to the "normal" operating temperature.

For those of you not interested in the technical intricacies of recovering disk space, the Mace REMEDY program is still a great idea. It allows you to test any disk — floppy or hard — to help avoid those "Abort, Retry, Rats!" error messages when DOS is unable to read or write something on a disk. In addition, the Mace Utilities is one of those indispensable software packages that all computer users should have. Although the well-known Norton Utilities has some of the same features, Mace has done some things longer than Norton, and that experience gives me a little more confidence in the Mace Utilities.

Mace Utilities

Apparently, there is more than one opinion as to how good the Mace Utilities really are because the package has been purchased by Fifth Generation Systems, developers of the popular FastBack hard drive backup and recovery software. Sales and technical support for Mace Utilities are now provided by Fifth Generation as listed at the end of this article. Many of you have written to me about how good the FastBack software is, and from your letters, I understand that it works on every Heath and Zenith system.

With the release of Mace version 5, there are actually two packages of Mace Utilities available. The first, called Mace 5, contains the standard utilities that have been enhanced from previous versions. The second, called Mace Gold, contains some extra utilities that are particularly oriented toward data recovery. As I've said before, Mace does not have the flash of some other kinds of utility software, but all programs are solid, reliable performers, and I use them on all my systems, including the Z-248, SupersPort 286, and Z-386/16. These latest packages work just fine with Zenith MS-DOS 3.3 Plus.

Mace 5 includes the almost-required utilities for just about any computer user, especially if you have a hard drive. The REMEDY program helps diagnose and mark bad sectors on any disk — floppy or hard — as I mentioned earlier. The UnFragment program performs the "disk optimization" function by moving files around so they are not fragmented, thus helping keep the performance of a hard disk system up to snuff. The Undelete program recovers deleted files. And the UnFormat is for those who still don't use a current version of Zenith MS-DOS and need to recover files from a disk that has been formatted. Current versions of Zenith MS-DOS include a "format protection" feature that requires an additional command (in DSKSETUP) to remove the format-protection flag before a hard drive partition can be formatted. From what I have seen, current PC-DOS versions still persist in attempting to format the default drive, which is why I still recommend entering the drive letter on any FORMAT command line, including Zenith MS-DOS. If you get into the habit of entering the drive letter, it is more difficult to make a mistake, no matter what version of DOS you are using. Of particular note to Zenith and Heath users, Mace 5 also includes the SYSTAT program which recognizes Zenith computers in its report on system statistics. Mace 5 has 17 useful programs, including MUSE (Mace Utilities Sector Editor) that help you perform nearly all of the data recovery features you would expect in a utilities package like this.

The second utilities package, called Mace Gold, includes all of the software in Mace 5 plus some additional utilities that focus on data recovery and protection. For those who use dBase and occasionally end up clobbering files for one reason or another, there is a dbFix utility that can help recover a data base. A unique utility, called TextFix, helps recover word processing files that have been corrupted. For those who don't want to spend the money on an Uninterruptible Power Supply (UPS) system, the Power-Out Protection (POP) program helps protect against power glitches. POP is a memory-resident program that requires a hard drive because the program uses up to 750 KB of disk space to automatically record an image of the computer's memory at preset intervals. My testing indicates that this program works fine on Zenith and Heath systems, but I still am *NOT* a fan of a lot of memory-resident programs — there is too much potential for conflict. And of course, no data recovery package would be complete without some kind of hard disk backup/restore utilities, and Mace Gold also includes Mace Backup, Mace Verify (to verify the backup), and Mace Restore.

All of these programs work fine on my Zenith computers, with the exception that the Mace Restore (MR) program drives my SupersPort 286 LCD screen nuts with crazy characters during some restores. The actual integrity of the restored files is not affected, but it is disconcerting to see all kinds of weird characters on the screen during a restore. The backup program (MB) has a nearly identical screen display, but oddly enough, it does not have any problem with the LCD display. In any case, both Mace 5 and Mace Gold are highly recommended software utilities that can help you save or recover valuable data.

HUG Software

One of the best HUG features is a va-

riety of incredibly useful and inexpensive software. Most of the HUG software is designed for nearly all of the Zenith and Heath computer systems, but there are some exceptions. Since many new HUG members are not quite sure what to order from the HUG Price List, it is important to know that all of the current Zenith and Heath computers are PC compatible. Any software listed in the HUG Price List under the "H/Z-100 and PC Compatibles" and "PC Compatibles" headings will run fine on any Zenith and Heath PC compatible computer, including laptops. Software listed under the other headings are hardware-specific to older (and discontinued) computer models, such as the H-8, H/Z-89 and H/Z-90, and the H/Z-100, and will *NOT* run on a PC compatible. All other computers, including all laptops and the orphaned eaZy PC, are PC compatible. Since Zenith changed model numbers, this has been especially confusing to new members, and perhaps the easiest way to figure out what you have is look at the front page of the Owner's Manual. PC compatible computers include all computers with the following series numbers: Z-130, Z-140, Z-150, Z-160, Z-170, Z-180, Z-200, and Z-300. In addition, all computers with an 80286 or 80386 processor (check the specifications page in the Owner's Manual) are PC compatible for HUG software purposes. Virtually all PC compatible HUG software will also run on other Zenith compatible systems, including IBM and Compaq.

If you are using a laptop computer, one of the best values around is the HUG Laptop Utilities. Even though there are new ROMs for the SupersPort that help fix the problem of the disappearing cursor that I mentioned last summer, the CURSOR program allows you to change the cursor size and blink rate to make it easier to see. The REVSCRN program reverses the image so that a graphics program, like GEM, will not look like a photographic negative on a laptop's LCD screen. CAPCON reverses the functions of the left CTRL key and the Caps Lock key so that the CTRL key is to the left of the A key, where it should always be. And the KEYS program changes the state of Num Lock, Caps Lock, and Scroll Lock keys . . . very useful in the AUTOEXEC.BAT file to turn off the Num Lock function so that the keypad can perform its "normal" cursor functions. These are just some of my favorite programs in the HUG Laptop Utilities package, and they work just fine on a desktop too. The May 1989 *REMark* (page 4) contains information about all programs included in this package. Highly recommended.

HADES II — the HUG Absolute Disk Editing System — is the enhanced version of the original HADES, which is still the best disk editor I've seen yet. This new version of HADES works just fine with Zenith MS-DOS 3.3 Plus and is required because of internal DOS changes. HADES is *much* easier to learn and use than all of the other disk editors I've seen, including Norton, Mace Utilities' MUSE, and the General Disk Utilities (GDU) program included with Zenith MS-DOS 3.3 Plus. Like the original program, HADES II does require some technical knowledge about disk formats, and it also works on the Z-100. Owners of the original HADES program can upgrade to the new version by returning the original distribution disk to HUG and including a modest $15 upgrade fee as listed at the end of this article. HADES II is the best in its class and is highly recommended.

The *Powering Up* book that I wrote last year contains all 15 of the articles published in *REMark*. This book was especially written for beginners and discusses various things that you need to know about computing and your computer. Some chapters include an introduction to various DOS commands that you must know. Other chapters are devoted to describing, in non-technical terms, how some of the computer hardware works and what you must know about it, especially if you want to upgrade or change it. And although the book includes specific information about Zenith and Heath computers, most articles also apply to just about any compatible. Additional information on both HADES II and *Powering Up* can be found in the May 1989 *REMark* (page 4).

The FlipFast Guides

As most of you know, I have written a few other books, and one of those is the *FlipFast Guide to MS-DOS*, which is specific to Zenith MS-DOS. This 544-page book includes DOS information for versions up to 3.20. It also includes every DOS command and error message for each one with practical explanations about how to fix whatever is wrong. A considerable number of useful examples is included with each command with explanations as to what each does. Approximately the last half of the book is dedicated to all kinds of technical information about DOS, such as interrupts (INT), function calls, disk formats, etc. MS-DOS commands for both Z-100 and PC compatible systems are included in the book.

The *FlipFast Guide to GW-BASIC* was written by Bill Barden, and I did some technical editing on that book to be sure it was "compatible" with the Zenith versions up to version 2. This 408-page book includes all kinds of handy information about how to use BASIC commands, statements, operators, in addition to how to write better and faster BASIC programs.

Because these books have been discontinued, I was able to get a special deal on the remaining inventory and am passing on the savings to you. The MS-DOS book originally sold for $24.95 and the GW-BASIC book was $21.95; however, I have a limited number of both books available for $9.00 (plus $3.00 shipping) that are distributed only as a set. Each set includes one MS-DOS book and one GW-BASIC book. Sets cannot be "broken" into two of the same book, and quantity discounts are available for more than five sets. This offer is good until March 31, 1990 or whenever I run out of books. If I am out of books when I receive your order, I will return your check or money order to you. Ordering information is listed at the end of this article.

Educational Software

Heath Company has all kinds of educational software ranging from basic electricity and electronics to robotics and computer fundamentals for both hardware and software. A lot of information about today's technology, such as lasers and fiber optics, is available in these courses. I have bought a number of these courses and have found them to be quite good. As a result, I wrote the current MS-DOS course that includes just about everything you ever wanted to know about DOS and then some. This 983-page course also includes 33 experiments that demonstrate how to really use the commands and concepts presented in the text. A program disk is also included with several programs that help demonstrate what a disk directory and File Allocation Table (FAT) is, and the List Directory (LD) program displays an alphabetically sorted list of file (including system and hidden files) information, including file attributes, of ALL files on a disk. LD also displays the cluster factor for any valid DOS disk, including a hard disk, and it can be used long after you have completed the course. If you are interested in learning about the details of how to really use DOS and its commands, I think you will find this course is the most complete of its kind.

Heath Company has a wide variety of courses available, and if you are interested in other subjects, you may want to write to Heath for a free catalog. The address is listed at the end of this article. All Heath products can also be ordered by phone using VISA, MasterCard, American Express or the Heath Revolving Charge. By the way, many of the Heath products and price information that I mention in this column are shown in the current catalog (#218). Products not specifically listed at the end of each article are usually discontinued, such as the Z-100, eaZy PC, and Z-171, even though they may be mentioned in an article.

The Imager

Data backup, especially on a hard disk system, is an important responsibility for a serious computer user, whether that

computer is used primarily for business or a hobby. Even floppy disks can fail after long use, so it's a good idea to keep data backups on a floppy disk system too. Unfortunately, I still find that many users don't take regular backups on a hard drive because it is quite painful and time consuming to sit in front of a computer and change floppies during the backup. You can ease that pain and reduce the time required by getting a tape backup system. The bad news is that most tape backup systems are expensive, and although the cost can easily be justified for a business computer, it is far more difficult to spend a considerable amount of money for a hobby system.

If you have a Video Cassette Recorder (VCR) for your television, you already have half of the system you need to back up a hard drive. Then, all you need is the Imager and its included software to set up your system for regular backups. In addition to its low initial cost, you can also save a considerable amount of money on backup tapes because even the highest quality VCR tapes can be purchased for about half of the usual cassettes required for other systems.

The Imager system consists of a full-size board that fits in an 8-bit slot on a PC compatible computer, a disk of software, connecting cables for the VCR, and of course, a manual. The Imager fits in a standard 8-bit slot, and it works with nearly all Zenith PC compatible computers (except laptops, of course) up to and including the Z-386. The only real requirement is that you must have space with an 8-bit slot for a full-size board which excludes the Z-148 and the eaZy PC systems. I am currently using Zenith MS-DOS 3.3 Plus on my Z-386, although that should not make any difference as far as backups with the Imager are concerned.

With today's high-capacity hard disks, it is important to know what kind of recording capacity is available, and I have included Figure 1 to show the basic capacities for common VHS tapes.

Tape Time	Backup Capacity
30 minutes	26 MB
60 minutes	55 MB
90 minutes	83 MB
120 minutes	110 MB

Figure 1
Imager VHS Tape Backup Capacity

I have found the Imager system to be extremely reliable, and it works great on my desktop systems. In order to have that kind of reliability, there are three important factors that must be considered. First, make absolutely sure that your VCR is in perfect shape before you begin to use it for backup. That includes such things as head alignment and other mechanical adjustments that are part of normal maintenance for these units. Although the Imager works with Beta-format VCRs, I don't recommend it because they seem to be quite fussy about head alignment for some reason, and that, in turn, seems to cause more reports of reliability problems. Remember that the Imager system simply converts the hard drive data into a "video-type signal" and transmits it to the VCR, so any reliability problem will almost certainly be caused by a problem with the VCR hardware or tape quality.

Speaking of tape quality, that is the second factor. It does not pay to use low-quality tape to back up a hard disk. Even though the software supplied with the Imager backs up each block of data in four redundant blocks on tape to help prevent problems, low-quality tape might still cause problems at the worst possible time — during a restore of a critical file. I have found that Fuji tapes and floppy disks are the most reliable for all my systems over the last couple of years. In short, be sure that your VCR is correctly aligned and adjusted, and use high-quality tape for your backups.

The third factor that you need to consider is what class of computer you are going to use the Imager with. This is part of the computer hardware design that includes something called Interrupt Requests, or IRQs, that are included in a future *Powering Up* article. IRQs seem to cause more computer problems because they are kind of obscure. IRQs are directly related to the class of computer you have. The PC/XT-class computer contains an 8088 microprocessor (e.g., Z-150 series), and the AT-class computer contains an 80286 or 80386 microprocessor.

In general, IRQs are used by the computer to communicate with the hardware. Specifically, the standard Imager board is configured to use IRQ2, which works just fine on a PC/XT-class computer. If you are using an AT-class system, like the Z-200 or Z-300 series, you should probably be sure that you request an IRQ configuration of IRQ5 (normally used for LPT2) or IRQ7 (normally used for LPT1). I mentioned IRQ5 first because few systems rarely need two parallel ports, and there is little possibility for an IRQ conflict. Using IRQ7 in most cases will probably not cause a problem unless you want to run the Imager and use LPT1 for a parallel printer at the same time. Yes, that's possible if you want to print reports during the backup process. The whole idea is to be aware of what IRQs your hardware (including a bus mouse) is using so that you can avoid the problem of an IRQ conflict. There have been some reports of problems with Zenith AT-class systems equipped with a numeric coprocessor (e.g., 80287) that have problems with the Imager's use of IRQ2. Neither of my two desktop systems have a coprocessor, and so the use of that IRQ causes no problem.

The software has also been improved and upgraded, and it has nearly all of the features you will need. The manual that I received some time ago (I haven't seen the latest manual, if any) with the Imager needed some improvement on the installation and software commands, although it is generally adequate. In any case, I strongly recommend you read Joe Katz's report on the "Imager Backup" in the March 1988 issue of REMark (page 51) BEFORE you begin installation. Despite some shortcomings in the manual I have, the Imager is highly recommended as an excellent and very cost-effective way to backup your hard disk.

Covox

I suppose that most HUG members like gadgets as much as I do, and I imagine that is why many of us started using computers in the first place. Perhaps you have seen the Covox advertisements in *REMark* for the Voice Master Key System and were struck with the idea that this is a new gadget. Actually, it is far more than just a gadget — it is a very useful tool that I am rapidly becoming addicted to.

What is the Voice Master Key System? It consists of a headset with a microphone, a board that fits in an 8-bit slot inside your computer, and some software. This equipment allows you to actually speak to your computer and tell it what to do. I am working on an article with more details that will be published early next year, but I have been working with the Voice Master Key System for over a month now and I am impressed. When you are working with a word processor, you can tell it to "bold" by just saying the word. With the included software, you can teach your computer to understand words, which in turn, are associated with macros. In WordStar, for example, I have taught my computer that the word "bold" means to enter the "macro" key sequence CTRL-PB (^PB in WordStar lingo). For SuperCalc, I can insert a row by saying: "Slash...insert...row...return". That may sound like science fiction, but you can do this today with virtually any software you are using.

The Voice Master Key System has obvious applications for the handicapped or for anyone who does not have full use of their hands. For the non-handicapped, it can help you do a lot of chores much faster. If you can type it and say it, this hardware can "translate" it to one or more keystrokes. But there is one very useful and interesting side effect that I have found in using the Voice Master Key System.

Although I thought I knew my usual software pretty well, I have found that I know how to use it much faster, even without the Voice Master Key System. Part of the reason for that is undoubtedly that I had to "teach" the software how to

interpret my words into keystrokes — a simple form of programming if you want to get right down to it. But I have found that I can do things faster without the Voice Master Key System too. When I travel with my SupersPort 286 and I use various kinds of software, I find that I can remember the appropriate keystrokes MUCH faster, almost without thinking. I think that must be a result of repeatedly saying the keystrokes when I use it, both for teaching it and for normal use. Perhaps you are familiar with the principle that if you say something repeatedly, you will remember it far longer, especially if you can see, hear, smell, taste or feel the results. Besides, it is absolutely fascinating to see the software (or the computer, if you prefer) actually DO something you have TOLD it to do. It seems to me that the Voice Master Key System would be a highly useful educational tool for helping students learn software.

As I said earlier, I am working on an article that will contain more details about using the Voice Master Key System, but my preliminary testing indicates that it works just fine with any software that I have tested: WordStar, SuperCalc 5, and even MS-DOS. Of course, the program that actually translates words into keystrokes must be memory resident, but that should not generally be a problem unless you have lots of memory-resident programs. As I have said before, it is good practice to keep the number of memory-resident programs to an absolute minimum, both to avoid program conflicts and reduce memory usage to a minimum. At this point, I would certainly recommend the Voice Master Key System as highly useful, and I suspect it will be highly recommended in my final review.

Powering Down

As most of you know by now, I include some information in each article about something specific to Heath and Zenith systems, either hardware or software. It will usually include something specific to a current version of Zenith MS-DOS, now 3.3 Plus, or it may be quite specific to certain hardware configurations. For those of you who have suggested a "Heath/Zenith Forum" or similar type of article or series, that is precisely what this column is intended to be. My objective is to help keep you informed on items of general interest (such as Zenith MS-DOS), as well as some (not all) of the latest technical developments in Heath and Zenith computers, both hardware and software. For example, next time we'll take a look at the General Disk Utilities programs (GDU and GDUTSR) that are supplied with Zenith MS-DOS 3.3 Plus and why it is a waste of memory and time to load GDUTSR as a standard practice. Unfortunately, some information is not quite as timely as most of us would like it to be because I spend considerable time testing software and hardware on my own systems. And of course, some time is required for typesetting, editing, and the layout of articles for *REMark* — about three months. If you have an idea for an article on a specific topic that would be of general interest, be sure and let me know about it because most of the topics in this column are designed to answer specific questions that HUG members have. More on that next month.

For help in solving specific computer problems, be sure to include the exact model number of your system (from the back of the unit) or series number from the first page of the Owner's Manual, the ROM version you are using (use CTRL-ALT-INS to find it), the DOS version you are using (including both version and BIOS numbers from the VER command), and a list of ALL hardware add-ons (including brand and model number) installed in your computer. The list of hardware add-ons should specifically include memory capacity (either added to an existing board or on any add-on board), all other internal add-on boards (e.g., modems, bus mouse or video cards), the brand and model of the CRT monitor you have, and the brand and model of the printer with the type of interface (i.e., serial or parallel) you are using. Also be sure to include a listing of the contents of the AUTOEXEC.BAT and CONFIG.SYS files unless you have thoroughly checked them out for potential problems (e.g. TSR conflicts). If the problem involves any application software, be sure to include the name and version number of the program you are running when the problem appears.

If you have any questions about anything in this column, or about Heath/Zenith systems in general, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, suggestion or comment.

Products Discussed

HUG Software

HADES II (885-3040) $40.00
Upgrade for original owners 15.00
Powering Up (885-4604) 12.00
HUG Laptop Utilities (885-6014) ... 20.00
Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (HUG Software only)

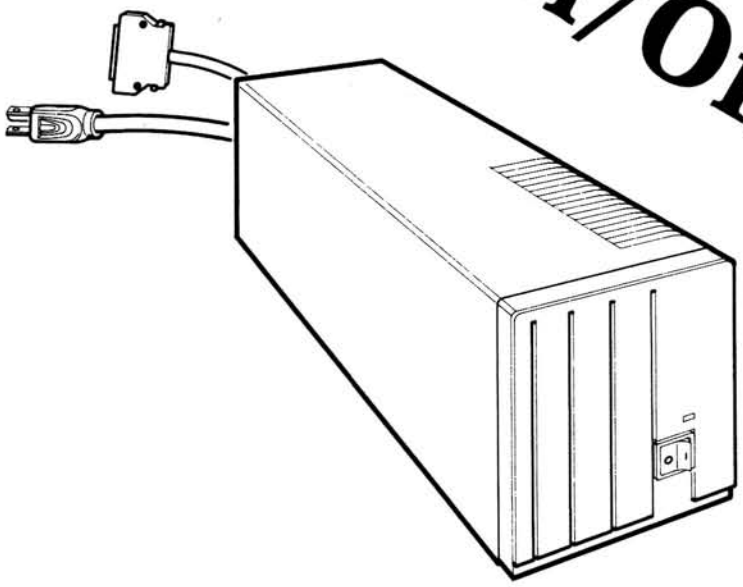
Hardware

Imager $295.00
HUG Members Only 195.00
The Light Pen Company
Box 45255
Los Angeles, CA 90045-0255
(800) 634-1967

Cont'd. on Page 15

The Complete Travel/Office System

Pat Swayne
HUG Software Engineer



A Look at the ZA-3034-EB Expansion Box

If you have a SupersPort 286 or TurbosPort 386 portable computer, you can create a versatile system that can be used in the office or on the road by adding the ZA-3034-EB Expansion Box. I recently had the opportunity to try one out with a TurbosPort 386 and an HVB-550 video card.

The ZA-3034-EB box contains a power supply, a backplane board with 4 XT-compatible slots, and adapter card. The adapter card plugs into one of the slots, leaving 3 for expansion. The adapter apparently converts the computer expansion bus to an XT-compatible bus, and also provides feedback to the computer to indicate that the expansion box is connected and working. One end of the supplied cable connects to a socket on the adapter card, and the other end plugs into the back of your SupersPort or TurbosPort.

The expansion box supports XT-compatible cards (8-bit data path) only, but the backplane board is laid out to accept AT-type card sockets. Perhaps this indicates that a future version will support AT-compatible cards. But that will not happen for the current models of portable computers, because their expansion busses only have an 8-bit data path.

To set up the ZA-3034-EB for operation, all you have to do is remove the cover, install your expansion card(s), replace the cover, and connect the cable to it and your portable computer. When you power up the system, you must turn on the expansion box first and then your computer. A green light on the front panel of the box blinks when you first turn it on, and then glows steadily when the computer comes on and sends a recognition signal to the box. It would probably be OK to plug both the computer and the expansion box into a power strip and use the switch on the strip to turn everything on, though I did not try that.

Before you can use the expansion box, you must run Setup on your computer. A nice feature of the SupersPort 286 and TurbosPort 386 computers is that there are two separate Setup screens, one for use without the expansion box, and one for use with it. Not only can you configure things likely to change, such as your video display, but you can configure just about everything else. For example, while you are traveling, you may want to disable your serial and parallel ports, and set your hard disk to power down after a while (to save battery power), but while you are in your office, you may need your ports, and it is inconvenient to have the hard disk power down. You can configure each of those things separately for use with or without the expansion box. Whether the computer is connected to the box or not, it will automatically use the correct Setup.

Although the manual that comes with the expansion box states that you cannot use a floppy disk controller in it, you probably can use the CompatiCard controller that was reviewed in the April 1989 issue of REMark (page 35). With that controller, you would be able to add 5.25-inch floppy drives (both the 360k type and the 1.2m type) to your system.

The ZA-3034-EB and HVB-550 that I tried out with the TurbosPort worked great (with an FTM monitor connected). In spite of the 8-bit data bus, video repsonse was quite adequate. Since the keyboard on the TurbosPort can be replaced with a standard (ZKB-2) keyboard, you can create a system that has the "feel" of a fullblown desktop system. But when you need to travel, all you have to do is unplug the expansion box, plug in your portable keyboard, and go! *

Graphics Printer or Epson FX Part 5

Control Files and Interference

John A. Day
5 rue Sauer
77500 Chelles, FRANCE

Disk Control Files

You can easily design your own menu-driven program to set print modes to any combination you want. But for settings you use often, it's quicker to have a disk file which you can copy to the printer from within a batch job, for example, to set the printer to 12 cpi for a compile or to 17cpi for an assembly, with appropriate tabs, or for printing a wide spreadsheet.

First of all, you need the keystrokes for some of the codes. Characters from 0 to 31 can mean straight values or control codes. 1, for example can be interpreted as ☺ if "Ctrl" is not depressed, or ^A if "Ctrl" is pressed as well. Unfortunately, many programs including DOS and EDLIN don't bother to check the "Ctrl" key; anything less than 32 will be handled as though it were a control character. This means that whether you use the "Ctrl" sequence or whether you hold "Alt" down while typing the ASCII value on the numeric keypad, you will get the same result. The correspondence between values is 0 = ^@, 1 thru 26 is ^A thru ^Z, then 27 = ^[, 28 = ^\, 29 = ^], 30 = ^^, and 31 = ^_ (^A means press "Ctrl", then "A" at the same time). So if you use a text editor to set up a printer control sequence, you are quite likely to get the "Ctrl" equivalents to what you type in. And, as you will quickly find out, several values have special meanings for DOS, and can't be keyed in anyway. Luckily, we have DEBUG. Key in your control file as best you can, using characters which are easy to recognize instead of the ones which jam; for instance, to set the printer to 12 cpi with tabs every 8 columns from 9 to 81, you want to end up with *ESC "M" ESC "D" 9 17 25 33 41 49 57 65 73 81 00*. For ESC - no way. This corresponds to the line cancel function with DOS and EDLIN, so all you'll get is \ and a new line. Well, for the moment you can key "E" for ESC, and "Z" for 0. We'll use the DOS COPY function to create a file called elite.ctl; using "Alt" and the numeric keypad for the numbers, you will get:

```
copy con elite.ctl
EMED      ^Q^Y!)19AIQZ
^Z
```

You end the lines with "Return", of course: <—. ^Z corresponds to 'F6', and returns you to command mode. The gap in the line is because 9 is the ASCII tab character; it's there all right, but you can't see it. You can check with the character tables in your printer handbook that ! thru Q are the right values for 33 thru 81. 17 and 25 print as ^Q and ^Y, but are only one character each in the file. Now, to proceed with DEBUG you need to know the hexadecimal values. If you're not sure how to convert 27 to hexadecimal, use BASIC:

```
PRINT HEX$(27)
1B
```

gives you 1B as the corresponding hex code. Now, you can use DEBUG to replace E (45 hex) with ESC (1B hex), and Z (5A hex) with 00. There will also be a line feed to take out; the "Return" at the end of your line inserted two printer codes, CR and LF — 0D 0A in hex — and you only want to keep the CR. This just means shortening the line by one character. Start DEBUG on your file:

```
debug elite.ctl
```

and, when you have the "-" prompt, get the length of the file, which is in CX:

```
-r cx<--
CX 0011
:10<--
```

This says that your file is 11 (hex) characters long, that is, 17 characters. You reply "10" (hex for 16) to shorten it from 17 to 16. You can look at its contents with

```
-d cs:100 I10 <--  (or L10 <--: length
"10" hex)
135B:0100  45 4D 45 44 09 11 19 21-29
    31 39 41 49 51 5A 0D
       EMED...!)19AIQZ.
```

The segment address, here 135B, depends on your system; within the segment, your binary file will always be at :0100. The 16 hex values are the current contents of your file. The text value at the right of the line gives those characters that can be printed, others are replaced by periods. Now, to change the two '45's to '1B's and the '5A' to '00', you use the 'e', or 'enter' command. DEBUG will give you the contents of your file, one byte at a time. To change a value, type the new value in hexadecimal. To skip to the next value, type a space. To end changes, press 'Return':

```
-e cs:100<--
135B:0100  45.1B 4D.   45.1B 44.   09.   11.
           19.   21.   29.   31.   39.   41.
           49.   51.   5A.00<--
```

As you can see, you just correct the three values you couldn't key in directly, and space over the others. Your file is now correct.

