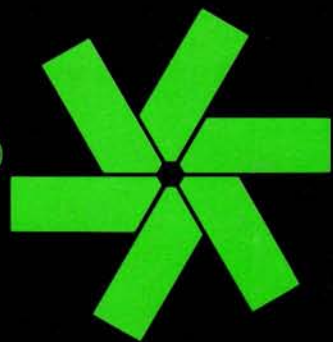
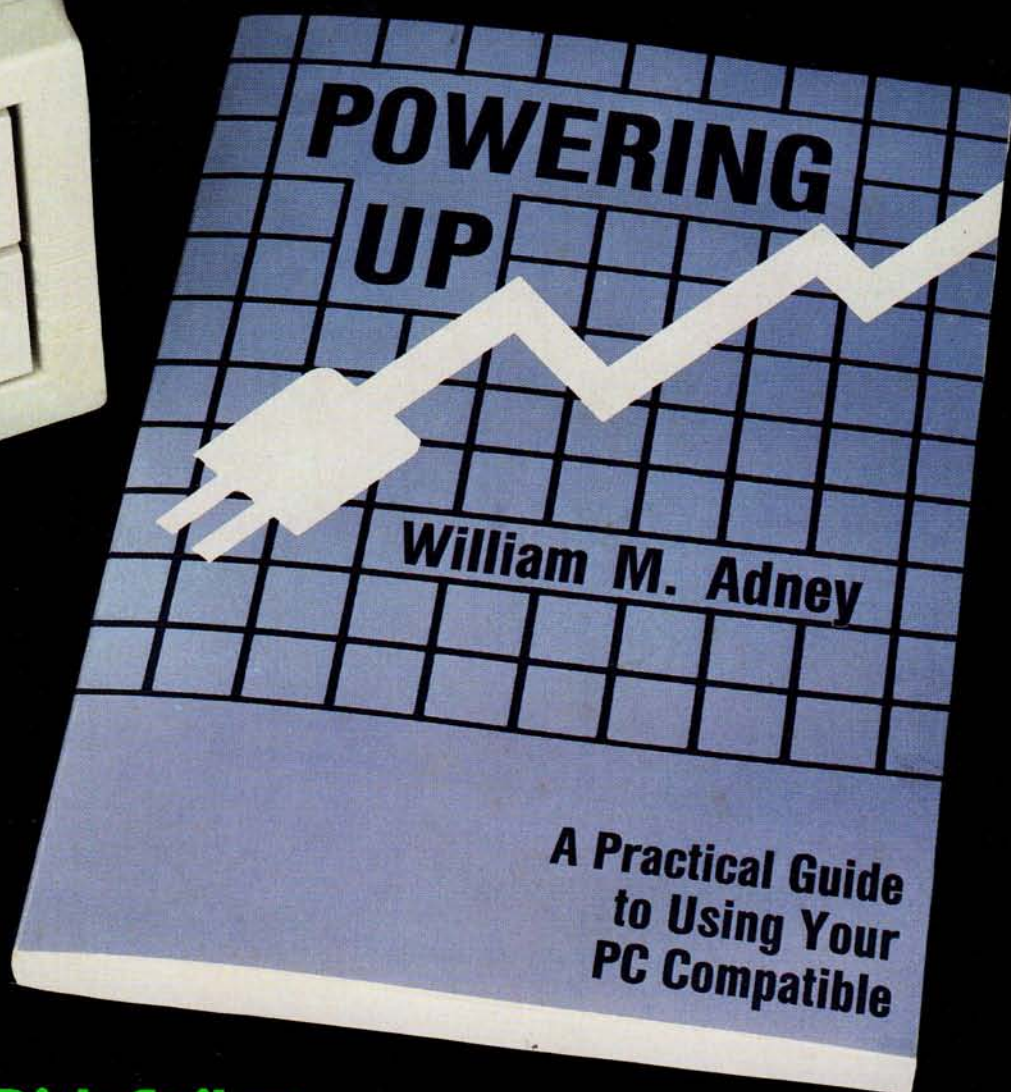


The Official **ZENITH** /Heath Computer Users Magazine

# REMark®



May 1989



## HADES II

Page 4

## Your First Hard Disk Failure

Page 7

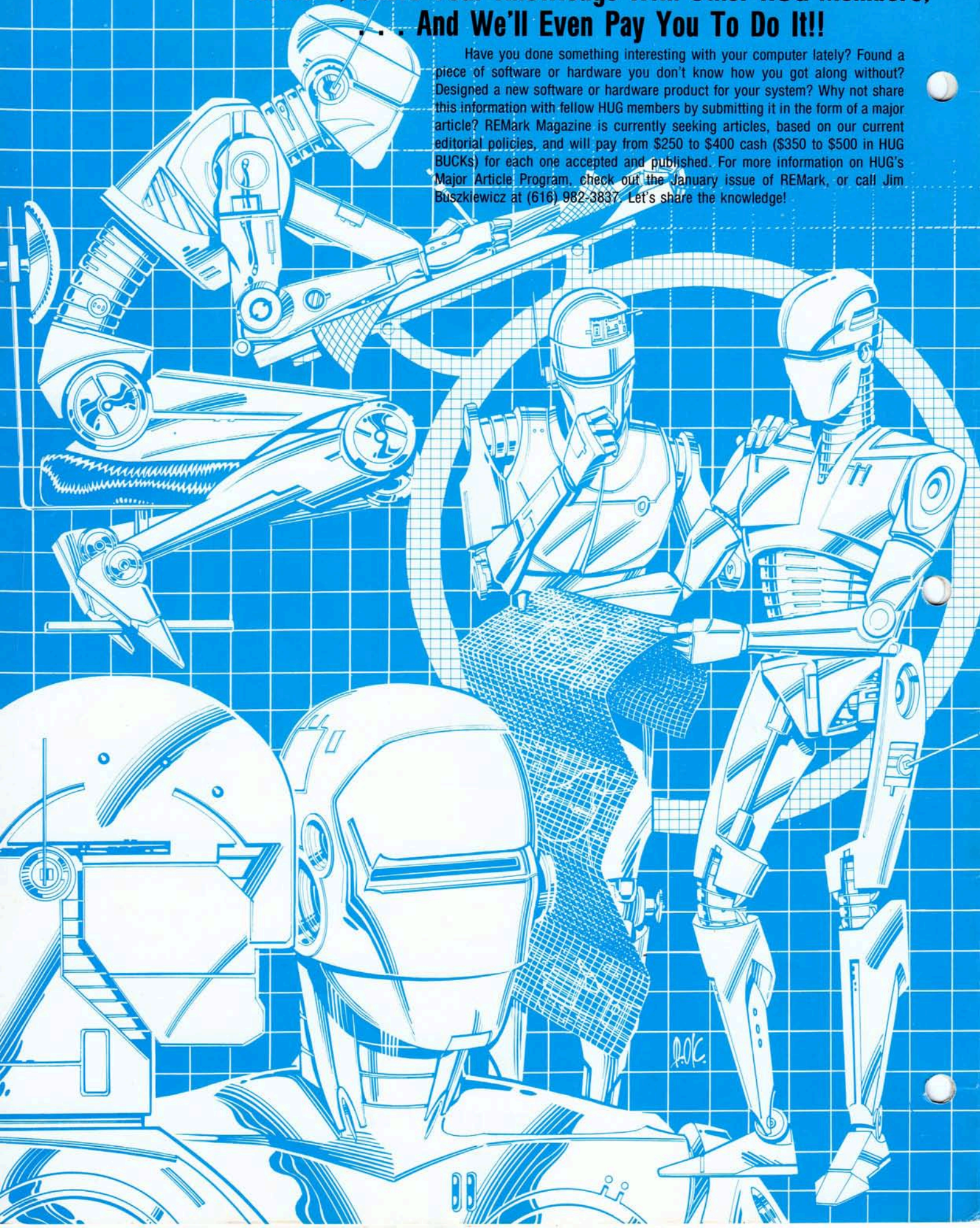
## Protect Your "Setup"

Page 45



# 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!





# **REMark**®

Volume 10, Issue 5 • May 1989

## On The Cover

It's finally here! **POWERING UP**, the book. This book is a practical guide to using your PC Compatible computer for the beginner to the advanced user. You've waited a long time for this, but it's here, available to you through HUG. P/N 885-4604.

## Resources

<b>HUG Price List</b> .....	<b>2</b>
<b>HUG New Products</b> .....	<b>4</b>
<b>Buggin' HUG</b> .....	<b>5</b>
<b>Glitches</b> .....	<b>46</b>
<b>Want Ad</b> .....	<b>56</b>
<b>H/Z Related Products</b> .....	<b>62</b>
<b>Bug Zapping</b> .....	<b>63</b>

## Reader Service

Reader Service No.		Page No.
196	Covox Inc.	46
197	DMA Technologies	61
198	Enable	15
104	FBE Research Company	40
105	First Capitol Computer	10
107	Paul F. Herman	62
137	Jay Gold Software	56
136	Lindley Systems	33
117	Payload Computer Services	34
130	Quikdata	20
121	Scottie Systems	16
195	Strategic Alliance	56
194	Weltec	46

## PC Compatibles

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

## PC Compatible

### Brief Review of WordStar Professional V5

Richard L. Mueller ..... **17**

### Improving Your Business Presentations with

Harvard Graphics Earl R. Zimmerman, Jr. .... **21**

**POWERING UP** William M. Adney ..... **25**

### Using the Mail Merge Feature of Microsoft Word

Jack W. Bazhaw ..... **41**

**Protect Your Setup** Patrick Swayne ..... **45**

## Z-100 and PC Compatible

**On the Leading Edge** William M. Adney ..... **11**

**ENABLE — Part 17** George P. Elwood ..... **57**

## Z-100 Only (Not PC)

**Z-100 Survival Kit #4** Paul F. Herman ..... **31**

### Turbo Pascal™ Versions 4.0 and 5.0

Common Ground for the Z-100 and

PC Compatible Keith Greer ..... **35**

## General

### Getting Started With . . . Your First Hard Disk Failure

William R. Rogers ..... **7**

### The Use of Computers in the Home

Thomas W. Putnam ..... **47**

**Programmers' Learning Corner** Don Keller ..... **53**

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

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) 982-3838

Although it is a policy to check material placed in REMark for accuracy, HUG offers no warranty, either expressed or implied, and is not responsible for any losses due to the use of any material in this magazine.

Articles submitted by users and published in REMark, which describe hardware modifications, are not supported by Heath/Zenith Computers & Electronics Center or Heath Technical Consultation.

HUG is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their usage of Heath/Zenith equipment. As such, little or no evaluation of the programs or products advertised in REMark. The Software Catalog, or other HUG publications is performed by Heath Company, in general, and HUG, in particular. The prospective user is hereby put on notice that the programs may contain faults, the consequence of which Heath Company, in general, and HUG, in particular, cannot be held responsible. The prospective user is, by virtue of obtaining and using these programs, assuming full risk for all consequences.

REMark is a registered trademark of the Heath/Zenith Users' Group, St. Joseph, Michigan.

Copyright © 1989, Heath/Zenith Users' Group

# HUG

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM		DESCRIPTION	PRICE
H8 - H/Z-89/90					
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
SOICALC	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
SPELLS	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
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 Veterinary 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.





# HUG NEW PRODUCTS



10 - Very Good  
9 - Good  
8 - Average

## TABLE C Product Rating

Rating values 8-10 are based on the ease of use, the programming technique used, and the efficiency of the product.

- 7 - Hardware limitations (memory, disk storage, etc.)
- 6 - Requires special programming technique
- 5 - Requires additional or special hardware
- 4 - Requires a printer
- 3 - Uses the Special Function Keys (f1,f2,f3,etc.)
- 2 - Program runs in *Real Time*\*
- 1 - Single-keystroke input
- 0 - Uses the H19 (H/Z-89) escape codes (graphics, reverse video)

**Real Time** — A program that does not require interactivity with the user. This term usually refers to games that continue to execute with or without the input of the player (e.g., 885-1103 or 885-1211[-37] SEA BATTLE.

## ORDERING INFORMATION

For VISA and MasterCard phone; telephone Heath/Zenith Users' Group directly at (616) 982-3838. Have the part number(s), description, and quantity ready for quick processing. VISA and MasterCard require minimum \$10.00 order. By mail, send your order, plus 10% postage/handling (\$1.00 minimum, \$5.00 maximum) to: Heath/Zenith Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. Orders may be placed, by mail only, using your Heath Revolving Charge account. Purchase orders are also accepted by phone or mail. No C.O.D.s accepted.

Questions or problems regarding HUG software or REMark magazine should be directed to HUG at (616) 982-3463.

## NOTES

When ordering any version of MSDOS software, you must specify what type of media you want the software supplied on. If you want 5-1/4" floppies, add a "-37" to the 7-digit part number. If you want 3-1/2" micro-floppies, add a "-80" to the 7-digit part number.

All special update offers announced in REMark (i.e., ZPC II update) must be paid by check or money order, payable to the Heath Users' Group. **NO CREDIT CARDS ACCEPTED.**

## P/N 885-4604 POWERING UP..... \$12.00 By Bill Adney

**Abstract:** The POWERING UP series of articles in this book is specifically designed to help a new computer user select, set up, and use a computer system. These articles are specifically designed to focus on the basic features and capabilities of the Heath and Zenith PC compatible microcomputer systems, and center on the "need-to-know" information with an emphasis on the practical aspects of using your system. For example, some articles include practical examples of specific DOS (MS-DOS and PC-DOS) commands, with explanations, so that even a novice or beginning user can immediately take advantage of advanced features without a lot of technical background or knowledge. Other articles include hardware-related subjects, such as how to add more memory to your computer (and what kind to add), how to set up a printer, and how to set up and use a hard disk. Even advanced microcomputer users will see some practical tricks and techniques that can improve their productivity significantly. And if you are a new user, the information provided in this book can help you graduate from the novice to the advanced (power) user!

These articles use the practical, how-to approach with very limited (if any) technical explanation as to why something works the way it does, except when necessary to help you troubleshoot a problem. In short, you will learn information that you must know in order to successfully set up and use a microcomputer system.

The 15 chapters in this book are titled as follows:

1. Setting Up Your Computer System
2. Powering Up Your System
3. Using File Names and Subdirectories
4. The DOS Command Line

5. Important DOS Commands You Must Know
6. Connecting Peripherals to Your Computer
7. Using Batch Files and CONFIG.SYS
8. Using Input/Output (I/O) Redirection
9. Understanding Video Hardware
10. Adding More Memory to Your Computer
11. How to Select Application Programs
12. How to Select Utility Programs
13. Selecting and Setting Up a Hard Disk
14. Other Useful DOS Commands
15. Maintaining Your Computer System

**Author:** William M. Adney

**Comments:** This is the book you've been waiting for. It's the PC-Compatible hardware/software manual for those people that don't like to read their manuals! It's the book for beginners and advanced users alike.

**TABLE C Rating:** (10+)

## P/N 885-3040 HADES II..... \$40.00

The original version of HADES (Hug's Absolute Disk Editing System) has been updated to HADES II. The original version of HADES, although works properly under new MS-DOS 3.3 PLUS, it cannot handle disk partitions larger than 32 megabytes, and will exit back to MS-DOS when attempting to do so. Now enters HADES II. This new version of HADES retains all of the features of the original version and adds a few extras described as follows:

HADES II will now work with MS-DOS partitions larger than 32 megabytes. Only limited by DOS, HADES II will now operate on disks as large as 0.5gb (512 megabytes) having as many as 1 megasectors!!

You can now manually recover an erased file by manually reconstructing it from recognizable sectors or clusters.

Continued on Page 24



# BUGGIN' HUG

## Poor Quality

Dear HUG:

I can't believe the *poor* quality of program listings that I find in the February 1989 REMark.

1. What happened to all of the comment delimiters in J. Stolarz' article on "QuickBASIC 4.0\* — BASIC Grows Up"? Such statements in the listing shown in the right-hand column on page 35 as:

```
***-----Start printer output
must begin with a ' — single quote. When I first saw that there were no quote marks starting delimiters on his "comment" statements, I thought he had discovered something new. So I had to try my own QuickBASIC — both versions 4.0 and 4.5 — and up — the statements (Comments) must begin with a single quote. e.g.,
```

```
****-----start printer output
```

2. The "Simple File Compression/Expansion Program in C" also needs help. e.g., Figure 4 on page 31 shows a listing of a C program wc.c — It has the following problems — at a minimum —

- the statement:

```
setmode(fileno(stdin), O_BINARY);
is missing a ") — probably after the word BINARY and before the ";"
• the 3rd line after the While statement reads as:
```

```
++n1;
It really should be "+n1;"
• the 5th line after the While statement reads
```

```
inword == NO;
It really should be inword = NO; with one equal sign as a replacement operator rather than with 2 equal signs which is a relational operator:
```

George W. Wilkinson  
66 Linda Avenue  
Portsmouth, RI 02871

## A Simple File Compression/Expansion Program

Dear HUG:

I just found the opportunity to play with the "C" programs listed in "A Simple File Compression/Expansion Program in C" (REMark, February 1989, p. 26) and would like to suggest the following corrections to the word count program (wc.c) on page 31.

The line which reads:

```
setmode(fileno(stdin), O_BINARY);
should read:
```

```
setmode(fileno(stdin), O_BINARY);
The line near the end which reads:
inword == NO;
should read:
inword = NO;
Additionally, the result is a little easier to read if the final printf statement is modified to read:
printf("Lines = %1d Words = %1d Chars = %1d\n", nl, nw, nc);
```

Since I wanted to use the word count program to calculate the actual number of words in my wife's new book, I quickly became annoyed with having to enter a new command for each chapter. So, using the LS.COM and APPLY.COM programs furnished by Zenith, I wrote the following simple batch file which accepts wild cards:

```
ECHO OFF
IF "%1"==" " GOTO HELP
IF "%2"==" " GOTO HELP
IF "%3"==" " GOTO BASE
LS %1%2 | APPLY - "WC < %1%% >> %3"
GOTO DONE
:BASE
LS %1 | APPLY - "WC < %% >> %2"
GOTO DONE
:HELP
ECHO ON
:Use: WCT [path] filename outputfile
The [path] parameter is optional. If it's not present, the batch file assumes all files are in the current directory.
:DONE
```

I also would like to say how pleased I am to see Paul Herman's new "Z-100 Survival Kit" column. It's reassuring to know that my two Z-110s are not yet defunct.

Sincerely,  
Richard Gugeler  
37 Shadowood Place  
Woodland Park, CO 80863

## Read Twice?

Dear Jim:

My "Chaos Microscope" article (Feb. '89) contains a few typos of no significance and one that may cause puzzlement. On page 11, right-most column, line 4, "3.300004" should be "3.300005".

A last-minute improvement: since Turbo Pascal's SQR function is faster than its Multiply, users will get measurably better speed by changing lines 259 and 262. On both lines, replace

```
Pop := Rate * Pop * (1 - Pop);
with
```

```
Pop := Rate * (Pop - SQR(Pop));
```

On the same principle, the reason I used SUCC at line 274 instead of just adding 1 is that PRED and SUCC in Turbo Pascal are 16% faster than -1 and +1. With calculations repeated many thousands of times, these little things add up.

By the way, why "Read this twice!"?

Was the piece *that* dense? Or just very rewarding? (Not that any author objects to being read twice.)

Sincerely,  
Hugh Kenner  
103 Edgevale Road  
Baltimore, MD 21210

ED: Hugh, the article was interesting.