-w<-- writes your file back to disk, with the message

```
Writing 0010 bytes for the new length
                                    of 16;
```

-q<-- winds up the DEBUG session.

Now, to switch your printer to 12 cpi, you just have to give the instruction copy elite.ctl prn. You can set up as many control files as you need this way; just type the values in as best you can, tidy up with DEBUG, and shorten the file by one character. Don't forget, though, that each one of these files takes up a full disk allocation group, which is at least 1 Kbyte. Although you can create the complete file just by using DEBUG, I never do — I would have to look up the manual to find out how the N instruction works, whereas I can modify an existing file without the book. File utility programs will let you modify your codes in full-screen mode, but shortening

the length by one character probably won't be easy; DEBUG is good on this.

We've already seen examples of mixing control codes and text in BASIC. dBASE accepts the same method; for programs that must be used with two or three different printers, it is worthwhile to set up a memory file (.mem) for each printer with the different codes equated to program variables. With this approach, switching printers is just a question of switching memory files.

Once you get down to programming your printer, you'll get all sorts of surprises. With MS-DOS, using copy or EDLIN, you can't key a whole series of values which correspond to systems functions:

0	doesn't key	13	is Return/Enter
3	is ^C, break	16	is ^P, print and display
6	is ^F, doesn't key	19	is ^S, stop scroll
8	is backspace	127	is Delete
10	is line feed		

Also, 26 and ^Z, is the end-of-file marker. Special care is needed with any file containing this character. Any *copy con filename* operation will terminate with the first ^Z found, whether the /b option is given or not. EDLIN will handle ^Z, but only if the /b option is used: *edlin filename /b*. And you must use the explicit binary copy *copy filename prn /b* to copy a ^Z to the printer. If your software sends a ^Z to the printer; for example, if you want to set a tab in column 26, it may or may not get through, depending on which system calls are used to access the printer. I have no problems with the BASIC interpreter, but with compiled BASIC I have had to shift a tab to column 25, because DOS refused to transmit the value 26.

MS-DOS is not the only wet blanket, you'll find most of your software ganging up on you to stop you from doing what you want. BASIC is quite pernicious, particularly because it seems so easy to introduce a *CHR$(n)* here and there. First of all, BASIC watches your line widths, and adds an automatic CR-LF sequence after 80 characters without a carriage return. Alas! BASIC doesn't think to look whether you are sending printing characters or not. A couple of underlined words is twelve extra characters on the line, three for each underline on and three for each underline off, and if you're using the full 80 printing characters, the last twelve will be axed and put on a new line. This one is fairly easy to get around: the instruction *WIDTH "LPT1:",255* suppresses all checking, 255 being a code value which switches checking off. After this instruction, be sure you stay within the limits of your printer page, because BASIC won't check what you are doing.

BASIC and the CR code, 13, is much more difficult. BASIC language specifications call for each CR to be followed by an LF, so whenever BASIC sees an ASCII 13, it checks whether it's followed by an ASCII 10 and if it isn't, it slips one in. There is one way I know to get around this: declare your printer as a relative file (!!), and use file writes instead of print writes:

```
L$ = CHR$(10)
OPEN "LPT1:" AS #1
WIDTH# 1,255
PRINT# 1,....
```

This lets you use ASCII 13 where you want, in control sequences for tabs, line spacing and so on. BASIC inserts a carriage return without line feed after each print. You may remember, I mentioned in one of the first articles that operations which require stopping and starting the print head several times on a line are best done as line overprints with no intermediate stopping. You will get this automatically. When you want a line feed, you will have to insert it yourself at the end of the line; this is the L$ variable defined at the beginning of the sequence:

```
PRINT# 1,"Dear Sir,"L$
```

After BASIC, dBASE seems quite easy. *CHR(n)* gives you any character you want, except zero. So, goodbye setting tabs, which need a zero to close the sequence (unless you set all 32 of them). For other sequences, there are a few ways around this limit. You can't use *ESC "!" 0* to restore pica print, but your printer may not be using codes 2, 64 or 128 — in which case you can use one of these instead of 0. *ESC "C" 0* n, for setting forms length in inches, can be replaced by *ESC "C" 128* n. To set superscripts or cancel underscore or enlarged print, you can use 48, the character "0", instead of the value 0: *ESC "S0", ESC "-0", ESC "W0"*:

```
store chr(27) to escape
? "this is " + escape + "S" + chr(1) +
   "underlined" + escape + "S0"
```

Quite often, 128 is accepted instead of 0, since the MSB (most significant bit) is not always used, and 128 in binary is 1000 0000.

If you're using a printer compose table, watch out for fun. Word processors know whether they are handling text or controls, and only use their compose tables for the former. General-purpose programs, like dBASE, where you mix controls and text yourself, don't know which is which and run the lot through the compose routines, as does my TSR program given in Part 1. If you suspect interference from remapping, disable your compose table and see if your control sequences work better. This one is difficult to get around; you may have to add compose entries to get values to map on to the ones you want, but you'll be better off if you can avoid this.

The Parallel Interface

The standard parallel interface is so easy to program that you can handle it in BASIC without a hitch. I even use BASIC to check out interface algorithms before coding them in Assembler. And if you program the port yourself, your program will be less portable, but you will have no problems with interference from systems software.

Each parallel interface occupies three consecutive ports on the pc-compatible computer. You handle these ports with *V% = INP(port%)* to input a value V% from port *port%*, and *OUT port%, V%* to output the value V%. There are 64K ports in the addressing scheme, but only a few are actually used. The BASIC instructions are for byte I/O only , so V% will always be less than 256.

Don't get confused between *PEEK* and *INP*. The first lets you access the 640K of memory directly, and is used with *DEF SEG* which indicates which 64K segment of memory to use. The second gives you direct access to the I/O ports, with no *DEF SEG* or anything.

You will need a *PEEK* to be sure which port is in use for the printer. From here on, I shall use a fair amount of hexadecimal notation: %H40 is the BASIC way of writing 40 hex, which is 96 decimal. And it is also the segment address of the DOS communications zone, which contains the addresses of all printer adapters present at power-on:

```
DEF SEG = %H40
LP1% = PEEK(8) + 256 * PEEK(9)
LP2% = PEEK(10) + 256 * PEEK(11)
```

LP1% now contains the base address — the first port — for LPT1 (or PRN), and if LPT2 is present, *LP2%* contains the second base address. LPT1 is usually at %H378 if you are using a separate graphics video adapter, but could be at &H3BC. The port addresses for LPT3 and LPT4 are at 12/13 and 14/15 if you need them.

First of all, let's look at the status of LPT1. This is always on the second port, %H379 or %H3BD.

```
PST% = INP(LP1%+1) AND &HF8
```

gives you the current status of the printer, including a lot of information you can't get through DOS. The *AND %HF8* is there to strip off the low-order three bits of the status byte, which aren't used. Now,

- if PST% AND &HB8 = &H80, the printer is switched off;
- if PST% AND &H80 = 0 the printer is not ready, and you can see why with:
- if PST% = &H58, it's just busy
- if PST% = &H78, the printer cable is disconnected;
- if PST% AND &H20 <> 0, the printer is out of paper;
- if PST% AND &H50 <> 0, the printer is in service but off-line;
- if it's not one of these four, then there is a hardware or firmware fault condition.

This means that in any BASIC program which uses the printer, you can check before you start whether the printer is plugged in and switched on, and these checks are immediate, not like MS-DOS errors where you have to wait 20 seconds

for a device time-out before the system will tell you something is wrong. The tests for "printer switched off" and "printer cable unplugged" are not in the book, but I have tried them on several different machines and they seem to work.

The other two ports are latched outputs. This means that any value sent to the port with an OUT instruction stays on the port until a new value is sent. The first port is the data byte to send to the printer. The third port handles synchronization. Bit 4 should be 0, to disable interrupts. Bits 3 to 0 feed pins 17, 16, 14 and 1 on the parallel connector, but are inverted except for bit 2; for the others, a 1 bit puts the pin low and vice-versa.

Bit 3/pin 17 is used to take the printer on and off line, and should stay at 1 (low). Bit 2/pin 16 is the printer reset signal; if it goes low (0), the printer does a hardware reset as when you boot. Bit 1/pin 14 should be kept high (0), otherwise the printer may line feed after printing. Bit 0/pin 1 is the "strobe" bit. If this pin flicks low, then back to high, the printer will read the data byte on the other port (if it's not busy). So you keep this bit at 0 until a data byte is ready and the printer is not busy, then flick it to 1 and back again. This is expressed by the binary code 0000 1100 between data transmissions (&H0C), and 0000 1101 to transmit (&H0D).

This is all we need to write a subroutine that will output a byte *Z%* to the printer. The program tests printer status; if anything is wrong, it returns with *PERR% = -1*; otherwise, it just waits. The printer can stay busy a moment, particularly if it's not buffering, but if there is no specific fault indication, then it should come ready after printing a bit. As soon as the printer is ready, the program puts the data byte on the first port, toggles the strobe bit on the third port, and returns. There's no sense in checking status after sending the byte; the microprocessors used in a printer are much slower than an 8088, and will be busy for several cycles storing the byte in the printer's internal buffers.

```
100 PST% = INP(LP1%+1) AND &HF8
110 IF PST% = &H58 THEN 100
120 IF PST% AND &H80 = 0 THEN PERR% = -1:
    RETURN
130 OUT'LP1%, Z%
140 OUT'LP1%+2, &H0D
150 OUT LP1%+2, &H0C
160 PERR% = 0
170 RETURN
```

I use a more sophisticated version of this routine in a BASIC program for graphics which is rather slow. If the printer is busy, the subroutine tries to hold the data byte on a 10-value stack. This means that for a line feed or a carriage return, the main program is not held up and can continue processing. With the standard BIOS access, the main program could be blocked for more than a second at a time just to issue a paper motion command. Whether this is worthwhile or not depends on how you are using your printer. I always have the printer input buffer set as a third character generator zone (download RAM, which we are coming to in the next article). Some printers manage to buffer quite well with what RAM memory is left, and never stay busy. Others (mine, in particular) stay resolutely busy for the whole width of a graphics print line. If you want to try stacking bytes like I do, loop a bit on the status INP function before you decide to save the byte and return. With programs in compiled BASIC, I find the printer stays busy anyway the time it takes to loop eight or ten times on the INP function. I only give up after about thirty loops, when it looks as though the print head has started across the paper. And whether or not you stack, you get *immediate* information on the printer status, and no nonsense from BASIC or MS-DOS on codes they don't want to transmit.

Next time, downloading and graphics, or how to do without the outboard character generators. ✱

Cont'd. from Page 11

Voice Master Key System $149.95
Covox, Inc.
675 Conger Street
Eugene, OR 97402
(503) 342-1271

Software

Flipfast Guides $12.00 (includes shipping)
FlipFast Guide to MS-DOS
FlipFast Guide to GW-BASIC
William M. Adney
P.O. Box 531655
Grand Prairie, TX 75053-1655

Heathkit Catalog (free)
Heath Company
Benton Harbor, MI 49022

MS-DOS Course (EC-1121-A) $59.95
Heath/Zenith Computer Centers
Heath Company
Hilltop Road
St. Joseph, MI 49085
(800) 253-0570
(Heath Catalog orders only)

Mace 5 Utilities $99.00
Mace Gold Utilities 149.00
Fifth Generation Systems, Inc.
11200 Industriplex Boulevard
Baton Rouge, LA 70809
(800) 873-4384 (Orders only) ✱

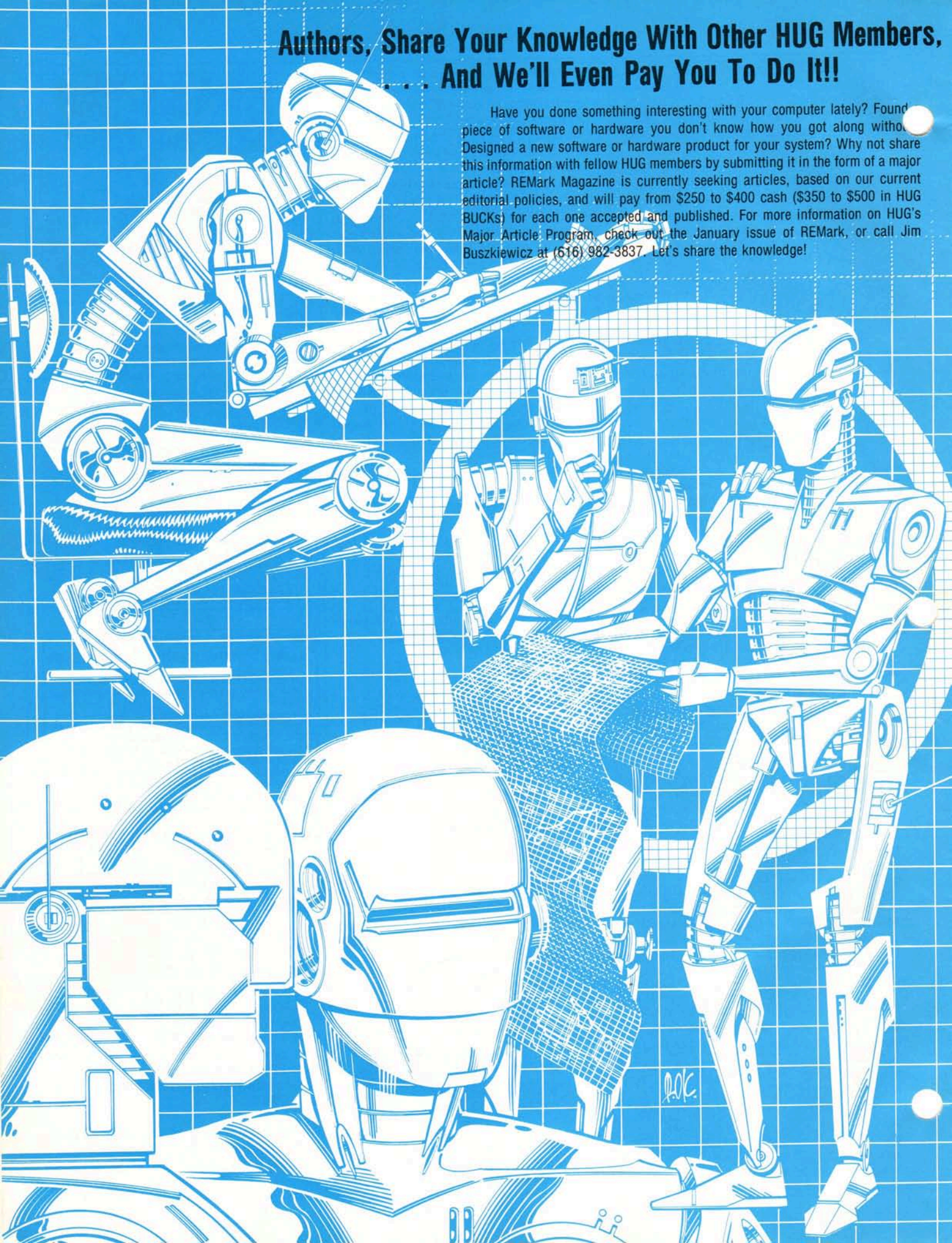
CLASSIFIED ADS

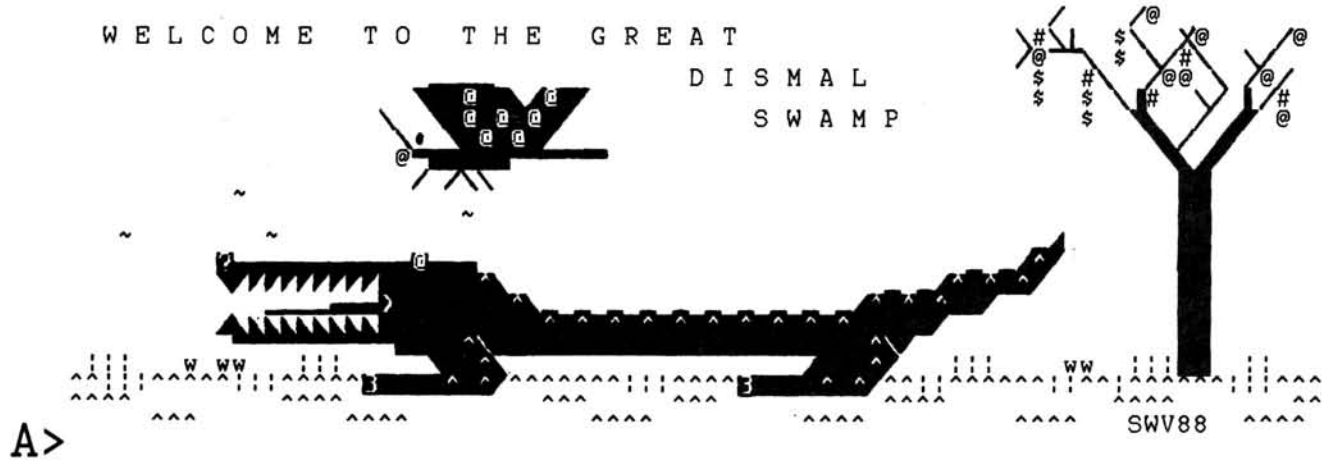
INTERESTED IN YOUR FAMILY TREE? F__TREE is a user-friendly genealogical data base management system for IBM PC compatible DOS 2.+ users. For free details, write MIN DAR Co., Dept. BR, Box 254, Elk River, MN 55330.

H/Z-100 COMPUTER WITH MONITOR AND 20M HARD DISK. Needs work. Software included. First reasonable offer accepted. Phone 1 (800) 553-8907 or (319) 285-4162 in Iowa.

Authors, Share Your Knowledge With Other HUG Members, . . . And We'll Even Pay You To Do It!!

Have you done something interesting with your computer lately? Found a piece of software or hardware you don't know how you got along without? Designed a new software or hardware product for your system? Why not share this information with fellow HUG members by submitting it in the form of a major article? REMark Magazine is currently seeking articles, based on our current editorial policies, and will pay from $250 to $400 cash ($350 to $500 in HUG BUCKs) for each one accepted and published. For more information on HUG's Major Article Program, check out the January issue of REMark, or call Jim Buszkiewicz at (616) 982-3837. Let's share the knowledge!





COLOR PAINT.ASM FOR THE H/Z-100!

Now H/Z-100s have a color version of "PAINT.ASM"! Spice up your games or your program's opening screen with COLOR! Using "HELLO.ASM" from my last article, show a color graphics screen on booting!

Background

The H/Z-100 was state-of-the-art when introduced to programmers. Normally capable of displaying 640 pixels (distinquishable dots of light on the screen) per line on 225 lines per screen, through reprogramming, it could double the scanning lines to gain a high-resolution mode of 640 pixels by 500 lines in the "interlaced mode."

Additionally, unlike many other computers which were character oriented (e.g., H/Z-89), the H/Z-100 did not have a non-graphics mode. It used bit-mapped characters and graphics that would permit programmers to turn on any of 144,000 pixels individually, in any of eight possible colors.

To produce color, separate planes (areas) of video memory contained character information for three main colors: red, green, and blue. Each pixel seen on the screen was composed of three superimposed color pixels, one of each color plane. Depending upon the status of each of the three color pixels, on '1' or off '0', eight different colors were possible:

Color Planes			
Green	**Red**	**Blue**	**Resulting Color**
0	0	0	-Black (no color)
0	0	1	-Blue
0	1	0	-Red
0	1	1	-Magenta
1	0	0	-Green
1	0	1	-Cyan
1	1	0	-Yellow
1	1	1	-White

Monochrome operation required that only the green plane have memory installed. However, if memory were installed in all three color planes, with a monochrome display, eight levels of intensity (brightness) could be produced, which corresponded to the above colors.

Despite the extensive color and graphics capabilities of the H/Z-100, however, there appears to be a tremendous lack of information in print. Going through my stack of "REMark" and "SEXTANT" magazines to the pre-dawn of the H/Z-100, I found only two articles that did more than just touch on the subject:

"Computer Graphics on the H/Z-100" by Randy Meyers of Corvallis, Oregon, does an excellent job describing its graphics capabilities in the May 1984 issue of "REMark". It discusses the processes involved in turning on or off specific pixels and includes an example program, "SETPOINT."

"Interlaced Anyone?" by Frank T. Clark of Zenith Data Systems addresses the H/Z-100's interlaced video mode, what it is, its use, and its inherent problems for programmers. Included is a small routine to turn on or off the interlaced video mode.

The H/Z-100 technical manual provides the most information on the theory of video operation, but little practical assistance to individuals who wish to explore these capabilities. Additionally, the other guys (i.e., IBM) subsequently chose different ways to improve resolution, leaving the H/Z-100's capabilities somewhat untapped and undeveloped by industry. Few individuals chose to vigorously explore these capabilities and, as a result, there are few programs that do the H/Z-100 justice.

Steven W. Vagts
2409 Riddick Road
Elizabeth City, NC 27909

Where is all this taking us? Well, first, I wanted to point out the reason few articles have apparently been written and encourage others to perhaps investigate and write about these untapped capabilities. Second, those programs that I've seen have primarily used bit graphics: draw a line, arc, circle, box, etc., then color it in. I'd found these programs highly sophisticated and complex, much more than I needed.

As with my previous H/Z-89/90 CP/M program, "PAINT.ASM", subsequently renamed to "PAINT89.ASM", I wanted to draw graphics screens or pictures in a relatively simple process. I felt it should be a natural progression from my previous program, using H/Z-89 graphics and remaining somewhat compatible with its predecessor.

The H/Z-100 program only uses the standard H/Z-89 graphics set, in eight colors, then saves them to memory or disk in precisely the same manner as the H/Z-89 version. Another advantage of using the H/Z-100, however, is that the graphics set can be changed to display or represent any graphics symbol you need. We'll discuss this next month.

Preliminaries

Over the next two months we'll discuss and list the entire program. I will also discuss a few other changes I have made to "PAINT89" since our last discussion:

1. A printer tabbing routine to move the printed picture horizontally anywhere on the paper, such as positioning the picture in the lower right quadrant for making greeting cards.
2. A routine to permit the display of the

opening screen of instructions at any time, without destroying the current work.

3. Better printer control — enhanced/expanded line spacing routine, compressed printing (144 CPL), through a second menu screen.

I must remind you that the printer routines are for the Panasonic KX-P1092 Multi-Mode, 9-Pin Dot Matrix Printer. I've found that for use on other printers, extensive modifications must be made. My previous articles in "REMark" discuss these.

The system I'm running is CP/M-85, Version 2.2, on an H-101-20 with 768K RAM and two double-density, double-sided disk drives.

The Program!

For those new at Assembly Language programming, we'll take the process slowly, discussing each step. I use "MAGIC WAND" to type in the program, but any other similar word processor or program editor would work. Begin typing in the program, Listing 1, as I discuss each part.

Chapter 5 of the CP/M-85 documentation discusses system interfacing in depth. Section 2 provides the Operating System Call Conventions — detailed information for performing direct operating system calls from user programs.

What does that do for us? Well, the operating system, CP/M in this case, already has routines to read characters from the keyboard, display characters on the screen, send characters to the printer, read and write from and to disks, etc. Rather than duplicate these in our own programs, these routines are available through these system calls. So those that are to be used in your program need to be listed at the beginning of the program. A routine in our program will take the info from these equates (EQU) and actually access CP/M to perform the appropriate function.

Also at the start, we must tell the program where to reside in memory — originate (ORG) at 100H. We next locate and set our own stack area and save CP/M's stack and flag values.

At this point, previous versions of "PAINT" used data (MSG's) at the end of the program to display an opening screen of instructions, line-by-line. This was a lengthy process and a common problem among programmers because not only the lines of characters had to be adjusted (located on the screen in the proper position), but also formatted (placed in reverse video and/or in graphics). MSG1 on page 59 of the March 1986 issue of "REMark" shows a simple example.

To add reverse video, graphics characters, and now color would require the addition of all the appropriate escape codes, tremendously complicating the task. Well, we'll avoid this entirely by completing our program first, then use it to develop our opening screen — a piece of cake! At the same time, we'll develop a second screen to list a menu for printer options.

The INPUT routine shows an example of the use of a system call. The first line places the value of CONIN (1) into register 'C', then we call BDOS (value of 5, which is the BDOS vector) and the CP/M Operating System does the required function, in this case, waiting for any key input.

We check this input for an ESCape character, or certain control (CTRL) characters and jump to other routines if these are detected. If an ESCape character is detected, we jump to XCAPE and check the next character using the CONSIO system call. Here we are primarily interested in detecting a function key.

Note that function keys F9 thru F12 transmit an ESC-'0' followed by a letter from I thru L. This is given with other symbols and codes in Appendix B of the Z-100 User's Manual. So we must not only test for the ESCape and the zero, but read another character if a zero is detected and check for one of the four letters — the function of the XCAPE2 routine.

Those of you who are more familiar with Appendix B are probably questioning why we don't use the single code for the Function Codes as listed on page B.12. Theoretically, we could, except CP/M-85 apparently doesn't recognize codes over 127 decimal (7F in Hex). The function keys give codes well above this range.

The ERASR routine is unique in that after we check for the appropriate code for the F0 function key, we also capture the next keystroke to determine what to erase. The ERA1 routine checks for the "HOME" key or the arrow keys.

Escape codes captured are converted to the appropriate code we need by RVID thru EALL routines, and the new codes passed to TXSPCL or TXSPC1 for sending to the screen. Some routines require that the default color be set, hence the difference between TXSPCL and TXSPC1.

The GRAFX routine places the graphics symbols of the lowercase letters on the top two lines of the screen for reference.

The COLR routine sets the color mode for subsequent characters or graphics, without changing what was already on the screen. Color on the H/Z-100 is added through an ESC-'m', followed by two numbers indicating the foreground and background color, immediately preceding each character.

The Mark routines, MARKR thru MKD1, are thoroughly discussed in my February 1987 "REMark" article, "Enhanced PAINT.ASM!!!." But things got considerably hairier. As discussed in that article, when we wanted to "MOVE" or "COPY" blocks of reverse video or graphics characters, it complicated those routines tremendously — you had to take the reverse video or graphics code with the block, but also leave it behind if that mode was to continue with characters to the right of the block.

Color codes had to be treated the same way. Keeping track of the number of characters, keeping in mind that color codes had one more number than the other codes, was a real pain. But we managed, the key being the ESCCH and ESCCH2 routines.

MSG50, DUMP, SAVE, and DSPLY routines worked much the same as in the previous "PAINT"s — articles in March and April 1986 issues of "REMark."

The HELP routine, called by function key F9 or the "HELP" key, causes the current work to be saved to memory, then displays the same file as the opening screen, "PAINT100.SC1." When finished, pressing any key will return the current work back to the screen.

The RDMENU routine works in a similar manner to the HELP routine, but calls up a different screen, "PAINT100.SC2", that presents the menu options for printing a screen. This screen would vary tremendously from printer to printer, and those familiar with my previous articles know the impact of this. This is an excellent example of using "PAINT" screens for displaying menus, however.

QUEST checks the keystrokes while using the menu screen until the letter 'Z' is detected, then the menu routine ends and the current work is redisplayed.

PICAP, ELITEP, COMPON, etc., are examples of sending codes to capable printers to set formatting options. Again, each printer will be different in capability and code sets. All these routines require special tailoring.

Of particular interest during this portion of the program, however, is that most of us cannot print color. We, therefore, must capture and eliminate the ESC-'m' and following numbers. This function is left to the SKIP and associated routines and again requires keeping track of number of characters.

This makes a good place to break for this month — plenty to mull over and review. Please don't forget to save your work.

Now for some homework. "HOMEWORK!?" you say? Well, next month we revisit the issue of bit-image graphics on printers — and H/Z-89 owners should take note. We'll simplify the process slightly, saving 2K of space in both the .ASM and .COM files.

Peak your interest? Good. Please reread my article "PAINT.ASM Again?" in the December 1987 issue of "REMark" for next time.

Closing

Upon request, I would be happy to send anyone the source code for "PAINT-89", "PAINT100", and "HELLO" for $8.00. I'll also send the finished products, .ASM & .COM files to all three programs, on disk for $15.00, if you include a preformatted disk with your request. Upgrades to prior customers will be $10.00. I now primarily use H/Z CP/M (H/Z-100) or CDR CP/M (H/Z-89) and double-sided, double-density extended, soft-sectored disks. However, I can still copy to the older 10 sector, single-sided disks. Please include a phone number in case I have problems.

As I've said on numerous occasions, my primary interest is to generate interest in Assembly Language. I'm hoping you will want to experiment and develop your own changes — unless your printer is similar to my Panasonic KX-P1092, you will have to modify the printer routines, as a minimum anyway.

In the process of coding, assembling and modifying these programs, if you have enjoyed the work, sweat, and, yes, some reddening of the eyeballs from the late nights when the gremlins just don't seem to cooperate, and finally the swelling of the chest with pride when the program finally works properly, then I have completed my goal.

Please feel free to distribute these programs to your friends as you see fit.

I'm interested in any improvements readers feel may be appropriate and I'd be happy to address questions if self-addressed, stamped envelopes are included.

Listing 1
PAINT100.ASM

```
; VAGTS H/Z-100 PAINT PROGRAM - UPDATED 3/10/89
; Mr. Steven W. Vagts
; 2409 Riddick Rd
; Elizabeth City, NC 27909
;    (919) 335-7487
;
; Following codes are from CP/M INTERFACE GUIDE
CONIN   EQU     1       ;CONSOLE INPUT
CONOUT  EQU     2       ;CONSOLE OUTPUT
LSTOUT  EQU     5       ;LIST DEVICE (PRINTER) OUTPUT
CONSIO  EQU     6       ;DIRECT CONSOLE I/O
PRINT   EQU     9       ;PRINT STRING
LINPUT  EQU     10      ;LINE INPUT FUNCT
OPEN    EQU     15      ;OPEN FILE
CLOSE   EQU     16      ;CLOSE FILE
RDFIL   EQU     20      ;READ FILE (SEQUENTIAL)
WRTFIL  EQU     21      ;WRITE FILE (SEQUENTIAL)
MAKFIL  EQU     22      ;MAKE FILE
BDOS    EQU     5       ;BDOS VECTOR
DFCB    EQU     05CH    ;DEFAULT FCB
DMA     EQU     080H    ;DMA AREA
ESCAPE  EQU     27      ;ESC CHAR

        ORG     100H
START   EQU     $
        LXI     H,0     ;HL REG=0
        DAD     SP      ;LOCATE STACK
        LXI     SP,STACK  ;SET OUR OWN
        PUSH    H       ;SAVE CP/M S STACK
        PUSH    PSW     ;SAVE CP/M S FLAGS
        JMP     RDSCN1  ;READ OPENING SCREEN

INPUT   MVI     C,CONIN
        CALL    BDOS    ;DIRECT INPUT
        CPI     01BH    ;ESC CODE?
        JZ      XCAPE
        CPI     008H    ;BACKSPACE?
        CZ      DELET
        CPI     07FH    ;DELETE?
        CZ      DELET
        CPI     00DH    ;CR?
        CZ      LF
        CPI     007H    ;CTRL-G?
        JZ      GRAFX
        CPI     004H    ;CTRL-D?
        JZ      DUMP
        CPI     00CH    ;CTRL-L?
        JZ      DSPLY
        CPI     010H    ;CTRL-P?
        JZ      PRNTR
        CPI     017H    ;CTRL-W?
        JZ      WDISK
        CPI     012H    ;CTRL-R?
        JZ      RDISK
        CPI     005H    ;CTRL-E?
        JZ      EXIT1
        CPI     002H    ;CTRL-B?
        JZ      RMARK
        JMP     INPUT

XCAPE   MVI     E,0FFH  ;REQ INPUT CHAR
        MVI     C,CONSIO
        CALL    BDOS    ;A=CHAR
        ORA     A       ;0?
        JZ      XCAPE   ;A=0 IF NO CHAR READY
        CPI     07EH    ;HELP KEY?
        JZ      HELP
        CPI     04AH    ;F0?
        JZ      ERASR
        CPI     053H    ;F1?
        JZ      RVID
        CPI     054H    ;F2?
        JZ      ERVID
        CPI     055H    ;F3?
        JZ      GRAF
        CPI     056H    ;F4?
        JZ      EGRAF
        CPI     057H    ;F5?
        JZ      RDMENU
        CPI     050H    ;F6?
        JZ      MARKR
        CPI     051H    ;F7?
        JZ      MOVBK
        CPI     052H    ;F8?
        JZ      COPYBK
        CPI     030H    ;F9-F12?
        JZ      XCAPE2
        JMP     TXSPC1

XCAPE2  MVI     E,0FFH  ;REQ INPUT CHAR
        MVI     C,CONSIO
        CALL    BDOS    ;A=CHAR
        ORA     A       ;0?
        JZ      XCAPE2  ;A=0 IF NO CHAR READY
        CPI     049H    ;F9?
        JZ      HELP
        CPI     04AH    ;F10?
        JZ      COLR    ;SET SCREEN COLOR
        CPI     04BH    ;F11?
        JZ      INPUT   ;NOT USED
        CPI     04CH    ;F12?
        JZ      INPUT   ;NOT USED
        JMP     TXSPC1

ERASR   MVI     E,0FFH   ;REQ INPUT CHAR
        MVI     C,CONSIO
        CALL    BDOS    ;A=CHAR
        ORA     A
        JZ      ERASR   ;A=0 - NO CHAR READY
ERA1    MVI     E,0FFH  ;REQ INPUT CHAR
        MVI     C,CONSIO  ;NEXT CHAR = ESC
        CALL    BDOS    ;GET NEXT
        ORA     A
        JZ      ERA1    ;A=0 - NO CHAR READY
        CPI     044H    ;LEFT ARROW
        JZ      ELNB
        CPI     043H    ;RIGHT ARROW
        JZ      ELNE
        CPI     042H    ;DOWN ARROW
        JZ      EPGE
        CPI     041H    ;UP ARROW
        JZ      EPGB
        CPI     048H    ;HOME
```

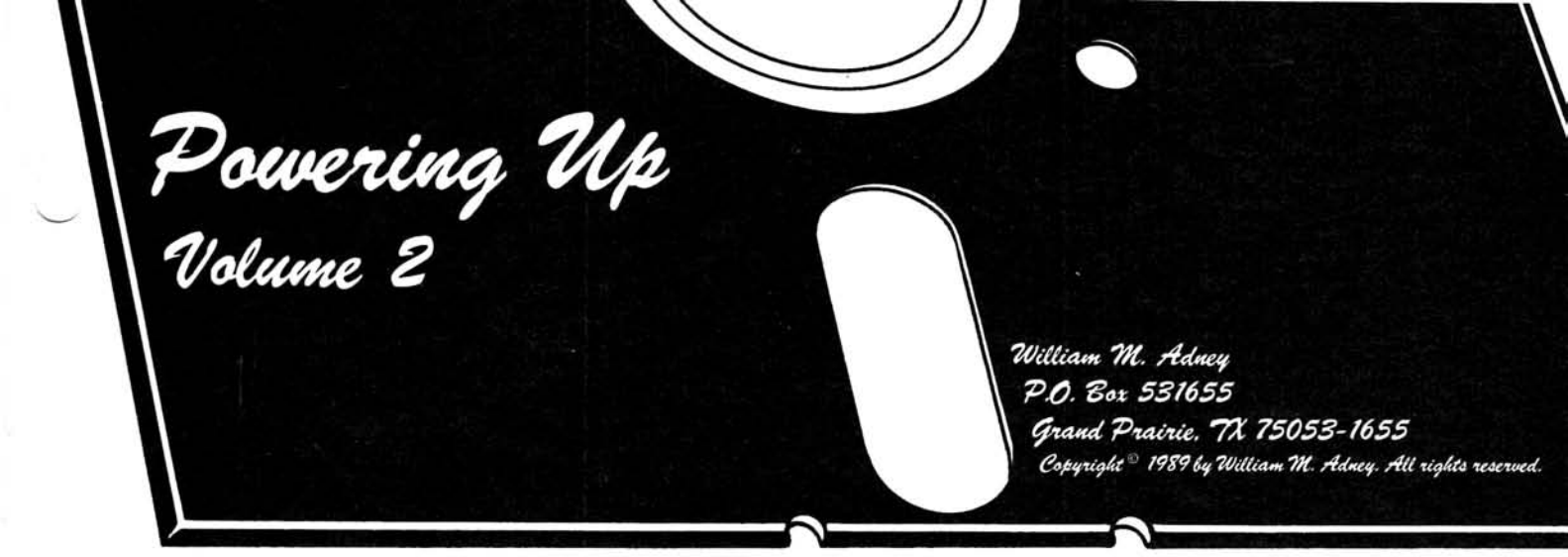
```
        JZ      EALL
        JMP     INPUT

RVID    MVI     A,070H  ;'p'
        JMP     TXSPC1
ERVID   MVI     A,071H  ;'q'
        JMP     TXSPC1
GRAF    MVI     A,046H  ;'F'
        JMP     TXSPC1
EGRAF   MVI     A,047H  ;'G'
        JMP     TXSPC1
ELNE    MVI     A,04BH  ;ERASE TO LINE END
        JMP     TXSPCL
ELNB    MVI     A,06FH  ;ERASE TO LINE BEGIN
        JMP     TXSPCL
EPGE    MVI     A,04AH  ;ERASE TO PAGE END
        JMP     TXSPCL
EPGB    MVI     A,062H  ;ERASE TO PAGE BEGIN
        JMP     TXSPCL
EALL    MVI     A,045H  ;ERASE SCREEN

TXSPCL  PUSH    PSW
        LXI     D,MSG43   ;SET DEFAULT COLOR
        CALL    PMSG
        POP     PSW
TXSPC1  STA     SPCL
        MVI     E,ESCAPE
        MVI     C,CONOUT
        CALL    BDOS
        LDA     SPCL
        MOV     E,A
        MVI     C,CONOUT
        CALL    BDOS
        JMP     INPUT

GRAFX   CALL    ESCJ
        LXI     D,MSG23
        CALL    GRAFX1
        LXI     D,MSG24   ;LIST GRAPHICS
        CALL    GRAFX1
        CALL    ESCK
        JMP     EGRAF

GRAFX1  CALL    PMSG
        MVI     B,33
        MVI     D,05EH
GRAFX2  PUSH    B
        PUSH    D
        MOV     E,D
        MVI     C,CONOUT
        CALL    BDOS
        MVI     E,020H
        MVI     C,CONOUT
        CALL    BDOS
        POP     D
        INR     D
        POP     B
        DCR     B
        JNZ     GRAFX2
        CALL    CRLF
        RET

COLR    CALL    ESCJ    ;SAVE CURSOR POSIT
        LXI     D,MSG39
        CALL    MSG50   ;FOREGROUND COLOR?
        MVI     C,CONIN
        CALL    BDOS    ;A=CHAR
        STA     MSG42+2
        LXI     D,MSG40
        CALL    PMSG    ;BACKGROUND COLOR?
        MVI     C,CONIN
        CALL    BDOS    ;A=CHAR
        STA     MSG42+3
        LXI     D,MSG42
        CALL    PMSG    ;SET COLORS
        JMP     EMSG

DELET   LXI     D,MSG43  ;SET DEFAULT COLOR
        CALL    PMSG
        LXI     D,MSG41   ;DELETE
        CALL    PMSG
        RET

ESCDEL  MVI     E,04EH  ;'N', DELETE CHAR
        JMP     TXCAPE
ESCE    MVI     E,045H  ;'E', CLEAR SCREEN
        JMP     TXCAPE
ESCJ    MVI     E,06AH  ;'j', SAVE CURSOR POSIT
        JMP     TXCAPE
ESCK    MVI     E,06BH  ;'k', SET CURSOR TO POSIT
        JMP     TXCAPE
ESCN    MVI     E,06EH  ;'n', CURSOR POSIT RPT
TXCAPE  PUSH    D
        MVI     E,01BH  ;SEND ESCAPE
        MVI     C,CONOUT
        CALL    BDOS
        POP     D
        MVI     C,CONOUT
        CALL    BDOS
        RET

MARKR   LXI     D,MSG43   ;SET DEFAULT COLOR
        CALL    PMSG
        CALL    ESCN
        LDA     MARKFG  ;CHECK FLAG
        ORA     A
        JNZ     MARK2
        MVI     A,031H
        STA     MARKFG
        LXI     H,MARKS
        MVI     B,4
        CALL    MK1
        CALL    ESCJ    ;SAVE CURSOR POSIT
        CALL    DUMP1   ;SAVE SCREEN
        LXI     D,MSG36   ;MOVE TO MARK 2
        CALL    MSG50
        CALL    ESCK
        MVI     E,02AH  ;'*'
        MVI     C,CONOUT
        CALL    BDOS
        JMP     INPUT

MARK2   MVI     A,032H
        STA     MARKFG
        LXI     H,MARKS+4
        MVI     B,4
        CALL    MK1
        MVI     E,02AH  ;'*'
        MVI     C,CONOUT
        CALL    BDOS
        CALL    ESCJ
        LXI     D,MSG37
        CALL    MSG50
        CALL    ESCK
        JMP     INPUT

RMARK   CALL    DSPLY1  ;ZERO MARKS SET
        JMP     MKDELT

MK1     PUSH    H
        PUSH    B
MK2     MVI     E,0FFH  ;REQ INPUT CHAR
        MVI     C,CONSIO
        CALL    BDOS
        ORA     A       ;CHECK IF 0
        JZ      MK2
        POP     B
        POP     H
        MOV     M,A
        INX     H
        DCR     B
        JNZ     MK1
        RET

MOVBK   MVI     A,000H  ;SET MOVFG TO MOVE
        STA     MOVFG
```

```
        JMP     BLOK

COPYBK  MVI     A,001H  ;SET MOVFG TO COPY
        STA     MOVFG
BLOK    CALL    ESCJ    ;SAVE CURSOR POSIT
        CALL    ESCN    ;REPORT CURSOR POSIT
        LXI     H,MARKS+8
        MVI     B,4
        CALL    MK1
        LDA     MARKS+2   ;ROW INFO OF MARKER1
        SUI     01FH    ;SUBTRACT ORIGIN-1
        STA     RSCNT   ;COUNT = # OF ROWS
        LDA     MARKS+3   ;COLUMN INFO OF MARKER1
        SUI     01FH    ;SUBTRACT ORIGIN-1
        STA     CSCNT   ;COUNT = # OF COLUMNS
        LDA     MARKFG  ;CHECK FLAG
        CPI     032H
        JNZ     NEGERR  ;NO 2 MARKS
        ANA     A       ;CLEAR THE CARRY BIT
        LDA     MARKS+2   ;ROW OF MARKER1
        MOV     B,A
        LDA     MARKS+6   ;ROW OF MARKER2
        SBB     B
        JM      NEGERR  ;RESULT IS NEG
        INR     A
        STA     RCOUNT  ;ROW COUNT DELTA
        ANA     A
        LDA     MARKS+3   ;COLUMN OF MARKER1
        MOV     B,A
        LDA     MARKS+7   ;COLUMN OF MARKER2
        SBB     B
        JM      NEGERR  ;RESULT IS NEG
        INR     A
        STA     CCOUNT  ;COLUMN COUNT DELTA
        LDA     MOVFG   ;COPY?
        ORA     A
        CNZ     DSPLY1  ;YES, REDISPLAY SCREEN
        LXI     H,BUFR  ;FIND FIRST CHAR
        LDA     RSCNT   ;ROW COUNT TO 1ST CHAR
        DCR     A
        JZ      COLCNT
        MOV     C,A
ROWCT1  MVI     B,80    ;CHAR COUNT/ROW
ROWCT2  MOV     A,M     ;GET CHAR
        INX     H
        CPI     ESCAPE
        CZ      ESCCH
        DCR     B
        JNZ     ROWCT2
        DCR     C
        JNZ     ROWCT1
COLCNT  LDA     CSCNT   ;COL COUNT TO 1ST CHAR
        MOV     C,A
        MVI     B,80    ;CHAR COUNT/ROW
COLCT1  DCR     C
        JZ      COLCT4
        MOV     A,M     ;GET CHAR
        CPI     ESCAPE
        CZ      COLCT2
        INX     H
        DCR     B
        JMP     COLCT1

COLCT2  CALL    CHOUT   ;CHAR TO CRT
        INX     H
        MOV     A,M
        CPI     06DH    ;'m'?
        JNZ     COLCT3
        CALL    CHOUT
        INX     H
        CALL    CHOUT
        INX     H
COLCT3  CALL    CHOUT
        INR     B
        INR     C
        RET

COLCT4  MVI     A,1     ;SET INSERT FLAG
        STA     INSFG
        PUSH    H       ;BUFR POSIT AT MEMORY BLOCK
        PUSH    B       ;LINE POSIT AT BLOCK START
        LXI     H,MARKS+8   ;CURSOR TO NEW
        CALL    SCRNP   ;INSERT POSIT
        POP     B
        POP     H
        LDA     CCOUNT  ;SET COLUMN COUNT
        MOV     C,A

BLKCT1  LDA     INSFG
        ORA     A
        JNZ     INSCH   ;INSERT CHAR ROUTINE
        MOV     A,M     ;A=CHAR
        INX     H
        CPI     ESCAPE
        CZ      ESCCH
        DCR     B       ;END OF ROW?
        JNZ     BLKCT1  ;NOT YET
        LDA     MOVFG   ;COPY?
        ORA     A
        CZ      DBLKLN  ;NO - DELETE BLOCK LINE
        PUSH    H
        LXI     D,MSG18
        CALL    PMSG
        POP     H
        LDA     RCOUNT  ;ROW COUNT
        DCR     A
        STA     RCOUNT  ;BLOCK DONE?
        JZ      MKDELT
        LDA     MARKS+10
        INR     A       ;INR TO NEXT ROW
        STA     MARKS+10
        LDA     MARKS+2
        INR     A
        STA     MARKS+2
        JMP     COLCNT

INSCH   MOV     A,M     ;CHAR=A
        PUSH    H
        PUSH    PSW
        PUSH    B
        MOV     E,A
        MVI     C,CONOUT
        CALL    BDOS
        POP     B
        POP     PSW
        POP     H
        INX     H
        CPI     ESCAPE
        CZ      ESCCH2
        DCR     B
        DCR     C
        JNZ     BLKCT1
        CALL    TDLY
        MVI     A,0
        STA     INSFG   ;RESET INSERT FLAG
        JMP     BLKCT1

DBLKLN  PUSH    H
        LXI     D,MSG43   ;SET DEFAULT COLORS
        CALL    PMSG
        LXI     H,MARKS
        CALL    SCRNP
        LDA     CCOUNT
        MOV     B,A
DCHAR   PUSH    B
        CALL    ESCDEL
        POP     B
        DCR     B
        JNZ     DCHAR
        CALL    TDLY
        POP     H
        RET

CHOUT   PUSH    H
        PUSH    B
        MOV     E,M     ;E=NEXT CHAR
```

```
        MVI     C,CONOUT
        CALL    BDOS
        POP     B
        POP     H
        RET

ESCCH   MOV     A,M     ;A=ESC CHAR
        CPI     06DH    ;'m'?
        CZ      ADDB
        CALL    ADDB
        RET

ESCCH2  MOV     A,M     ;A=ESC CHAR
        CPI     06DH    ;'m'?
        CZ      ADDCB
        CALL    ADDCB
        RET

ADDCB   INR     C
        INR     C
ADDB    INR     B
        INR     B
        RET

NEGERR  LXI     D,MSG38   ;NEG RESULT
        CALL    MSG50
        CALL    DELAY
        LDA     MOVFG   ;CHECK MOVE FLAG
        ORA     A
        CZ      DSPLY1  ;DISPLAY SCREEN
MKDELT  MVI     A,000H  ;0 MARKFG
        STA     MARKFG
        LXI     H,MARKS
        MVI     B,12
        MVI     A,000H
MKCLR   MOV     M,A
        INX     H
        DCR     B
        JNZ     MKCLR
        JMP     EMSG

SCRNP   MVI     B,4     ;SCREEN POSIT
MKD1    MOV     E,M
        PUSH    H
        PUSH    B
        MVI     C,CONOUT
        CALL    BDOS
        POP     B
        POP     H
        INX     H
        DCR     B
        JNZ     MKD1
        RET

MSG50   PUSH     D
        LXI     D,MSG2  ;ERASE & ENABLE LINE 25
        CALL    PMSG
        POP     D
        CALL    PMSG
        RET

DUMP    CALL    ESCJ
        LXI     D,MSG43   ;SET DEFAULT COLOR
        CALL    PMSG
        CALL    DUMP1
        JMP     EMSG
DUMP1   LXI     D,MSG3  ;DUMP TO MEMORY & TXMIT PAGE
        CALL    MSG50
SAVE    LXI     H,BUFR
TXCH    PUSH    H
        MVI     E,0FFH  ;REQ INPUT CHAR
        MVI     C,CONSIO
        CALL    BDOS    ;A=CHAR
        ORA     A       ;0?
        POP     H
        JZ      TXCH    ;NOT YET
        MOV     M,A
        INX     H
        CPI     15Q     ;CR SENT AT END OF SCREEN
        INZ     TXCH
        RET

DSPLY   CALL    DSPLY1
        JMP     EMSG

DSPLY1  CALL    ESCE    ;CLEAR SCREEN
        LXI     D,MSG4  ;RECALL SCREEN FROM MEMORY
        CALL    MSG50
        LXI     H,BUFR
DSPLY2  MVI     B,80
DSPLN   MOV     A,M     ;GET CHAR
        MOV     E,A
        INX     H
        CPI     ESCAPE
        CZ      ESCCH
        PUSH    B
        PUSH    H
        MVI     C,CONOUT
        CALL    BDOS
        POP     H
        POP     B
        MOV     A,M     ;GET CHAR
        CPI     15Q     ;CR? MEANS END OF SCREEN
        RZ
        DCR     B
        JZ      SCRLF
        JMP     DSPLN

SCRLF   PUSH    H
        CALL    CRLF
        POP     H
        JMP     DSPLY2
HELP    CALL    DUMP1   ;SAVE SCREEN
        LXI     D,MSG1  ;DISABLE LINE 25, CLS
        CALL    PMSG
        CALL    CLRFCB
        MVI     B,11
        LXI     D,DFCB+1  ;POINT TO BEGIN OF FCB
        LXI     H,PSCRN1  ;FILE NAME ADDR
        CALL    RDNAME
        JZ      RD1ERR  ;OPEN FAILED
        CALL    RLOOP   ;OPEN OK
        JNZ     RDMER1  ;READ FAILED
        CALL    CLOSDK
        CALL    DELAY
        JMP     DSPLY

RD1ERR  LXI     D,MSG21   ;PAINT100.SC1 NOT ON DISK
        CALL    PMSG
        JMP     RDMER2

RDMENU  CALL    DUMP1   ;SAVE SCREEN
        LXI     D,MSG1  ;DISABLE LINE 25, CLS
        CALL    PMSG
        CALL    CLRFCB
        MVI     B,11
        LXI     D,DFCB+1  ;POINT TO BEGIN OF FCB
        LXI     H,PSCRN2  ;FILE NAME ADDR
        CALL    RDNAME
        JZ      RDMERR  ;OPEN FAILED
        CALL    RLOOP   ;OPEN OK
        JNZ     RDMER1  ;READ FAILED
        CALL    CLOSDK
QUEST   LXI     D,MSG34
        CALL    MSG50
        MVI     C,CONIN
        CALL    BDOS    ;A=CHAR
        CPI     031H
        JZ      PICAP   ;SET PICA
        CPI     032H
        JZ      ELITEP  ;SET ELITE
        CPI     033H
        JZ      COMPON  ;SET COMPRESSED TEXT
        CPI     034H
        JZ      COMPOF  ;RELEASE COMPRESSED TEXT
        CPI     035H
```

```
        JZ      EMPHS   ;SET EMPHASIS
        CPI     036H
        JZ      EMPHR   ;RELEASE EMPHASIS
        CPI     037H
        JZ      DBLS    ;SET DOUBLE PRINT
        CPI     038H
        JZ      DBLR    ;RELEASE DOUBLE PRINT
        CALL    MUC     ;MAP TO UPPER CASE
        CPI     04CH    ;'L'
        JZ      LPTLS   ;SET LPT LS
        CPI     054H    ;'T'
        JZ      LPTAB   ;SEND TAB TO PRINTER
        CPI     05AH      ;'Z'
        JZ      DSPLY   ;NO CHANGE/DONE
        JMP     QUEST

RDMERR  LXI     D,MSG22   ;PAINT100.SC2 NOT ON DISK
        CALL    PMSG
        JMP     RDMER2

RDMER1  CALL    NOREAD
RDMER2  CALL    DELAY
        JMP     DSPLY

PICAP   MVI     A,0
        STA     PICAFG  ;SET PICA PITCH
        LXI     H,MSG25
        JMP     SETMSG

ELITEP  MVI     A,16
        STA     PICAFG  ;SET ELITE PITCH
        LXI     H,MSG25+3
        JMP     SETMSG

COMPON  MVI     A,52
        STA     PICAFG
        LXI     H,MSG25   ;SET PICA PITCH
        CALL    LPMSG
        MVI     E,15    ;SET COMPRESSED TEXT
        CALL    PRNTCH
        JMP     SETMSG1

COMPOF  MVI     A,0
        STA     PICAFG
        MVI     E,18    ;REL COMPRESSED TEXT
        CALL    PRNTCH
        JMP     SETMSG1

EMPHS   LXI     H,MSG26
        JMP     SETMSG

EMPHR   LXI     H,MSG26+3
        JMP     SETMSG

DBLS    LXI     H,MSG27
        JMP     SETMSG

DBLR    LXI     H,MSG27+3
        JMP     SETMSG

PRNTR   CALL    ESCJ
        LXI     D,MSG5  ;PRINT SCREEN MSG
        CALL    MSG50
        CALL    SAVE
        LXI     H,PICAFG
        MVI     A,80    ;80 CHAR/LINE
        STA     SCPL    ;# OF SCREEN CHAR/LINE
        ADD     M       ;ADD LINE LENGTH DELTA
        STA     PCPL    ;# OF PRINTER CHAR/LINE
        LXI     H,BUFR
        LDA     PICAFG  ;COMPRESSED PRINTING?
        CPI     52
        JZ      PRESSP  ;YES
PRLP    LDA     PCPL
        MOV     C,A     ;COUNT CHAR/LINE
        CALL    LPTABR  ;TAB TO RIGHT
        LDA     SCPL
        MOV     B,A     ;BOTH B&C COUNTERS ARE SET
PRLP1   MOV     A,M     ;GET CHAR
        CPI     15Q     ;CR? MEANS END OF SCREEN
        JZ      LPDONE
        MOV     E,A
        CPI     ESCAPE
        CZ      ADD1    ;ALLOW FOR ESC CODES
        JZ      PRLP1
        PUSH    B
        PUSH    H
        LDA     LPRVFG
        ORA     A
        JNZ     PRLPRV
        LDA     LPGFFG
        ORA     A
        JNZ     PRLPGFX
        CALL    PRNTCH
PRLP2   POP     H
        POP     B       ;RECALL B&C
        DCR     C       ;END OF LPT LINE?
        JZ      PRLP3   ;YES, FINISH BUFR LINE
        INX     H
        DCR     B
        JNZ     PRLP1   ;END OF SCREEN LINE?
        JMP     PRLP4   ;YES, FINISH LPTR LINE

PRLP3   MOV     A,M     ;DISCARD REMAINING BUFR LINE
        CPI     ESCAPE
        CZ      SKIP
        INX     H
        DCR     B
        JNZ     PRLP3
        JMP     PRLP5   ;DONE

PRLP4   MVI     E,020H  ;PRINT SPACE
        CALL    PRNTCH
        DCR     C
        JNZ     PRLP4
PRLP5   CALL    PCRLF
        JMP     PRLP

PRESSP  LDA     PCPL
        MOV     C,A     ;COUNT CHAR/LINE
        CALL    LPTABR  ;TAB TO RIGHT
        LDA     SCPL
        MOV     B,A     ;BOTH B&C COUNTERS ARE SET
PRESP1  MOV     A,M     ;GET CHAR
        CPI     15Q     ;CR? MEANS END OF SCREEN
        JZ      LPDONE
        MOV     E,A
        CPI     ESCAPE
        JZ      SKIP1   ;SKIP ESC CODES
        CALL    PRNTCH  ;FOR COMPRESSED TEXT
        DCR     C       ;END OF LPT LINE?
        JZ      PRESP2  ;YES, FINISH BUFR LINE
        INX     H
        DCR     B       ;END OF SCREEN LINE?
        JNZ     PRESP1
        JMP     PRESP3  ;YES, FINISH LPTR LINE

PRESP2  MOV     A,M     ;DISCARD REMAINING BUFR LINE
        CPI     ESCAPE
        CZ      SKIP1   ;SKIP ESCAPE CODE
        INX     H
        DCR     B
        JNZ     PRESP2
        JMP     PRESP4  ;DONE

PRESP3  MVI     E,020H  ;PRINT SPACE
        CALL    PRNTCH
        DCR     C
        JNZ     PRESP3
PRESP4  CALL    PCRLF
        JMP     PRESSP

SKIP    INX     H       ;SKIP ESC CODE & FOLLOW CHAR
        MOV     A,M
        CPI     06DH    ;'m'(COLOR) HAS TWO CODES
        CZ      ADDH
```

```
        INR     B
        RET

SKIP1   INX     H       ;SKIP ESC CODE & FOLLOW CHAR
        MOV     A,M
        CPI     06DH    ;'m'(COLOR) HAS TWO CODES
        CZ      ADDH
        INX     H
        JMP     PRESP1

PRLPRV  CALL    RVIDC
        JMP     PRLP2

PRLPGFX CALL    GRAFC
        JMP     PRLP2

PRNTCH  PUSH    PSW
        PUSH    H
        PUSH    B
        MVI     C,LSTOUT  ;E=CHAR
        CALL    BDOS
        POP     B
        POP     H
        POP     PSW
        RET

ADD1    INX     H
        MOV     A,M
        CPI     070H    ;"p"?
        JZ      LPRVID
        CPI     071H    ;"q"?
        JZ      ELPRVID
        CPI     046H    ;"F"?
        JZ      LPGRAF
        CPI     047H    ;"G"?
        JZ      ELPGRAF
        CPI     06DH    ;"m"? COLOR?
        CZ      ADDH    ;CAN T PRINT IT OUT
ADDCD   INX     H
        MVI     A,0
        ORA     A
        RET

ADDH    INX     H
        INX     H
        RET

LPRVID  MVI     A,1
        STA     LPRVFG  ;RV FLAG
        JMP     ADDCD

ELPRVID MVI     A,0
        STA     LPRVFG
        JMP     ADDCD

LPGRAF  MVI     A,1
        STA     LPGFFG  ;GRAPHICS FLAG
        JMP     ADDCD

ELPGRAF MVI     A,0
        STA     LPGFFG
        JMP     ADDCD
```