## Chaos Microscope

Dear Jim:

WOOPS! The code you published for my "Chaos Microscope" (Feb. '89 REMark) won't run as listed. Line 31, delete the extra comma following the 'y'. Line 163, insert a missing left parenthesis after 'VAL'. Anyone familiar with Turbo Pascal might have caught those. BUT: of crucial importance: in line 265, XVal in the IF statement should be YVal. That's not easy to catch, and it keeps any graph from appearing on the screen at all. My thanks to D.C. Shoemaker of Seattle for isolating these fixes.

Yours sincerely,  
Hugh Kenner  
103 Edgevale Road  
Baltimore, MD 21210

## Math Checking Motivating

Dear HUG:

In your July 1988 issue, there was a letter that included a short BASIC program for checking math. Boy, did that hit the spot. Last month my daughter's class started into long division and, of course, Dad had to check her homework for her. I remembered reading the letter and went and put the program into my computer. Once she saw what the computer would do she could hardly wait to do her homework. Thanks HUG.

I bought ENABLE for my Z-100 when it first came out. Of course, my computer has two floppies and so there was a lot of disk swapping. I'd like to tell you how I reduced some of the swapping.

Firstly, one of the features of MS-DOS is a Memory Disk Drive better known as MDISK.DVD. In my manual, (Zenith MS-DOS version 2) this is found under System Component Features. I made a disk and named it "I" and made it 64k. Then, I created a Batch File to load ENABLE. One of the parameters that you can add to the ENABLE start-up commands is "M". The function of "M" is to hold non-resident programs in memory. As the ENABLE manual says: "This parameter can increase the speed of some ENABLE operations by retaining non-resident programs in memory, eliminating them only when



memory shortage occurs."

My Batch file to start ENABLE is as follows:

```
REM Copy data files from drive A to Drive I
copy a:\%1\*. * =I:
PAUSE Insert Util Disk in A and Oper Disk in B
enablez (a:,b:,,,I:),VC,M
```

I keep my batch file on the same disk as my data file and to start Enable, I type in Enable followed by a space and then the name of the directory on the data disk that I want to use, for example, mydata. The batch file reads all the files in that directory (no more than 16 files, no more than 64k total) and then pauses and tells you to insert the utility disk in drive A and the operation disk in drive B. Then, the Enable start command, *enablez (a:,b:,,,I:),VC,M* tells the computer to look for the utility disk in drive A and any other enable disk in drive B and the data disk in drive I. I set mine this way because most of the time I am using the word processor and the spreadsheet and once I start I almost never have to switch disks. If you use the data base, you will have to switch disks in drive B. The M on the tail end tells the computer to hold non-resident parts of the program in memory, reducing disk access and speeding things up. Once I had my batch file set up I found that Enable is quite good.

Now, for a couple of questions. How do you increase the size of the memory disk? I've not been able to figure that out and the manual doesn't say. But I'd like to make drive "I" larger if I could.

I have a ZP-150 laptop. It is one of the early laptops that Zenith made. Zenith never made any accessories that I know of. My question is: Can an external disk drive be connected to it? I would use it more, I think, if I had an external disk drive. As it now stands, I have to connect it by cable to my Z-100 and download any material through the file transfer utility.

Thanks for any information you can give me.

Sincerely,  
Joseph G. Nikolai  
Box 5876  
APO NY 09012

### Hero Jr. Accessories

Dear HUG:

I need your help. I am an advisor for a gifted and talented class on robotics and a Boy Scouts of America Explorer Post. I need either the Hero Jr. BASIC accessory or the Hero Jr. Programming Language accessory so I can use my Hero Jr. to teach programming to the two groups. I would also like to get a couple of old out of service Hero Jr.s to let my kids use for parts to build their own robots. A commented ROM listing would also be very useful. If

anyone would be interested in selling any of the items, please contact me.

Sincerely,  
Ken Boone  
Route 6 Box 336D  
Raleigh, NC 27613

### Bug in MS-DOS 3.3 Plus

Dear HUG:

I spent a lot of time, but now have concluded without doubt that the problem is a bug in Zenith's MS-DOS 3.3 Plus (OS-31-3). I loaded four separate copies of this on four Zenith computers (my H-248, a Z-248/12, a Z-148 and a Z-183) and got the same problem on each. The problem went away when I reloaded Version 3.21.

The problem is that if you use the F3 function key to repeat a command (such as a simple DIR directory listing), the computer will disregard the NEXT command if it is a CTRL-something combination (such as CTRL-P to toggle on the echo print). For example:

```
DIR      (list directory on screen)
F3 RET   (repeat list of directory on
screen)
CTRL P   (should toggle on printer echo
- IT DOES NOT)
DIR      (should list directory on
screen AND print - with 3.3
Plus, only the screen list
occurs)
```

You have to enter CTRL-P a second time (after using F3 to repeat ANY command) to have it recognized. The above sequence will work only if you actually re-enter manually DIR in the second line.

Sincerely,  
Harry A. Cole, PhD, PE  
Dept. of Civil Engineering  
P.O. Drawer CE  
Mississippi State, MS 39762

### Help With Hudson Z Support Board

Dear HUG:

I recently acquired a HUDSON AND ASSOCIATES Z SUPPORT BOARD, Rev. A. The board contains an Intel 8088-2 and an Intel 8087-3. It is populated with thirty-two 64K chips. I understand the board was Hudson and Associates' approach to utilizing the 8087 co-processor for the Z-100. I have no documentation or software support for the board nor do I have an address for Hudson and Associates. The board has a forty pin "interface" female connector. Is there an interface cable connection required? Does anyone have an address for Hudson and Associates? Any help will be greatly appreciated.

Tom Hall  
5523 Morgan Road  
Ypsilanti, MI 48197

### Packkit-232 and the H-89

Dear HUG:

I am a radio amateur and I have been building Heath gear for over 30 years. In recent years, I became interested in RTTY and built an HD-3030 interface to work with my H-89 (built in 1985) and SS-9000. Works fine, but I decided to upgrade with the Heath Packkit-232 and take advantage of its many modes of operation. I soon recognized there was a problem with the Fax mode and its incompatibility with my H-25 printer.

Matt Adrian, KD8ZB, of the Heath Amateur Radio Division was contacted and he detailed the reasons for the incompatibility. 1) The HK-232 has a parallel interface printer requirement whereas the H-25 is serial. 2) The H-25 did not have the requisite Epson graphics to work with the HK-232. However, Matt did not give up on a solution and Larry Bollman, KB8EJM, entered into the conversation and, to make it brief, they suggested that I might try the following: 1) Obtain a parallel to serial converter (Centronics) for the HK-232 to H-25 connection. 2) Substitute certain ICs within the H-25 with those supplied by Fina Software. I made those changes and I now copy Weather Maps. I do not understand them, but it all works fine.

What Matt and Larry did for me illustrates the fine cooperation and intelligent assistance in support I have received from Heath over many years of interesting and rewarding kit construction.

Although the HK-232 has a great many features that I do not have with the HD-3030, in operation with the H-89, it does not have the essential capability of transmit buffer with split screen. The transmit buffer only operates from the software and Heath only offers that which is PC compatible. The AEA PK-232 (from which the HK-232 was cloned) also offers the software for the C-64/128. The Heath User's Manual refers to operation with the Heath/Zenith 89 and 90, but full operation requires the necessary software. I am not a programmer, but some hope is indicated from the descriptive material in Amateur Electronics Supply Fall 1988 catalog, page 44, where the AEA states:

"The Host mode (a kind of computer to computer shorthand) of the PK-232 also allows programmers to write special terminal programs, like PC Pakratt and Com Pakratt, for full featured multimode operation . . ."

The Heath User's Manual on page 8-63, after a brief description of the Host Mode, states:

"Refer to the Technical Manual for more information about the Host Mode."

In any event, I carefully looked over the Technical Manual and I do not be-

Continued on Page 24



# Getting Started With.....

William R. Rogers  
4525 Martin Dr.  
Boulder, CO 80303-6003



## Your First Hard Disk Failure

When I purchased my Z-248 in Oct. 86, it came with a Seagate ST-225 20 megabyte hard disk installed. Getting away from the "floppy disk shuffle" was great. The Seagate's performance wasn't dazzling, but it had a reputation for reliability, and with careful management it provided plenty of space. I thought about upgrading to a disk with "true AT performance", but couldn't decide whether to spend the money.

Well, one bright Monday in June the Seagate died. It wasn't a spectacular death; it just didn't power-up with the rest of the system. I had listened to the system power-up sequence every day for nearly two years, and my ears provided the first clue: I didn't hear that characteristic whine of the Seagate spinning up to speed. The boot process hung for several minutes before giving me a disk error message, then the boot failed anyway because the hard disk was my boot disk. Suppressing an urge to scream, I set to work.

I reset with CTRL-ALT-INS, responded to the error message, and at the ROM prompt entered the command to boot from a floppy using my diagnostics disk. The diagnostic programs, available from Heath, thoroughly check out the hardware and identify any errors found. I chose the hard disk diagnostics from the menu, and got quick confirmation that the disk wasn't responding.

Since the 248 keeps its configuration data in ROM, including instructions to boot from the hard disk, I had to make some changes so the system would boot properly. I did a second CTRL-ALT-INS reset to get into the ROM, and selected the SETUP program to alter the ROM configuration data. I deleted the hard disk from the configuration and changed the boot option to floppy disk 0. Then I powered down the system and pulled out the hard disk. I was back to a two-floppy system. I had forgotten what it's like - I'm spoiled by the hard disk!

I knew already that failed hard disks are seldom repairable. Usually they are just replaced. I didn't really want to spend a lot of money, but I did want a faster disk, and this incident was forcing me into a classic quandary: to replace or to upgrade. I thought about my system's performance with the small, slow ST-225. I had no real complaints. Any hard disk is better than none, and this one had been a great improvement over dual floppies on my H-161 (and the H-89 for that matter). I had learned to periodically weed out old files so that I always had some working space, and speed was only occasionally a problem, and only because I had used faster systems and I'm occasionally impatient. The ST-225 would be relatively cheap to replace, and since it would be simply a component swap there was no doubt that it would work just fine.

I had two kinds of upgrade possibilities: a larger, faster AT-type disk that would work with my existing disk controller, or a new sub-system that included a new disk controller with the new disk. The controller is a circuit card that matches the computer to the disk so that they work properly together. Controllers are usually identified by their two connecting interfaces. The bus interface determines the type of computer a controller will fit into, such as the S-100 buss for the Z-100 series computers, the PC buss for the Z-100 PC series, and the AT buss for the Z-200 series.

The other connecting interface is for the disk drive itself. The most common interface is the ST-506, which is old and relatively slow as such things go, but is solid, proven technology. I have an ST-506 controller which uses "Modified Frequency Modulation" (MFM), a recording method for writing bits to the disk. One alternative would be a new "Run Length Limited" (RLL) system. RLL controllers use the ST-506 interface but they use a different recording method to achieve greater storage in a given space. The increased density requires a matching RLL-type drive. The price difference between MFM and RLL systems is small; if you're thinking about a new installation, a RLL system might provide more storage for the money. If you're replacing an existing drive be sure to get the right kind of

drive, since MFM and RLL are incompatible.

Other, newer drive interfaces are ESDI and SCSI. They are much faster than the ST-506 and are incompatible with it and with each other. They use more "intelligence" than ST-506 systems, are more expensive, and are not yet common. Hard disk cards are also available which have the controller and disk in one easy-to-install unit. My controller seemed to be working, and I didn't really want to spend the money for a new one, so I limited my options to simply getting a new ST-506 MFM drive.

This really isn't much of a limit. Drive choices include sizes from 5 megabytes (MB) through 300 MB, with prices that increase accordingly. Drive sizes are usually listed as formatted size; formatting uses some of the capacity of a "raw" drive. Drives also come in two different physical sizes, half height and full height. The full height size is based on the size of older floppy drives, and half height is the size of the floppy drives used in the PC series machines. Many newer machines won't accept full height drives, but the 248 has plenty of room for them. Speed of access is another parameter, and of course faster drives cost more. Also, faster access speeds are usually found in larger drives. Speeds on common disks run from 80 milliseconds for 10 MB "XT" disks down to 28 milliseconds for larger (30 MB and up) "AT" disks. The speed measured is an average time to access a random sector on the disk, so faster times mean better performance. Some advertising features the track-to-track speed. This is only the speed that the heads are moved across the surface, called "seek" time. Average access also includes the time for a given part of the disk surface to rotate under the heads, called "latency", and it is always higher than track-to-track times. Some product reviews use average times that include controller-induced and software-induced delays, which gets very confusing. If you compare two drives, be sure to compare the right numbers. Disk speed can seriously affect the apparent speed of the system when you are doing work that requires a lot of disk access.

Additional parameters of hard disks that may be needed include the number of platters (disks), the number of heads, the pre-compensation cylinder and the parking cylinder. These numbers usually don't mean a thing to the average user; you only need to know them when you install a new disk yourself. If a dealer installed your disk, you can ignore them. The 200 series computers have a program in ROM, called SETUP, that includes these numbers for the drives that Heath sells. Since I was planning to buy and install my new disk myself, I had to pay attention to them.

I checked the hard disk specifications

in the SETUP program to see what my options were, then called the Denver Heath store and talked to them. Their prices seemed high, but I knew I would also be buying a lot of support. That support is nice to have; they're right there if you have any questions or problems and you KNOW the installation is going to work.

I then looked through the mailers and catalogs I had on hand. Since I subscribe to several computer magazines, I get plenty of advertisements and catalogs, and comparison shopping at home is easy. I found that Priority One Electronics, Chatsworth, CA, was selling a discontinued 40 MB (35 MB formatted) drive from Quantum for \$300. ST-225 drives were selling for about \$270, so this looked like a good deal. I knew from the catalog description it didn't match the specifications for any drive listed in the SETUP selections. A drive that matched one of these specifications would be so much easier to install. I mulled it over for an hour before I called them. They assured me that the drive would work fine with my controller. I expected some trouble during installation, so I also got from them the numbers I expected to need: number of heads, number of cylinders, shipping cylinder (the drive automatically parks the heads on power-down), and formatted size (based on the Seagate the size in the SETUP listing is after formatting). I was taking a chance with this drive, since it was not supported in the ROM and I didn't know if I could get it to work. I was on my own - Priority One has good telephone technical support but I couldn't take the system to California if I ran into trouble. If I couldn't get the disk to work, I was going to lose a lot of time, and if I damaged the disk I would lose the money too.

As for performance, the Seagate's average access time is 65 milliseconds and the Quantum boasted 40 milliseconds, so I was getting a faster drive but not top performance of around 28 milliseconds. Overall, I decided it was a good compromise for the money. Actually, I was delighted with the cost/performance ratio, and the size!

While I waited for the new drive, I read the system manuals to get familiar with the installation process, including PREP, PART, and FORMAT. PREP prepares the surface of the disk for later use by PART and FORMAT. It establishes the cylinders and sectors, called low-level formatting, and it checks the condition of the disk media as well. Some vendors ship disks with low-level formatting already done, but I didn't expect that in this case. PART then allocates space on the disk to one or more partitions, where each partition is used by MS-DOS as a separate disk drive. It also allows you to designate a partition as the boot drive. A partition is

limited by MS-DOS to 32 MB, but by partitioning a larger drive you can effectively use very large physical drives in Zenith machines. Other machine vendors don't always include a PART utility, so it is often sold with hard disk systems as an extra cost option. Thanks to Zenith, we don't need to buy it. After PART, FORMAT prepares each partition for file storage within MS-DOS. Since the Quantum drive offered 35 MB after formatting, I planned to allocate two partitions with PART, about equally dividing the space. The Zenith MS-DOS manuals describe these and other hard-disk utilities in great detail, and you should read those sections carefully before buying a hard disk. Don't forget to reboot after running any of these hard disk utilities.

The drive arrived in exactly one week (to the hour!), and I began the process of installation. There was a warning about static electricity packed in with the drive; static electricity is not a problem in my house but I was very careful anyway throughout the procedure. The first step was physically mounting the drive. This was tricky because of all the cables, but I found out that removing floppy drive B and its support bracket made things much easier. The Seagate had been mounted under floppy A in the inner bracket. The support brackets are removed by taking out one screw at the bottom front of the case and working the bracket backwards out of the strong spring clip on the bottom of the case. The drives are mounted on the brackets. After pulling the cable connectors off I had both floppy drives, the hard drive, and the mounting brackets out of the machine and many cables hanging loose inside. I strongly recommend making notes about the locations of all those connectors; I used small gummed labels on the cables near each connector. I installed the new drive in its bracket, beneath floppy A. The Seagate was a half-height drive, but the 248 allows full height drives like the Quantum to be installed so this was no problem. I put the brackets back inside the case and re-connected the cables. It's easier to connect the cables before locking the brackets in place.

That done, I fired up the machine to PREP the new disk. The system simply refused to acknowledge the new drive, hanging up during the boot process just like it did that day the old drive died. I pulled it out again to recheck everything and insure that the "drive select jumper" was on DS1, as specified in the Owner's Manual and the Technical Reference Manual. There was no doubt in my mind that I was doing it right. I stepped through the installation process again to be sure all the cables were on the right connectors, and tight, and tried again. Still no response. Finally I checked the Seagate, and decided on the basis of the drawing in the



Owner's Manual that it was set for DS2. I set the new drive for DS2, and the system found it! After more than two hours with the hardware I finally closed up the case. I was ready to move on.

I expected problems in PREP because the drive wasn't directly supported in the ROM. I had talked to Denver Heath again, and was told to "just run PREP with the /Q option and it will ask for the right numbers". I chose a disk description in the ROM that was close to the one I had, loaded the back-up of my MS-DOS distribution disk and entered "PREP /Q". I got the introductory screens, but no questions, and PREP began to work. I knew something was wrong and aborted the process, then very carefully tried it again. After the second abort I called Heath again, and got the word that "/Q" only worked with a newer release of MS-DOS. I have release 3.1.

So now I was stuck! I could upgrade to MS-DOS 3.3 for another \$150, or... what?? I reviewed the SETUP options again, and looked carefully at number 25. Its specifications matched my drive except for fewer cylinders and smaller total size. I had a 42.7 MB drive that should format to 35.7 MB. With #25, I could get a formatted 32 MB. This was a guess on my part, as I was not sure at all that the software (PREP, PART, and FORMAT) would work this way. If I was right, and lucky, I now knew the "or" part: losing 3.7 MB of disk space. I could get that 3.7 MB for \$150. I decided not to for now, and pressed ahead. Since the ROM disk selection limited the apparent physical size to 32 MB, I decided to use only one 32 MB partition.

I changed the SETUP disk selection, and started again with PREP. PREP does take a while to work. Then I ran PART, selecting one 32 MB DOS-16 partition and the boot option. The next step was FORMAT, with the System option, still from my copy of the distribution disk. Finally, with MS-DOS on the hard disk, I rebooted. What a feeling! The boot process was fast - even after a week of floppy boots it seemed faster than the old drive. I had eaten supper during the PREP, so I refilled the coffee cup and pressed on with installing the software on my new disk.

I had prided myself on my back-ups. No wonder, as I had never had to do much with them. I felt wonderfully virtuous every time I updated the back-up floppies, but I am a careful user and seldom needed them. I occasionally needed to restore a file or two, but always a current one, and the back-up was on my "current stuff" disk kept close for quick back-ups at the end of a session. Now I needed a real good system, and mine wasn't.