William M. Adney
P.O. Box 531655
Grand Prairie, TX 75053-1655

Understanding Zenith ROM Error Messages

Unlike most of the articles in this series, this article is intended to be primarily used for reference. Many of these error messages and some explanations are described in the Owner's Manual for your specific Zenith computer model, but some are not. For example, ROM error messages for the Z-525 Cache Card are not listed in my Z-386 Owner's Manual.

Many of the Zenith ROM error messages are can be easily identified because they are displayed with special characters, typically three plus signs (+++) before and/or after the message. For ease of reference, these special characters are omitted from this article, and all error messages are listed in alphabetical order in bold type. Although some of these error messages use acronyms or abbreviations (e.g., CRC or DMA), I have not attempted to describe the details of what these mean; rather, I have described what the general problem is and what steps you need to take to further isolate the problem or fix it. Virtually all of these error messages will normally occur either during the POST (Power On Self Test) or initialization of your system as described in the previous article. Because the DOS is not loaded when these error messages are displayed, you should generally not suspect a new DOS version or any other software that has been recently added to your system. But before we get into the details of each error message, a review of some general troubleshooting techniques is in order.

General Hardware Troubleshooting

In most cases, you will find that these ROM error messages will be displayed immediately or shortly after you have made some kind of hardware change to your system. When this happens, it usually means that at least one connector or circuit board has not been plugged in correctly and completely. Sometimes correcting the problem is simple and obvious, such as just plugging in the keyboard or disk drive connector. When an explanation of an error message suggests checking cables, connectors or boards, be sure to double-check ALL connectors because you may have "bumped" or pulled something loose that was not part of a hardware change. Check to make sure that all pin connectors are connected to ALL pins — it is easy to skip one row of pins when connecting a cable to a disk controller. Also check to be sure that the connectors, especially the pin connectors on the disk controller, have the correct orientation. In most cases, you will find that pin 1 is on the "top" row of the disk controller (it's labeled on the board — check it), and most of the flat ribbon cables use the convention that pin 1 has a colored stripe (e.g., red or blue). Many of these problems normally occur after you have made some kind of hardware modification to your system, and it is really a result of an incorrect connection.

To help avoid this kind of problem, I recommend that you have a permanent black felt tip marker, such as a Magic Marker, in hand when you disassemble your computer. Many connectors, especially those for disk drives, must be connected in a certain way. When you disassemble your computer, use the marker liberally to indicate both the connector name and the orientation of the connector on a board. On a disk controller end of a drive cable for example, I mark the connector name (e.g., J-1) and an "up arrow" to indicate its orientation. On the other end, I mark a drive letter for a floppy drive (e.g., A or B) or a hard drive number (e.g., 0 or 1), and I mark some kind of indicator directly on the drive itself to show which side the striped edge is on. I also mark the power supply connectors with an "A" or "0" to show the connections when I build a computer kit because some of these cables are shorter than others. These are only suggestions, and you should do whatever kind of cable marking that is meaningful to you. Just keep it clear and simple so you can remember it a year from now when you add a new hard drive or whatever. The important point here is to do this the FIRST time you disassemble your system (or when you build it), and if you do it clearly, you will not have to refer to any instructions on which cable goes where and how the next time you need to take things apart.

If you see one of these error messages and have not disassembled your computer for some time, keep in mind that a connector still may have been jarred loose. When performing troubleshooting, it is very important to have a systematic procedure, and I always suggest checking the simple things first. You might be amazed at how often an easy check will reveal the problem quickly.

After you have double-checked all the printed circuit board and cable connections and the problem still occurs, there are several possibilities. Sometimes the socketed chips on various boards will not be making good contact after some use. All you need to do is remove each board, and gently but firmly, perform a "chip massage" by pressing each chip into its socket. It is NOT necessary to remove the chip. A chip massage will frequently correct many of the "memory" failures.

Like all mechanical and electronic devices, computer systems will occasionally have a hardware failure. Except for a mechanical failure on a hard drive which usually occurs immediately after you power on the system, many failures will occur after the system has been running for some time. When this occurs, it is usually because there is a "marginal" chip (e.g., memory) in the system that works fine when it is "cold," but fails after heat accumulates inside the computer. Depending on what the failure is, you will probably want to use the ROM TEST com-

mand or the Zenith Disk Diagnostics program as appropriate.

Keep in mind that all Zenith test and diagnostic programs may not work with non-Zenith hardware. For example, the disk controller tests in the Zenith Disk Diagnostics will almost certainly NOT work with non-Zenith disk controllers. Depending on your technical knowledge of your computer and your budget, you may want to consider this when you buy additional hardware for your system. If your system contains Zenith disk controllers and memory, you can do a lot of this testing yourself and perhaps save repair costs for troubleshooting in the long run. Unfortunately, many of these Zenith upgrades have outrageous and uncompetitive list prices, although you can get substantial discounts if you are a HUG member. You will have to balance the disadvantage of higher cost in the short term against reduced maintenance and repair costs in the long term.

In summary, be sure to first check all of the printed circuit boards and all connectors. For memory failures, try a chip massage. If you have just installed a hardware change, remove it, and check the system again. For all other errors, run the Zenith ROM TEST command (see previous article) and/or the Zenith Disk Diagnostics as appropriate. If you still have a failure, you will probably need to contact a dealer for assistance in troubleshooting and repairing the computer.

Types of ROM Error Messages

There are really four types or categories of ROM error messages: Disk Error, Error, SETUP error messages, and other ROM error messages that usually indicate some kind of user error. Because the SETUP error messages are self-explanatory and only indicate a user error (e.g., Invalid time), they are not included in this article.

Disk Error messages usually indicate a specific point of failure for the disk controller (floppy or hard), the drive hardware itself or the disk media. For some reason — which is usually either mechanical or electronic — the floppy or hard disk drive system is not responding as it should, and this will normally occur when you power on or reboot the system. Disk Error messages are generated by the ROM because DOS is not yet loaded, and there is no way to display the usual DOS "Abort, Retry" errors until the system is booted. In many cases, Disk Error messages can be easily fixed because the cause is frequently a loose or reversed cable connection, or a card that has been inserted improperly. My experience is that most Disk Error messages occur right after I have disassembled my system and failed to make sure that everything is reconnected properly. Although it is obviously possible that the Disk Error messages may indicate a "real" hardware or electronic failure, I have never had that happen, even through half a dozen hard drive crashes. Disk Error messages are indicated by an error message preceded by ÷DISK ERROR:÷, but I have included that at the end of the message, where appropriate, to preserve the alphabetical listing.

Plain old "Error" messages are preceded by ÷ERROR:÷, and they are typically caused by a hardware element's failure to successfully pass the POST. These Error messages are most frequently the result of a loose or improperly inserted circuit board or connector. You will generally find the error message itself will help pinpoint the problem area, but these error messages are, of necessity, quite general. For example, any of the error messages indicating a memory, parity, CPU, ROM or timer failure may really be caused by a printed circuit board or chip not being completely or correctly inserted into a slot or socket. Be sure to keep that in mind if you are troubleshooting a problem based on one of these error messages.

The third type — other ROM messages — are displayed after the system has successfully passed the initialization and POST, but the boot attempt failed. These other ROM messages are standalone messages with no special characters (i.e., +++) or indicators, and they indicate some kind of user error. These messages include: "No bootable partition", "Not a bootable partition", and "No system".

There is another completely different category of error message that is not generated by the ROM, but may be displayed during system boot. These error messages may be generated by virtually any program based on a command line contained in the AUTOEXEC.BAT file. The most common error messages are "Invalid date" and "Invalid time" which mean that the response to the DATE and TIME commands were invalid. These error messages are usually the result of a typing error and are easily corrected. Perhaps the most confusing is the use of the TIME command based on the 24-hour clock for those not used to that, and the most common error is to attempt to enter an "a" (for AM) or a "p" (for PM) because that's the way times are displayed by the DIR command. For the time value, it must be entered in an hours:minutes:seconds (i.e., hh:mm:ss) format separated by colons. To convert the PM hours to the 24-hour clock for the TIME command, just add 12 to the PM value — 3:00 PM is 15:00, 6:00 PM is 18:00, 9:00 PM is 21:00, and so on.

Now it's time to take a look at the different ROM error messages. All error messages are listed in alphabetical order which does not include certain words like ÷ERROR:÷ or ÷DISK ERROR:÷, except for the first one where no error message is displayed at all. Depending on the ROM version, the specific computer model, and the type of error message, some may not be prefixed by these words.

No Error Message — No Nothing

This message generally indicates a failure on the boot disk, either a floppy drive or a hard drive. Observe the drive lights, and wait for about a minute for the system to report disk problems. For a floppy-drive system, insert another bootable disk, and reboot the system. If the problem continues, run the ROM TEST command and/or the Zenith disk diagnostics as appropriate. Otherwise, you should contact a dealer for assistance. For a hard-drive system, use a bootable floppy in drive A, reboot using the ÷BF0÷ command described in the previous article, and try to access the hard disk with the DIR command. If that works, back up all files on the hard drive, FORMAT it, and reload all files. If the problem continues, the hard drive may have failed. Run the Zenith disk diagnostics to check out both the hard drive and controller. Otherwise, you should contact a dealer for assistance.

Bad configuration information found in [auxillary] CMOS! (ERROR:)

As a matter of information, the word "auxillary" is not spelled correctly in this error message — it should be "auxiliary." The information stored by the SETUP program in the Complimentary Metal Oxide Semiconductor (CMOS) chip does not correctly reflect the current hardware (e.g., memory, video, disk drives, etc.) or is invalid. This can happen during powering up a brand new system, when you have removed one or more of the printed circuit boards from the computer, or when the backup battery needs to be replaced (See "Please replace the back-up battery!" error message). Run the ROM SETUP command to reset all information to correctly reflect the current hardware configuration.

Bad disk controller! (DISK ERROR:)

Usually indicates a bad disk controller on the boot drive (floppy or hard drive). May also indicate a bad disk drive. Verify that all printed circuit cards and cable connectors are correctly placed and firmly seated. Run the ROM TEST command and/or Zenith Disk Diagnostics to attempt to isolate the problem to the disk controller or drive. If a hardware change was recently made, reverse the change (e.g., remove the board), and if the problem disappears, contact the dealer or manufacturer of the hardware (non-Zenith hardware) for assistance. Otherwise, contact an authorized Zenith dealer or service center for assistance.

Base memory size error!
SETUP: xxxK ACTUAL: xxxK

The ACTUAL amount of memory found in the system is not exactly the

same as that specified in the ROM SETUP program. Run the SETUP command to update the configuration with the correct base memory value shown as ACTUAL (up to 640K). Verify that all memory boards are correctly inserted in their sockets because this message may indicate a computer memory problem. Run the ROM TEST 3 (BASE MEMORY TEST) or the Zenith Disk Diagnostics to check out base memory.

Cache circuit card DATA failure!
Chip: U7xx
Cache circuit card failure!
Cache circuit card RAM failure!
Chip: U7xx
Cache circuit card TAG failure!
Chip: U7xx

These four error messages indicate a problem in a Zenith hardware Cache Memory Card, such as the Z-525 for the 16 MHz Z-386 (Z-386/16) computer. Early ROMs for the Z-386/16 could not correctly initialize the Cache Memory Card (usually displaying "Chip: U707" as the culprit) which can be corrected by updating to at least version 2.6E or later. Replacing the indicated chip (labeled on the card as U7xx, where the x represents a digit) ÷may÷ correct the problem, but if it does not, the entire system should be serviced by an authorized Zenith service center.

Cannot reset drive! (DISK ERROR:)

The system is unable to access (i.e., reset) a disk drive during the boot process because a disk is not correctly inserted in the drive (or try another bootable disk) or the drive door is not completely closed. May indicate a defective disk or disk drive. Correct the problem and reboot.

CMOS Memory Failure! (ERROR:)

Indicates that the Complimentary Metal Oxide Semiconductor (CMOS) chip, usually located on the I/O card for an 80286 or later system, has failed the ROM POST or initialization because of a bad chip. Verify that all boards — especially the I/O board on computers that have one — are correctly inserted in their sockets. Power off the system, wait about a minute, and power up again. If you see this error in combination with other error messages, and the problem persists, the entire system should be serviced by an authorized Zenith service center.

CPU failure! (ERROR:)

The Micro Processing Unit (MPU) chip (e.g., 8088, 80286, etc.) has failed the ROM POST or initialization. Verify that all boards (especially the CPU board) and chips are correctly and firmly inserted in their sockets. Power off the system, wait about a minute, and power up again. If you see this error in combination with other error messages, and the problem persists, the entire system should be serviced by an authorized Zenith service center.

CRC error! (DISK ERROR:)

The Cyclic Redundancy Check (CRC) on a disk failed to pass during system start-up. Try a different bootable disk. Also try to boot from a different disk drive. Verify that all disk drive cables are correctly and firmly seated on the controller(s) and drives. If you have a Zenith drive controller, run the Zenith Diagnostics Disk to help isolate the problem. If the problem persists, the entire system should be serviced by an authorized Zenith service center.

Data corrected! (DISK ERROR:)

The system is unable to access a disk drive during the boot process because the disk is not correctly inserted in the drive (or try another bootable disk) or the drive door is not completely closed. This message may also indicate a defective disk drive or controller which may require repair service.

Disk not bootable! (DISK ERROR:)

The ROM Boot Loader program was unable to successfully read the complete contents of sector zero (0) on a disk or the disk does not contain all files required to make it bootable. Copy or backup all information on that disk, FORMAT it again with the /S (System) switch, and copy or restore all information back to the original.

Divide by zero! (ERROR:)

A processor (i.e., either the MPU, such as an 80286, or a numeric coprocessor, such as an 80287) has encountered an unexecutable command that results in an attempt to divide by zero. May be caused by a program that has encountered an unusual condition which causes the error. May also be caused by an MPU failure, numeric coprocessor failure, system memory problem or an error in the ROM itself. Verify all printed circuit boards are inserted correctly. If the problem only occurs with a single program, contact the manufacturer of that program for assistance. If the Zenith Disk Diagnostics do not report a problem, contact an authorized Zenith dealer or service center for assistance.

DMA overrun! (DISK ERROR:)

Indicates a problem with Direct Memory Addressing (DMA) usually associated with a bad disk controller on the boot drive (floppy or hard drive). May also indicate a bad disk drive. Verify that all printed circuit cards and cable connectors are correctly placed and firmly seated. Run the ROM TEST command and/or Zenith Disk Diagnostics to attempt to isolate the problem to the disk controller or drive. If a hardware change was recently made, reverse the change (e.g., remove the board), and if the problem disappears, contact the dealer or manufacturer of the hardware (non-Zenith hardware) for assistance. Otherwise, contact an authorized Zenith dealer or service center for assistance.

Drive not ready! (DISK ERROR:)

Occurs when computer attempts to boot a disk, and the drive indicates a "Not ready" signal, usually because the drive door or latch is not closed properly. Verify that the disk is bootable and that the drive door or latch is properly closed, and reboot with CTRL-ALT-DEL. If this does not work, try another bootable disk or attempt to boot the system from another drive. May also be caused by a poor or broken cable connection.

Errors found!
Please press <Esc> to continue

An information message usually following another error message. Press the ESC key as instructed, and see the appropriate error message in this article.

Errors found! Please unlock keyboard, then press <Esc> to continue (ERROR:)

The keyboard lock is active (i.e., locked) with the key. Locate the key, unlock the keyboard, and press CTRL-ALT-DEL to reboot the system.

Expansion memory size error!
SETUP: xxxK ACTUAL: xxxK (ERROR:)
Extended memory size error!
SETUP: xxxK ACTUAL: xxxK

Both messages indicate that the SETUP command has one value defined for the memory size, but the ACTUAL amount of memory recognized during testing is different. Activate the ROM Monitor with CTRL-ALT-INS, run the SETUP command, and update the Expansion Memory Size to the value shown as ACTUAL in the error message. Save the configuration and reboot the computer.

FATAL: Internal Stack Failure, System Halted

Usually occurs when a program is defective or attempts to execute more hardware interrupts than the default number allowed by DOS (9 in most versions). Make a note of the program you were using and the last action you took within that program (e.g., the last keypress or program command you entered). Power off the computer for at least 30 seconds, then turn the system on again. Update the CONFIG.SYS file with the command line: ÷STACKS=18,128÷, which will double the number of defined stack frames. Restart the program which initiated the problem, and attempt to recreate the problem by entering the same command or keypress that occurred before the failure. If the error message is still displayed,

contact an authorized Zenith service center for assistance or check with the manufacturer of the program causing the problem.

Fatal Error: Cannot Continue! (ERROR:)

An unexpected error occurred that is not specifically identified by another error message. Power off the computer for at least 30 seconds, then turn the system on again. If the error message is still displayed, contact an authorized Zenith service center for assistance.

Fatal Slushware RAM Error

Slushware is a special Zenith technique of keeping some of the ROM "program information" (i.e., firmware) in a computer's RAM, and this technique improves system performance. This process is also referred to as "RAM shadowing" and is typically used in 80286 and 80386 computers that have a *standard* 1 MB of memory. The error message indicates that some part of the RAM shadowing process failed. Power off the computer for at least 30 seconds, then turn the system on again. If the error message is still displayed, contact an authorized Zenith service center for assistance.

Incorrect video configuration - Please run SETUP! (ERROR:)

The SETUP command has a value defined for the video configuration, but the actual configuration recognized during testing is different. Activate the ROM Monitor with CTRL-ALT-INS, run the SETUP command, and update the Video Display field to the correct value based on the video card installed in your computer. Save the configuration and reboot the computer.

Invalid address mark! (DISK ERROR:)

A bad or incorrect address mark, normally written on a disk by the FORMAT program, has been detected, and the disk cannot be read. This message usually indicates a defective disk or disk drive. Try a different disk. Also try using a different disk drive. Verify that all disk drive cables are correctly and firmly seated on the controller(s) and drives. If you have a Zenith drive controller, run the Zenith Diagnostics Disk to help isolate the problem. If the problem persists, the entire system should be serviced by an authorized Zenith service center.

Invalid command! (ERROR:)

A processor (i.e., either the MPU, such as an 80286, or a numeric coprocessor, such as an 80287) has encountered a command that causes a problem. May be caused by a program that has encountered an unusual condition which causes the error. May also be caused by an MPU failure, numeric coprocessor failure, system memory problem or an error in the ROM itself. Verify all printed circuit boards are inserted correctly. If the problem only occurs with a single program, contact the manufacturer of that program for assistance. If the Zenith Disk Diagnostics do not report a problem, contact an authorized Zenith dealer or service center for assistance.

Invalid data read! (DISK ERROR:)

The system is unable to access a disk drive during the boot process because the disk is not correctly inserted in the drive (or try another bootable disk) or the drive door is not completely closed. This message may also indicate a defective disk drive or controller which may require repair service.

Invalid/No keyboard code received! (ERROR:)

The keyboard is not responding correctly to the ROM check. Usually indicates the keyboard connector has been unplugged or loose and should be checked. May indicate that a keyboard switch is not set to the correct position (usually PC/XT or AT) for use with a specific computer. May also indicate that a ROM upgrade is required to use a 101-key keyboard with an "older" computer.

Keyboard lock active: the keyboard is disabled

The keyboard is disabled because the lock has been enabled. Find the key, unlock the keyboard, and reboot the system.

Keyboard not responding or not connected! (ERROR:)

The Power On Self Test (POST) was unable to verify that the keyboard was available to the system. Usually indicates the keyboard connector has been unplugged or loose and should be checked. May indicate that a keyboard switch is not set to the correct position (usually PC/XT or AT) for use with a specific computer. May also indicate that a ROM upgrade is required to use a 101-key keyboard with an "older" computer.

Memory parity failure! (ERROR:)

Indicates a potential hardware problem with memory parity checking and may be a hardware failure. Run the Zenith Disk Diagnostics to attempt to isolate the problem. If the message occurs when the computer is "hot" (i.e., it has been powered on for some time), chances are that the hardware failure is heat related. If the Zenith Disk Diagnostics do not report a problem, contact an authorized Zenith dealer or service center for assistance.