My problems fell into two general types: depending on my memory, and re-

lying on re-installing my software from distribution disks if it was ever needed. First, my weak memory. I really believed that as often as I typed in file names I would remember them. That much abused DOS interface would force me to know my files, right? Wrong! I couldn't remember even a few of the file names. There were just too many files, and too many strange names, for me to begin to remember them all. I could remember project names, but only a few files matched up that way. I could recall what a file was when I saw its name on a DIR display, in context with other files associated with it, but since I didn't have even that now I was lost. I had not ORGANIZED my back-ups, so I had several hundred files on many floppies but no systematic way to restore what I needed to rebuild my system. A tough lesson learned the hard way: don't back-up for backing up, back-up for restoring! Think through a system, practice it, then use it. If I had ever rehearsed a massive restore I wouldn't have been stuck with this mess.

Re-installing the commercial software packages seemed like a good option, and could have worked if I had noted somewhere, perhaps in the manual, what choices I had made, what deviations I had devised, and how all of the batch files I built interacted to provide a coherent system. I did none of that, and now faced the task of re-reading all of appropriate manual sections, making the same mistakes again, and re-thinking the low-level stuff hidden in the batch files. This has eaten more time than I ever imagined, and I can't claim much improvement from the first time. I had slowly customized everything to my liking, and now feel like my own machine is a stranger.

I got my DOS and UTILITY sub-directories restored that night, since both of them were easy to reconstruct, and decided to finish the rest the next day (Tuesday). By Saturday night, after five days of installing software on the new disk, I had only the C compiler left to do, and it was to be an upgrade since I never installed the last release.

I wish I could justify a tape back-up system. It would make this part of using a computer much easier. But the budget is already strained, so the goal now is to devise an effective, but cheap, back-up and restore system that meets my needs. I have tried the BACKUP and RESTORE utilities, but I think they are clumsy and slow. I want more flexibility. I want something that works the way I think. I want to do a one-time back-up of installed software, then back-up only data and documents on a daily basis.

I spent most of Saturday on the one-time backup of installed software. First, as I had rebuilt my first level sub-directories during the week I made careful notes of how I intended things to work. This is, in

effect, a plan for organizing the disk. Also, as each package was installed, I noted in the installation section of each manual what I had done and why. This will save a lot of time next time, since I won't have to re-read the manuals and make the decisions all over again. In most cases the manual steps you through the installation, and I simply noted which options were chosen and why. I made note of any files on the distribution disks that were not on the hard disk when the installation was completed. For some packages, such as Microsoft Word, this list is quite lengthy. Also, batch files I wrote along with the installations were printed out and inserted into the manual. It was in building the batch files that I made most of my mistakes, and lost a lot of time. When things got confusing, I referred to my notes on sub-directories to remember the overall plan. When each installation was finished, I copied the files that were new or changed to a floppy dedicated to that software package. It was easy to determine which files to copy by checking the dates on a DIR listing. Anything with today's date got copied. I then printed a directory listing of the sub-directory by piping the display to the printer as "DIR > PRN". I wrote notes on the directory printout which was then stored with the floppy disk. Finally, I printed the entire directory structure using the TREE utility and stored this listing with the backup floppies.

With this system, any sub-directory can be rebuilt by copying the distribution disks to the sub-directory, eliminating unneeded files, then copying the back-up floppy. If this level of back-up fails, I can fall back on the printed information and manually reconstruct what I need relatively quickly. I did a quick test of the system by wiping out a few sub-directories and restoring them. It was fast and easy. But this test was on a clean system with the problems fresh in my mind. I'll test it again later in the year to see how it really works. I think I can simplify the restore process with some batch files, and that should save time, too.

I'm not yet satisfied with a back-up method for data files and documents. I'm currently using a floppy for each system, which I update after each session, but this is tedious and I don't always remember. I have in the past ignored file management utility programs, but one of them is probably a good solution to this problem just waiting to be found. What I think I want is a utility to run through the directory structure at the end of the day and copy to a floppy any file that was created or changed that day. But I don't want them put into an archive format; I'd like to be able to retrieve them individually. If I can't find a program to do this, I'll have to write one, but it will be a few months before I

Continued on Page 52



## Some People Are Still Searching (and Searching) for a Better Way to Upgrade Their Systems than First Capitol Computer's Full System Upgrades...

send it to us to minimize the lost time, and we'll make every effort to turn the machine around within 48 ours of receipt.

### Still Searching for Impossible Dreams?

We can wait, but pricing and availability on the best upgrade for you might not! Offers subject to change without notice! Better to lock your upgrade in now, while there's still time! *Prices listed below are based upon models available at the time this ad was placed.* Call for current prices, availability and full details.

### Upgrades Currently Available

#### -Zenith Data Systems Upgrades-

**FCC-150/286-UP:** Z-151/2/8/9 or Z-161 to Z-286 80286 AT compatible (16 bit). \$1295.  
**FCC-150/248-UP:** Z-151/2/8/9 or Z-161 to Z-248/12 12MHZ 80286 AT compatible. \$1895.  
**FCC-150/386-UP:** Z-151/2/8/9 or Z-161 to Z-386 16MHz 80386 computer. \$2495.  
**FCC-248/386-UP:** Z-241/248 to Z-386 16MHz model (32 bit processor). \$2095.

#### -First Capitol Clone Upgrades-

**FCC-150/CL-AT:** Z151/2/8/9 or Z-161 to First Capitol 12MHz CL-AT (16 bit) Clone. \$995.  
**FCC-150/CL3825:** Z151/2/8/9 or Z-161 to First Capitol 25MHzCL-386/25 (32bit)Tower.\$2295.  
**FCC-248/CL3825:** Z-241/248 to First Capitol 25MHz CL-386/25 (32 bit) Tower. \$1795.

**Call for Prices on Z-386/25 Upgrades!**  
**These Also Work on Kit Models!**

Add 2% shipping and handling (plus 6.725% sales tax, if a Missouri resident).

### What Could Be Better than a Full System Upgrade?

Full System Upgrades aren't mere circuit boards, modifications, or kits! We take your original drives (and other components) and move them into a new chassis. The result is an almost entirely new machine indistinguishable from a unit right off the production line! No compatibilty problems, & all work is performed by factory authorized service technicians.

### The Cost? A Small Portion of What a New Computer Would Cost You!

That's right! Since First Capitol uses parts from your old machine in the new cabinet, you only pay for what you need - and you receive a built-in trade-in credit for your old machine! We'll even provide trade-in credit for any third-party products which won't work in the faster machines towards replacements - and install them for you at no extra charge!

### So, What's the Catch?

The catch is that since these upgrades involve components from your old computer, you'll lose use of that machine for about a week (counting shipping time in both directions). However, we will schedule your system upgrade before you

**First  
Capitol  
Computer**

#16 Algana  
 St. Peters, MO 63376  
 Orders: 1-800-TO-BUY-IT  
 Technical: 1-314-447-8697



## Heath/Zenith Computer Model and Series Numbers, Upgrading the Keyboard Z-386 Memory Boards,

William M. Adney  
P.O. Box 531655

Grand Prairie, TX 75053-1655

Copyright © 1989 by William M. Adney. All rights reserved.

There seems to be an increasing number of HUG members who have computers and aren't quite sure what they have in terms of the articles in REMark. To help solve this continuing problem, I talked to Jim Buszkiewicz some months ago about the idea of including something about the Heath/Zenith model numbers in the REMark Table of Contents for each issue. You saw the first implementation of that idea in the January 1989 REMark, but it has become quite clear to both me and Jim that the listing provided is not quite sufficient by itself. For example, Jim McArthur (Seattle, WA) recently said: "I have a Z-159 model 13. What do I have?" And Linda Wheatley (Bawlf, Alberta, Canada) mentions that she has a Z-100 PC series computer with 256 K of memory, and she wants to add more memory, but does not know what is required to do so. I hope that the March 1989 Powering Up answered her question of how to add memory, but this general problem still seems to be related to the wide range of Heath/Zenith model numbers.

To some extent, the problem is not unique to Heath and Zenith systems, and you can find the same difficulties with other manufacturer's models, too. For example, if I write about an IBM model 339 or model 80, I suspect that very few people would know what I'm talking about. If I mention the IBM 8 MHz AT computer (model 339), nearly everyone who knows anything about computers knows what

that is. And although I wrote about this basic question in the Introduction to the Powering Up book (not published in my column), it is clear that the subject needs to be addressed in considerable detail in REMark. I am currently working on a major article that discusses all of these model numbers and series, but that has turned into more of a research project than I expected. Part of the answer includes trying to anticipate what questions might arise in conjunction with the model number, and there are a lot of them. As you will see later in this article, there are two major classes of microcomputers, but there are also significant differences within the same class. Although I will not try to give a full explanation of all model numbers in this article, there are a couple of critical things that all owners of Heath or Zenith computers should know.

### Model and Series Numbers

When Zenith announced the first computer — the Z-100 PC series — in the PC compatible series, I thought: "Oh no! That designation will confuse nearly everyone." And it has. Trying to keep all of this straight is a real chore, even if you know the model numbers. If you don't, it is incredibly confusing. At the time, I suspected that was a marketing ploy to help ease the transition from the then-current Z-100 computer to the PC compatibles. That was a dumb move, and I said so at the time in this column. I thought that someone at Zenith partially recognized

the problem with the introduction of the Z-200 series computers, but I never did understand why the 6 MHz unit was called the Z-241. (The first model?) The Z-248 model did imply that it was an 8 MHz system, so the first model should have been a Z-246? By the way, the Z-200 model series also seemed to imply that the system used an 80286. Then the Z-286 was introduced, and I was nearly convinced that the problem had been solved in a reasonable way. However, that was not to be the case because Zenith has introduced the Z-2526 which makes no sense unless you actually know what it is. It happens to be an 80286-based computer that runs at 12 MHz. So, I will say it again. Whoever is defining these incredibly stupid model numbers needs to explain them so we can all understand what they mean (if anything). Why not simply call this new computer a Z-286/12 to keep things simple? And if Zenith releases a 25 MHz 80386 computer, why not call it a Z-386/25?

Unfortunately, these ideas do not solve the current problem, and there are several things you should do if you don't really know what kind of computer you have. The first thing is to determine exactly what model of computer you have. On most Heath and Zenith computers, you will find the exact model number on a label on the back of the computer. It will say something like Z-158 (or H-158 if it was a Heathkit) or Z-148 or Z-386. Now you know what model you have.

The next step is to become familiar with the manual that came with the computer. This manual is generally referred to as the "Operations Manual" or the "Owner's Manual," depending on exactly what series computer you have. For example, my reference library includes a copy of the hardware manual supplied with the original Z-151, and it has a title of "Z-100 PC Series Computers Operations Manual." For my Z-241/248, the manual has a title of "Z-200 PC Series Computers Owner's Manual." When you look for this manual, remember it is NOT the MS-DOS manual; it is the manual that is specific to the computer (usually the series), not the operating system. These are just examples based on my own library, and your manual may say something a little different. Your manual may refer to the Z-130 series, even though the Z-138 was the only specific model in that particular model series. Now you know what series computer you have. For discussion purposes, I have limited this description to PC compatible computers because I assume that Z-100 (not PC) owners know about the differences as a result of past REMark articles.

After you have found the manual for your computer, you should spend some time scanning through it. I am finding, in many cases, that most of the common questions can be answered by reading the "Owner's Manual." In particular, you need to read the chapter that discusses hardware, and early in that chapter you should find a page that lists the "Specifications" for your computer. For the Z-100 PC manual I have, the specifications are listed on page 2.5, and the Z-200 specifications are listed on page 5.5. For purposes of understanding the keyboard discussion later in this article, you will want to pay particular attention to the "Processor" specification which will state that your computer has something like an "Intel 80826 16-bit Microprocessor." In any case, I strongly recommend that you spend some time becoming familiar with this manual because it may be essential in figuring out how to work with or change something in your computer. More on this general subject in a future article.

### Z-386 Memory Boards

I have received a couple of letters from David Brown (Deerfield Beach, FL) about what 32-bit memory boards (besides Zenith) will work in the Z-386. Although the cost of memory appears to finally be decreasing, the 1 MB Z-505 and 4 MB Z-515 Zenith memory boards are still quite expensive. In my first letter to him, I mentioned that "At the present time, no third-party manufacturer provides 32-bit memory boards for the Z-386," but David pointed out that an article in the May 31, 1988 (page 270), *PC Magazine* that reviewed the Z-386 indicated that it had an

"Intel-standard slot" for 32-bit memory. Unfortunately, we all make mistakes like that, and although all writers try to avoid publishing bad or misleading information, it still happens sometimes. Suffice it to say this particular point in that article is simply wrong because it implies that there is some kind of "standard" about the physical connections on a 32-bit memory slot. That is, in part, what some of the furor about the EISA versus MCA bus is all about — a standard. If the term "Intel-standard slot" really means that the DATA PATH was designed in accordance with the Intel standard, that's fine.

For those of you with a Z-386 computer, I have done some additional research on this question, and here's what I found. At best, the *PC Magazine* article information on page 270 is at best misleading on the subject of 32-bit memory for the Z-386. So far as I can determine, I believe that it should have been listed as "Intel-standard INTERFACE", not "Intel-standard slot." The Z-386 essentially has what amounts to an AT-compatible bus with 8-bit and 16-bit slots available for display adapters, disk drives, internal modems, fax boards, and mice as mentioned in the article (on page 274), but any suggestion of memory boards is significant by its absence in this text.

On the Z-386, Zenith has implemented a 32-bit data path for some slots using a proprietary 32-bit bus that can currently use only Zenith-designed memory, CPU, and I/O cards. Compaq has done the same thing for memory cards, but with a different proprietary 32-bit bus. IBM has nothing to do with this discussion because ALL of their current computers (except for the "new AT" PS/2 model) use a still different bus configuration that is referred to as Micro Channel Architecture or MCA. You can't even insert MCA cards in a "standard" AT-type slot because they are different sizes and have different pin connections. Even if you could, IBM 32-bit cards for the MCA (or other third-party equivalents, if any) will simply not work in a Zenith or Compaq proprietary bus.

Although I have attempted to identify a third-party manufacturer that makes a Zenith-compatible 32-bit memory board for the Z-386, I have not been successful. At least two popular memory board manufacturers have advised me that they have no intention of developing boards for the proprietary bus for either Zenith or Compaq 80386 systems. If anyone knows of a Zenith-compatible 32-bit memory board, be sure to let me know because I would definitely like to review it in my column as an alternative to the Zenith memory boards. I should also note that this memory board problem is "disappearing" to some extent because all manufacturers, including Zenith, are providing SIMM sockets on the "motherboard" for

up to 6 MB of memory, such as in the latest 25xx series of 80286 computers.

### Powering Up and the Z-100

The response to the "Powering Up" book published by HUG has been tremendous and was surprising, even to me. This book is a complete set of all the Powering Up columns that I completed last year, and the last article in the series will be published this coming August. Although the primary focus of Powering Up was directed toward the Zenith PC compatibles, many of the articles apply to most DOS-based systems, including the Z-100. The two primary exceptions to that are the article on video hardware (published in the February 1989 issue) and adding memory (published in the March 1989 issue). The article on hard disks was also specific to the PC compatibles, but many of the subjects covered also apply to the Z-100.

One of the major reasons that Powering Up focuses on the Zenith PC compatibles is that most of the Z-100 items, such as hard disks, have already been covered in this column. And although there are clearly some unique topics related to the Z-100 (such as video boards), there is not much to add so far as the hardware itself is concerned. Besides, Paul Herman is doing an excellent job with his articles on the Z-100, and I suspect he will cover many subjects of interest to Z-100 owners (including me).

As I have said many times, the Z-100 design is unique, and in my opinion, still ranks as one of the best-designed microcomputers ever. It doesn't matter whether you want to talk about its dual-MPU architecture or how easy it is to disassemble. And its keyboard functionality, touch and feel are still unsurpassed by any computer or keyboard manufacturer. I have had quite a chuckle out of IBM's introduction of the "enhanced" 101-key keyboard with 12 function keys when the Z-100 had that, and more, back in 1983. For touch typists, the layout of the IBM enhanced keyboard is very poor. It is too bad that Zenith apparently believed that it was important to have the same layout for "compatibility" reasons.

### The Best PC Keyboard

I have not used a Zenith keyboard on my '248 for some time, and while I tried the 101-key keyboard that came with my '386 for a few minutes, I adopted the Northgate OmniKey/102 for both systems last year. The OmniKey/102 is far superior to any other PC compatible keyboard I have tried, and I believe that Northgate has finally resolved all of the various compatibility problems. Still, you may have to upgrade the ROM in your Z-200 ('241 or '248) or Z-386 to get the keyboard to work properly because there was evidently a glitch in some early ROM versions for



these units. And if your current system originally had one of the older 84-key keyboards, you absolutely MUST upgrade the ROM in order to get the 101-key keyboard to work for any brand of computer — IBM, Compaq or Zenith. Even some of the Zenith systems that originally supported the new keyboard, such as the Z-386, apparently had at least one ROM bug that caused a problem with the OmniKey/102. As far as I know, all that has finally been fixed, and I use this keyboard on both systems. It is unfortunate that Zenith has evidently abandoned innovation in keyboard design in favor of being compatible with the "standard" design. Most other computer manufacturers have too. Interestingly enough, the terrible design of the standard 101-key keyboard has created a significant after-market for keyboard sales.

If you are not a touch typist, then the keyboard layout probably does not matter all that much. How well you like your keyboard also depends on exactly what software you are using, too. If your software is heavily function key oriented, like Word Perfect, you will probably prefer a keyboard that has function keys on the left side like the usual 84-key keyboard. For that kind of software, the placement of the function keys is important because you also need easy access to the CTRL, SHIFT, and ALT keys because they are used to modify function key commands. This kind of keyboard layout is very important to touch typists because there is minimal hand and finger movement from the main keyboard's home row position. And anything that requires more than minimal hand/finger movement from the home row can have a considerable impact on typing speed. In contrast, the standard 101-key keyboard has the function keys arranged along the top of the keyboard, and using any of these function keys with CTRL, ALT or SHIFT is a two-handed operation — unless your hands are the size of the Jolly Green Giant's or you have fingers the length of Freddie's (Nightmare on Elm Street). Most of us don't.

There is one other factor that is important in how fast a touch typist can type: the keyboard response — how it "feels" (the tactile response) and how it sounds. Evaluating keyboard response is a very subjective and personal matter because it depends on a number of variables, especially the keyboard that you originally used when you learned to touch type. Many of us learned on an IBM Selectric, which is usually accepted as a standard for keyboard response. If you learned to type on a manual typewriter, you may prefer a keyboard with a very heavy touch like the usual IBM keyboards. And if you ask 10 people about their evaluation of keyboard response, you will probably get at least three differ-

ent opinions with no real consensus.

Some studies have shown that many touch typists prefer a keyboard with a "lighter" touch because it tends to increase typing speed. Most Zenith PC compatible keyboards have this light-touch feature, probably for that reason. To some (including me), the Zenith keyboards also feel slightly "mushy", and generate virtually no sound.

The sound of a keyboard can also be important to a touch typist because of the audible feedback provided when a key is pressed. Some manufacturers have even developed a TSR program that provides a "click" on the computer's speaker to emulate a typewriter to some extent. I use Borland's SuperKey program that also provides an option to turn on a click on the speaker.