Must run SETUP to boot from Winchester! (DISK ERROR:)

The system attempted to boot from a winchester hard drive, but the drive type was not defined in the SETUP program. Typically occurs after removing a board from a computer or when the backup battery fails. Activate the ROM Monitor with CTRL-ALT-INS, run the SETUP command, and update the Drive Type field to the correct value based on the hard drive installed in your computer. Save the configuration and reboot the computer.

No bootable partitions

The system attempted to boot from a hard drive (e.g., drive C), but no bootable partition was defined by the PART command. Run the PART command to define a bootable partition (normally partition 1) that has been formatted with the FORMAT command using the /S (System) switch and reboot the system. If the partition has already been formatted without the /S switch and any software or data has been loaded, be sure to backup the entire partition to floppy disks before running the FORMAT command again (using the /S switch) because the FORMAT command will destroy all information on the hard drive.

No system
Non-system disk

Usually occurs when attempting to boot a floppy disk system using a disk that does not contain the required system files created by the FORMAT command with the /S (System) switch. The problem can normally be fixed by inserting a bootable floppy disk and rebooting the system. This error message may also indicate a defective drive or loose cable connections and may occur when attempting to boot a hard drive that has not been formatted with the /S switch.

Non-maskable interrupt! (ERROR:)

A invalid "command" (called an interrupt) was issued from some source. May be caused by a program problem, power interruption, hardware problem, or a ROM failure of some kind. Verify all printed circuit boards are inserted correctly. If the problem only occurs with a single program, contact the manufacturer of that program for assistance. If the Zenith Disk Diagnostics do not report a problem, contact an authorized Zenith dealer or service center for assistance.

Not a bootable partition

The system attempted to boot from a hard drive (e.g., drive C) partition, but it was not bootable. Be sure the partition (normally partition 1) was defined as bootable by the PART command. Also be sure that the partition has been formatted with the FORMAT command using the /S (System) switch. If the partition has already been formatted without the /S switch and any software or data has been loaded, be sure to back up the entire partition to floppy disks before running the

FORMAT command again (using the /S switch) because the FORMAT command will destroy all information on the drive.

Overflow! (ERROR:)

A processor (i.e., either the MPU, such as an 80286, or a numeric coprocessor, such as an 80287) has encountered a command that causes a problem. May be caused by a program that has encountered an unusual condition which causes the error. May also be caused by a system failure to pass the POST, an MPU failure, numeric coprocessor failure, system memory problem or an error in the ROM itself. Verify all printed circuit boards are inserted correctly. If the problem only occurs with a single program, contact the manufacturer of that program for assistance. If the Zenith Disk Diagnostics do not report a problem, contact an authorized Zenith dealer or service center for assistance.

Parity failure! (ERROR:)

Indicates a potential hardware problem with memory parity checking and may be a hardware failure. Run the Zenith Disk Diagnostics to attempt to isolate the problem. If the message occurs when the computer is "hot" (i.e., it has been powered on for some time), chances are that the hardware failure is heat related. If the Zenith Disk Diagnostics do not report a problem, contact an authorized Zenith dealer or service center for assistance.

Parity hardware failure!
Address: hhhh:hhhh, Bit b, Chip Unnn (ERROR:)

Displayed by the Monitor ROM when a hardware failure is detected during the Power On Self Test (POST) at the hex address shown. Usually indicates a hardware failure in the computer. Record the address information displayed, and see the "RAM failure!..." error message for additional information. If the problem continues, contact an authorized Zenith dealer or service center for assistance.

Please replace the back-up battery! (ERROR:) Press any key to continue

Usually indicates that the backup battery in the computer is worn out and must be replaced. Normally accompanied by a series of additional error messages indicating that the SETUP parameters have been lost. After replacing the battery, activate the ROM Monitor with CTRL-ALT-INS, run the SETUP command, and update the SETUP parameters to the correct values based on the hardware actually installed in your computer. Save the configuration and reboot the system.

RAM failure! Address: hhhh:hhhh, Bit: b, Chip: Unnn (ERROR:)

Typically occurs during the POST because the computer is unable to read from or write to memory at the hex address indicated. RAM chips on Zenith memory boards are identified by a "U" number (e.g., U701) on each printed circuit board which is displayed in this message. Carefully remove (using the instructions in the Owner's manual) the memory board(s) from the computer, and examine the pins on the suspect chip to see if they are bent or not making good contact in the socket. Place the memory board on a firm surface with the "chip side" up, and press the suspect chip firmly into its socket. This technique is called a "chip massage" and will frequently eliminate RAM failure and other similar problems with socketed chips. When you remove a board from a computer, you may want to perform a chip massage on all socketed chips on that board because sometimes corrosion or dirt may prevent the pins from making proper contact, especially when all components expand slightly from heat. Sometimes a chip will really fail, and it must be replaced, but try a chip massage first. If you are not able to or are uncomfortable with the idea of disassembling your computer, contact an authorized Zenith dealer or service center for assistance.

ROM checksum failure! (ERROR:)

The actual "programs" contained in the Monitor ROM did not match an internal numeric value called a checksum that is used for self-testing. Can be caused by a ROM chip that was "zapped" by static electricity because of poor installation procedures or by unauthorized modification to the ROM programs. Power off the system for about 30 seconds, then power up again. If the problem persists, the Monitor ROM will probably need to be replaced. Contact an authorized Zenith dealer or service center for assistance.

Seek failure! (DISK ERROR:)

A disk drive was unable to locate (i.e., "seek") a specified sector header on a track. May be caused by a defective disk or faulty disk drive. Verify that the drive door is closed and the disk is inserted properly. Make sure that all cables connectors and printed circuit boards are correctly and firmly inserted. Use the Zenith Disk Diagnostics to check out the suspect drive. If the problem persists, contact an authorized Zenith dealer or service center for assistance.

Sector not found! (DISK ERROR:)

The computer could not locate a specific sector on a disk, usually a floppy. This message usually indicates a defective disk or disk drive. Try a different disk. Also try using a different disk drive. Verify that all disk drive cables are correctly and firmly seated on the controller(s) and drives. If you have a Zenith drive controller, run the Zenith Diagnostics Disk to help isolate the problem. If the problem persists, the entire system should be serviced by an authorized Zenith service center.

System Control Processor failure! (ERROR:)

The System Control Processor (SCP) has failed to perform a required task and is not responding correctly. The SCP performs a number of essential functions, and its failure will probably result in a number of other symptoms, such as no keyboard response. SCP failure may also cause a system freeze, and this error message may also indicate a general software or hardware problem, such as the ROM or SCP incompatibility with a new ROM version. In most cases, you can power off the machine, wait about 30 seconds, and then power up again in an attempt to "clear" the error. If the problem persists, the entire system should be serviced by an authorized Zenith service center.

Timer interrupt failure! (ERROR:)

Virtually all computers contain a "timer" chip and associated circuitry that are used to synchronize vital computer functions. This error message indicates a failure in the Input/Output circuits (e.g., an I/O board for some computers) which may be a result of a major hardware failure. Carefully remove (using the instructions in the Owner's manual) the board(s) from the computer, and examine the pins on each socketed chip to see if they are bent or not making good contact in the socket. Place the board on a firm surface with the "chip side" up, and press each socketed chip firmly into its socket. This technique is called a "chip massage" and will frequently eliminate this problem and other similar problems with socketed chips. When you remove a board from a computer, you may want to perform a chip massage on all socketed chips on that board because sometimes corrosion or dirt may prevent the pins from making proper contact, especially when all components expand slightly from heat. Sometimes a chip will really fail, and it must be replaced, but try a chip massage first. If you are not able to or are uncomfortable with the idea of disassembling your computer, contact an authorized Zenith dealer or service center for assistance.

Wild hardware interrupt! (ERROR:)
Wild interrupt! (ERROR:)

The computer has received an invalid "command" (called an interrupt) that cannot be processed. Usually caused by running software that was not designed to run on your computer, but may be caused by a software problem. Contact the dealer from whom you purchased the software for advice because it is doubtful that you will be able to run the program on your computer.

Z-319 Card not present! (ERROR:)

The Z-319 card was available for the Z-150 series computers and was used

Cont'd. on Page 45

QUIKDATA - 12 YEARS OF H/Z SUPPORT!

For all your H/Z 8-bit and PC/XT/AT needs

ACCELERATE YOUR PC!

From Sota Technologies, Inc., the fastest and most proven way to breath new life in your Heath/Zenith PC/XT computer, giving it -AT compatible speeds! Turn your turtle into a -286 rabbit with a 12.5 Mhz 80286 accelerator board. Complete with 16K on-board CACHE RAM for dramatic speed increases.

There are other ways to speed up your H/Z PC/XT computer, but you spend too much and get too little. The **286i** is the effective solution, making your H/Z150/160/150/158/159 series of computers, or any general PC/XT computer faster, in many cases, than a standard IBM AT type computer! Simple half-card plug-in installation with switchable speed control. **You won't believe your stop watch!**
EXP-12 - $359
EXP386 - $459 Much faster 16Mhz 80386 version

MEMORY UPGRADES

Note: All memory upgrades come without memory chips. Call for current chip pricing.

Z150MP - $19 Will allow you to upgrade your H/Z150/160 to up to 704K on the main memory board, using up to 18 256K DRAM chips.

MEGARAM - $43 Upgrades your H/Z150160 series with up to 704K of main memory, and about 512K for RAMDRIVE memory. Includes documentation, software RAMDRIVE disk, PAL and jumper wire. For the full 1.2 megs total memory, 45 256K DRAM chips are required.

ZMF100 - $53 Will allow you to upgrade your H/Z110/120 (old motherboards; with p/n less than 181-4918) to 768K system RAM. Requires 27 256K DRAM chips.

Z100MP - $76 Similar to ZMF100 above, but for new motherboards with p/n 181-4918 or greater.

WINCHESTER UPGRADE KITS

PCW20 - $289 Complete winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes 21 meg formatted half-height Segate ST-225 65ms drive, WD WX1 winnie controller board, cable set, documentation.

PCW30 - $349 32 meg with 38ms Segate ST-138.
PCW40 - $409 42 meg with 37ms Segate ST-251.
PCW80 - $649 80 meg with Segate ST-4096 full size drive.

EWIN20 - $525 Complete 21 meg winchester package for the Z100. Includes S100 host adapter card, WD winchester controller, Segate ST-225 winchester drive, cables, software and documentation.

BARE WINCHESTER DRIVES

ST-225 - $225 65ms half height bare winchester with 21 megabyte storage capacity.

ST-251 - $325 37ms autopark half height bare winchester with 42 megabyte storage capacity. Good choice for -AT compatibles.
ST-251-1 - $349 Same as ST-251 but 28ms speed.

ST-4096 - $595 28ms autopark full height bare winchester with 80 megabyte storage capacity. Best choice for -AT compatibles.

ANY DRIVE IN YOUR PC/XT/AT

With the CompatiCard, you can install up to four additional drives, of any type in your PC/XT/AT computer. Add a 1.2 meg 5" floppy, or a 1.44 meg 3.5" floppy, or any other drive, including 8" to your system. The CompatiCard I will handle up to two drives, and the CompatiCard II will handle up to 4 drives. Additional cables and external enclosures may be required.
CCARD2 - $89
CCARD - $109

DOES YOUR COMPUTER KNOW THE TIME?

We carry the SMART CLOCK for any PC/XT computer, including the Z100 series. Simply plug it in the system ROM socket, and plug the ROM into the clock module. Self contained sealed lithium battery for 5-10 year life. Includes software for the clock/calendar.
SMARTCLK - $35 (add $2 for Z100 installation for spacers)

ZENITH COMPUTER SAMPLE

Zenith AT STARTER. Start with this basic unit and build your own. Includes 12Mhz Z248 (80286 processor) with two serial ports, one parallel port, 1MB RAM, 5" 1.2 meg floppy, no video card. If you desire we can install a 42mb 28ms hard disk for an additional $345, or a 80MB 28ms hard disk for an additional $589. VGA card for $189, EGA card for $119.
Z24812S1 - $1595 (starter system)

Zenith Z159 8Mhz, 640K RAM, 101 keyboard, EGA video, dual floppies.
Z159-S1 - $1295 ($1549 for single floppy 21mb hard disk unit)

Zenith MinisPort notebook computer with 1MB RAM - $1395

Call or write for a quote on any Zenith computer system.

OTHER STUFF

Quikdata also carries spike protection filters, backup power supplies, modems, printers, disk drives, drive enclosures, cables and connectors, laptop batteries, monitors, memory cards, memory chips and ICs, joysticks, accessory cards, serial and bus mouse, the complete Zenith line of computers, parts and accessories, a wide variety of useful and most popular software and much more!

8-BIT

We carry a full line of hardware and software products for the H8/H89/90 computers. This includes diskettes, printers, H89 working boards and parts, some H8 parts, etc. Call or write to obtain a catalog of all our 8-bit goodies. Here's a sample.

H37 SOFT SECTOR CONTROLLER	$169
H8 SOFT SECTOR/WINNIE CONTROLLER PACKAGE	$249
TM100-4R Tandon DS DT 96 TPI refurb disk drive	$89
SIEMENS DRIVES (refurbs) for H8/H89	$49
Z47 REMEX DRIVES slaves and masters - special	$89

H-SCOOP

Of course, don't forget about the only independent source of Heath/Zenith related information you can obtain - our monthly newsletter, **H-SCOOP!** Just $24 for a 12-issue year ($28 Canada, $35 foreign), it will help you get the most from your computer investment. Get sound technical advice, helpful hints, find out what the problems are, fixes, reports, reviews, information from other subscribers, classifieds and much more.

Call or write in to place your order, inquire about any products, or request a free no obligation catalog. VISA and Master Card accepted, pick up 2% S&H. We also ship UPS COD and accept purchase orders to rated firms (add 5% to all items for POs). All orders under $100 add $4 S&H. Phone hours: 9AM-4:30PM Mon-Thu, 9AM-3PM Friday. Visit our bulletin board: (414) 452-4345

QUIKDATA, INC.

2618 PENN CIRCLE
SHEBOYGAN, WI 53081
(414) 452-4172

Reader Service #130

TK SOLVER PLUS

Part 2

Edwin G. Wiggins
13 Clare Drive
E. Northport, NY 11731

Introduction

In the April 1989 issue of REMark I described an equation solver software package called TK Solver Plus. That article covers the basic features of TK Plus including tables, graphs, unit conversions, and numeric formats. This second article covers advanced features including: conditional rules, built-in functions, user defined functions, programming constructs, advanced print formatting, and interchanging files with other software packages.

Following these, the TK Solver Plus model for the solution of a complicated problem is presented. Although many of you will not be familiar with the problem itself, it will demonstrate just how powerful TK is.

Conditional Rules

TK Solver Plus includes an IF... THEN...ELSE capability that is similar to the one in BASIC. For example:

```
IF x<5 THEN y=3x ELSE y=2x
```

will yield 12 for y when x=4 and 12 again when x=6. The test condition (x<5 in the example above) can involve the following equalities and inequalities:

<	less than
<=	less than or equal to
=	equal to
>=	greater than or equal to
>	greater than
<>	not equal to

Built-in Functions

TK Solver Plus has a rich selection of built-in functions. The standard trigonometric functions: sine, cosine and tangent appear in two different forms. One form works with an angle in radians, and the other with an angle in degrees.

There's a square root function and also a general root function. The latter takes the form ROOT(Y,3), which will compute the cube root of the variable Y.

Exponential and logarithmic (both natural and base 10) functions are provided. So are hyperbolic sine, cosine and tangent.

Even some rather exotic functions are provided. These include the Gaussian error function and the gamma function. Although Bessel functions are not listed in the reference manual, these functions are actually available. The syntax is BESSEL(A,B,C) where A is the kind of Bessel function (1 or 2), B is the order of the Bessel function, and C is the value at which the function is to be evaluated. Bessel functions may not be important to most readers, but to a few they are very important.

The DOT function can be very helpful. If you're familiar with vectors, think of this function as taking the dot product of two lists. For readers not familiar with vectors, DOT can still be useful. It multiplies the first element of the first list by the first element of the second; does the same with succeeding elements; and then adds everything together. For example, if L1 is a list composed of the following elements: 1,2,3 and L2 is a list composed of 10,11,12, then DOT('L1,'L2) represents the following calculation:

```
1*10+2*11+3*12
```

Note the single quotes in the command syntax. This indicates that L1 and L2 are names of lists rather than variables with unique values.

The Boolean functions AND, OR, and NOT are also available. These come in very handy in combination with IF... THEN...ELSE structures. Suppose y=x+10 between x=5 and x=9, and y=0 if x is less than 5 or more than 9. The appropriate rule would be:

```
IF   AND( x>=5 , x<=9 )   THEN y=x+10
   ELSE y=0
```

A function named GIVEN provides a nice way to set default values of variables. Calculations involving pressure are most often done at 14.7 pounds per square inch, but they are also done at other pressures. In order to save a few keystrokes, the function:

```
GIVEN('p,p,14.7)
```

tells TK to use the value of p typed by the user if there is one. If the user has not typed a value, TK will automatically assign the value 14.7.

Polar-to-rectangular and rectangular-to-polar coordinate transformations are provided. There are two versions of each

```
==================== LIST FUNCTION: LIST1 ======================

Comment:              LIST FUNCTION EXAMPLE
Domain List:          X
Mapping:              Table
Range List:           Y
Element-- Domain------------- Range---------------
1         1                   'SMITH
2         2                   'JONES
3         3                   'ADAMS
4         4                   'JOHNSON
5         5                   'WIGGINS
6         6                   'ALLEN
7         7                   'EAST
8         8                   'WEST
9         9                   'NORTH
10        10                  'SOUTH



==================== VARIABLE SHEET ==============================

St Input---- Name--- Output--- Unit----- Comment ----------------
             y       'SOUTH
   10        x



==================== RULE SHEET ==================================

S Rule ------------------------------------------------------------
  y=LIST1(x)
```

Figure 1
List Function Example

transformation. One uses polar angles in radians, the other polar angles in degrees. RTOPD(1,1) (rectangular coordinates x=1, y=1) yields (1.414, 45) (polar coordinates radius=1.414, angle=45°).

User Defined Functions

User defined functions are a very powerful feature of TK Solver Plus. These functions come in three types: list functions, rule functions, and procedure functions.

A simple list function is illustrated in Figure 1. At the top of this figure we see the List Function Subsheet for the function named LIST1. In the middle we see the Variable Sheet for the problem, and the Rule Sheet appears at the bottom of the figure. The list function itself relates numbers to various words. The number 10, for instance is related to the word 'SOUTH. Note on the Variable Sheet the variable x has been assigned the input value 10. The Rule Sheet shows the rule:

```
y=LIST1(x)
```

This means that the value of y is given by whatever is matched in the function LIST1 with the number assigned to x. The number assigned to x on the Variable Sheet is 10. The word 'SOUTH is matched with the number 10. Thus the "value" of y shown on the Variable Sheet is 'SOUTH.

In this example, the Domain list is composed of numbers and the Range list is composed of words. Actually, either list may be of either type. It wouldn't make much sense in this case to assign the value 1.8 to the variable x. Note near the top of the List Function Subsheet it says:

```
Mapping:      Table
```

When mapping is set to table, TK will not attempt to evaluate the function unless x is an exact match for an entry in the Domain list.

If, however, both lists were numbers, we might want TK to interpolate in the table. In this case, we can set the mapping to linear, for linear interpolation, or cubic, for cubic interpolation.

The use of a rule function is illustrated in Figure 2. The Rule Function Subsheet shown at the top of this figure contains the rule:

```
T=R*S-5
```

In the heading of this subsheet R and S are declared as argument variables, and T is declared as a result variable. This means that values for R and S are passed (in that order) to the rule function and a value T is passed back. The names R, S and T are valid only within the Rule Function Subsheet. The Rule Sheet at the bottom of the figure shows:

```
a=RULE1(b,c)
```

This means that the values of variables b and c (in that order) should be passed to the rule function RULE1, and the value that the rule function passes back should be assigned to the variable a. In this example, b=2 and c=5 are passed to RULE1. When they arrive, the values 2 and 5 are assigned to the local variables R and S, respectively. RULE1 carries out its calculation (2*5-5) and passes the value 5 back. This value appears in the output column of the Variable Sheet in the row belonging to variable a.

If we assign values to the variables a and b, the rule function will still work. It will pass the value of a as T in the Rule Function Subsheet. Similarly, it will pass the value of b as R. The rule function will calculate a value for S and pass it back as c on the Variable Sheet.

The heading of the Rule Function Subsheet also contains a row for Parameter Variables. In our example, this row is blank. Parameter variables are variables that are common between the Rule Sheet and the Rule Function Subsheet. If you are familiar with subroutines, all this looks

```
==================== RULE FUNCTION: RULE1 ======================

Comment:              RULE FUNCTION EXAMPLE
Parameter Variables:
Argument Variables:   R,S
Result Variables:     T
S Rule ------------------------------------------------------------
  T=R*S-5



==================== VARIABLE SHEET ==============================

St Input---- Name--- Output--- Unit----- Comment ----------------
             a       5
   2         b
   5         c



==================== RULE SHEET ==================================

S Rule ------------------------------------------------------------
  a=RULE1(b,c)
```

Figure 2
Rule Function Example

```
==================== PROCEDURE FUNCTION: PROC1 ==================

Comment: PROCEDURE FUNCTION EXAMPLE
Parameter Variables:
Input Variables:       R
Output Variables:
S Statement-------------------------------------------------------

  FOR I=1 TO 10
      'A[I]=R^I
      IF 'A[I] > 1E6 THEN RETURN
  NEXT I



==================== VARIABLE SHEET ============================

St Input---- Name--- Output--- Unit----- Comment-----------------

   10         a



==================== RULE SHEET ================================

S Rule------------------------------------------------------------

  CALL PROC1(a)



==================== LIST: A ===================================

Comment:
Numeric Format:
Display Unit:
Calculation Unit:

Element-- Value---------------
1         10
2         100
3         1000
4         10000
5         100000
6         1000000
7         1E7
```

Figure 3
Procedure Function Example

very familiar.

Rule functions and procedure functions (discussed below) are very handy when the same equation is to be solved numerous times with different values of the input variables. This is illustrated later.

Figure 3 illustrates the use of a procedure function. A procedure function is really a program. The rules (equations) in a procedure function are really assignment statements like the equations in BASIC or FORTRAN. A single variable name must appear alone to the left of the equals sign. This fact is emphasized by the use of the := operator instead of the usual equals sign.

One of the main things I like about TK Solver Plus is that I can solve problems without writing a program. I just tell TK what equations to solve and let it decide how to proceed. Sometimes tighter control over the flow of execution is needed. In a procedure function, the statements are executed in the order written. FOR... NEXT loops and GOTO statements can be used to alter the flow of execution.

Values of all input variables must be passed to a procedure function, and a value of the output variable will be passed back. It is not possible to specify a value for the output variable and get back a value for one of the input variables. Note how this differs from a rule function.

The procedure function shown in Figure 3 fills a list named 'A. It begins with a FOR statement just like BASIC. The syntax:

```
'A[I]=R^I
```

means that list element A[I] will take on a value given by the variable R raised to the power I. Note that the value for R is passed to the function as an input variable.

In order to prevent numeric overflow, the next statement in the function:

```
IF 'A[I] > 1E6 THEN RETURN
```

will terminate the function if it computes a value greater than 1,000,000. The statement:

```
NEXT I
```

increases the value of I by one and returns control to the beginning of the loop. This process continues until the value of A[10], the tenth element in the list, has been calculated, or execution is terminated by the IF statement.

The Rule Sheet in Figure 3 contains the statement:

```
CALL PROC1(a)
```

This statement invokes the procedure function and passes the value of the variable a to it. In our example, a is assigned the value 10 on the Variable Sheet. The CALL statement passes this value to the procedure function, which assigns the name R to this value. Note that there are no output variables in our procedure function. This means nothing is passed back to the Variable Sheet. Of course there is some output, but the values of 'A[I] go directly into the list 'A. They do not appear on the Variable Sheet.

At the bottom of Figure 3 we can see the list 'A. It contains the values of 101, 102, 103, 104, 105, 106 and 107. That's only seven values, not 10. The function was terminated by the IF statement after it calculated the first value greater than 106. This kind of execution control is possible only in procedure functions.

Note that our procedure function was invoked by a CALL statement rather than a statement like:

```
y=PROC1(a)
```

Actually a procedure function can be invoked either way, and so can a rule function.

Advanced Print Formatting

We saw how to print tables and graphs in the first article. Any sheet can be printed by entering the command /P, responding to the prompt by highlighting what is to be printed, and pressing ENTER. In this way, all or part of a sheet can be printed.

TK prints to the printer by default. You can create an ASCII print file instead. Go to the Global Sheet (type =G) and move to the line that says "Printer Device r Filename." The default entry here is PRN or LPT1. They mean the same thing. To create a print file, change the entry to a valid filename such as REMARK. As long as this Global Sheet setting remains in place, all print commands will be directed to a file named REMARK.PRF. The first print operation will create the file. Subsequent print operations will overwrite the old file. If you will be saving several sheets to print files, you can tell TK to append instead. Just enter the filename REMARK+ on the Global Sheet.

Print enhancements such as compressed print, bold facing and underlining can be made when printing to the printer (but not when printing to file). The Global

```
==================== VARIABLE SHEET ============================
St Input---- Name--- Output--- Unit----- Comment----------------
                                         Psychrometric Chart
                                         by J.A. Harbach and
                                            E.G. Wiggins
                                                 5/28/87

          Pb        14.7       psia      Barometric Pressure
                                            [default 14.7]
  2       POINTS                         # points on chart (1 or
                                            2) [default 1]

                                         *** INLET CONDITIONS

  95      DB1                  F         Dry Bulb Temperature
  90      WB1                  F         Wet Bulb Temperature
          DP1       88.73      F         Dew Point Temperature
          RH1       82.28      %         Relative Humidity
          W1        .03002     lb/lb     Humidity Ratio
          Pdb1      .8224      psia      Partial Pressure @ Dry
                                            Bulb Temp.
          Pdp1      .6767      psia      Partial Pressure @ Dew
                                            Point
          h1        55.92      BTU/lb    Total Enthalpy
          V1        14.66      ft3/lb    Specific Volume

                                         *** OUTLET CONDITIONS

  75      DB2                  F         Dry Bulb Temperature
  60      WB2                  F         Wet Bulb Temperature
          DP2       49.51      F         Dew Point Temperature
          RH2       40.73      %         Relative Humidity
          W2        .007566    lb/lb     Humidity Ratio
          Pdb2      .4338      psia      Partial Pressure @ Dry
                                            Bulb Temp.
          Pdp2      .1767      psia      Partial Pressure @ Dew
                                            Point
          h2        26.28      BTU/lb    Total Enthalpy
          V2        13.65      ft3/lb    Specific Volume

                                         *** PROCESS VARIABLES

1,000     CFM                            Air Flow
          delDB     -20        F         Change in DB Temperature
          delW      -.02245    lb/lb     Change in Humidity Ratio
          delh      -29.64     BTU/lb    Change in Enthalpy
          Mdot      68.21      lb/min    Mass Flow
          Qt        -2,022     BTU/min   Total Heat
          Qs        -345.8     BTU/min   Sensible Heat
          Ql        -1,676     BTU/min   Latent Heat
```

Figure 4
Variable Sheet for Humid Air

Sheet contains a line labelled "Printer Setup String." The syntax of printer setup strings is not like Lotus 1-2-3. If you remember printer setup strings in VisiCalc, you're all set. Otherwise, consult the reference manual for the string you need. Some of the commonly used codes for Epson and IBM printers are:

^HOF	Compressed mode on
^EE	Emphasized mode on
^EA^H18^E2	Double spacing on

Compressed mode can be especially useful for printing rules that are longer than the screen can display. To do this, enter the compressed mode code and also change the printed page width setting on the Global Sheet to something like 132.

Interchanging Files

In addition to its native files, TK can read and write .WKS, .WK1 and .DIF files. When TK reads a .WKS or .WK1 file, it creates a list from the contents of each column of the spreadsheet. Equations in the spreadsheet are ignored. When TK writes a .WKS or .WK1 file, only the lists in the current TK model are saved.

This file interchange capability can be handy if you set up a problem on a spreadsheet and then find the need for the greater power of TK. You can transfer all the data (numbers) to TK and set up a TK model by typing the necessary equations on the Rule Sheet.

A Complicated Problem

An air conditioner takes in warm, humid air. It cools the air and may condense some of the moisture from it. The analysis of this problem involves calculation of dry bulb temperature, wet bulb temperature, dew point and relative humidity. It is important to know how much heat the air conditioner must remove in order to achieve the desired temperature and humidity.

Complicated equations have been developed that relate the various quantities involved in the calculation. These equations must be solved twice: once for the warm, humid air and once for the air leaving the air conditioner.

A TK model has been developed to solve this problem. The Variable Sheet for this model is shown in Figure 4. The inlet dry bulb temperature is 95°F, the inlet wet bulb temperature is 90°F, the outlet dry bulb temperature should be 75°F and the outlet wet bulb temperature should be 60°F. Air flows through the air conditioner at a rate of 1000 cubic feet per minute. All these values appear in the input column of Figure 4.

Although we elected to give values for wet and dry bulb temperatures at the inlet and outlet of the air conditioner, we could have given values for any two variables at these locations. For instance, we could have given values for dry bulb temperature and relative humidity instead of dry bulb temperature and wet bulb temperature.

TK has calculated that the total heat that must be removed is 2022 BTU per minute. This value appears in the third line from the bottom of the figure. The negative sign indicates heat is being removed from the air. Most of the heat removed, 1,676 BTU per minute, is associated with condensing moisture. Only 346.8 BTU per minute is associated with reducing the air's temperature. The original relative humidity was 82.28%. Leaving the air conditioner, the relative humidity is 40.73%.

Figure 5 shows the Rule Sheet for this model. Note that the first two equations make use of the GIVEN function. The first equation sets a default value for the barometric pressure (Pb) of 14.7 pounds per square inch. No input value for Pb was entered on the Variable Sheet, so TK used this value. The second equation sets a default value for the number of places that properties are to be calculated. The default value is 1. Since we want to calculate properties at two places (inlet and outlet), the value 2 has been entered on the Variable Sheet for the variable POINTS.

All of the equations that calculate property values are contained in rule functions. Thus, the INLET CONDITIONS portion of the Rule Sheet shows nine CALL statements. For instance, the statement:

```
CALL RH1(Pdp1,Pdb1;RH1)
```

invokes the rule function RH1 (relative humidity at location 1), passes values of Pdp1 (pressure at the dew point temperature at location 1) and Pdb1 (pressure at dry bulb temperature at location 1), and gets back a value for the variable RH1. A semicolon separates argument variables from result variables. Note that the rule function RH1 and the variable RH1 are re-

```
=================== RULE SHEET ================================

S Rule

   Pb=GIVEN('Pb,Pb,14.7)
   POINTS=GIVEN('POINTS,POINTS,1)

   "          INLET CONDITIONS

   CALL PRESSURE(DB1;Pdb1)
   CALL RH1(Pdp1,Pdb1;RH1)
   CALL W(Pdp1,Pb;W1)
   CALL AorB(WB1;A1)
   CALL AorB(DB1;B1)
   CALL H(DB1,W1;h1)
   CALL RH2(DB1,WB1,A1,B1,Pb;RH1)
   CALL V(DB1,Pdp1,Pb;V1)
   CALL PRESSURE(DP1;Pdp1)

   "          OUTLET CONDITIONS

   IF POINTS=2 THEN CALL PRESSURE(DB2;Pdb2)
   IF POINTS=2 THEN CALL RH1(Pdp2,Pdb2;RH2)
   IF POINTS=2 THEN CALL W(Pdp2,Pb;W2)
   IF POINTS=2 THEN CALL AorB(WB2;A2)
   IF POINTS=2 THEN CALL AorB(DB2;B2)
   IF POINTS=2 THEN CALL H(DB2,W2;h2)
   IF POINTS=2 THEN CALL RH2(DB2,WB2,A2,B2,Pb;RH2)
   IF POINTS=2 THEN CALL V(DB2,Pdp2,Pb;V2)
   IF POINTS=2 THEN CALL PRESSURE(DP2;Pdp2)
*  IF POINTS=1 THEN WB2=0
*  IF POINTS=1 THEN DP2=0

   "          PROCESS VARIABLES

   delDB=DB2-DB1
   delW=W2-W1
   delh=h2-h1
   Mdot=CFM/V1
   Qt=Mdot*delh
   Qs=Mdot*(.24+W1*.45)*delDB
   Ql=Qt-Qs
```

Figure 5
Rule Sheet for Humid Air

lated but distinct.

The same rule function, PRESSURE, is used twice in the INLET CONDITIONS area of the Rule Sheet. Both the first and last CALL statements in this area invoke the PRESSURE function, but different argument and result variables are involved. The first CALL passes the dry bulb temperature, DB1, and gets back the dry bulb pressure Pdb1. The second call passes the dew point temperature, DP1, and gets back the dew point pressure, Pdp1.

All of the rules in the OUTLET CONDITIONS area of the Rule Sheet involve IF...THEN constructs. Only the last two rules will be executed if the value of the variable POINTS is 1. If POINTS equals 2, the first nine rules will be executed.

The last three rules in the PROCESS VARIABLES area of the Rule Sheet calculate the heat that must be removed by the unit. There are seven rule functions in this model. Figure 6 shows the Function Sheet, which serves as an index of these functions. Each individual rule function has its own Rule Function Subsheet. These resemble Figure 2 except that the actual equations are much more complex.

The Global Sheet for this model is shown in Figure 7. Note the places for turning page breaks and page numbering on or off. Page length and width can also be specified. "Printer device or Filename:" has been set to A:\PERSONAL\REMF4+. This means that all print operations will be directed to a file named REMF4 in the PERSONAL subdirectory of the disk in drive A:. The + sign at the end of the file name means that print operations will be appended to what's already in the file. "Printer Setup String:" has been set to ^EE. As long as print operations are being directed to file, this has no effect. If "Printer Device or Filename:" is changed to LPT1 (the printer), this setup string will set the printer to emphasized print mode.

```
=================== FUNCTION SHEET ============================
Name---------- Type----- Arguments-- Comment-------------------
PRESSURE       Rule      1;1
AorB           Rule      1;1
H              Rule      2;1
RH1            Rule      2;1
RH2            Rule      5;1
W              Rule      2;1
V              Rule      3;1
```

Figure 6
Function Sheet for Humid Air

This model illustrates rule functions, CALL statements, the GIVEN function, and IF...THEN structures that were discussed earlier.

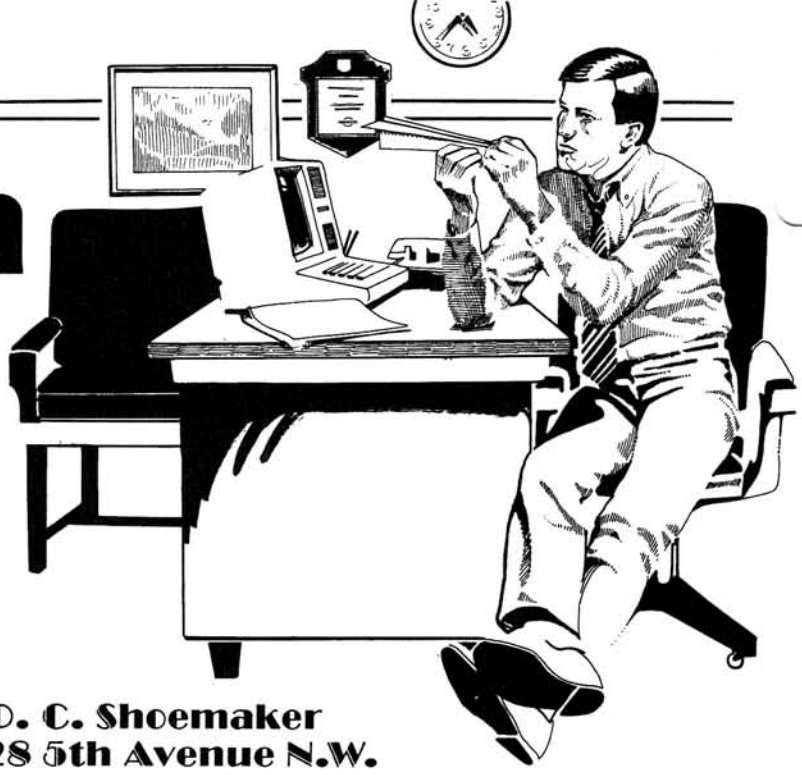
Conclusion

My first article described basic TK Solver Plus features. Those features may be sufficient for solving most problems. This article has described a number of advanced features of the software. As an engineer, I find this program to be a very powerful problem solving tool. Thanks to TK, I seldom need to resort to a programming language. Problems can be solved much more easily with TK, because it takes care of the flow of execution. I need only specify the equations to be solved.

The amount of time and effort required to learn the software is comparable to what's required to learn Lotus 1-2-3, but for most engineering problems TK is far more useful.

Cont'd. on Page 45

Problem Solving in the Office



D. C. Shoemaker
5828 5th Avenue N.W.
Seattle, WA 98107

With DOS

Problem solving with DOS? For a great many microcomputer users, DOS is a part of the problem, not a part of the solution. I'd like to change that perception with three of the less-used, but most useful, DOS commands. My thrust is toward the business user and the office environment, where the problems are often the greatest and the help the most lacking. Most hackers will know lots of other neat things you could do with these commands, but frankly, most of us in the office will never need them.

Most of today's high-powered professional software gives you all the help you need to set up your hard disk, install the software, customize the printer and screen, and do useful work. But there are some exceptions. The time will come when you will need to do something that your application software will not easily support, or perhaps will not do at all. You may need to turn to your disk operating system, DOS.

For most of us, DOS is just the environment you need to make the computer work, and the application software (Lotus 1-2-3, dBASE III, Microsoft Word, and so on) are the real problem-solvers. Unfortunately, they can also be a part of the problem, not a part of the solution.

Many of you may know that I teach microcomputer software applications here in the Pacific Northwest, and I'd like to share a few of the DOS tricks that my business-oriented students have found useful. It's partly a case of "If the shoe fits..." but you never can tell when a bit of DOS background might come in handy.

A classic example of using DOS as a problem-solver is the limitation of dBASE III. This most widely used of all data bases is older than DOS. First written to run under CP/M, it was designed to be able to use up to five disk drives as storage for data bases, program files, indexes and the dBASE program itself. Moving from one disk to another was easy under program control, and if you had a large dBASE data base, you probably used several disk drives. When DOS arrived from Microsoft via IBM, dBASE users immediately took advantage of the hard disk capabilities to store vast data files. Unfortunately, dBASE's ability to recognize disk drives, not disk subdirectories, caused great problems. It wasn't readily apparent how to keep your dBASE program in one directory, the data bases in another, and other related files elsewhere.

The SUBST Command

The folks at Microsoft foresaw this problem, and beginning with DOS Version 2 provided the ASSIGN command. In the current version of DOS, Version 3, this command is split into two separate commands, SUBST and JOIN. Since that's the version you're most likely to have, that's the one we'll discuss.

SUBST allows the dBASE III user to substitute an unused disk drive letter (A: to E:) for any other disk's subdirectory, and make DOS use that subdirectory whenever the appropriate drive designation is used. For instance, let's assume that you have a typical Zenith PC with one floppy disk, Drive A:, and one hard disk, Drive C:. In the DOS scheme of things, you have three unused drives, B:, D: and E:. The dBASE program itself can use only five disk drives, so the fact that you could use up to 26 drives with DOS Version 3 doesn't figure in our discussion.

Let's further assume that you would like to organize your hard disk files in such a way that one collection of dBASE data base files is stored on Drive C: in a subdirectory called C:\DBASEDAT. You may have others; the following procedures will work for any set of directories.

Since there's no Drive B: in your computer, you could use the SUBST command in the following way:

```
SUBST B: C:\DBASEDAT
```

From now on, whenever you direct some disk operation to Drive B:, DOS will perform that operation with the subdirectory C:\DBASEDAT. How can you take advantage of this? Use the dBASE command SET DEFAULT TO B:. This command tells dBASE to use Drive B: and DOS tells the computer to use the appropriate subdirectory. When you no longer want DOS to make the substitution, use the DOS command SUBST B:/D where the /D means disengage.

The SUBST command could be included in your AUTOEXEC.BAT file so that the substitution always takes place at boot time, but that's rather crude, and

means that if you want to undo the substitution or make another one, you'll have to type in another command. Here's an alternative. Create a batch file called DBASE.BAT that contains the following commands:

```
CD \DBASE
SUBST B: C:\DBASEDAT
DBASE
CD \
SUBST B:/D
```

The first line changes the directory to the one that contains dBASE III. The second line makes the substitution to allow you to use the data base stored in C:\DBASEDAT. Next, the batch file starts the dBASE III program running. The batch file then "goes to sleep" and waits for you to complete whatever dBASE tasks you have, then returns you to the root directory of Drive C:. Finally, the last line disengages the substitution. This last command isn't really needed unless you want to make another substitution later in the operating session. There's no limit to the number of substitutions you can make, but remember that dBASE III can only recognize five drives.

Some business systems have more drives than our hypothetical computer. A full-house system might have a 360K Drive A:, a 1.2 meg Drive B: and a 60 meg hard disk partitioned into three twenty meg "drives," C:, D: and E:. Can we still work our DOS magic to allow dBASE III to use subdirectories? Yes, with a catch. The catch is that while in most systems you will be able to use the SUBST command to make the initial substitution, you may not be able to disengage with the /D option. It depends on the built-in ROM BIOS, the version of DOS, the height of the tide, the phase of the moon, and possibly other imponderables. In the worst case, you may have to reboot the system to disengage. For many of you, this will be a small price to pay.

Incidentally, the earlier DOS versions of WordStar, such as Version 3.X, all share the same problem with dBASE III, and for exactly the same reasons. Later versions of WordStar, and the newly released dBASE IV, do away with these old-time restrictions. Nevertheless, there are hundreds of thousands of folks still using the older software, and if you're one of them, then this may be for you.

As for the JOIN command, the other half of the ASSIGN command, that does the opposite. It allows you to join a subdirectory designation to a drive designation. You could use this command to temporarily refer to a subdirectory and force DOS to use a specified disk drive. There may be a good use for this, but I haven't found one in three years of teaching.

The MODE Command

Have you ever looked at the MODE command and wondered what in the world it's supposed to do? It's another appendix-like program that dates back to the old days when an IBM PC with one floppy disk and 128K of RAM might cost $3,000. A color video monitor would run as high as $750, and that was too steep for lots of people in 1982, when that was Real Money. To save on the initial investment, IBM worked up a video system that let you use your color television set as a monitor, but the TV wouldn't display 80 characters across the screen. It did relatively better with 40 characters (shades of Apple and Radio Shack) so Microsoft worked another bit of code into DOS and gave us the MODE program. This allowed you to change from the normal 80 character video output to 40 characters, and back again. It did a lot of other video-related magic such as changing from color to monochrome, and centering displays.

Most of us no longer use television sets as monitors, but the MODE command did something else. It allowed you to send a few special printer codes to the "IBM-compatible" parallel-port dot-matrix printers. Since Epson made all the original dot-matrix printers that IBM sold, the codes that MODE generates for the printer will work with almost any modern-day dot-matrix printer, and a great many others, including various laser printers.

The MODE command allows you to configure the right kind of printer to six or eight lines per inch, and to 80 or 132 characters per line. Since this can be made to happen regardless of any other printer control program you may have, it might be useful to add to your DOS toolkit. The commands are as follows:

MODE LPT1: ,6 — Sets the printer on LPT1: to 6 lines/inch.

MODE LPT1: ,8 — Same as above, for 8 lines/inch.

MODE LPT1: 80 — Sets the printer to 80 character/line.

MODE LPT1: 132 — Same as above, for 132 characters/line.

In the examples above, it's assumed that your parallel dot-matrix printer is plugged into printer port LPT1:, the first parallel port. Most computers give you the option of three parallel ports, LPT1:, LPT2: and LPT3:, and you can use these commands on any port by substituting the right number. Also, note the commas. Some versions of DOS will recognize that you're talking about six or eight lines per inch without the comma (the number is too small to be anything else.)

More than one option can be set at once by combining commands. In fact, that's the only way you can do it. If you type MODE LPT1: 8 and then type MODE LPT1: 132, you may well reset the eight lines per inch setting to the standard default of six lines per inch. So do it this way:

```
MODE LPT1: 132,8
```

Any settings you make will remain in effect until you reset them with another command, or the program you're running resets them internally, or you turn off the printer power. To undo any settings without powering off, you could use the command MODE LPT1: and your printer would be reset to 80 characters per line and six lines per inch. If you must continually set the printer's characteristics, you could create a batch file, or add the appropriate lines to your AUTOEXEC.BAT file.

The XCOPY Command

No, XCOPY isn't some experimental program Bill Gates or Gordon Letwin slipped into your DOS files. It provides several file copying functions the standard COPY command won't do. The main feature of XCOPY is that it's subdirectory oriented. You can use it to copy the contents of an entire subdirectory from one disk to another. For example, if you keep business contract files in a subdirectory called C:\CONTRACT, you can use XCOPY to copy the entire set of files to Drive A:, creating the appropriate new subdirectory "on the fly." The command would work like this:

```
XCOPY C:\CONTRACT A:\CONTRACT
```

DOS would ask you if the destination was to be a file or a directory, and as soon as you type the appropriate letter, XCOPY makes the new destination directory, reads all the source files, and copies them to the new directory.

What could you do with that? It's certainly one way to make a backup copy of your files. It has the advantage over other backup methods in that the copies are immediately readable and usable. If you use the DOS BACKUP command to do the same thing, the backup files created can be used for only one purpose, to RESTORE from. They cannot be read directly. So if you've used BACKUP to back up your contract files, you'll have to RESTORE them to your disk before you can use them again. With XCOPY, the backup disk can be transferred to any other PC and any one of them can be read immediately, without the RESTORE process.

As usual, there's a batch file approach that can be used to partially automate the XCOPY command. To continue with our example, suppose you want to periodically make a complete backup of your contract files. You would prefer, perhaps, not waiting for DOS to ask if your new destination is to be a file or a directory, so we can use the DOS ECHO command in our batch file, like this:

```
ECHO D ¦ XCOPY C:\CONTRACT A:\CONTRACT
```

This small batch file could be called BACK.BAT, and whenever you call it up by name, it will automatically perform the XCOPY. The ECHO D ¦ part of the batch file places a "D" in the command buffer, just waiting for DOS to ask the question. As soon as it does, the "D" is ready to respond, and XCOPY proceeds to make the destination a directory.

The ECHO feature can be used in other situations, where it might be desirable from a security standpoint to remove a set of files, or even a complete directory from a disk after it's been backed up to a floppy for safe storage. You might do something like this:

```
ECHO Y ¦ DEL C:\CONTRACT
```

DOS will proceed immediately to do the erasing, and not wait for you to answer the "Are you sure?" question. Needless to say, a bit of care should be used here to insure the right files are deleted. Rolling it all together, the completed batch file for backing up those sensitive contracts might look like this:

```
C:
CD \
PAUSE Place a disk in Drive A:
ECHO D ¦ XCOPY C:\CONTRACT A:\CONTRACT
ECHO Y ¦ DEL C:\CONTRACT
RD \CONTRACT
```

The first line insures that you're on Drive C:, and the second line insures that you're at the root directory. Both are cheap insurance that your batch file is going to do its work where you expect it to. Remember, it's automatic, and will cheerfully erase the wrong files if you set it going in the wrong place. The PAUSE command will prompt you for the floppy disk in Drive A:, just to be sure everything's ready. The next two lines we already know about, but the last one removes the directory, too. Now there's no obvious record that the contracts were ever on Drive C:.

I say "obvious" because you're probably aware that there are tools which would allow someone to recover the erased directory and files. They're not a part of DOS, but come with software packages like the Norton Utilities, PC Tools or the Mace Utilities. But that's another story.

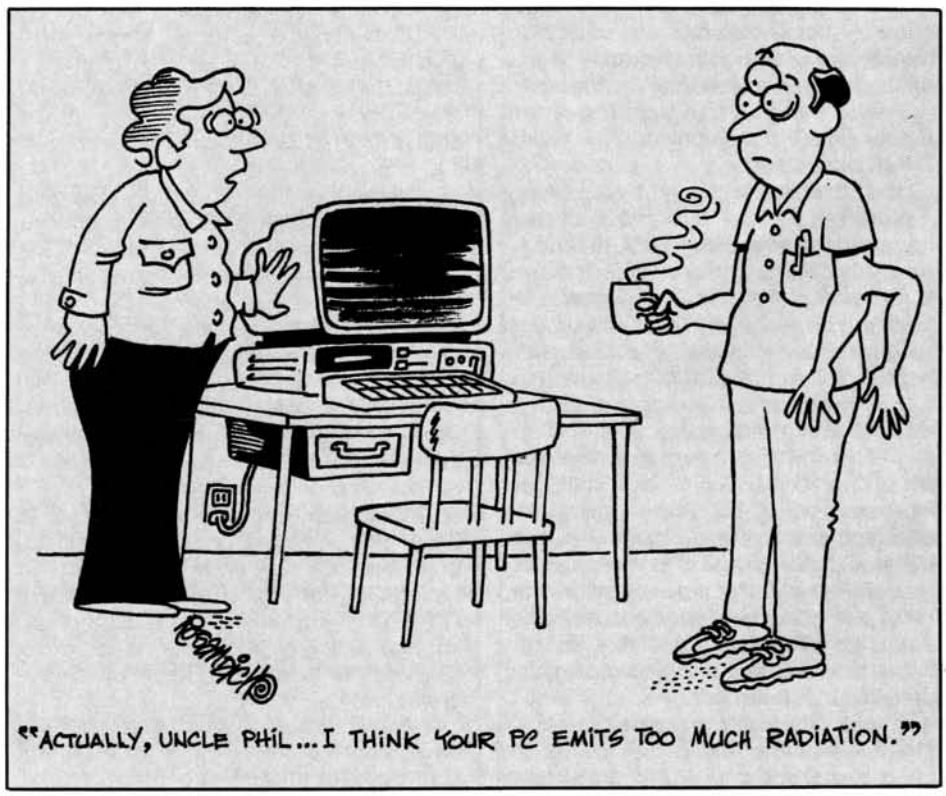
Organizing for Business

First, note that all of the commands we've looked at above are classed as external commands. That means that it's a program file that must reside in your DOS subdirectory, which is probably on your boot disk. All those dozens of DOS files need to be somewhere, and if you routinely boot your computer from Drive C: that's the best place. Make a subdirectory called C:\DOS, and copy all the files from your DOS distribution disks into it. Then, to be sure that DOS can find those external commands when you need them, add a line to your AUTOEXEC.BAT file like this:

```
PATH = C:\DOS
```

There may, of course, be other things in your AUTOEXEC.BAT file, but that's another story, too.

To recap, then, we've looked at three seldom-used, but very useful, DOS external commands, SUBST, MODE and XCOPY. Depending on the software you use in your business, you may never need them. On the other hand, one of them might just do the trick and solve a nagging problem that spoils your concentration and fun. And I always tell my students that if you're not having fun, you're probably not doing it right. *



Z-100

Paul F. Herman
3620 Amazon Drive
New Port Richey, FL 34655

SURVIVAL KIT

ZPC Revisited — By Popular Demand

In the very first installment of this column I stated that I didn't want to get too deeply involved in describing ZPC patch information. But that was before the letters began coming in. My mail has been running about 10 to 1 in favor of continued support for ZPC patch information for new PC programs.

There are several factors at work here. First of all, Pat Swayne's ZPC Update column has virtually disappeared from the pages of REMark. For several years, Pat kept us abreast of the patches required to run the latest PC software under ZPC. Occassionally, another ZPC update article filters down to us, but patch information for new programs, and new releases of old programs, are beginning to get away from us. Many of you are continuing to discover how to run your PC programs under ZPC, but there isn't a public forum for exchange of that information.

Even though Pat has been too busy to continue the ZPC Update tradition, he has silently been at work trying to keep track of information people send to him. He has periodically added new patch information to the PATCHER.DAT file on the ZPC Upgrade Disk (HUG #885-3042-37), and has tried to keep track of bug reports and fixes to the ZPC program itself, that users have sent to him. I too, have received quite a bit of patch information for running PC software under ZPC.

By popular demand, I have talked with Pat Swayne and offered to carry the torch for ZPC patch information into the indefinite future. But now the question is this; how should all this information be collected, organized, and distributed to the Z-100 community?

Must We Choose Sides?

At this point, I can't help but put in my two cents worth about the subject at hand. It seems like whenever I mention running ZPC on the Z-100, I get a lot of angry letters pro and con, so I might as well stir things up real good. My associations with Z-100 users lead me to believe that there are three distinct attitudes that prevail regarding the use of ZPC.

First, there are those who spend every spare moment in the trivial pursuit of PC compatibility. Every public domain, shareware, or commercial PC program they can get their hands on presents a new challenge, whether the program has any useful features or not. These people are only interested in the "conquest". Once they figure out how to run the program under ZPC, they will probably lose all interest. There is some hacker spirit here. This isn't a quest for practical use of the Z-100; it is a hobby. We have these people to thank for much of the ZPC patch information that has already been compiled.