I have used a large number of different keyboards for a wide variety of microcomputers, not to mention mainframe terminals, such as the IBM 3278 and 3279. For my money, the best is still Northgate's OmniKey/102, and it seems silly to spend several thousand dollars for a fast computer system only to put up with a tacky keyboard that slows you down. That can be nearly as much fun as driving a Corvette without a steering wheel, but it will also be slightly less exciting. Zenith would do well to consider using a keyboard like the OmniKey/102. But before you decide that the OmniKey/102 is for you, there are some things you should know about keyboards, in general, and the OmniKey/102, in particular.

### Keyboard Differences

All keyboards are NOT the same. In general, there are two classes of keyboards: the PC/XT class and the AT class. For Heath/Zenith computers, the PC/XT class includes all desktop models in the Z-130, 140, and 150 series of computers. If you are not sure what specific model you have, look at the back of the computer to find the exact model number. For example, the Z-150 series includes the Z-151, 157, 158, and 159; and it is also referred to as the Z-100 PC series as I mentioned earlier. All of these series can also be identified by looking at the specifications page in the Owner's Manual which will show that these computers contain an Intel 8088 processor chip. For purposes of this discussion, all of these Heath and Zenith computer models and series are in the general PC/XT class of microcomputers.

The AT class of computers contains at least an Intel 80286 processor and also includes the 80386, as well. This includes models such as the Z-241, 248, and 386, plus all of the newer models too, like the 2500 series.

For purposes of looking at keyboards, this discussion also applies to virtually all other manufacturer's computers, too —

IBM, Compaq, and just about any other compatible you can find. To figure out the class of system you have, you need to check the specification sheet for the type of Micro Processor Unit (MPU). If it is an 8088, you have a PC/XT system. If you find an 80286 or 80386, you have an AT class system. Since that was relatively straightforward, let's take a look at how the IBM "standards" have managed to really confuse everyone on which keyboard can be used with what system.

Without going into all of the hairy technical details, suffice it to say that each class of computer also has a special keyboard interface (including the system ROM) with the computer's hardware. When you press a key on a PC/XT class keyboard, the ROM in that system expects to receive a "key code" of one byte. When you press a key on an AT class keyboard, the ROM in that system expects a TWO byte-key code. Obviously, a PC/XT keyboard cannot talk to an AT class computer and vice-versa, because of the difference in the length of the key code. And for those of you who know something about programming and PC compatible keyboards, the "key codes" I am talking about here are NOT the same as the "scan codes", although there are also differences in the scan codes, too.

Fortunately, this problem is not difficult to overcome, and many manufacturers, including Zenith and Northgate, provide a switch that allows you to easily change the key code configuration to match your system. For the Zenith enhanced keyboard, the switch is located underneath the ZDS nameplate in the upper left-hand corner. On the Northgate OmniKey/102 that I have, the switch is located on the underside of the keyboard.

To specifically relate this discussion back to Zenith computer models, it should be clear that you cannot generally take a keyboard from an 8088-based desktop computer, such as a Z-158, and expect it to work on an 80286 or 80386 system, such as the Z-386. It won't work, and you will probably see an error message like: "Keyboard not connected." Unfortunately, the problem with the key-code differences is only the tip of the iceberg.

### 84-Key Keyboards

When you talk about keyboards, it is usual to define them in terms of how many keys they have. The first group of keyboards generally has about 84 keys, although a specific unit in this group may actually have 83, 84 or even 88 keys, depending on the manufacturer. For discussion purposes, I will call this the 84-key keyboard. These keyboards are usually easy to identify because they have the 10 Function Keys on the left-hand side like the original IBM PC keyboard. Unfortunately, not all 84-key keyboards are the

same.

Like computer systems, there are two classes of 84-key keyboards: the PC/XT class and the AT class. The PC/XT class uses the 1-byte keycode, and the AT class uses the 2-byte keycode as mentioned before. Even though the keyboards look about the same, they cannot be used with a different class of computers as mentioned earlier.

Keyboards that contain about 84 keys were supplied as original equipment on virtually all of the 8088-based (PC/XT class) Heath and Zenith computers. Similar keyboards were also provided with all AT-class Z-241 computers, and some early Z-248 computers (including mine). These statements also apply to nearly all other keyboards by other manufacturers, so long as the class of computers is the same.

Because these keyboards look very much the same, it is frequently difficult to tell which is which just by looking at it, but there is one common difference. The AT-class keyboards have a "Sys Req" (System Request) key, and the PC/XT keyboards generally do not. The System Request key provides a special feature that is used with some operating systems like OS/2, but you will probably not find much, if any, need for it with other software. By the way, don't press this key during system operation because it can freeze a DOS-based system under certain circumstances. Besides, it only works with software that can use the System Request function, just like the function keys.

Computer manufacturers, like automobile manufacturers, continually change things, and changes to the keyboard have been no exception.

### 101-Key Keyboards

The 101-key keyboard was introduced by IBM a couple of years ago (about April 1987 as I recall). At the time of its introduction, it was also called the "Enhanced" keyboard, and I have a friend who also likes to refer to it as the "Enchanted" keyboard. I suspect that this reference has to do with the fact that, although this keyboard does have some very good features, many users were extremely disenchanted with, and quite vocal about, the new layout. Touch typists in particular were not pleased because the new layout was not especially helpful in doing things quickly on a system. IBM evidently forgot the key fact that a keyboard is THE interface to a computer. The keyboard is also the very slowest part of a computer because a human must use it, and anything that affects that already-slow interface is definitely not welcome, especially by touch typists. But this keyboard does have some distinct advantages.

In particular, the 101-key keyboard essentially duplicates the original 84-key

unit plus a new cluster of "function keys" (for cursor control, etc.) inserted between the "main" keys and the keypad. The new design makes it much more convenient to use the keypad to enter numbers, especially in spreadsheets, because you don't have to keep switching back and forth in the Num Lock mode. This keyboard also features 12 Function Keys (instead of 10) arranged along the top of the keyboard. There are two sets of ALT and CTRL keys — one set on each side of the spacebar — that supposedly makes them easier to use. And the CAPS LOCK key has been moved to just above the Left Shift key where the CTRL key used to be. Although that particular design may be appropriate for a typewriter, it is not particularly helpful on a computer keyboard.

The movement of the CTRL key from its original standard position has irritated so many people that there are lots of "fixes" for it. For example, Pat Swayne developed "A Software Fix for the Misplaced Caps Lock and Ctrl Keys" published in the February 1989 REMark (page 17). Borland's SuperKey also allows you to fix the keyboard, and you can also change most of the other keys to suit your own preferences. And WordStar version 5.0 also includes a program that will "exchange" the functions of the CTRL and CAPS LOCK keys. I have tried all of these programs, and although each will exchange these key functions, they also have some minor disadvantages because it is usually not possible to correct the entire problem with software. For example, some programs may not be able to correct the display of the CAPS LOCK status light (LED), and Pat's program cannot change the key click. To me, those are really minor problems compared to how clumsy the standard 101-key keyboard really is. If you really want to cure the problem, however, there is a hardware approach that works much better.

On the Northgate OmniKey/102 keyboard that I have, there is a switch on the underside of the keyboard that allows the functions of the CAPS LOCK and one of the CTRL keys to be interchanged. I understand that all current models of the OmniKey/102 are now hard wired so that the CTRL key is permanently in the correct position, and the CAPS LOCK key is just to the left of the spacebar. That is not a disadvantage in my opinion because that's where the CTRL key should be anyway.

If you really use your system a lot and want a better keyboard, I have found nearly everyone who has tried it likes the Northgate OmniKey/102 as much as I do, especially touch typists. Unfortunately, adding this new keyboard to your current system is not quite so simple as just plugging it in.

### Adding a 101-Key Keyboard

In general, it is fairly easy to add a 101-key keyboard to any Zenith desktop computer system, but there are a few things that you must know. There was also a big flap last summer about whether or not the Northgate keyboard would work with all Zenith computer models. Much letter writing and CompuServe time was expended to describe the problem, and there was a lot of finger-pointing as to who caused the problem. More on that in a minute, but first let's look at the general problem.

The introduction of the new 101-key keyboard by IBM also involved some very subtle and little-known technical changes in a computer's hardware, especially the system ROM. I recall receiving a letter from one Huggie who decided that he would not buy another Zenith computer because he could not add a 101-key keyboard to his Z-148 without the added expense of changing the system ROM. That kind of view illustrates some of the confusion surrounding the addition of the 101-key keyboard to Heath/Zenith computers. The problem also is a clear demonstration that there are some technical details that you must know about your computer system: in this case, the ROM.

When we looked at the 84-key keyboards earlier in this article, I pointed out that they were used with both the PC/XT and AT class computers. In both classes, the system ROMs were designed to be used with the appropriate keyboard which consisted of 84 keys. At the time these systems and their ROMs were designed and built, there was no such thing as the new 101-key keyboard. Obviously, those older systems and their ROMs were not able to handle the new keyboard because it did not exist. Although IBM originally introduced the 101-key keyboard in April 1987, many other computer manufacturers, including Zenith, did not introduce a similar keyboard until much later that year. Part of that delay was due to the ROM changes that all manufacturers had to make for the new keyboard, not to mention the design and manufacturing changes associated with the keyboard itself.

If you bought a Heath or Zenith computer before 1988, it is almost certain that you will also need to buy a new system ROM for your computer. One possible exception to that is if your system originally had a 101-key keyboard, the OmniKey/102 may work fine. But even that is no assurance that a Northgate keyboard will work with your system, as you will see in a minute.

If you have any of the older Heath or Zenith computers in the PC/XT class, which includes all of the 8088-based systems, it is almost certain that you will need to upgrade your system ROM to use a new keyboard. I have a local friend who



has upgraded the ROM in his '151 (to version 3.0 B), and he has been using the OmniKey/102 with his system for over six months with no problem. The point is that all older systems were originally equipped with the 84-key keyboard, and their ROMs must be changed to a newer version that handles the 101-key keyboard. Why is that true?

There are many reasons, but let's take the simple and obvious one. The old 84-key keyboard had only 10 Function Keys; the new 101-key keyboard has 12. ROMs made only for the old keyboard will not handle the two new Function Keys: F11 and F12. In general, however, ROMs designed to handle the new 101-key keyboards will also work just fine with the older 84-key units.

Unfortunately, it became quite clear last year that some Heath and Zenith computers originally equipped with a Zenith 101-key keyboard did not work with the Northgate OmniKey/102. This problem involved at least the Z-248 and Z-386 computers. What happened?

#### The OmniKey/102 Flap

Joe Katz and I had extensive discussions about the new Northgate keyboards, and he wrote about them in his "Mainstream Computing" columns that appeared in the May and June 1988 RE-Mark issues. His June 1988 column specifically mentioned the Northgate keyboard, and at the time the column was written, the statements made were completely accurate. By the time the column was actually published, however, Northgate had changed keyboard manufacturers, and the "new" OmniKey/102 did not work with the Z-248 and Z-386 systems, regardless of the ROM version. This "new" OmniKey/102 was different than the one Joe and I had originally tested, and although the original model worked fine in all our systems, the new one did not. In some cases, a "Keyboard not connected" error message was displayed when the system was booted; in other cases, the system would freeze with no error message at all.

Because I like to troubleshoot this kind of technical problem, Northgate kindly sent me a "new" OmniKey/102, and I was able to duplicate both of the reported problems. Without getting into too much technical detail, suffice it to say that there is also a "ROM" inside each keyboard, and these problems were apparently the result of a change to the keyboard hardware. The "Keyboard not connected" error was displayed when the keyboard failed the system diagnostics when the system was powered on. These diagnostics, generally called POST (Power On Self Test), generally check the system hardware for errors, and this includes memory testing, as well as making sure that a keyboard is connected.

The system freeze problem appeared to be slightly different. Once a system passes the POST diagnostics, system hardware is "initialized." If you have a 101-key keyboard, the initialization sequence includes setting the Num Lock to "On" in accordance with the IBM standard. Although I was able to isolate these two specific problems based on a lot of testing, I was never able to pinpoint precisely which problem was related to exactly what system and ROM version.

There was a big flap about this because a lot of people were not able to get the OmniKey/102 keyboard to work with their systems. And a number of people were quite displeased with Northgate because the "new" keyboard still did not work on their systems. To be fair to Northgate, I have never been completely convinced that it was all a Northgate problem — I suspect that some of the problem may still be due to a glitch in some Zenith ROM versions.

If you decide that you want to get the OmniKey/102 keyboard, I suggest one approach that should help you avoid most, if not all, of these problems. First, order a new ROM set for your Zenith desktop system, regardless of what model you have. New ROMs are available for virtually all of the older Zenith computers, and I have checked with a usually reliable Zenith source who tells me that all new ROMs support the 101-key keyboard. To

give you an idea of the upgrade cost, I have included two part numbers that are listed in the current Heathkit catalog at the end of this article. If the part numbers for your computer are not listed, you can still order ROMs for your system at the toll-free number shown, but be sure that you have the model number (from the label on the back of the computer) handy before you call. Even if you have a Z-386 computer, you still may find that you need the very latest ROM version to get the OmniKey/102 keyboard to work properly.

I think the OmniKey/102 keyboard falls into the "highly recommended" category, subject to making sure you have the latest Zenith ROM version in your system. If you decide that you need the OmniKey/102, I recommend that you order and install a new ROM before you order the keyboard. Although I believe that the currently available model of the OmniKey/102 will work with all Heath and Zenith computers with the latest ROMs, I can't make any ironclad guarantee because I have not tested everything. Fortunately, Northgate provides a 10 day satisfaction guarantee on the keyboard, so you can get a refund if you are not pleased. I think that is an excellent, and quite reasonable, guarantee; and if you have any problems, it will only cost you some shipping charges to try it.

However, there is one other interest-

#### ATTENTION US Government Employees Special Offer From Enable Software



For a limited time, US Government Employees can purchase a copy of Enable for home use at a special price of only \$100.

Enable/FE (2.14) is the same award-winning software program--with word processing, spreadsheet, graphics, database management and telecommunications--that most government employees now use in their office. Enable/FE includes all the powerful features and capabilities you are accustomed to working with day after day, with the exception of Perspective 3-D graphics.

To order your copy, send your check or money order, payable to Enable Software, with your name, federal government ID number, and the address and phone number of your office and home. Mail to Enable Software, Federal Buy, Northway Ten Executive Park, Ballston Lake, NY 12019. Specify 5 1/4" or 3 1/2" disks.

Shipping Costs included for continental US and APO, FPO addresses.

**enableFE™**  
Version 2.14

Reader Service #198

ing side effect that you may find if you upgrade the ROM in your computer, regardless of whether or not you also decide to upgrade the keyboard, too.

### ROM Upgrade Side Effect

When IBM announced the 101-key keyboard, it drove computer manufacturers nuts trying to keep up with the hardware and ROM changes. But most people don't seem to be aware that the required ROM changes also had a side effect on the "compatibility" of some software. This change also affected some software manufacturers, too, especially in one category generally called "keyboard enhancement" programs, such as Borland's SuperKey.

Because of the way that most of these "keyboard enhancement" programs work, many of the older versions of these programs simply will not work with the new ROM versions. In my '248, for example, I upgraded the ROM from 1.8D to 2.1A. This change was required so that I could run OS/2, and the later ROM also allowed me to use the OmniKey/102 keyboard, too. But I also had SuperKey version 1.15 which would not work with the new ROM, so I had to upgrade to SuperKey version 1.16A. Other keyboard enhancement programs, such as ProKey and SmartKey were similarly affected, and I imagine there are some others that I don't know about. If a program fools around with the keyboard on a relatively low level like these programs do, chances are they will be affected too, and you may need a later version.

### Powering Down

The weather in Texas has been really bad, and we recently had an ice storm that pretty much closed down the Dallas/Ft. Worth metroplex area on February 3 last Friday). Like the rest of the nation, we

have had some extremely cold weather in the last week, and I am certainly looking forward to May when it will be warmer. As I write this, we are still in the middle of some very frosty weather, at least for Texas.

For help in solving specific computer problems, be sure to include the exact model number of your system (from the back of the unit), 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 printer) 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 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, comment or request.

### Products Mentioned

POWERING UP (Book) \$12.00  
Heath/Zenith Computer Centers  
Heath/Zenith Users' Group

P.O. Box 217  
Benton Harbor, MI 49022-0217  
(616) 982-3463 (HUG Software only)

HS-386 Computer Kit \$3349.00  
1 MB Memory Board for Z-386 (Z-505) 799.00  
4 MB Memory Board for Z-386 (Z-515) 2999.00  
Cache Memory Board for Z-386 (Z-525) 599.00  
FTM Monitor (ZCM-1490) 999.00  
Heath/Zenith Computer Centers  
Heath Company Parts Department  
Hilltop Road  
St. Joseph, MI 49085  
(800) 253-7057  
(Heath Catalog orders only)

OmniKey/102 \$99.00  
Northgate Computer Systems  
13895 Industrial Park Blvd., Suite 110  
Plymouth, MN 55441  
(800) 526-2446

In the February 1989 issue of REMark, Bill mentioned a company called Lightgate, who marketed "Felix". This company has gone out of business, but Altra has taken over for them with this product. Their address is:

Altra  
1200 Skyline Drive  
Laramie, WY 82070



**Are you reading  
a borrowed copy of REMark?  
Subscribe now!**



### IBM XT in an H/Z-100

**Scottie Board W/ZPC \$149.**

#### OPTIONS AVAILABLE

- PC Compatible Serial Port W/Cable \$50
- 2nd Port W/Cable \$45.
- Clock/Calendar \$45.
- No Solder H/Z-100 MOD Kit \$5.

Requires H/Z-100 MS-DOS 768 of RAM, H/Z-100 Modification.

#### FACTORY NEW...THE REAL THING

- Z-217-1 W/DOCS, HDWE, CABLES, INSTRS
- With 20 Meg ST-225 Drive \$599.
- With 40 Meg MS3650 Drive \$729.

#### \*\*NEW\*\*

- Pair of AT Drives - External, Self Contained, Complete
- Uses 1.2 Meg AT Floppy Drives • W/Power, Cables,
- 34 pin to 50 pin adapter, Instructions - \$299.
- 34 pin to 50 pin adapter alone - \$23.95

IBM XT is a registered trademark of IBM Corp. • ZPC is a product of HEATH USER GROUP.

#### PC/XT/AT and WORKALIKES

- LOGITECH Scanman 4" x 10" Scanner - \$199.
- C7 Serial Mouse - \$ 79.
- HI RES BUS Mouse - \$ 89.
- Complete PC Hand Scanner - \$169.
- GRAVIS Joystick - \$ 39.

#### HARD DRIVES - All Makes/Models at BEST PRICES

- ADD ON CARDS, VIDEO, Controllers, Adapters
- Tape Backup 40/60 MEG, 2 MEGS/minute.
- Colorado Memory Systems DJ-10 \$299.
- VCR Tape Backup "IMAGER" \$199.

#### BEST PRICES ON:

- TURBO XT, 80286, 80386 Computers -
- BUYING SERVICE - any item you wish us to purchase COST + 10%
- GIVE US AN OPPORTUNITY TO BID ON YOUR NEEDS

#### Scottie Systems, Inc.

1609 S. Main St.  
Milpitas, CA 95035  
(408) 262-5021

VISA & MC ACCEPTED, ALL PAYMENTS SUBJECT TO APPROVAL



---

# BRIEF REVIEW of

## WORDSTAR PROFESSIONAL V5

**Richard L. Mueller, Ph.D.**  
11890-65th Ave. N.  
Maple Grove, MN 55369

### Background/Overview

It's that time again for all you Wordstar lovers. MicroPro has just released V5.0 of its Wordstar Professional word processing software package. Just as with V4.0 of Wordstar, I had the opportunity to Beta Test Version 5.0 as well. Last Fall in REMark, issue 93 (October 1987), I covered the features of Wordstar V4.0. Now I would like to cover Wordstar V5.0.

However, instead of briefly discussing all 300+ new features, I would like to cover just a few of the major features but in more detail, plus I will just mention some of the other features. This article would be too lengthy if each and every one of the new or enhanced features were covered in detail.

Previous releases of Wordstar Professional were aimed at the current Wordstar users. However, this release, with some major enhancements, is aimed not only at the faithful user but at new users as well, as you will see shortly. V5.0 of Wordstar is being released for PC-type compatible microcomputers, just as with V4.0.

MicroPro is abandoning us old H/Z-100 users. Pat Swayne, the HUG Software Engineer, took V4.0 and made a few patches to get it to work on an H/Z-100. Hopefully, he can do the same with V5.0. I still use my H/Z-100 a lot and have V4.0 installed on it. I am running V5.0 on both my Z-160 (two floppy drives) and my Z-386. I must give MicroPro a lot of credit for making this new version still work with two-floppy-disk systems. One cannot say this for most of the new versions of other software products. They require a hard-disk to run. Cheers for MicroPro.

Let me now cover some of the neat features of Wordstar Professional V5.0.

### User Interface

As all you old faithful Wordstar users know, there are 3 levels of Help within Wordstar. Help 3 displays the most information to the user and uses about a third of the screen. Wordstar Professional V5.0 adds a fourth level which is the new default. Level 4 adds the Pull-Down Menus

which are popular in many of today's software products.

The Pull-Down Menus cover file handling, editing, layout, style, calling other facilities such as TelMerge and MailList, etc. See Figure 1 to get an idea of what the Pull-Down Menus look like. According to the MicroPro documentation, these menus conform to the IBM SAA (Systems Application Architecture) Common User Interface specification. Although I have seen articles in magazines talking about SAA, I have never really seen a copy of the actual spec. So I'll take MicroPro's word that it conforms to that spec.

The main purpose, in my opinion, for adding the Pull-Down menus and Dialog boxes (yes, these are new too - more about this later) is to make Wordstar Professional easier to learn and use, particularly for users new to microcomputers. MicroPro simply wants to attract new users and at the same time keep us old die-hard current Wordstar users.

For current users, the familiar control sequence commands (classic commands as MicroPro calls them) are still there. They are always active whether you are using the Pull-Down Menus or not. In fact, when you pull down a menu, you will see the corresponding classic commands to the right of each of the menu options (see Figure 1). I mentioned earlier that the Pull-Down Menus were the default as Help Level 4. At anytime this can be changed by the user while running Wordstar to another Help level. Level 3, for example, is the default in Wordstar V4.0 and earlier versions. This level gives the user the familiar command screen. To permanently change the default to another Help Level, the user can use WSCCHANGE.

Connected with Pull-Down Menus, is another User Interface feature called Dialog boxes (boxes of various sizes that pop up on the screen when information is needed). Dialog boxes have replaced some of the Prompts in Version 4.0 and earlier versions. The one that I like best is the one for printing a document. In Version 4.0 (and earlier versions), the user

was asked a series of questions before printing the document. That is not bad except if you made a mistake. There is no way of backing up to a question already answered, except to abort and start the printing request over.

The Printing Dialog box lets the user see all the parameters at one time together with the defaults. The user can then change only those that need to be (see Figure 2). Once all the needed changes have been made, the file is printed as in earlier versions.

In addition to making Wordstar Professional easier to learn and use, the Pull-Down Menus free up more of the screen so more of the document can be seen at one time. With Help Level 3 selected, about a third of the screen is taken up by the command screens. The Pull-Down Menus only take up a Menu Bar line at the top of the screen till a menu is requested.

The Pull-Down Menus can help the new user to learn and use Wordstar, and the classic commands can then be used later for speed. As with any major enhancement, its tough to change. Some of you may learn to like the Pull-Down Menus, but for now I am still satisfied with the classic commands.

### Advanced Page Preview

MicroPro went all out when they implemented the Advanced Page Preview feature. This type of feature has been out for some time in other word processing packages, but for Wordstar Professional, this is new. Its called the "Advanced" Page Preview feature because it goes beyond what is available today. In fact, it's probably the best in the industry at this time.

This feature allows the user to view his/her document in many different ways. Without this feature, one could only display a document on the screen as part of a page. Scrolling up and down was required to view more of the same page or another page.

The Advanced Page Preview feature allows one to reduce the size of a page to

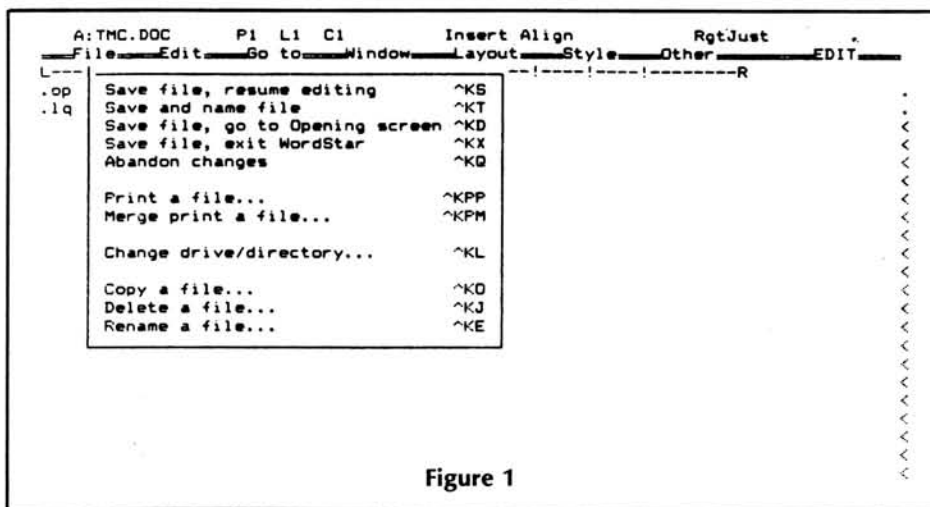


Figure 1

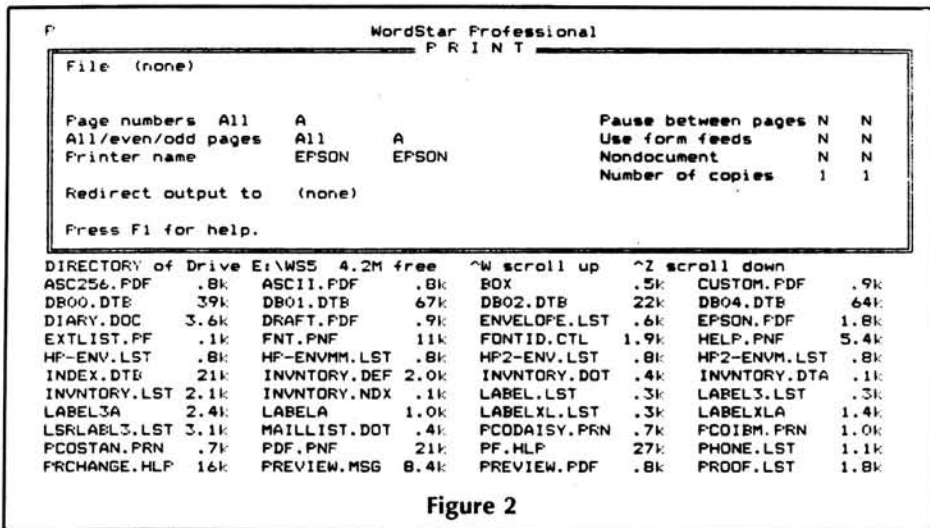


Figure 2

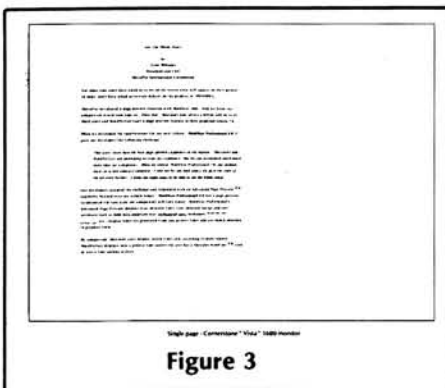


Figure 3



Figure 4

see the entire page on the screen at one time (Figure 3). I must apologize for the Figures for the Advanced Page Preview feature; I tried to get good screen copies but couldn't because of the graphics mode used. Instead of putting in my own screen copies of these Figures in this article, I received permission from MicroPro to use their Figures from the "What's New" booklet. Reducing the size of the document to see the entire document on the screen allows the user to see the placement of headers, footers, superscripting, sub-scripting, side-by-side col-

umns, spacing, different fonts, etc. Even on a CGA monitor, one can see facing pages on the same screen (Figure 4). Some of the text in either situation above may not be readable but it's the "overall look" that's important here.

Before I mentioned that, one can reduce a page to see an entire page at one time. The user can also do the opposite. One can use the "zoom-in" feature to enlarge or magnify a section of a document (2X or 4X the reduced size). The user can now see in much detail what the docu-

ment will look like when printed. Using a small window outline on the screen, the user can move it to select any part of a document to be enlarged.

The opposite of "zoom-in" is "zoom-out" which provides four different sized options. I already talked about the single page and facing pages. There is also a multiple page option, displaying 6 or more pages on the screen depending on the monitor and graphics adapter (Figure 5), and finally, there is what is known as a "thumbnail" view (Figure 6). using an EGA adapter and monitor, the multiple page option will display up to 6 pages at one time. Just as with the facing page preview, the reading is difficult as well. In fact, very little, if any, can be read but it's the overall look that's important.

The Thumbnail View displays even more pages on the screen than the multiple view option. Again this gives the user more pages to look at at one time. For an EGA monitor and adapter, this number is 18. For a Cornerstone Vista 1600 monitor, the number of thumbnail pages is 144. While viewing thumbnail pages, the user can select zoom-in to enlarge any of the pages to get a detailed look at the page.

One nice thing about MicroPro's implementation of this Page Preview feature is that it's FAST. One can scroll forward and backward through a document, scan only specified pages in a document, request a specified page, and exit to the page that you select. All the Advanced Page Preview functions that I mentioned above, can be selected from the Page Preview View Menu. However, to speed things up, MicroPro added some hotkeys or special command key sequences. These hotkeys allow the user to start and stop Page Preview as well as zoom-in and zoom-out. This feature deserves another applause for MicroPro. Another feature well done.

## Other Files

Wordstar V5.0 lets the user insert information from Lotus 1-2-3 and Symphony (both from Lotus Development Corporation), or Quattro (the new spreadsheet program from Borland International) into a Wordstar document. Wordstar V5.0 recognizes the file type by the file name extension and reads the file accordingly. For mail merging where the user has a list of names and addresses, in addition to Wordstar files, the user can use information in Lotus and dBASE III files (from Ashton-Tate). What I am trying to point out here is that Wordstar Professional V5.0 can now interface with other vendor products.

## Notes

Earlier I said that the Advanced Page Preview was probably the best in the industry, I also have to say that for the Notes capability. Included in this capabili-



ty are Footnotes, Endnotes, Annotations, and Comments.

First of all, let me say a few words about Comments. I would have to say in most cases, there isn't a need for comments in a document. However, when there is that need, Wordstar Professional V5.0 provides it. The purpose of this capability is to allow the user to specify some instructions or state some comments to another user who may be updating the document. The comments are only seen on the screen and NOT in the printed document.

To go a step further, comments are only displayed on the screen when selected by the user. What the user normally sees is a "Comment Tag" in the document. The user can then request the comments to be displayed.

Next there are Footnotes and Endnotes. When the user wishes to make a comment that WILL appear on the printed page about some item in the document, Footnotes and Endnotes are used. Associated with each of these notes is a number automatically assigned by Wordstar. Footnotes will appear at the bottom of the same page as the item referenced. Endnotes will appear on the last page.

The last note capability is Annotations. I look at this note capability as a special case of the Footnote. This is used in the case where several items in a table or chart refer to the same note at the bottom of the page.

Just as one can create notes, one can edit them as well. This means making changes to the note text itself, deleting notes, converting from Footnotes to Endnotes (and vice versa), and positioning where the Endnotes should be if other than on the last page. Fonts can also be set for the notes.

#### Additional Programs

TelMerge and Maillist are two programs tied to Wordstar Professional V5.0. By that I mean that they can be called from within Wordstar. However, both of these programs can also be called directly from the DOS prompt. This provides flexibility. I will not discuss these since both are currently in V4.0. Enhancements have been made to improve usability.

MicroPro also included two other programs with the release that are not callable from within Wordstar but currently are stand-alone programs, namely, ProFinder and PC-Outline. ProFinder is a utility that allows the user to manage and organize files and directories on a disk. Although ProFinder can be used with floppy disks, it is much more useful for hard disks.

Some of the features of ProFinder are: display all files in a directory, print list of files, start programs including Wordstar from within ProFinder, look at the contents of individual files, perform searches

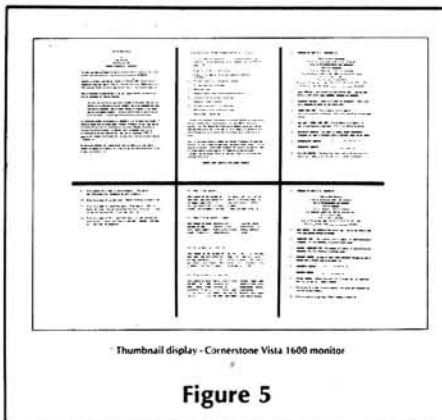


Figure 5

on text, attach a descriptive comment to each file, sort list of files, etc.

The other program is PC-Outline from Brown Bag Software. This program provides some of the best outlining capabilities that I have seen. Although PC-Outline was designed originally as an outlining program, it can provide many more tasks such as: project planner, daily schedules, simple word processor, scratch pad, lists of things to do, phone lists, etc. PC-Outline is called from the DOS prompt and it can run either as a memory resident program or as a non-memory resident. When running it as a memory resident program, it can be activated while running other applications including Wordstar V5.0. In this way, it's at your fingertips when it is needed.

#### Other Features/Enhancements

Earlier, I said that there was just too much to cover in a single article if all enhancements and new features were discussed in detail. Here I would just like to mention some of the other features and enhancements: there are over 30 new classic commands, the thesaurus has now been incorporated into Wordstar, files now can be saved automatically every so many minutes while editing, some printing enhancements including the print Dialog box discussed earlier and an increased number of printers supported, paragraph style guides have been included to increase productivity, for legal documents lines and paragraphs can be numbered, two documents can be displayed on the screen at the same time in separate windows, and the list goes on and on.

#### Customer Support

While other software vendors are starting to charge for support, MicroPro has taken the opposite approach and then some. They provide seven days a week support with a toll-free number. That's right, even Saturday and Sunday support. Now that is what I call "Supporting the User." I personally don't know of any other vendor providing such a service. MicroPro wants their customers

to be happy and when they are in trouble, help is there any day of the week.

#### Conclusion

This should give all you Z-Machine owners some idea of what the new release of Wordstar Professional has to offer. As I pointed out above, there are some major enhancements such as the Advanced Page Preview and Pull-Down Menus. There are improvements in existing commands, many new commands are added to make your life easier, new programs added, and last but not least, the super customer support policy. Try it, you'll like it. I'm enjoying the new features.

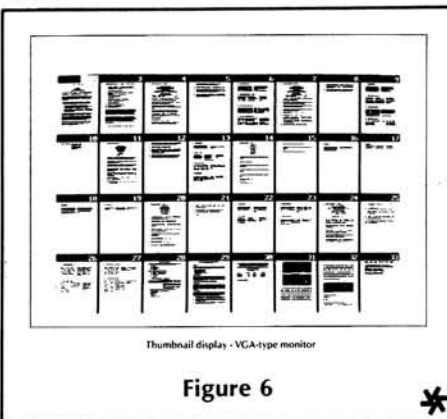


Figure 6



## ACCOUNTING & TAX

Not sure if you need the expensive 'Chinese Flower 1-2-3', or 'Spanish Numeral Four' spreadsheet programs? Then find out for only \$20! "CheapCalc" will do double precision addition, subtraction, multiplication, division, power, SUM, and roots (using fractional powers). CheapCalc has many other functions too numerous to mention (just like the expensive spreads)! CheapCalc is available for all Heath/Zenith computers and operating systems. For more information, check out page 58 of the Software Catalog Update #1, or call HUG and order your copy today.

# QUIKDATA - THE ONLY NAME TO REMEMBER!

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

## ACCELERATE YOUR PC!

From PC Technologies, the fastest and most proven way to really speed up your Heath/Zenith PC/XT computer, giving it -AT compatible speeds! Turn your turtle into a -286 rabbit with a 12Mhz 80286 accelerator board.

There are other ways to speed up your H/Z PC/XT computer, but you spend too much and get too little. The **286 EXPRESS** is the effective solution, making your H/Z150/160/150/159 series of computers, or any general PC/XT computer in many cases faster than a standard IBM AT type computer! Simple half-card plug-in installation with keyboard software switchable speed control.

Complete with 16-bit -AT -286 processor, CACHE memory for dramatic speed increase, 12Mhz 80286 processor, and software. **You won't believe your stop watch!**

**\$379 SPECIAL** (\$645 list)

## ACCELERATE YOUR 241/248

With the Aox Z-MASTER **20Mhz 80386**, your Z241/248 computer will be faster than the Zenith Z386 and more compatible. The Aox 20Mhz Z-MASTER 386 is a one-board CPU card that replaces the Zenith 80286 CPU board of the Z241 and Z248. With support for up to 8MB of onboard 32-bit memory and 80387 math co-processor on one single card. The Z-MASTER 386 comes standard with an efficient 8KB cache providing 0 wait state access to the onboard 32-bit memory and optimum access to existing 16-bit Zenith and conventional memory. The Phoenix ROM BIOS Plus is also standard providing a higher level of software and hardware compatibility. All existing 16-bit memory is accessed efficiently via the onboard cache controller. Installation is simple with only one board to replace, approximate installation time is only 10 minutes. See H-SCOOP #107, or request a copy for a report on this board, or inquire for more information.

**AOX386 - SPECIAL - \$1295** 0K RAM  
**APX386-1 - \$1695** with 1MB RAM

## 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/Z150/160 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** Allows you to upgrade your H/Z110/120 (new motherboards with p/n 181-4918 or greater) to 768K system RAM. Requires 27 256K DRAM chips.

## WINCHESTER UPGRADE KITS

**PCW20 - \$295** 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.

**PCW40 - \$449** 42 meg with 37ms Segate ST-251.

**PCW80 - \$695** 80 meg with Segate ST-4096 full size drive.

**EWIN20 - \$555** 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. We also have this in 42 megs (\$669), and 80 megs.

**We also have winchester systems for the H8 and H/Z89/90.**

## BARE WINCHESTER DRIVES

**ST-225 - \$229** 65ms half height bare winchester with 21 megabyte storage capacity.

**ST-251 - \$395** 37ms autopark half height bare winchester with 42 megabyte storage capacity. Good choice for -AT compatibles.

**ST-251-1 - \$429** Same as ST-251 but 28ms speed.

**ST-4096 - \$629** 28ms autopark full height bare winchester with 80 megabyte storage capacity. Best choice for -AT compatibles.