At the other end of the scale are the Z-100 purists. These are the people who insist that PC emulation represents a degradation of their Z-100. Even though they admit that ZPC is a dandy program, it is for others to use, not them. Most of the purists believe that a Z-100 can do anything better than an IBM-PC can. Not coincidentally, this group consists mostly of programmers who are able to make the Z-100 do what they need it to do.

The mainstream Z-100 user is somewhere in between these two extremist groups. He uses ZPC when he needs to, but prefers to use native Z-100 software when available. This is obviously the most well-rounded approach.

I would consider myself to be somewhere near the top of "mainstream", and occasionally slipping into the "purist" camp. Actually, my "purist" affiliation is not so much because I dislike PC emulation. It has more to do with the fact that I have an AT clone sitting next to my Z-100. So if I want to run PC software I just run it on the PC. It seems silly to go to the trouble of emulating it on the Z-100, under the circumstances.

But I realize that many of you don't have a second computer (perhaps because your spouse would raise an eyebrow). And I certainly wouldn't want to suggest that you should out-and-out trade your Z-100 for a lowly PC compatible clone. That would probably be a mistake, unless you can afford to go for an 80286 or 80386 machine. So that brings us back to the subject at hand — ZPC emulation of your favorite PC software.

ZPC Patch Information — Some Problems

As most of you know, the key to success in using ZPC, many times depends on developing patches for the PC software. Sure, some programs will run straight up under ZPC, but many will not. The problem of patching software to run under ZPC is something that is unavoidable. Some things just can't be emulated with software, and therefore, the program must be changed.

The HUG User's Manual that comes with the ZPC Emulator is an excellent source of information about how ZPC works, and about the type of problems encountered in PC software which need to be patched. I expect that many of the

questions Pat and I receive regarding how to make patches to programs could be avoided if everyone would simply read the manual!

Being an ingenious lot, many Z-100 owners have read the instructions, and have mastered the techniques of applying patches to programs so they can run under ZPC. Information about many of the patches you have developed have been sent to HUG, and to me . . . elaborately described in lengthy letters, or scrawled on the back of an envelope.

But there are several problems with the present scheme (or lack thereof) for submitting patch information. The biggest problem is trying to make sure that the file that is being patched is, in fact, the proper version of the program. Even though the present PATCHER program reports "Your Program was successfully patched", it really doesn't have any way of knowing whether the patches were applied correctly.

Sometimes, hints about using the program under ZPC are not available along with the patch information. For example, the program may need to be run in PC mode 7, as well as be patched. It would also be nice to have some information about the logic of the patches which were developed, to help other users find patches to future versions.

And lastly, it would be nice to have a list of the software that didn't require patches, along with any special instructions for its use. A data file containing patch information is nice to have, but software titles not included in the patch list leave a question in the user's mind. Does it work okay without patches, or is it just that patches have not yet been developed?

Let's Get Organized

What I have in mind is a single all-encompassing data file which would contain all the information you might need to know about patching programs for ZPC. Something similar to the present PATCHER.DAT, but bigger and better (the new file will be named PATCHIT.DAT). This file will be accompanied by a new utility, named PATCHIT.EXE, which will be used to patch programs, and provide information about their use under ZPC. I am beginning development on PATCHIT.EXE now (I'm writing this in August), and will be compiling all the patch information I have at present. By the time this column is printed, everything should be ready to go.

I am depending on your support to make this project a success for all of us. Continue to send in information about patches you have developed, or programs you have found to work without patching. The new PATCHIT program has the capability to do patch verification, and give user information about the patched program. But in order to provide this type of support to ZPC users, patch developers (that's you!) will need to provide more complete information when patches are submitted.

```
------- New Entry Field ------------------------------------------------
*
        Any line beginning with an asterisk '*' signifies the start of a
        new PATCHIT.DAT File entry.

------- Program Name Field ---------------------------------------------
N textstring
        where "textstring" is the name of the program.

------- Requirements Field ---------------------------------------------
R [N][P][Z][C]
        where N,P,Z,C are single letters indicating any requirements for
        proper operation of the program.  More than one letter requirement
        may be listed, except when requirement N is shown.  The letters
        indicate the following requirements;
                N = none, program works without patches or ZHS support.
                P = program requires software patches.
                Z = program requires ZHS circuitry for proper operation.
                C = program requires PC style COM port for proper operation.
        Note that when it is questionable as to whether the ZHS support
        circuitry is required, the Z requirement will be shown until proven
        unnecessary.  Users who run the programs on a ZHS equipped Z-100 may
        not be able to tell if the program would run on an unmodified system,
        therefore they should show requirement Z if any doubt exists.


If the Requirements Field 'R' contains requirement P, the following fields
must be present;

------- Filename Field -------------------------------------------------
F filename
        where "filename" is the name of the file to be patched.  If more than
        one file needs to be patched for this application, each file to be
        patched must be declared with an F field line.

------- Patch Field ----------------------------------------------------
& hexaddress s1,s2,s3,...sn > d1,d2,d3,...dn
        where;
                hexaddress      = a hexadecimal address of up to five digits.
                                  This is the file offset where the bytes to
                                  be patched are located.
                s1,s2,s3,...sn  = two digit hexadecimal values of the original
                                  bytes to be patched.  PATCHIT uses these
                                  values as an integrity check, and to
                                  determine if the file has already been
                                  patched.  These values are optional.  If
                                  original byte values are not known, the
                                  entire s1...sn series may be replaced by a
                                  single question mark '?'.
                d1,d2,d3,...dn  = two digit hexadecimal values of the patches
                                  to be applied at the specified offset.

        Each '&' field line specifies a series of contiguous bytes to be
        patched.  If more than one area of the file must be patched (which
        is likely), multiple '&' field lines must be used.  In other words,
        each 'F' field line may be followed by as many '&' field lines as
        are necessary to provide the required patch information.
```

Figure 1
Required PATCHIT.DAT Fields

Following is a detailed description of the format which will be used in the PATCHIT.DAT file. Your program patch submissions may be provided in this format, or you may simply submit the required information, and I will make up the file entry.

PATCHIT.DAT File Format

Each program entry in the PATCHIT .DAT file will have several fields describing the program or file, and giving information about any patches that may be required. Each field will be described on a separate line of the file (in other words, the fields are separated by a CRLF). Some of the fields are required fields, and must be present in order for PATCHIT.EXE to patch a program. Other fields are optional, and may be present to give information about the program or file, and to help insure that patches are applied correctly.

Each field is designated by a single character in the first column position of a line, followed by a space. Lines are limited to 80 characters in length.

Required fields for each PATCHIT .DAT file entry are shown in Figure 1. Notice that a valid PATCHIT.DAT file entry could be composed of only the informa-

```
------- Publisher Field -----------------------------------------------------
P textstring
        where "textstring" is the name of the program's author or publisher

------- Version Field -------------------------------------------------------
V textstring
        where "textstring" provides version or release information about
        the program.

------- Summary Field -------------------------------------------------------
S textstring
        where "textstring" is a brief description of the program.

------- Instruction Field ---------------------------------------------------
I textstring
        where "textstring" provides instructions or other information which
        should be displayed on the screen after patches have been successfully
        applied to the file(s).

------- Contributor Field ---------------------------------------------------
C textstring
        where "textstring" is the name of the person, company, etc.
        responsible for providing information about this entry in the
        PATCHIT.DAT File.
```

Figure 2
Optional Fields for Each Program Entry

```
------- File Date Field -----------------------------------------------------
D filedate
        where "filedate" is the date of the file to be patched.  This is the
        DOS file creation date shown by a disk directory listing.  The date
        should be in the form mm/dd/yy or mm-dd-yy.

------- Check Field ---------------------------------------------------------
? hexaddress s1,s2,s3,...sn
        where "hexaddress" is a hexadecimal address of up to five digits
        which gives an offset into the file.  The series s1...sn are
        contiguous byte values which should match the bytes at the specified
        offset in the source file.  PATCHIT uses the values given in the
        Check Field to help determine if this is the correct file to patch.
        Note that the patch field 'P' also allows for byte verification.
        The check field is provided for more thorough checking, where
        necessary.

------- Comment Field -------------------------------------------------------
' textstring
        where "textstring" may be any comment.  PATCHIT ignores comment
        fields.  They are provided so that patch logic may be documented.
        PATCHIT will also ignore any line which is totally blank, or begins
        with a white space character (space or tab).  Therefore, if you
        prefer, comments in the PATCHIT.DAT File may begin with a space or
        tab character.
```

Figure 3
Optional Fields for Each File to be Patched

tion available in a present PATCHER.DAT file (that is, file name and bytes to patch). And, in fact, most of the entries in our first PATCHIT.DAT file will have only basic information, since the extended information will not have been provided. So we'll need more information about many existing entries, as well as information about new programs. Hopefully, many of you who have provided patch information in the past will help fill in the missing information for users who will need it.

The PATCHIT program has the capability of using optional information, if it is provided. The fields described in Figure 2 are optional, and if included, should immediately follow the program name 'N' field. The fields shown in Figure 3 are optional, and if included, should immediately follow the Filename 'F' field for each file to be patched.

Figure 4 gives a brief summary of all the different PATCHIT.DAT fields, and shows which ones are required. The reason that only a few fields are required is so that the existing data base of ZPC patches (from HUG's PATCHER.DAT) could be utilized. When providing new submissions, the information should be as complete as possible.

Figure 5 shows a couple of examples of how the PATCHIT.DAT fields can be used to provide information for a program entry.

Suggestions for Contributors

When you contribute information about a program patch (or a program that does not require patches), provide as much information as possible. After all, most of the information requested by the PATCHIT.DAT fields will be readily available to you while you are researching the patches. Of particular importance is the version of the program, the file date of any files that need patching, and the original values of bytes which are patched. PATCHIT.EXE will use this information to verify that a correct patch has been applied.

If no original values for patched bytes are provided, PATCHIT.EXE will simply display the message "Program has been patched". But if the PATCHIT.DAT file contains original values (in the Patch field '&' or Check field '?'), it may report "Program has been patched successfully".

You may be as verbose as you like with your comments. I will edit them, if necessary, to keep the PATCHIT.DAT file size reasonable.

Some programs may warrant more than one entry in the PATCHIT.DAT file. For example, if the program will run without patches with the ZHS circuit, but requires patches to run on a plain Z-100, then a separate entry should be made for each case. One entry would not contain any patch information, but would inform the user (use the 'R' field) that the ZHS circuit was required. The other entry would show that the ZHS circuit was not required, and include the appropriate patch information.

Spreading the News

Now that we have a specification for submitting ZPC patch information, and a

```
* New entry                                                          Required
N Program Name                                                       Required
P Publisher's Name
S Summary of program
V Version
I Instructions
R Requirements  (N,P,Z,C)                                            Required
C Contributor's name

For each file to be patched (if any)...

F Filename to patch                                                  Required
D Filedate
? Check         (address s1,s2,s3,...sn)
& Patch         (address s1,s2,s3,...sn > d1,d2,d3,...dn)            Required
' Comment
```

Figure 4
PATCHIT.DAT Field Summary

```
**********************************************************************
N Norton Utilities, Standard Edition
P Peter Norton Software
V 4.5
S MS-DOS disk utilities
R N
' The line above indicates that no patches or ZHS circuit is required

**********************************************************************
N LogiCADD
P Generic Software
V 2.0
S This is the Logitech version of Generic CADD
R P
I In order to use a mouse with LogiCADD, a PC style COM port will be required.
C Paul F. Herman Inc.

F CADD.EXE
D 10/02/86
   Change all instances of ... MOV BX, F000 ...to... MOV BX, B000
& DCCB F0 > B0
& DD35 ? > B0
& DD72 ? > B0
& DDDB ? > B0
   Change all instances of ... MOV SI, FA6E ...to... MOV SI, 0
& DE2C 6E,FA > 0,0
& DE94 ? > 0,0
& DF0F ? > 0,0
```

Figure 5
PATCHIT.DAT Program Entry Samples

volunteer (who, me?) to collect, organize, and distribute it, the only remaining question is how to make it available to other Z-100 users. Since I must support a family and run a business, as well as write this column, I have to be able to cover my expenses of handling and distributing the PATCHIT disk. If you want a copy of the latest version of the PATCHIT disk (which includes the PATCHIT.EXE program and the PATCHIT.DAT file) send me your request along with a check for $10.00. No phone orders or VISA/MC orders will be accepted. The contents of the disk may be considered to be in the public domain for anyone to use, so feel free to give it to anyone you please, or upload it to your favorite bulletin board.

As most of you know, Z-100 specific articles are getting harder and harder to find, so I'm not really willing to squander valuable column space describing lengthy patches to programs so they will run under ZPC. If I published patch information in this column, there just wouldn't be much space for more interesting stuff. So the preferred method of distributing patch information for now will be on disk, as described above. From time to time, I may break down and do a column about new programs that we've been able to get running under ZPC, or other interesting news on the ZPC scene.

The Future for ZPC

Although some will skoff at the idea, I dare say that Pat Swayne's ZPC is one of the most important pieces of software that has ever been written for the Z-100. It is a prestigious piece of work which takes software PC emulation way beyond the point most people thought was possible. I wish I had written it.

What does the future hold for ZPC? I tread lightly in this area, because when all the cards are played, the fact is that the future of ZPC is for Pat Swayne to determine. It's been several years since the last ZPC Upgrade disk, and in the mean time, lots of changes to the ZPC program itself have been put forth by users. Pat acknowledges these modifications, and agrees that some of them offer substantial improvements in PC emulation, or program operation.

Some of the modifications have been published in REMark and SEXTANT magazines. And many are available on the Heath Bulletin board for all to explore. Will another ZPC upgrade be forthcoming? We'll just have to wait and see. *

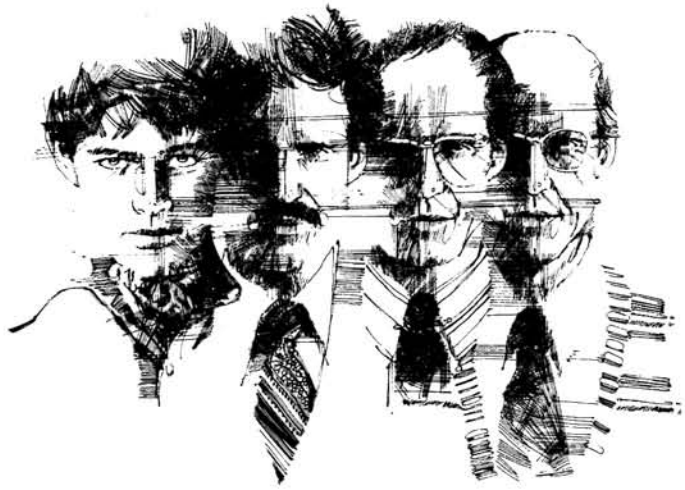
HUG Discount List

CB-5063-50	10%	TM-158	10%	ZA-181-19	10%
DC-1000	10%	TM-159	10%	ZA-181-24	10%
HA-2860-1	10%	TM-170	10%	ZA-3034-CS	10%
HA-2860-2	10%	TM-180	10%	ZA-3034-EB	10%
HA-2860-4	10%	TM-240	10%	ZA-3034-ME	10%
HS-40A	10%	TM-380	10%	ZA-3034-NP	20%
HS-42	10%	TM-2300	10%	ZCA-2300-EF	10%
HS-386C	10%	TM-2500	10%	ZCA-2300-MG	10%
HS-2526A	10%	TM-3034	10%	ZCM-1390Z	20%
HS-2860	10%	TMP-200	10%	ZCM-1400-1	10%
HS-3629	10%	Z-207-7	10%	ZCM-1490Z	20%
HS-3860	10%	Z-416-2	10%	ZD-12	10%
HS-3860-M	10%	Z-416C	10%	ZD-14	10%
HV-2000	10%	Z-445	10%	ZD-800	10%
HVB-550	10%	Z-449	10%	ZKB-2	20%
HWD-420	10%	Z-505	10%	ZMM-149A	20%
HWD-440	10%	Z-515	10%	ZMM-149P	20%
HWD-4028	10%	Z-516	10%	ZMM-1470G	20%
PMK-121	10%	Z-525	10%	ZUS-386	10%
PMK-130-H	10%	Z-549	10%	ZVM-1200-1	10%
SK-203-H	10%	Z-605-1	10%	ZVM-1240	20%
SK-209	10%	ZA-180-65	10%	ZVM-1300-1	10%
SK-211	10%	ZA-180-66	10%	ZVM-1330	20%
SWA-4300	20%	ZA-180-67	10%	ZVM-1380C	20%
TM-140	10%	ZA-180-83	10%		
TM-150	10%	ZA-181-7	10%		

FOR ALL ASSEMBLED COMPUTERS, CALL FOR PRICE. *

The Evolution of DOS

Robert C. Brenner
9282 Samantha Court
San Diego, CA 92129



The history of DOS mirrors the history of the personal computer. DOS is a reflection of a technology revolution that began just a few years ago and that has forever changed the way people work, live and play. This evolutionary leap in electronics began in 1975 with the Altair 8800 and skyrocketed with the introduction of the Apple Computer in 1977 and the IBM Personal Computer in 1981. Hardware improvements made it possible, but software improvements made it happen.

Disk operating systems were born when Apple introduced the Apple II computer with its disk drive interface. DOS is the special program that controls all disk operations. By the time the IBM PC was born, Apple DOS was in its third version. DOS (as we in the MS/PC-DOS world think of it) was born in 1981 with the introduction of the IBM PC.

DOS (as we in the MS/PC-DOS world think of it) was born in 1981 with the introduction of the IBM PC.

In the pre-DOS days, if we didn't have an Apple II, we depended on crude system monitors and wrote custom drivers to interface the computer with the keyboard, displays and disk drives.The development of DOS gave us a set of file management and utility programs that interface application programs to the Basic Input/Output System (BIOS) and physical hardware. DOS works with BIOS to handle mundane computer housekeeping chores and to optimize the interface between the microcomputer hardware and user programs.

But DOS goes beyond BIOS by providing file management tools for interfacing with the floppy and hard disk mass storage units installed in your system. DOS lets you format disks, copy, erase, and rename files, generate a directory, and a myriad other functions related to the mass storage interface. During the boot process, BIOS downloads DOS from a hard disk or floppy disk into your computer's random access memory (RAM). Together, BIOS and DOS oversee system operation and direct the flow of information between keyboard, screen, memory, and mass storage.

In the early days, microcomputers were called "appliances" and used mainly to play games and serve as program development workstations. When Dan Fylstra, Dan Bricklin and Robert Frankston introduced "VisiCalc," they transformed the hobbyist computer toy into a small business personal computer causing even the sleeping giant, IBM, to sit up and take notice.

Gary Kildall of Digital Research in Monterey, California developed an operating system called CP/M (control processor/microcomputer) for the Apple computer, and CP/M quickly became the standard operating system against which all other 8-bit computers were judged.

Then word leaked out that IBM was planning to enter the small computer arena with a product of its own. This information hit the industry like a whirlwind. Up to that time, the fledgling personal computer industry had been growing steadily in the home and education marketplaces, but it was not getting serious attention in the business market. Suddenly, the business world looked at the microcomputer with renewed interest. IBM's involvement suggested important value in those small desktop computers with their shiny keyboards, illuminated screens, and (even better) low investment cost.

As the world entered the decade of the eighties and word circulated that IBM was working on its own personal computer, "experts" speculated that IBM would choose the popular CP/M operating system software from Digital Research for its new entry. However, in the hush of innovation, Seattle Computer Products (SCP) was quietly working on a new operating system for 8086-based computers. Tim Paterson, the author of 86-DOS wrote the software while an employee of SCP. SCP began shipping 86-DOS with their hardware in August 1981, a month after Tim Paterson was hired by Microsoft. Microsoft bought all rights to 86-DOS just before IBM announced the IBM PC in August 1981. Minor changes were made to 86-DOS and it was released by IBM as PC-DOS. Microsoft retained 86-DOS and began licensing their own version called MS-DOS (Microsoft Disk Operating System). IBM licensed PC-DOS (Personal Computer Disk Operating System) while the rest of the PC-compatible world adopted MS-DOS as "the" standard operating system. Each company then customized their own version of MS-DOS (e.g., Zenith MS-DOS, DOS 3.3 Plus, etc.). The

relationship between SCP and Microsoft resulted in a $60 million lawsuit filed by SCP against Microsoft. But DOS was a reality and the world was ready.

The introduction of the IBM Personal Computer made history. Backed by the solid reputation of IBM, the microcomputer toy became the "personal computer" and PCs were quickly accepted by corporate America. Even home users and hobbyists took notice, and purchased. Overnight, IBM gave credibility to the microcomputer and established a major standard against which all other computers would be measured. The term "IBM compatible" became synonymous with quality and capability. To be IBM compatible meant MS-DOS/PC-DOS compatible. Within a few short years, the industry moved beyond "compatibility" to accept (and expect) IBM compatibility as a standard feature. Over five hundred new businesses were formed to support the PC, and IBM clones surfaced everywhere. The IBM PC was the catalyst that released a torrent of pent-up creativity and innovation, irrevocably establishing the PC standard.

DOS 3.0 provided support for the 1.2 Mbyte high-capacity floppy disk drives.

In the late 70s and early 80s non-IBM-compatible computers came and went like the seasons. Names like Osborne, Attache, PET, and Rainbow passed on into history, but IBM and IBM-compatible PC sales continued to surge. From a base of about 1.5 million computers in 1981, today over sixty million personal computers are in use worldwide.

As new hardware technology appeared, the MS/PC-DOS operating system rapidly changed to exploit the added capability. The IBM PC of 1981 included PC-DOS 1.0 and supported only single-sided disk drives (160K). A few bugs in version 1.0 were corrected in DOS 1.05, but DOS 1.1 provided the first major improvements including support for double-sided disk drives (320K), faster disk access, date and time file recording and better serial communications.

In 1983, IBM abandoned the 64K RAM memory limit of the PC and introduced the PC extended technology PC/XT and the next major upgrade to PC-DOS (DOS 2.0). Microsoft began licensing MS-DOS 2.0 for PC-compatible computers. DOS 2.0 provided improved data management utilities, and added tools for partitioning hard disk drives and establishing multiple directories and subdirectories. DOS 2.0 also changed the floppy disk format to a 9-sector formatting scheme increasing the 160KB/320KB capability to 180KB/360KB per floppy disk. A DOS 2.1 upgrade added support for half-height floppy drives and for the graphics and disk access speed requirements of the short-lived PCjr.

In early 1984, the world anxiously awaited the introduction of the IBM PC AT. The AT was considered a counter to AT&T's introduction of a multiuser supermicro 32-bit computer.

As summer 1984 approached, rumors spread that Digital Research, creator of CP/M, was writing a new operating system for the PC AT. But IBM again worked with Microsoft and bundled PC-DOS 3.0 with the PC AT. Microsoft simultaneously introduced MS-DOS 3.0. DOS 3.0 provided support for the 1.2 Mbyte high-capacity floppy disk drives and large 40 Mbyte hard disk drives. It also provided the ability to create a RAM disk, to specify alternate keyboard layouts for several foreign countries, and added a data security level that let users create read-only files. It also caused new data to be stored on the oldest discarded data space thus preserving recently erased data as long as possible so users could recover deleted files.

However, DOS 3.0 did not provide multitasking and multiuser capability, so the new DOS couldn't take advantage of the Intel 80286 CPU's "protected mode." This mode allows several tasks to run simultaneously in isolated segments of memory with access to the full 16M of the 80286 address space in the PC AT. A PC version of UNIX called XENIX was produced that could access three Mbyte of memory and support three users, but compatible software was slow to come.

In early 1985, DOS 3.1 added network service capability with file sharing and data locking. This gateway gave the AT access to mainframe computers. DOS 3.1 also allowed a subdirectory to be aliased so it could appear as a separate disk drive to certain programs. However, an ability to change the environment size held some less than 100 percent compatible IBM clones at pre-DOS 3.1 operating levels.

Technology was rapidly changing, and disk operating system designers raced to keep current. MS/PC-DOS 3.2 incorporated DOS 3.1 and added support for 3.5 inch floppy disk drives. It also added logical drive support. MS/PC-DOS 3.3 replaced all prior DOS versions except DOS 2.1.

A March 1988 trade publication article reported that IBM was preparing to release DOS 3.4 to support hard disk partitions larger than 32 Mbytes and to include drivers to convert extended memory to conform to IBM's proprietary XMA memory expansion specifications. It was purported to also add mouse control. For several years PC users had read about a DOS upgrade that would incorporate multitasking and swallow IBM's TopView windowing package. It was also expected to remove the 640K memory address stumbling block.

However, the business strategy of IBM was in transition and the computer industry was about to experience a major power struggle. In January 1987, IBM cut prices on its PC AT setting the stage for the introduction of its new Personal System/2 (PS/2) microcomputer line. Reverting back to its mainframe computer mentality, IBM decided to redefine the market, abandon the existing PC hardware and software standards and establish new standards with Operating System 2 (OS/2) and the PS/2's Micro Channel Architecture (MCA) bus interface. The 32-bit PS/2 computer family represented a radical change in design. Except for the PS/2 Model 30, which lacks the Micro Channel, the PS/2 family is incompatible with the PC and AT interface structure. IBM's

PC-DOS 4.0 will only interface with IBM hardware and software.

apparent expectation that its followers should abandon their substantial investment in PC/XT/AT technology to purchase the new PS/2 products caused widespread controversy. Millions of MS/PC-DOS machines are in use worldwide and few users wish to accept the cost and expense of abandoning their MS/PC-DOS computer system investments.

IBM began a closeout process for its PC/XT/AT line, but industry observers reportedly felt that IBM had miscalculated the marketplace and its influence in the industry. Instead of quickly signing up to buy PS/2 machines, thousands of users opted to keep their PC/XT/AT and compatible personal computers in place and wait out the impending competitive storm that was rapidly building. IBM was criticized by almost everyone in the industry. MCA was not setting the world on fire although IBM bragged that more than half of the three million PS/2 computers shipped worldwide contained MCA. Competitors contended that many of the MCA-bus PS/2s were still sitting on dealer shelves.

While the PS/2 is a desirable step in personal computer evolution, it produced an era of innuendo, arrogance and challenge. IBM stock declined in price with poor PS/2 sales throughout 1987. In September 1988, a group of nine competitors led by Compaq Computer and including Zenith Data Systems, shocked the world by announcing an intent to develop machines designed to their own Extended Industry Standard Architecture (EISA) standard interface, instead of adopting the Micro Channel interface. EISA is compatible with existing MS-DOS operating systems.

The nine primary members of EISA were shortly joined by other computer, semiconductor and software manufacturers including Intel (still partly owned by IBM) and Microsoft (IBM's partner in DOS development). Recently, Intel introduced a fully EISA-compatible chip set.