## PRINTERS

We have a full variety of Panasonic printers; narrow and wide carriage dot matrix, 24-pin dot matrix and daisywheel. We also have cables, buffers, ribbons, tractors when not included, etc:

<b>KXP-10911</b>	80 column 160 cps parallel std	\$225
<b>KXP-10921</b>	80 column 240 cps parallel std	\$359
<b>KXP-1595</b>	132 col 240cps ser/par std	\$489
<b>KXP-1524</b>	132 col 240 cps 24pin par std	\$579
<b>KXP-1124</b>	80 col 192 cps 24pin parallel	\$369
<b>KXP-3151</b>	132 col daisy 22cps par std	\$539

## 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.**

<b>H37 SOFT SECTOR CONTROLLER</b>	\$179
<b>H8 SOFT SECTOR/WINNIE CONTROLLER PACKAGE</b>	\$275
<b>TM100-2R</b> Tandon DS ST 48 TPI refurb disk drive	\$79
<b>H89 4MHZ</b> - Double your H/Z89/90 CPU speed	\$43
<b>Z47 REMEX DRIVES</b> slaves and masters - special	\$95

## H-SCOOP

Of course, don't forget about the best source of Heath/Zenith related information you can obtain - our monthly newsletter, **H-SCOOP!** Just \$24 for a 12-issue year (\$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.

POB 1242  
SHEBOYGAN, WI 53082-1242  
(414) 452-4172



# IMPROVING YOUR BUSINESS PRESENTATIONS WITH HARVARD GRAPHICS

EARL R. ZIMMERMAN, JR.  
169 SPINNING ROAD  
DAYTON, OH 45431

## Introduction

We all have heard how a picture is worth a thousands words. This is especially true in the business world. Often, managers are forced to present a large amount of data, that is easily understood, in an impressive format, in a short period of time. Even under the best of circumstances this is a difficult task. This task can be further complicated if the graphics software is inflexible and difficult to use. Fortunately, there is a software program that takes the worry out of meeting all the aforementioned challenges — Harvard Graphics 2.1, by Software Publishing. This article will briefly discuss the suggested system configuration and the most important features of Harvard Graphics.

## System Configuration

To take advantage of all program features, you need an IBM or IBM compatible hard disk computer with 2 megabytes of remaining hard disk space, 640K of memory, a monochrome or color monitor, and a supported plotter and/or a dot matrix or laser printer. While a monochrome monitor and dot matrix printer are more than enough for the average home user, I suggest a color monitor to take full advantage of the slide show capabilities, and a laser printer to produce graphics shop quality hard copies of your presentation. I also suggest a mouse because it makes drawing symbols and making other chart enhancements easier. At work, I use a Zenith 248, Zenith 1380 EGA monitor, an LN03+ Postscript Laser Printer, and a Logitech mouse.

## Features

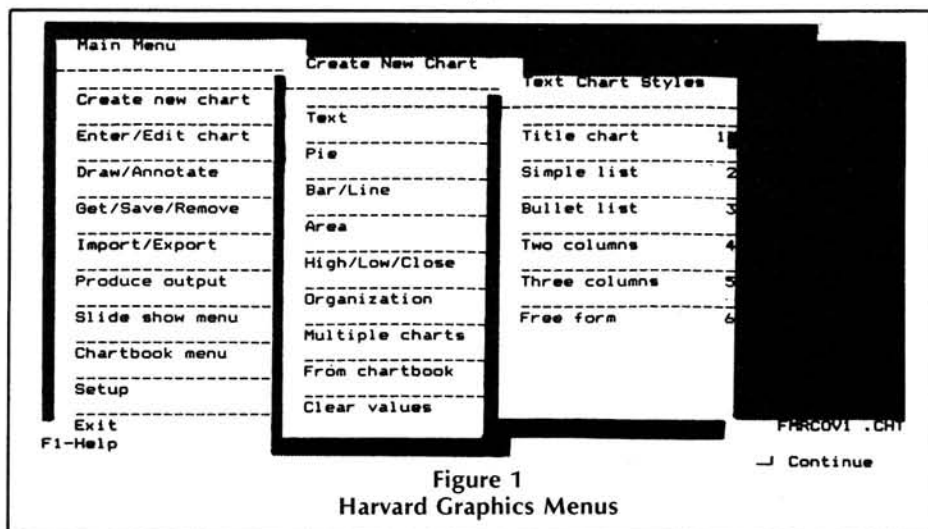
**Ease of Operation.** Harvard Graphics is a pleasure to use. To begin creating charts, you type "HG" at the DOS prompt in the subdirectory you created. The Main Menu then appears on the screen. You can select what function you want by typing the number or moving the bar cursor

over the function and depressing the left mouse button. All the available function keys are displayed at the bottom of the screen at all times during the program. All changes to charts can be accomplished through option screens, pop-up windows, and menus. At any point in time, you can view your chart by depressing the F2 key. On-line help is also available by depressing F1. See Figure 1 for how the menus appear.

give you an idea of Harvard Graphics capability, I've created eight sample charts for the fictitious ERZ Company. My fictitious company makes aircraft.

Figure 2 is an Enhanced Title slide. It was enhanced with the Draw/Annotate feature and symbol libraries which are discussed later in the article.

Figure 3 is a Bullet List Chart, which is a sub-chart of the text chart. You have your choice of circles, dashes, check



To give you an idea of how easy it is to use, the Aeronautical Systems Division at Wright-Patterson AFB contracts out Harvard Graphics training. A student, with no prior experience with the program, easily learns to create all types of charts in four hours.

**Large Variety of Charts.** Harvard Graphics is both text and graphics oriented. The type of text charts that can be created include: title charts, simple lists, bullet lists, two column charts, three column charts, and flexible free form charts. Graphics charts include: pie charts (simple and multiple), bar/line, area, high/low/close, organizational, and multiple. To

marks, squares or numbers for bullet shapes.

Figure 4 is an Enhanced Freeform Chart. I first typed the text and drew the boxes and lines with the Draw/Annotate feature. The arrows were created using the line option feature of Draw/Annotate.

Figure 5 is a Three-dimensional Pie Chart enhanced with a few symbols. Some of the features associated with pie charts are: (1) ability to cut out slices or hide slices, (2) linking pies, (3) creating proportional pies, and (4) sizing and rotating them.

Figure 6 is a Two Pie Chart. You have a choice of displaying data in a pie and/or



## FINANCIAL MANAGEMENT REVIEW



PREPARED BY: EARL R. ZIMMERMAN JR.  
DATE: SEPTEMBER 21, 1988

**Figure 2**  
**Enhanced Title Chart**

columnar form, as in this chart. This chart also shows how pies look when linked. A more advanced feature is adding text before or after your data. In this figure, I added the word "Aircraft", and the parenthesis around the data in the columnar portion.

Figure 7 is a Three-dimensional Bar/Line Chart. Variations on this chart include: cluster bars, overlapping bars, stacked bars, 100% bars, stepped bars (histograms), and paired bars. You can control the type of data table, tick marks, legend placements, and axis scale. You can also hide data.

Figure 8 is an Enhanced Organizational Chart. Data can be displayed horizontally or vertically. You can also shadow boxes and graph staff positions.

Particularly impressive is the ability to create multiple charts. A person could create a separate chart, of various varieties, and combine them into one chart.

See Figure 9 for an example of a Multiple Chart.

*Numerous Chart Enhancement and Drawing Features.* The basic charts can be enhanced to grab the attention of your audience. Using the Draw/Annotate feature, you can add text, lines, boxes and circles, or create your own symbols with polylines or polygons. These features can be modified (moved, sized, or placed in front or back of a portion of the chart), copied, and deleted. You can also add a grid that helps you draw straight lines or line up your enhancement.

The user can also control certain default chart features, such as chart orientation (landscape or portrait), chart borders, (none, single line, or double line) and font. Font selections include: Executive, Square Serif, Roman, San Serif, Script, and Gothic.

*Symbol Libraries.* Harvard Graphics contains 16 different symbol libraries that

## AGENDA

- SIGNIFICANT EVENTS
- REVENUE FORECAST
- EXPENSE FORECAST
- PRODUCT DEVELOPMENT
- PLANT RETOOLING STATUS
- EMPLOYMENT AND PAYROLL STATUS

**Figure 3**  
**Bullet List Chart**

total 200 symbols. Some of the symbols libraries are arrows, buildings, maps, currency, people, industry, office, and transportation. These symbols can be added to your graphs and modified. Portions of symbols, i.e., states in a map of the United States can be ungrouped, modified, and saved as new symbols. For instance, the ERZ Company logo in Figure 2, was created by ungrouping Pennsylvania from a map symbol and adding other symbols and text with the Draw/Annotate feature. While these symbols are quite useful, there simply aren't enough of them. Software Publishing should develop a utility program that converts scanned images, that are compatible with Pagemaker and Ventura Publishing, to CGM format. The user who doesn't have the time or talent to create their own symbols could add these images to the Harvard Graphics library.

*Spelling Checker.* To avoid the possi-

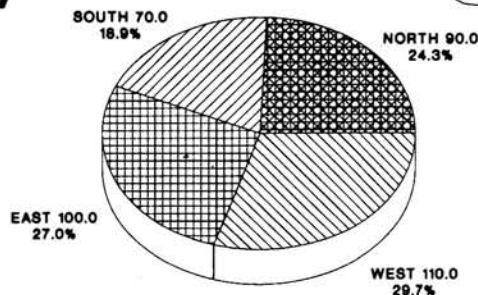
## SIGNIFICANT EVENTS (OCT- NOV 88)

EVENT	OCT	NOV
DAYTON AIR SHOW	↑	
ROLLOUT - 1000th ERZ-1		↑
STOCK HOLDERS MEETING		↑

**Figure 4**  
**Freeform Chart**



## AUGUST SALES



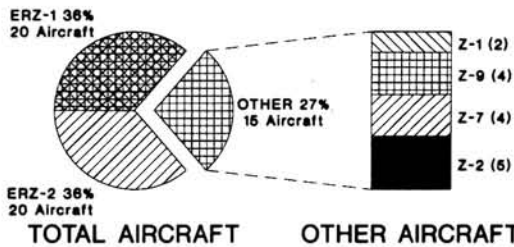
### SALES BY REGION

(\$ IN MILLIONS)

**Figure 5**  
**Enhanced Simple Pie Chart**



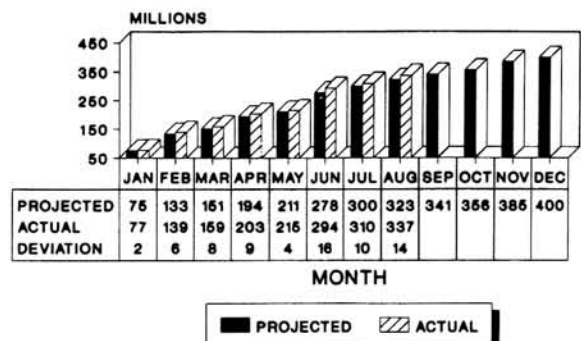
## TYPE OF AIRCRAFT SOLD (NORTH SALES REGION)



AS OF:31 AUG 88

Figure 6  
Two Pie Chart

## PROJECTED REVENUE- ERZ COMPANY (CUMULATIVE - FY 88)



AS OF:31 AUG 88

Figure 7  
Bar/Line Chart

## ERZ COMPANY ORGANIZATION CHART

AS OF:21 SEP 88

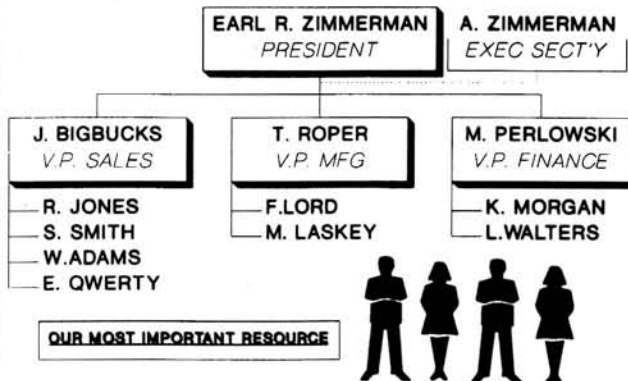


Figure 8  
Enhanced Organizational Chart

## SIGNIFICANT EVENTS (OCT - NOV 88)

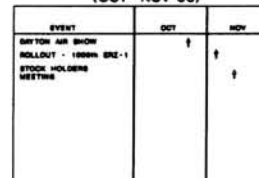
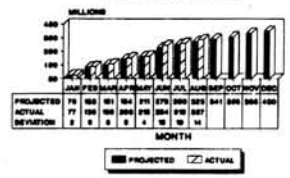


Figure 9 - Calendar Chart

## PROJECTED REVENUE- ERZ COMPANY (CUMULATIVE - FY 88)



AS OF:31 AUG 88



Figure 9 - Enhanced Simple Pie Chart

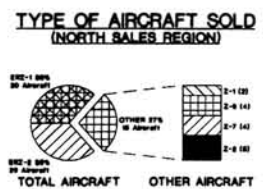


Figure 9 - Two Pie Chart

Figure 9  
Multiple Chart Example

ble embarrassment of displaying a slide with a spelling error, you can spell check a slide or an entire slide show. To spell check a chart you have created, all you do is return to the main menu and depress F4. The spelling checker works like those found in word processing programs. If the spelling checker finds a word it doesn't recognize, an overlay will appear and ask you if you want to continue (accept as is), add the word to the dictionary, or type in a correction. It will also suggest a word if it has one similar to the misspelled word in its dictionary. To choose one of the proposed actions or suggested words, place the cursor over the option or word and depress Enter.

**Import/Export Capability.** Harvard Graphics can import data from Lotus 1-2-3, ASCII files, PFS:Graph, and PFS:Professional Plan. It can also import files from programs that create CGM metafiles. Har-

vard Graphics can export files to PFS:Professional Write and Harvard Professional Publisher, Pagemaker, and Ventura Publisher. Through the use of supplied drivers, it can also export to Freelance Plus, Manuscript, and Super Image.

**Slide Show Features.** A major strength of Harvard Graphics is its ability to display and print up to 90 charts in an impressive manner. The slide show feature is used to build a briefing and to batch print slides. By creating a slide show, you can determine how well your presentation ties the main points together. Creating a slide show is particularly useful if you must brief someone before the actual presentation. You can add, delete, or rearrange your presentation to suit the individual very quickly. It also saves you time because you don't have to create a hard copy of your charts for these prebriefs.

Once the slides meet your satisfac-

tion, you can print out the entire briefing or any portion of it. You don't have to call up each individual slide and print each one separately. You can do other office tasks while the slides are printing. This feature is especially valuable if you are printing the full size charts in high quality, as each slide can take up to ten minutes to print out.

You can really dazzle your audience if you add special effects to your presentation. You can set how the chart should be drawn, the direction it should be displayed, and how it should be erased.

Some of the more unique ways to display charts are overlay, weave, rain, and iris. Overlay displays slides element-by-element. If the previous slide isn't erased, the next slide in the briefing will displayed with it. Weave entails the slide coming in from two directions at one time

SLIDE # NAME DESCRIPTION

8 MULTICHT.CHT MULTIPLE CHART EXAMPLE

This is an example of a practice card. I selected the four lines I wanted to display and the order I wanted them displayed in. You have your choice of doing this with three charts, four charts, or you can customize the chart to your liking. The information you are reading now is automatically word warped. You can type up to 11 lines. If you were preparing an actual practice card it would be best to use short bullet statements to jar your or the speakers memory.

DISPLAY TIME: 0:50

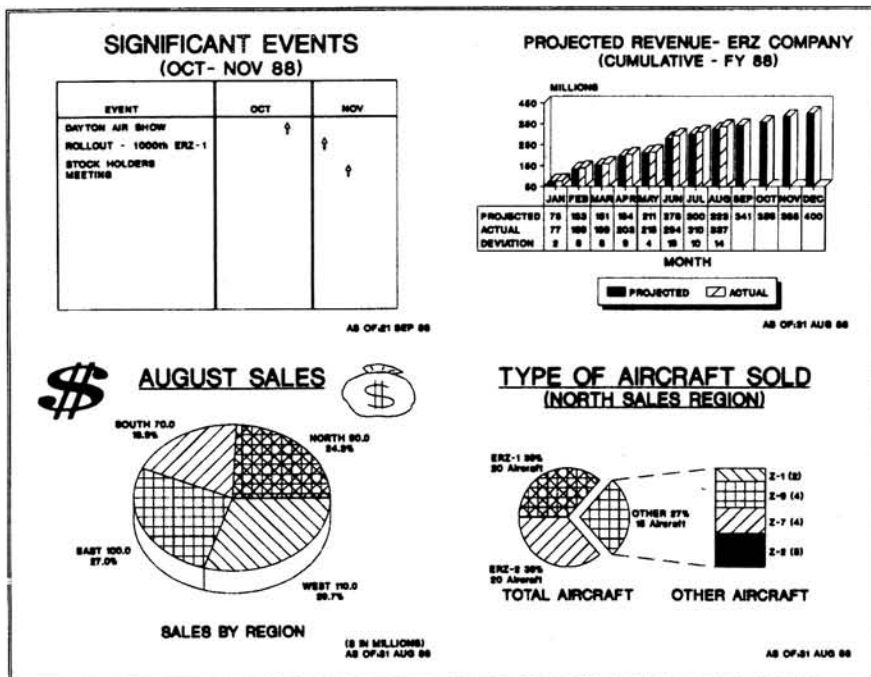


Figure 10  
Example Practice Card

to form the final chart. Rain gradually fades the slide on or off the screen, from the top down. It's sort of like rain forming a puddle. Iris closes the screen from all sides toward the center or opens out from the center to all sides. If your company is of the more conservative nature, you can select the replace option, which replaces the entire slide at the same time, you can scroll a slide off or on the screen, or just finally wipe it off the screen horizontally or vertically.

If you speak from index cards, or are preparing a briefing for someone else, you can create practice cards for the briefing. Harvard Graphics allows you to enter eleven lines of information about the slide. The practice card also tells the briefer the slide number, slide name, and how long the slide will be displayed if it is timed. It will also print out a half-size portrait picture of the slide. See Figure 10 for an example.

**Macro Capability.** Experienced users of Harvard Graphics may wish to take ad-

vantage of Harvard Graphics macro feature. Macros speed up routine operations by reducing the number of keystrokes. They are especially well suited for operations such as printing charts, selecting particular slides for update, and selecting what type of chart you want to create. Creating macros is done with the MACRO command. Typing MACRO causes an overlay to appear. MACRO is a TSR program that requires 60 K of memory. If you depress ALT-0, the macro menu will appear. If you wish to record a macro, you depress R and you are then prompted to enter a macro name. You then begin performing the tasks. Depressing ALT-0 and then S stops the recording of the macro and saves it. This macro can be played back, edited, or deleted at a later time.

You can save even more keystrokes by naming the macro A-Z or 1-9. The macro can be played back by depressing ALT and then the letter or number. It isn't necessary to go through the MACRO menu.

You can customize macros by creating variable fields or adding messages. In addition, you can change the rate of playback, specify a timed pause during playback, prevent Harvard Graphics from skipping keystrokes during playback by setting the polling delay, and set the time for a macro to begin running.

### Conclusion

Harvard Graphics 2.1 is a top rate graphics program. Its ease of operation, combined with its flexibility and data show features, makes it a wise choice for any business or personal user in the market for a powerful graphics program. \*

Continued from Page 4

HADES II no longer uses the disk header sector to obtain its internal information. This means that HADES II will now work with 'headerless' MS-DOS disks, such as those created with "Fast-back".

HADES II now has an option in the main menu to display all the important internal information about the currently logged disk drive. Some of this information includes: disk capacity, sectors per disk, sectors per cluster, number of FATs, starting sector numbers for the FAT, root directory, and data, bytes per sector, etc.

HADES II still works without modification on any computer that uses MS-DOS, including the Z-100!! NOTE: The Z-100 is still limited by MS-DOS to a maximum 32 megabyte partition.

The original abstract for HADES can be found in the January 1989 issue of REMark, on pages 34 and 35.

Older versions of HADES can be updated to HADES II by returning your original HADES distribution disk, along with a check or money order for \$15 to: The Heath Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. \*

Continued from Page 6

lieve that I overlooked it, but I can find no additional information on the Host Mode.

Aside from a few letters, my "old" H-89 is mainly used in conjunction with radio amateur operation. I am not saying that sometime I will never go PC compatible, but it does seem that, especially for radio amateurs, there is a blending of the old with the new and sales of new items may be dependent on support of the old. Of course, as I have an all Heath station, I would like to keep it all Heath.

So to the bottom line, is there somebody out there in the Heath organization, HUG or one of its contributors, that can come up with the required CW and RTTY software (CP/M) to make the HK-232 with the H-89 fully multimode operational? If this could be accomplished, then I would like to be the first customer.

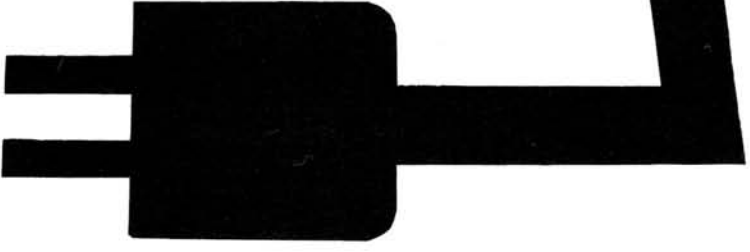
Sincerely,  
A.C. Stoddard (W8KLN)  
1502 Briarwood Road  
Lansing, MI 48917 \*



---

---

# POWERING UP



William M. Adney

P.O. Box 531655

Grand Prairie, TX 75053-1655

## How to Select Utility Programs

Computers can do so many things that it is impossible to describe them all. In the last article, you saw some suggestions on helping you select and learn how to use a major application program such as a word processor or spreadsheet. This article will deal with some specific suggestions on selecting what I call a utility program. To keep things simple, I have defined a utility program as any software outside of the major applications that we discussed last time. In addition, many utility programs are designed to provide single-purpose functions, such as disk organization and/or maintenance, within a major category, such as disk-related utilities. Other utility programs provide a solution to a common need such as an automated way to manage your checkbook or translate files between two word processor formats. Since you have already seen a general way to evaluate software in the last article, this article will discuss specific software that can help you solve a common problem. Let's begin with the most common problem of all.

### Recovering from a Mistake

Loss of data or data integrity in a computer system is a common problem. It can be caused by something over which you have no control, such as a power failure, or it may be caused by a mechanical problem with a disk drive. Despite what you may read or hear from the media, the most common cause is NOT the so-called computer "hacker" who deliberately sab-

otages a system. The most common cause of data loss or a compromise in data integrity is user error.

Rebooting the computer at the wrong time is one way to make an unintelligible hash out of a computer file, but the most common cause of data loss on any computer simply amounts to deleting the wrong file or group of files. This kind of user error occurs on all computer systems from mainframes to micros.

In a microcomputer system, of course, this is usually the result of what I call the "dreaded DEL \*.\* command." When this happens to you, it is usually because you forgot which drive and/or subdirectory you were in. Sometimes it happens because of a simple typo on the command line because you were in a hurry. Regardless of what mistake you made, however, the result is the same — a number of important files have been erased. It is inevitable that you need one of those files NOW, and the mistake occurred at the worst possible time. And because you were in a hurry, you did not have time to even make a backup of that important "new" file. This kind of thing happens to all computer users sooner or later. If it hasn't happened to you yet, it will. But now that you have erased that file, what do you do?

One way to recover is to go back and try to reconstruct the entire file or the changes you have made to an existing file assuming that some kind of backup is available. That is usually not a viable option because, in addition to being time

consuming in the first place, you may not be able to do it or you may be in a hurry because of a deadline. Fortunately, there is a better way.

Many software manufacturers have recognized this common problem and developed programs that can "unerase" or "undelete" a file or group of files. This works because the DEL command does not actually delete or overwrite a file's contents on the disk; it just "flags" a directory entry so that DOS can use the disk space that the file occupied. The only restriction in recovering an erased file is that you must do it IMMEDIATELY after the mistake or DOS may overwrite the disk area that actually contains the data.

No matter what other utility software you may be considering, I strongly recommend that your next purchase include a package that contains a program to recover deleted files. Two of the most popular and reliable packages are the Mace Utilities and the famous Norton Utilities including the advanced version. Although both have programs that can recover a deleted file, my personal preference is Mace Utilities because I believe it is slightly better, and Mace includes other programs that I often use on my hard disk. If money is no object, you might want to consider getting both because each has its own set of programs and features.

Another program that can recover deleted files is HADES — Hug Absolute Disk Editing System. In addition to recovering deleted files, you can also use this program to explore and change the contents

of a disk, change a file's attributes, and some other functions. HADES is an excellent program that is generally designed for those with a more technical interest and background.

Once you have a program that can help you recover accidentally deleted files, you can then move on to other kinds of useful programs. One extremely popular category is the desktop utility program.

### Desktop Utilities

In its most general form, a desktop utility usually provides a calculator, notepad, a phone list, and an automatic phone dialer that can be used if you have a modem. Most of the desktop utility software is usually referred to as a Terminate-and-Stay-Resident (or TSR) or more simply as a memory-resident program. In case you are not familiar with TSR programs, here's how they usually work.

First, you type in the appropriate command, and this loads all of the program into memory. Unlike most other programs, a TSR program STAYS in memory until you do something specific like reboot or power-off the system. The whole idea behind a TSR program is that you can use it in addition to a specific application. For example, let's say you were using a word processor that did not have a capability to perform any math calculations (most of today's word processors have this). You might be working on a document that contained some numbers that you needed to perform a simple calculation on: add, subtract, multiply or divide. With a TSR desktop utility, you simply press a "hot key" that "pops up" a calculator inside your word processor, perform the calculation, and enter an exit command to return to your word processing. The real advantage is that you do not have to exit from whatever you are doing and return to the DOS prompt to run another program — a TSR can normally be used in addition to an application program.

The entire market for desktop utility software was created single-handedly by a company called Borland which released the famous SideKick program. The original SideKick included the usual features: a calculator, notepad, phone list, and a dialer. In its latest release, called SideKick Plus, Borland has significantly improved the program and added lots of other features. Perhaps it is interesting to note that the notepad commands and features have been enhanced to the point that I am told many people don't even use a word processor for most documents — the notepad has all of the usual features and capabilities that most word processors have.

SideKick Plus also has a powerful file manager that helps in disk maintenance by providing a command interface that can copy, move, delete, rename, and

print files; change a file's attributes, FORMAT a disk, and provide a disk directory listing in various sorted orders. The notepad includes all kinds of features including block manipulation, search and search/replace, and even has a sort command. The "outlook" function is a full-featured outline processor that provides a unique way to develop and format an outline. "Phonebook" maintains your names and addresses, and can dial a number if you have a modem. SideKick Plus' Time Planner function is excellent and is ideal if you are using a laptop computer — you don't need to carry a separate schedule or diary because all of the necessary functions are provided. And there are four calculators (business, scientific, programming, and formula) that provide the capability to perform just about any kind of function you will ever need. It displays an "electronic tape" too.

It should not surprise you that a program which has all this power REQUIRES a hard disk. Three program files require about 1.3 MB of disk space, and there are other files, including some examples, that push the total requirement to about 1.7 MB of disk space. A minimum amount of 384K of system memory is required, although the default requires about 72K when loaded. You can also configure SideKick Plus to use either extended or expanded memory. SideKick Plus is an excellent program and is highly recommended, especially if you are using a laptop computer that has a hard disk.

Because I have introduced the subject of memory-resident programs using SideKick Plus as one example, there are a few things you should generally know about programs that generally fall into this category: how to troubleshoot problems that seem to be related to any TSR program.

### Troubleshooting TSR Problems

TSR programs have two specific advantages that I have already mentioned: they can be activated while you are using another application, and they are fast. But, you may find an occasional problem that occurs when you use any memory-resident program, and sometimes the causes of these problems are difficult to pinpoint. Let's take a look at how to isolate the cause of some common problems that you can have with memory-resident programs and how to fix them.

Even if you have a maximum of 640K memory in your computer, you can still run out of memory. When this occurs, you may find that an application program suddenly refuses to load completely (or at all), and your system may freeze. The only way to recover is to reboot or power-off the system in this situation. Sometimes this occurs with no warning on an application that you may not have used for some time, and you can't figure out why every-

thing did not work as before.

There are five major causes of system problems that can occur when using a TSR program that are listed in Figure 1.

1. "Insufficient memory"
2. Conflict with other TSR program(s)
3. Conflict with application program
4. Program bug in TSR or application program
5. System/DOS version incompatibility

**Figure 1**  
**Major Causes of System Problems with TSRs**

In many cases, the root cause of the problem is that you have "insufficient memory" to load all of the programs. Here's the way to begin troubleshooting the problem.

First, reboot the computer, and enter the command to load the TSR that seems to be causing the problem. Run CHKDSK, and make a note of the "bytes free" for memory from the CHKDSK display. Then, divide that number by 1024 to obtain the number of kilobytes of remaining memory that can be used for your application. Refer to the application program's manual to find the memory requirements for that program. If you have more than enough free memory to run that application, then there is another problem. If your calculated memory capacity is less than the application program's requirements, your system simply does not have enough memory for all programs, and you can try to free up some memory. In either case, the next step is the same, but let me take a minute to discuss the reason for doing these steps.

When an application program does strange things — causes a system freeze, does not respond normally to commands or generates odd error messages at strange times — there are a number of things that can cause that problem. When that occurs, it's time to get back to basics. Memory-resident programs in particular can cause all kinds of strange problems. Memory-resident programs are also called TSR programs because they use the DOS Terminate-and-Stay-Resident function. TSR programs are probably the most common cause of weird problems.

These problems can be kind of tricky to find because they are usually installed in an AUTOEXEC.BAT file, and you may forget about them because the ECHO OFF command was included in the batch file. For that reason, I suggest you be especially cautious when installing ANY new software on your system. Sometimes the installation program will ask you if you want it to update your AUTOEXEC.BAT (or CONFIG.SYS) file for that new application. Sometimes an installation program will update one of those files without asking (and without your knowledge). If



asked, I usually allow an installation program to update one or both of those files, and after the installation is complete, I use the TYPE command to look at both of them to see what changes were made if the file date has been changed. Because the changes are sometimes not as obvious as a new line in the file (e.g. a new path added to your existing PATH command), I compare the "new" file with a backup of that file or a printed listing. Be sure to keep some kind of record of the current contents of your AUTOEXEC.BAT and CONFIG.SYS files because you never know when they will "disappear."

Even more obscure and trickier to find are the problems caused by various device drivers that are implemented in CONFIG.SYS in the DEVICE= commands. I once spent hours trying to find a problem that was caused by a conflict with the Microsoft Mouse device driver (MOUSE.SYS) that I had installed in CONFIG.SYS. The point is that you should never assume that anything works right with anything when you are trying to troubleshoot one of these strange problems.

So, the next step is to get rid of everything at once to see if the software does work in a plain "vanilla" system without the AUTOEXEC.BAT and CONFIG.SYS files. The easiest procedure is to RENAME AUTOEXEC.BAT to AUTOEXEC.SAV, and COPY CONFIG.SYS to CONFIG.SAV. This gives you a backup copy of these files that you can easily restore with the COPY or RENAME command when you are finished with your testing. Now edit the CONFIG.SYS file and delete ALL lines containing the DEVICE= command. Save the file and reboot the computer. Be sure the suspected TSR is loaded, and try your application program again. If you had a mouse device driver, such as MOUSE.SYS, you will find that the mouse does not work, but everything else should. The idea behind deleting all of the DEVICE= commands is that each represents a device driver which is permanently loaded into memory each time the computer is booted. In other words, a device driver is also a TSR or memory-resident program too, and the objective is to reduce the problem to the suspected TSR and application program.

In most cases, you will now find that the suspected TSR and application program work fine together, and the basic problem is resolved. If they don't, I suggest reading the manual (and the "README" file on disk, if any) for the TSR program again to see if there are any known problems with any specific application programs. You may find that there is a problem with a "hotkey" conflict because the TSR's hotkey is also a command key for that application. Most of today's TSR programs allow you to change the hotkey for that reason. If read-

ing the documentation does not enlighten you as to a possible cause of the problem, then contact the TSR's manufacturer for some technical support. Be sure that you have already gotten rid of potential problems with the AUTOEXEC.BAT and CONFIG.SYS files as mentioned earlier before you contact the manufacturer of the TSR. They should be able to advise you if there is a known conflict or bug in your application, a bug in the TSR or a special problem relating to system hardware/DOS version incompatibility (problems 3, 4, and 5 in Figure 1).

If you have gotten this far, you KNOW that you have some kind of conflict between the suspected TSR and at least one other memory-resident program you are using. Now you can systematically isolate the problem to see where the specific conflict is. The easiest way to begin the process is to "restore" the CONFIG.SYS file to the original. Here, you can just COPY CONFIG.SAV back to CONFIG.SYS. Reboot the computer, load the suspected TSR, and start the application again. Take some time in testing this combination to be SURE there are still no conflicts. If the problem does occur, then you know that there is a conflict between the suspected TSR and at least one of the memory-resident device drivers. If you have only one DEVICE= command in the CONFIG.SYS file, then you know the suspected TSR has a conflict with it. It's probably time to call the manufacturer's technical support for some help, but before you do, take a minute to consider the problem. For example, if the purpose of the TSR is to provide disk caching for better system performance, and you already have a DEVICE= command that implements ZCACHE.SYS (in current versions of Zenith MS-DOS), it should be obvious that you cannot have two memory-resident programs which perform the same function. It's almost a sure bet that they will conflict with each other, and you will have to choose one or the other.

When you have more than one DEVICE= command in the CONFIG.SYS and the problem still occurs, edit the CONFIG.SYS file and delete the DEVICE= command lines until only one remains. Reboot the computer, load the suspected TSR, and start the application again. Again, take some time in testing this combination to be SURE there are still no conflicts with that device driver. If the problem does occur, then you know that there is a conflict between the suspected TSR and the last DEVICE= command in the file. If the problem does not occur, restore another DEVICE= command and test it. Keep doing this until you finally have the complete CONFIG.SYS file that you originally had. If you did find a problem with one of the device drivers, it's probably time to call the manufacturer's technical support for some help as be-

fore, but before you do, take a minute to consider the problem and follow the procedure suggested in the previous paragraph.

Since you have now thoroughly checked out possible problems with memory-resident device drivers in the CONFIG.SYS file, the problem is almost certainly caused by some command in the AUTOEXEC.BAT file. Now you can use the RENAME command to rename AUTOEXEC.SAV to AUTOEXEC.BAT. Use an editor to display the file, and examine each line carefully to see what memory-resident programs are loaded during the AUTOEXEC batch file processing. Lines containing batch file commands (e.g. ECHO, REM, PAUSE, etc.) can be left alone as can any of the DOS built-in commands (e.g. PATH, PROMPT, DATE, TIME, etc.). For all other command lines (i.e. memory-resident commands) you find, comment out each one by inserting a REM command at the beginning of each line. That includes DOS commands such as GRAPHICS, MODE, and PRINT, and ZSPOOL because they also have memory-resident code. If you have a "MOUSE" or similar command in the AUTOEXEC.BAT file, you may want to leave it alone for now so that your mouse will still work with an application.

Save the edited file, reboot the computer, load the suspected TSR, and start the application. There should be no conflicts because you have only run standard built-in DOS commands, and a mouse command of some kind should not cause any conflicts in any program. Follow the same procedure as before — test your application thoroughly, read the manuals, and contact the manufacturer of the TSR if you have a problem.

Now you can systematically "restore" each line in the AUTOEXEC.BAT file by simply deleting the REM that precedes the next TSR command in the file. Save the edited file, reboot the computer, load the suspected TSR, and start the application. Follow the "standard" procedure of testing your application thoroughly, read the manuals, and contact the manufacturer of the TSR if you have a problem.

When you find the TSR command that is apparently causing the problem, don't stop there. Comment it out with a preceding REM command. Then, continue restoring the TSR commands on a line-by-line basis, and test each time, just in case there are TWO existing programs that conflict with the suspected TSR so that you can report both problems to that manufacturer. This kind of testing is pretty boring, but it is about the only way you can isolate the problem. In virtually all cases, using this procedure, or one similar to it, will allow you to isolate the problem TSR or application programs. Even though you won't be able to fix the problem, you can probably figure out a way to work

around it. Now let's return to the original subject of choosing utility software.

### Calculating

Maybe you don't need a program that has all of the features and capabilities of SideKick Plus. Perhaps all you need is a calculator that can be used for just about anything such as quick calculations when you are writing or programming. If that is what you are looking for, then HEPCAT — the HUG Engineer's and Programmer's CAIculation Tool — is one of the best programs you can find anywhere.

HEPCAT is a memory-resident calculator that provides the usual capabilities to perform engineering and programming calculations. HEPCAT provides standard floating point calculations for sine, cosine, tangent, arc-sine, arc-cosine, arc-tangent, and natural and base 10 logarithms with anti-ln and anti-log functions. As you would expect, HEPCAT also supports scientific notation.

Programming calculations and conversions can be done in various radices: decimal, hex, and binary. Logical operations in binary such as AND and OR are also supported. And you can easily convert from English to metric units using HEPCAT's built-in functions. In short, HEPCAT is a neat program that can help you perform all kinds of calculations without the need for a desktop calculator.

### DOS Shell Programs

One of the functions of a shell program is to help "insulate" you from all the vagaries of the DOS commands and their many options. A good shell program can help you perform various tasks much faster because they usually contain commands and functions equivalent to the COPY, RENAME, DEL, MD, CD, and RD commands to mention a few. Some shells provide far more than that, and I have used one called AutoDex for years because it allows me to enter a 40-character description of each file. Unfortunately, it is no longer available, and the version I have been using does not understand how to use subdirectories, which makes it more and more awkward to use.

One advantage of a shell program is that it can help you perform disk maintenance. It also makes it easy to create regular backup files which was one of my primary uses of AutoDex. Virtually all of the shell programs use the "point and shoot" technique which basically allows you use the cursor keys to move a highlighted block to "point" to a file name in a list of files on that drive or subdirectory. Then you can easily execute a command (or "shoot") against that file name. To give you an idea of the general capabilities of shell programs, I will mention four with a short description of their capabilities.

If you have any trepidation about

learning some of the DOS commands, like COPY, DEL, and RENAME, you will find that the HUG File Manager (HFM) is an extremely useful shell. Its primary claim to fame is that it displays all of the files on the disk (or DOS subdirectory) and allows you to perform some easy file maintenance. Since the concept is relatively simple, I will take a moment to explain.