Given the cloudy future caused by the EISA/MCA controversy, PC-DOS 3.4 became vaporware and was never introduced. Instead, in December 1988, IBM introduced PC-DOS 4.0. PC-DOS 4.0 will only interface with IBM hardware and software. Requiring 1 Mbyte of storage, the five disks containing about 84 files of DOS 4.0 can only be installed on systems with hard disk drives formatted with PC-DOS (not MS-DOS). The user must define and then maintain one of three configurations: floppy disk only, minimum hard disk and maximum hard disk. DOS 4.0 also provides copy protection by recognizing a diskette's serial number, and it supports the extended memory standard (EMS). In addition, PC-DOS 4.0 won't recognize the Hercules Graphics Plus card and will only recognize an IBM mouse or IBM printer. IBM's centralizing its PC family to IBM-only peripherals is pushing many vendors and compatible suppliers to the MS-DOS-only marketplace.

Microsoft quickly introduced MS-DOS 4.0, and since December 1988, both IBM and Microsoft have corrected several DOS 4.0 bugs through their own versions of DOS 4.01.

The uncertainties and confusing realities of today are dramatically changing the computer world, but, one thing is certain: The use of existing PC/XT/AT-compatible products will continue for the foreseeable future. IBM's entry into the microcomputer market forever legitimized the personal computer as a serious business tool and produced an impact on business that has been felt around the globe. Applications that seemed impossible (at least for a desktop computer) are now commonplace.

The personal computer has become a part of our culture. And in these turbulent times, many companies prefer PCs for most business functions. As microcomputer technology advances, operating systems will improve to exploit these new capabilities. With new MS/PC-DOS upgrades, the PC/XT/AT family will be a useful part of business, education and the home for a long, long time. ✱

Cont'd. from Page 29

with a special version of MS-DOS to enable the Z-150 computers to run Z-100 software. This error message usually means that you have attempted to run this special MS-DOS version without having a Z-319 card installed in the computer.

Powering Down

Next time we will look at the special key sequences, such as CTRL-S, that are available on most DOS-based computers. In addition, you will see how and when to use some of the unique Zenith key sequences, such as CTRL-ALT-RETURN.

If you have any questions about anything in this column, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, suggestion or comment. ✱

Cont'd. from Page 35

```
==================== GLOBAL SHEET ==============================
Display Intermediate Values: Yes
Stop on List Error:          No
Use Automatic Iteration:     Yes
Comparison Tolerance:        .000001
Typical Value:               1
Maximum Iteration Count:     10

Global Numeric Format:       DEFAULT
Append Variable Names:       Yes

Use Page Breaks:             Yes
Number Pages:                Yes
Form Length:                 66
Printed Page Length:         60
Printed Page Width:          80
Left Margin:                 0
Printer Device or Filename:  A:\PERSONAL\REMF4+
Printer Setup String:        ^EE
Plot Output Filename:        TKPLOT.OUT

Use Color:                   No
Slow Redisplay:              No
Solid Line Headings:         No
Bottom Prompt Line:          No
```

Figure 7
Global Sheet for Humid Air ✱

GETTING STARTED WITH . . .

AMI

ALAN NEIBAUER
11138 HENDRIX STREET
PHILADELPHIA, PA 19116

Windows, like the Disk Operating System, is an environment in which you work with other programs. It provides a graphic interface that includes pull-down menus, icons, and mouse support — much as you'd find on the Apple Macintosh computer.

Some of the programs that you may be using are Windows "smart." This means they can take full advantage of Windows graphics and printer support. Programs such as AMI have been specifically designed to work within Windows: they get much of their power and versatility from Windows itself. So to appreciate how AMI works requires a basic understanding of the Windows environment.

Rather than have their own printer and screen drivers, programs such as AMI utilize the drivers installed in Windows. Since they must be used with Windows, they're sold with a special run-time version in case you don't have Windows already. The run-time version includes the basic interface, and screen and printer drivers needed to run the application.

Other programs can still be made to work with Windows, but they don't really use the full features that Windows has to offer.

Microsoft Windows provides a graphics-based environment that rivals OS/2. It is even capable of multi-tasking, especially if you have extended or expanded memory. Because Windows comes in many platforms, from basic PC through Windows 286 and 386, it offers a common interface that can be shared across the PC world.

One advantage of Windows, is that it can handle all I/O tasks. For example, once you install your hardware, all features are available in every Windows application. This is particularly powerful if you have a laser or other printer capable of handling softfonts. Once you install the softfonts in the Window's driver, along with corresponding screen fonts, they can be accessed in your applications.

Printer vs Screen Fonts

Screen fonts are files of binary data with the FON extension. Like softfonts, each file contains the data necessary to describe each character and symbol. But in this case, the file is used to reproduce the font on the screen, not on the printer.

Windows comes with a number of screen fonts already installed. These font files are totally independent of the printer softfonts: you can't download them to the printer and, as supplied, they have no relation to softfonts.

Two types of fonts are supplied — stroke and raster — in several typefaces, sizes and sets. Stroke fonts are used for displaying text on the screen, printing to a plotter or printer in a graphics mode. Raster fonts are device-specific. That is, the fonts used depend on the resolution of your screen and your particular printer.

Windows comes with three strokes fonts — Roman, Modern, and Script — and five raster fonts — Helv, Courier, Tms Rmn, System, and Terminal. The last two fonts are built into the Windows environment and are not separate FON files.

In order to keep track of which fonts should be used for your hardware, Windows separates the fonts into six sets:

Set 1: The stroke fonts
Set 2: Raster fonts for systems with 640 × 200 screen resolution
Set 3: Raster fonts for systems with 640 × 350 screen resolution
Set 4: Raster fonts for printers with 60 dpi resolution
Set 5: Raster fonts for printers with 120 dpi resolution
Set 6: Raster fonts for systems with 640 x 480 screen resolution

How you use these fonts depends on your application and hardware. With programs like Windows Paint you can use the system and terminal screen fonts in your drawings. When you print your creation, Windows will convert the screen fonts into bitmap data that will be printed as graphics, not as high resolution text.

Windows Write, on the other hand, won't be able to access most screen fonts at all unless you have a related printer font — either built into your dot matrix printer or available as a cartridge or softfont with a laser printer.

If you do have a laser printer, you can create screen fonts that match your printer fonts. A number of commercial programs are available for just this purpose. Additional FON files are available from public domain sources such as Compuserve and Genie, creating a truer WYSIWYG display.

By the way, you can add or delete screen font support using the Control Panel's Installation Add Font and Delete Font options.

Windows Applications

While not all PC software can take full advantage of Windows, more and more applications are becoming available that do. Aldus Pagemaker is perhaps one of the most well known Windows applications.

But recently, the AMI word processor has joined the ranks. While AMI doesn't have the full range of power as programs such as WordPerfect and Word, its link with Windows actually provides a combination desktop publisher and word processor in one inexpensive package.

While AMI is not advertised as a desktop publishing program, it makes the distinction between word processor and publisher even more unclear. Like WordPerfect and Word, it allows graphic files to be merged and printed along with your document. But through Windows, it presents a WYSIWYG environment that more powerful word processors cannot provide.

It is here where you have to consider the tradeoffs with such a package. If you write book-length manuscripts and want a comprehensive word processor, then look elsewhere. But if you write basic documents, leaflets, flyers, and other documents — and you need basic desktop publishing without the expense — then AMI might be the answer.

Like Ventura Publisher, AMI uses stylesheets and paragraph tags (called styles) to quickly format text. You can select from several style sheets supplied with AMI or create your own. As a word processor, AMI has the capabilities of controlling page size and layout, formatting paragraphs, and using the features of your Window's printer driver.

Before starting, keep in mind that I am using Windows 386 with a number of

LaserJet printer and screen fonts installed. The fonts have been installed in the Windows LaserJet printer driver, and the screen fonts added to WIN.INI, the configuration file that Windows checks every time it starts. On your own hardware, you might not have the same range of printer or screen fonts.

As an example, I'll run through the procedure for importing a file from WordPerfect, then adding headlines and graphics. Then I'll summarize some of the other features available.

While you can type and create documents within AMI, (I wrote this article using it) its capability to directly import files from other formats is a powerful feature. If you have another word processor, you can use its own features for writing and editing the document, then use AMI for final graphics formatting and printing. This is particularly useful if you have a word processor not fully Window's capable.

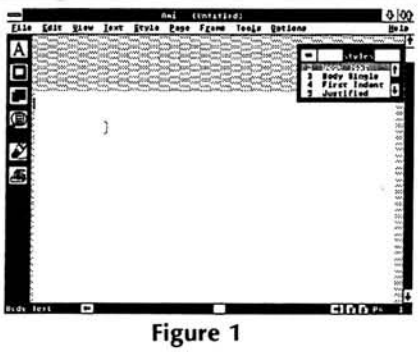


Figure 1

Figure 1 illustrates the initial AMI screen. The styles menu on the right lists paragraph formats that are part of the default stylesheet. Notice the Windows pull down menus along the top and the scroll bars at the right and bottom. The icons on the left allow for quick selection of functions without the need to first pulldown a menu.

Stylesheets

If your word processor doesn't provide stylesheets, you might not understand their usefulness. Stylesheets are used heavily in Ventura Publisher and Microsoft Word. Even WordPerfect now incorporates stylesheets in version 5.0, although I suspect not many WordPerfect users take advantage of the feature.

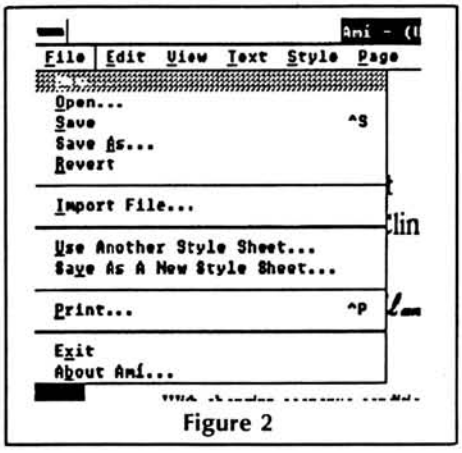


Figure 2

A stylesheet is a collection of formats stored independently of any text. AMI comes with 26 stylesheets for formatting documents from simple letters and memos, to proposals and multi-column newsletters. Something called the Default stylesheet is loaded automatically when you start AMI. However, if you wanted to create a two column newsletter, for example, you'd pull down the File menu (Figure 2) to access two options that deal with styles. Use Another Style Sheets lets you load the formats of a stylesheet into the current document. Save As A New Style Sheet allows you to save a new or edited stylesheet on the disk.

Once a stylesheet is loaded, however, you can quickly format lines, paragraphs, pages, or the entire document in a few keystrokes. As you can see in Figure 1, each of the styles is associated with a function key number. To create a "first indent" format, you merely place the cursor in the paragraph and press F4. Most stylesheets have formats for body text, indented paragraphs, and bulleted paragraphs. A small manual shipped with AMI, the Style Sheet Guide, illustrates each of the stylesheets available.

But more important than those provided, you can create your own stylesheets to conform to formats used in your office, school or household. Suppose you like a certain letter or memo style. Rather than repeat the formatting instructions with each memo or letter you type, create and save stylesheets for each. Your documents will now be consistent without having to worry about formatting each time.

This is particularly useful in office settings where standardized formats are frequently used.

Importing Files

Once you select a stylesheet, or decide to use the default one, you can enter text or import a file from another application. Selecting Import File from the File pulldown menu shows a list of formats the program recognizes. Figure 3 shows some of the formats. Not listed in the figure, however, are the Advancewrite, ASCII, DCA/FFT, Microsoft Word, and Multimate options.

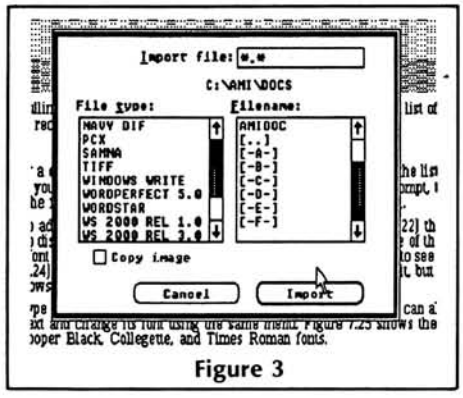


Figure 3

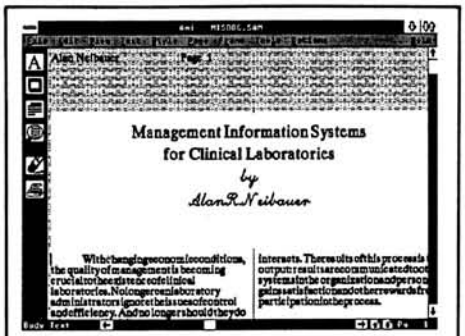


Figure 4

Click on the format of the file in the left-hand list box, enter its path at the Import File prompt, then press Return. A list of files will appear in the right-hand list box. Double-click on the file's name to start the conversion process. In a moment, the file will appear on screen using the default Courier screen font (Figure 4).

This shows a WordPerfect 5.0 imported and converted into AMI. Not only were the centered headline and indented paragraph formats maintained, but the WordPerfect header was placed in the AMI header area. The automatic page numbering was lost, but is easily added by selecting Page Numbering from the program's Page pull-down menu.

Formats such as boldface and underlining are also converted automatically, so while you have to check the document before printing, little extra work will be required.

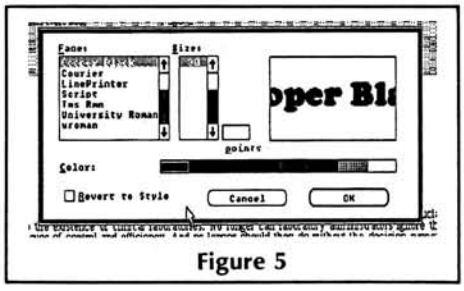


Figure 5

But now that we have the document in AMI, let's use its powers to create a professional looking document. Selecting FONT from the Text pull-down menu shows a dialog box listing the printer fonts installed in the Windows printer driver (Figure 5). The fonts are named in the left-hand list box. When you select one (the reverse selection bar doesn't appear properly in the figure), available sizes are listed in the center list box, and a sample of the font appears in the right. Proper samples require having screen fonts that match the printer fonts. In this case, we're looking at 30-point Cooper Black.

Before seeing how the document appears with fonts, look at the options for page layout in Figure 6. Using this menu you can set the margins, sizes and num-

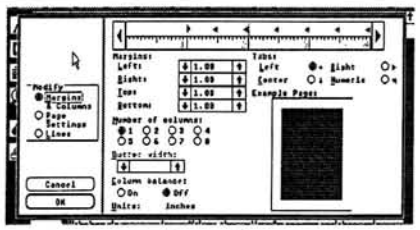


Figure 6

ber of columns, and line spacing. After doing some formatting, the imported file appears as shown in Figure 7. The title is formatted in 18-point Times and the author's name in 20-point Script. The rest of the text is 12-point Times in two columns, separated with a vertical line.

Figure 7

Notice that the text appears differently in this figure compared to Figure 4. The earlier example shows what AMI calls the Working view. All text is in the default Courier screen font, with the full line displayed on the screen. In the standard view, as in Figure 6, the appropriate screen fonts are used for all text, and the lines are spaced in proportion. Other views are Full Page, Enlarged, and Draft. In Full Page you can't read the text, but you see the general layout of the entire page.

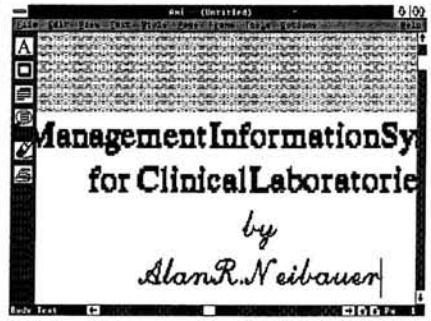


Figure 8

Enlarged display distorts certain screen fonts, as shown in Figure 8, but is useful when working with graphics. The draft mode displays text in perhaps the most distinct way, a dark san serif screen font. Carriage returns are shown by the << symbol, tabs with an arrow like ---->.

In Draft mode, all text is shown starting at the left, even centered and right aligned formats. In a way, draft mode display resembles an ASCII text file in which all formatting is removed. While it doesn't take advantage of the Windows environment, draft mode provides the quick keyboard and screen response to your editing. So it might be useful to write and edit in draft mode, then change to one of the Layout modes (full page, working, standard, or enlarged) when you want to format.

Working with Graphics and Frames

To add emphasis to the document, you can add text and graphic frames. Frames are rectangular sections of the document that can hold drawn and scanned images, or text. While the frame doesn't have to be surrounded by a border, the lines set the frame off from the rest of the text, as shown in Figure 9. Here

Figure 9

a sample TIF formatted figure has been loaded into a frame in the first column. You can also merge files in PCX format, a popular style used by many drawing and painting type programs. Once imported, the graphic can be reduced or enlarged, or automatically made to fit the frame in which it is placed.

In addition to the features mentioned here, it also includes a comprehensive spelling checker, one of the easiest to use that I've seen. Just place the cursor at the start of the document, pull down the Tools menu, and click on spelling.

AMI lacks some power user word processing features, such as floating footnotes, index and table generation. As an author specializing in word processing software, I've grown quite used to the more powerful applications on the market. But not everyone needs these powers. AMI's blend of utility with the graphic interface, make it an excellent tool for those needing an "executive word processor."

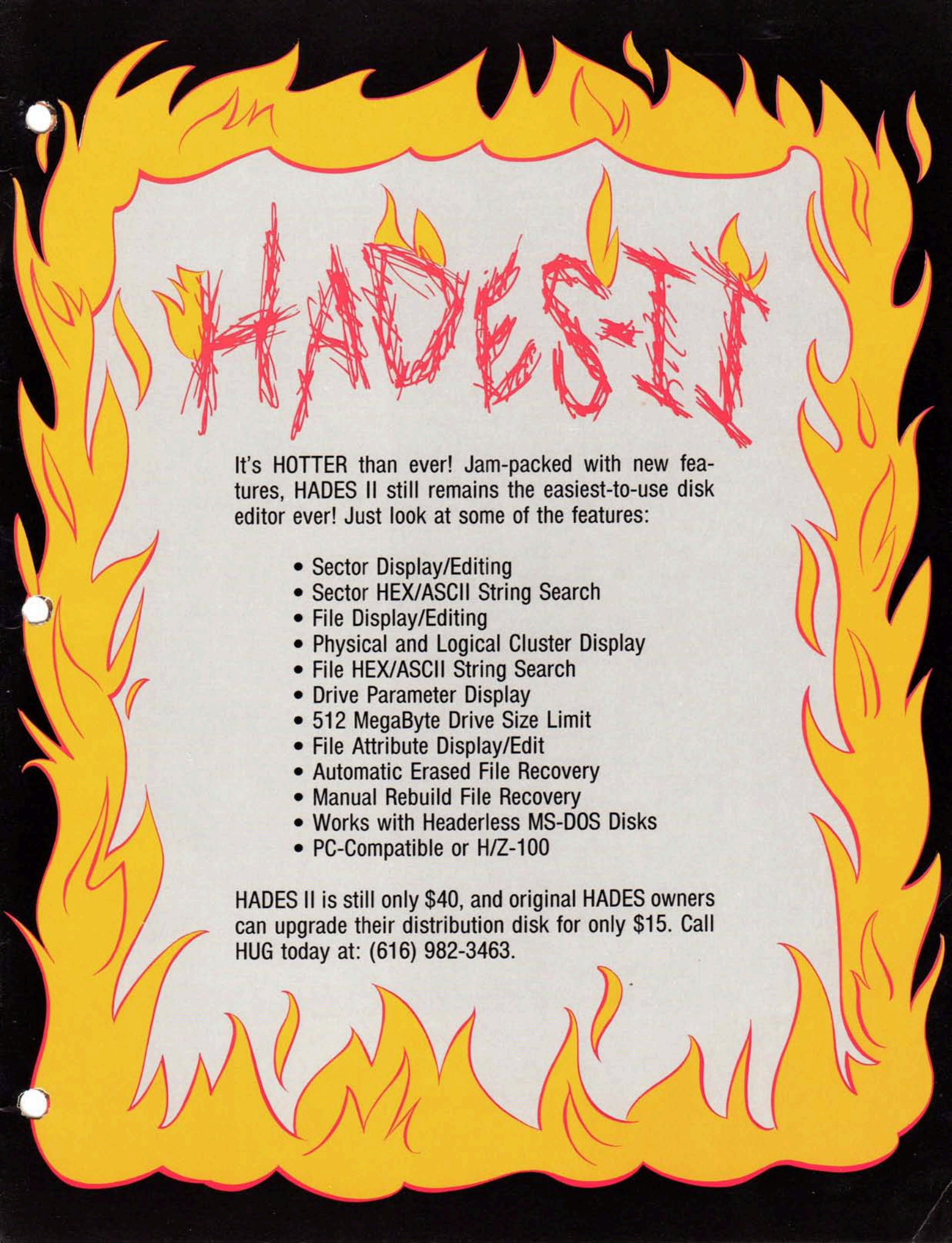
Combined with Window's screen fonts and a laser printer's softfonts, AMI offers some exciting possibilities. *

Glitches

In the November 1989 issue of REMark, Page 23, there was a statement made about calling for Parts orders at Heath Company on the toll free number. This statement is incorrect.

Heath Company Parts orders should be made using (616) 982-3571. Heath Catalog orders only should be made on the toll free number, (800) 253-0570.

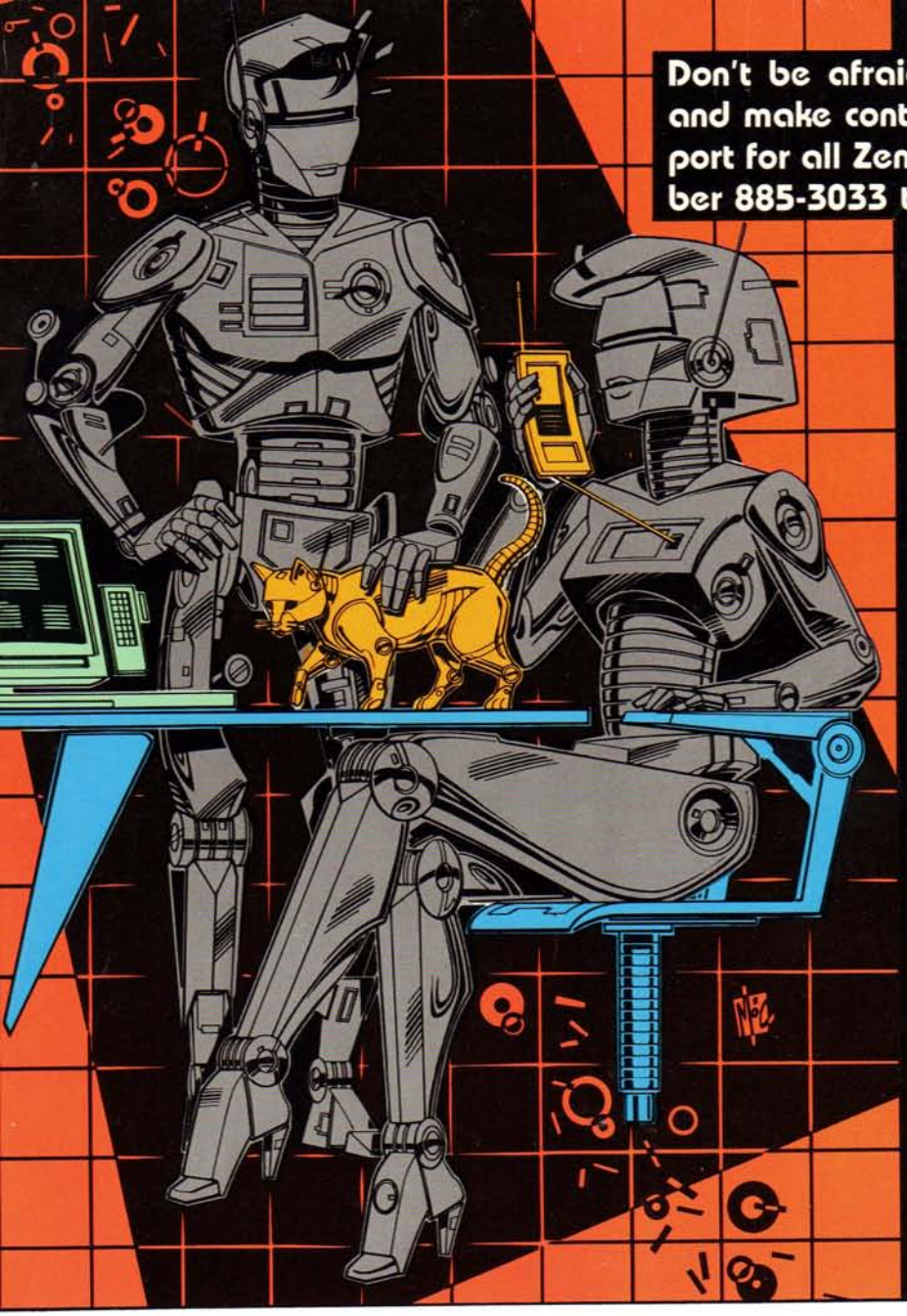
We're sorry for any inconvenience this may have caused.



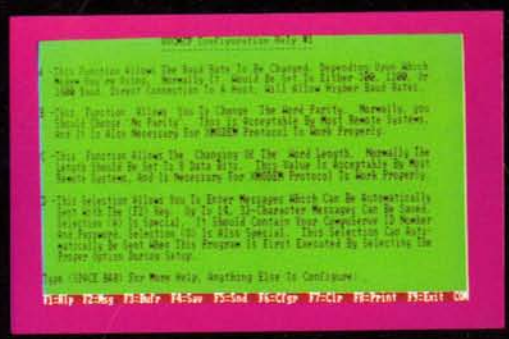
It's HOTTER than ever! Jam-packed with new features, HADES II still remains the easiest-to-use disk editor ever! Just look at some of the features:

- Sector Display/Editing
- Sector HEX/ASCII String Search
- File Display/Editing
- Physical and Logical Cluster Display
- File HEX/ASCII String Search
- Drive Parameter Display
- 512 MegaByte Drive Size Limit
- File Attribute Display/Edit
- Automatic Erased File Recovery
- Manual Rebuild File Recovery
- Works with Headerless MS-DOS Disks
- PC-Compatible or H/Z-100

HADES II is still only $40, and original HADES owners can upgrade their distribution disk for only $15. Call HUG today at: (616) 982-3463.



Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all Zenith Laptops, order HUG Part number 885-3033 today.



Heath/Zenith Users' Group

P.O. Box 217
Benton Harbor, MI 49022-0217

BULK RATE
U.S. Postage
PAID
Heath Users' Group

POSTMASTER: If undeliverable, please do not return.

$2.50
P/N 885-2119