HFM presents a simple screen with a list of all files on the disk (or subdirectory). One file shows up in reverse video, which is a marker (the "point") that can be moved with the cursor keys. As an example, you can copy that file by pressing C (the "shoot"), and the program asks for the destination disk and/or subdirectory. But what if you want to "shoot" a number of files? HFM also allows you to "flag" any of the files (indicated by an asterisk on the screen). You then press C as before and specify the destination as usual.

HFM also has a number of other features. You can also create and delete subdirectories, Flag (multiple files), ID (for disk label), Newdisk (change disks/drives), Print, Quit (HFM), Sort (file names), Type, and Unsort (file names). This program is not memory resident.

WindowDOS is a memory-resident program that provides all kinds of features in addition to those mentioned in HFM. WindowDOS is also a file manager that has lots of additional features. Although it can be run like any program, it was designed to be memory-resident so that you can use its features any time. Once loaded, it is activated by CTRL-INS that displays a directory. The cursor keys are used to move around in the alphabetically sorted file list that also includes subdirectories. As each entry is highlighted by the cursor key movement, the top of the display shows the file's creation date, time, and attributes. You can change subdirectories by simply moving the cursor key to a directory name and pressing RETURN. This is an invaluable feature for moving through a number of subdirectories on a hard disk so that you can delete unwanted files.

WindowDOS also includes commands that can be executed by a single letter such as COPY, DIR, ERASE, FORMAT (floppies only), GLOBAL, LIST, MKDIR, RENAME, SORT, TREE, and VIEW. These commands have some really neat features. For example, ERASE will not only erase a file, it will also "erase" a subdirectory name when it is empty like DOS should have done. All 10 Function Keys are used for various purposes including an on-line help function. If you have a hard disk, you can even use WindowDOS to back up files with no problem because it can copy files to multiple disks. You can easily "tag" multiple files for any function (like copy or erase) by pressing the plus (+) key on the keypad. All in all, a nice

program that is useful to just about any computer user.

In the last year or so, DOS shells have become more popular, and the Norton Commander is one of the newest ones. It is not a memory-resident program, but it can act like one. One of its nicest features is a "split screen" that can display directory listings for two drives or subdirectories side by side. You can also "Point and Shoot" data files by establishing what program should be executed for that file based on its extension. For example, SuperCalc would be executed against any file with a CAL file type or Microsoft Word could be executed for any file with a DOC file type. This setup is accomplished by including each file type and its associated extension in the NC.EXT file. You can also set up a custom menu for your applications or DOS commands in the NC.MNU file.

I have been looking for a replacement program for the AutoDex software that I mentioned earlier, and I finally found an excellent replacement in an unexpected place. I recently purchased the WordStar update to version 5.0, and a program called ProFinder was included. In addition to the expected shell commands, ProFinder also allows you to easily define a custom menu using what are essentially batch file commands in a file called USERMENU.PF, and you can even define submenus if you wish. It also contains a "Locate" command that searches for text much like GOFER does (see below). But the feature I like best is that it allows you to enter a 39-character file description for each file so that you can type in a reasonable description of a file's contents. This is a dynamite program, but unfortunately, it is only available with WordStar version 5.0. If there is enough demand, perhaps MicroPro will sell this program separately.

### Fixing the Keyboard

Many of today's applications have the so-called "macro" feature that allows you to enter a series of that application's commands or a literal text string by typing only a couple of keystrokes. For example, if you have a general form letter, you could store it in a special file and display it on the screen by some unique command sequence such as ALT-F. To generate a letter, you could then just press ALT-F, type in the appropriate information (e.g. recipient's name and address), and print it. In this context, the ALT-F command is called a macro, and it can save you LOTS of typing. You can also use a macro to do repetitive and/or complicated tasks to further ease your chores. It took some software manufacturers a long time to recognize that the macro capability was something that should be included in their software.

There is a similar problem related to



the keyboard, especially the Function Keys and other special purpose keys. Even as recently as a couple of years ago, many software manufacturers seemed to think that their specific assignments of the Function Keys and other special-purpose keys were "perfect" for all users. Some still do. And when IBM released their so-called "enhanced" 101-key keyboard, many touch typists (including me) were extremely disappointed by the poor design layout of the keys. To this day, a lot of touch typists still dislike that IBM-style keyboard because it is clumsy. You can find considerable evidence of that dislike by looking at various keyboard advertisements. Fortunately, there are a couple of programs that can help fix these problems.

One way to fix the keyboard problem is to effectively "move" (or remap) the function key and other commands around so they are the best for you. For example, Word Perfect assigns the "Block Mark" command (used for marking blocks) to ALT-F4. To me, that is really clumsy because I use it so often. Why not have the Block Mark command as simply F4 and have the Indent function as ALT-F4. Although many current versions of these applications, including Word Perfect and WordStar, allow the user to assign Function Keys as desired; some still don't.

If you are having this kind of problem, you may want to consider a keyboard remapping program such as HUG's KEYMAP. KEYMAP is implemented by using the KEYCON program which allows you to customize the function keys to your liking. If you simply want to use the program, several configured versions are also provided on the disk which include variations for WordStar, Basic, and DOS commands. Since some programs do not "like" resident utilities, the UNMAP program provides a way to unload the KEYMAP program from memory. If you want to be able to reconfigure your function keys and commands, this is an excellent program. It also will give you some insight as to how this is actually done which is valuable just for general information. Most of them allow a maximum of 20 characters input, which is more than enough for most uses. But there are some problems that KEYMAP can't solve.

Perhaps the worst problem is that 101-key keyboard that I mentioned earlier. For at least 10 million computer users, the CTRL key is in the wrong place, and the Caps Lock key is now in that location. Indeed, one keyboard advertisement mentions that "the CTRL key is where God intended it to be." Although you can buy a new keyboard for \$100 or so, it would be nice to have some software that could change it.

Fortunately, Borland recognized all of these problems a long time ago, and they

can be easily corrected with SuperKey. SuperKey is a keyboard "enhancement" program that does an incredible number of things. In addition to saving macro commands and remapping the keyboard, it has a data encryption feature for file security, a screen saver that shuts off the CRT display to prevent "burn in", a keyboard layout editor that allows you to set up the keyboard to your preference (e.g. switch the CTRL and Caps Lock keys), and an easy way to include a customized "help" system for any software.

SuperKey is a memory-resident program that I have used for several years, and I have found it compatible with just about all software except the Samna word processor. You may recall from the last article that I mentioned, an easy way to learn new software is to create a cheat sheet. After I have spent the time doing that, I usually type that information into a SuperKey "help" file so that I can call it up any time. I also use SuperKey to set up all of the function keys in both WordStar and Microsoft Word so they are the same, or at least as close as possible. That makes it easy for me to switch between those two word processors because I don't have to remember a lot of different commands. And if I forget something, I can always call up my help file to find a command or function. Even though WordStar 4.0 and later makes it easy to customize the Function Keys, I still use SuperKey because I don't have to change them inside WordStar; I just use the appropriate SuperKey file. SuperKey is an excellent program that solves a number of problems. Perhaps you will find it as useful as I have.

### Finding Things

Studies show that the most common application used on a computer is word processing. The power of word processing is that it makes the written word easy to edit and revise, but it doesn't take very long before you have created a lot of files containing those documents. Unfortunately, the DOS limitation of eight-character filenames and three-letter extensions does not readily lend itself to finding specific information in a file, even if you have a standard file naming convention. Perhaps you wrote only one letter to an individual about a specific subject, but you don't have the foggiest notion as to what the file name is. Now, you need that information quickly, and the only apparent alternative is to look through every file on your system. If you keep your floppy disks organized by subject, you may be able to isolate the file to 100 or less. If you use a hard disk, the sky may be the limit in terms of time required to locate a specific file.

There is an easier and faster way. If

you can remember a specific phrase or a couple of keywords — name, subject, date or whatever — you can locate that "missing" file with a program called GOfer. All you need to do is specify a phrase, or a keyword or two, and GOfer will "go for" a match on those items. GOfer can look for various combinations using the and/or/not Boolean operators, and the closer your specification, the more exact the search. As GOfer finds a match, he can display the document on your CRT so you can quickly scan it to see if that is the one you are looking for.

GOfer can be used with most popular word processors such as Microsoft Word, WordStar, and Word Perfect as well as other programs like dBase III, Framework, Symphony, and SideKick. It is usually run as a memory-resident program, but it can be run in a "stand-alone" mode too. Unlike some similar programs, you don't have to create a special "index" for use with GOfer to perform the search functions — you can use your word processing files exactly as they are. GOfer is a very flexible and useful program that can help you find things in your word processing files.

### Miscellaneous Information

I used to keep lots of "important information" on various scraps of paper at my desk. Although I tried to keep these paper-clipped together, I usually found that the specific piece of paper with that critical information had already been eaten by the information monster, and I could not find it. It may have been an important name and phone number that I had not entered in my phone list or it may have just been a time-critical item on my running To Do list. Or, I may have found some hardware or software that I wanted to include in my next column. Regardless of what it was, it had a tendency to get buried to the point that I forget about it totally or forgot it until I needed it. Or it just disappeared.

Trying to keep track of information is one thing, and trying to find it WHEN you need it is another. I have found a program that can help you cope with all of these need-to-know (and FIND!) tidbits of information: TORNADO. This program is basically a mini-database that allows you to record all kinds of free-form (i.e. text and numbers) information in various files that are equivalent to scraps of paper. You can look through each file, top to bottom, and each note is displayed as a window. When you need information from one of those notes, TORNADO has a search function that can scan a file for any information that is recorded on a note.

Like its physical paper counterpart, you can "bury" a note in the bottom of the pile, mark it as "important" or throw it away. You can even use TORNADO as a "clipboard" to grab information from a

foreground program (e.g. a word processor or spreadsheet) to store as a note or you can copy information from a note to a foreground program. TORNA-DO is a memory-resident program that requires about 20K of memory in its default configuration, and it can store a maximum of 25,000 notes. It comes with a few example files to help get you started "Torna-doing." An extremely useful and highly recommended program that can help you keep track of all that random information that has become so important today.

### File Translation

One of the major problems I have found is the situation where I need to use many different word processors. Even though the disk format can easily read by any DOS-based computer, nearly every major word processor has its own unique FILE format. That means a file generated by one word processor cannot usually even be read by another. And although every company seems to have a "standard" word processor, many users do not follow that standard because they prefer another word processor. The problem is: How do you translate one word processor's file format to another?

I have looked at and used a number of file translation programs, and the best one I've found is Software Bridge. I have used this program to regularly translate files to and from WordStar, WordStar 2000, Word Perfect, Sprint, Microsoft Word, DisplayWrite, MultiMate, Samna, Sprint, DCA/RFT, and DCA/FFT. Virtually all formatting (e.g. bold, underline, hanging paragraphs, columns, etc.) is preserved, although "comments" (called hidden text in some word processors) sometimes are translated as regular text depending on the source and target word processors. In a heavily formatted file, some slight "cleanup" may be required, but it is minor and can be done quickly. In addition, the program can also translate files for Wang PC, CEOWrite, WordMArc, Navy DIF, Volkswriter, and DEC/DX, although I have not tried these translations. . . yet. If you need to solve the file translation problem between any of the word processors mentioned, this program is highly recommended.

### Watching Your Checkbook

There are a plethora of programs available that can help you keep track of and balance your checkbook. I have seen a lot of them, and one of the best I have found is ExpressCheck. It is a neat little program that does just about everything you would expect. One of the nicest features is that its displayed check register appears in the identical format as my printed register. Check transactions are recorded on a check-like form displayed on the screen, and the program automatically increments the check number as you

write multiple checks. If you are so inclined, you can even order pin-feed check stock for your printer.

This program has all of the standard features you would expect. At the end of each month, you can mark the cleared checks in the register, and reconcile the account. The account reconciliation function shows you the calculated difference (hopefully it is 0.00 unless you or the bank made a mistake) between the balance in your checkbook and the bank's statement. You can sort transactions in date or number sequence. ExpressCheck has all kinds of reports including the obvious check register report. Special categories can be established for tax purposes, and you can generate a report by category (e.g. house payment, utilities, etc.) with automatic totals. This highly recommended program can be found on some bulletin boards as shareware or can be ordered direct at the address shown at the end of this article.

### Next Time

The purpose of this article has been to give you an idea of the different kinds of problem-solving software that are available. I have described some of the problems I have found and how these programs have helped solve them. There are many kinds and categories of utility programs, and the ones included here only scratch the surface, but they do help with a number of very common problems that you are likely to have.

In the next article, we will explore the mysteries of selecting, setting up, and using a hard disk. If you have ever looked at some advertisements for hard disks, you may have wondered how to get it up and running. The next article will help you do so.

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.

### Products Discussed

#### HUG SOFTWARE

HADES (885-3040)	\$40.00
HEPCAT (885-3045)	35.00
HFM (885-3014-37)	20.00
KEYMAP (885-6001-37)	20.00
Heath/Zenith Users' Group	
P. O. Box 217	
Benton Harbor, MI 49022-0217	
(616) 982-3463 (HUG Software only)	

#### OTHER SOFTWARE

ExpressCheck	\$29.95
David Berdan	
Expressware Corporation	
P.O. Box 230	
Redmond, WA 98073	
(206) 481-3040	

GOfer	\$79.95
Microlytics, Inc.	
One Tobey Village Office Park	
Suite #547	
Pittsford, NY 14534	
(800) 828-6293 (except NY)	
(716) 248-9150 (NY only)	

Mace Utilities	\$99.00
Paul Mace Software, Inc.	
400 Williamson Way	
Ashland, OR 97520	
(800) 523-0258 (Orders only)	

Norton Utilities	\$100.00
Norton Advanced Utilities	150.00
Norton Commander	79.95
Peter Norton	
2210 Wilshire Blvd., #186	
Santa Monica, CA 90403	
(213) 826-8032	

SideKick Plus	\$199.95
SuperKey	99.95
Borland International	
4585 Scotts Valley Drive	
Scotts Valley, CA 95066	
(800) 255-8008 (Except California)	
(800) 742-1133 (California only)	
(800) 237-1136 (Canada only)	

Software Bridge	\$149.00
Systems Compatibility Corporation	
401 North Wabash	
Chicago, IL 60611	
(312) 329-0700	

Tornado	\$99.95
Tornado with Library	149.95
Micro Logic Corp.	
100 2nd St.	
POB 174	
Hackensack, NJ 07602	
(800) 342-5930 (except NJ)	
(201) 342-8101 (NJ only)	

WindowDOS	\$49.95
WindowDOS Associates	
Box 300488	
Arlington, TX 76010	
(817) 467-4103	

WordStar Version 5.0	\$495.00
MicroPro International	
Attn: Customer Service	
P.O. Box 7079	
San Rafael, CA 94901-7079	
(800) 227-5609 (Orders only)	



**Are you reading  
a borrowed copy of REMark?  
Subscribe now!**



# Z-100

# SURVIVAL

# KIT #4

## Writing Text Directly to the Z-100 Video Memory

The last issue of Z-100 Survival Kit was all about different ways to write text on the screen of the Z-100. We discussed how to display text with high-level routines, DOS function calls, BIOS calls, and MTR-100 monitor ROM calls. The last way of doing it is to write the text directly to video memory, and that is the subject of this installment of Survival Kit.

## Understanding the Z-100 Video Memory Map

Before we start talking about writing text to video memory, it might be wise to take a quick look at the Z-100 video memory layout. The Z-100 is an all-graphics machine (i.e., there is no text mode) which has three separate planes of video memory — one for each primary color (red, green, blue). Each plane is allocated a 64K chunk of the system RAM memory map, as follows . . .

C0000 to CFFFF Blue video RAM bank  
D0000 to DFFFF Red video RAM bank  
E0000 to EFFFF Green video RAM bank

The numbers given above are five digit hexadecimal values representing the offset into the system memory map. The memory starting at C0000 (blue bank) is commonly described as segment C000, based on the way this memory must be accessed using the 8088 CPU segment registers.

Each text character position on the screen is composed of 9 bytes of viewable data, which are the nine bytes of the font design. For instance, here is a representation of the capital 'A' font character...

```
0 0 0 0 0 0 0 0 0 00H
0 0 0 1 1 1 0 0 0 1CH
0 0 1 0 0 0 1 0 0 22H
0 0 1 0 0 0 1 0 0 22H
0 0 1 1 1 1 1 0 0 3EH
0 0 1 0 0 0 1 0 0 22H
0 0 1 0 0 0 1 0 0 22H
0 0 1 0 0 0 1 0 0 22H
0 0 0 0 0 0 0 0 0 00H
```

The Z-100 screen is organized into 25 text lines each containing 9 scan rows. The beginning of each scan row is offset 128 (80H) bytes from the start of the previous scan row. Since there are only 80 displayable columns on the screen, this scheme leaves an invisible area of 48 bytes at the end of each scan row.

The start of each text line is offset 2048 (800H) bytes from the start of the previous one. This is enough memory for 16 scan lines ( $16 \times 128 = 2048$ ), but since only nine scan lines are used for each text line, this leaves 1152 (480H) bytes unused at the end of each text line.

It may seem strange that the video layout leaves all these unused 'holes' in the memory map. But this was done to facilitate address calculations. For instance, if the scan rows were contiguous, the calculation to find the start of a scan row would involve multiplying by 80. But since each scan row is actually offset by 128, the calculation can be done by bit shifting, which is much faster. The same type of argument also applies for the start of each text line — shifts are faster than multiplying by nine.

## Calculating the Video Offset for A Text Character

Here is a formula which can be used to calculate the video RAM base offset for a particular text character . . .

$$((\text{Line} - 1) \times 2048) + (\text{Column} - 1)$$

This formula assumes that the top line is line one, and the left-most column is column one. The first byte of the font character goes at this base address, and each successive byte (nine in all) is offset 128 from the last. For example, if you wanted to put the letter 'A' (see font example above) at the 21st column position of the 8th text line . . .

$$\text{base address} = ((8 - 1) \times 2048) + (21 - 1) = (7 \times 2048) + 20 = 14356$$

Therefore, the nine bytes of the font character would be placed at . . .

00H --->	offset 14356 (3814H)
1CH --->	14484 (3894H)
22H --->	14612 (3914H)
22H --->	14740 (3994H)
3EH --->	14868 (3A14H)
22H --->	14996 (3A94H)
22H --->	15124 (3B14H)
22H --->	15252 (3B94H)
00H --->	15380 (3C14H)

## Colored Text

You might have noticed that we haven't been talking about which segment (video RAM bank) to use. This is because the address calculations are the same, regardless of which color bank you write to. Typically, all three banks would be written whenever you are putting text on the screen. Different color text is generated by writing the font pattern or clearing the character position of each bank. For instance, if you want white text, you would write the font pattern to every bank. Or if you wanted green text, you would write the font pattern to the green bank (E000) and clear the other two banks. The Z-100's video RAM port allows an easier way to write more than one bank of memory at a time, but our space here doesn't allow a complete discussion of that feature right now. We'll cover how to program the Video RAM port in another installment of the Z-100 Survival Kit.

## Assembly Language Character Output Routine

Now that we have some of the preliminaries out of the way, let's get on with the show. Listing 1 is an assembly language routine that can be called to display a text character on the screen.

The character to display is passed to the routine on the stack, a method which is used by quite a few high language compilers. This routine was written specifically to interface with Microsoft 'C', but it should be useful with other compilers, and from other languages, as well. The character position on the screen, and the

text colors, are passed using public variables.

Here is an example of how to use the `__prtc` routine from an assembly language program ...

in your data segment ...

```
fore_color    db    ?
back_color    db    ?
text_line     dw    ?
text_column   dw    ?
```

code to call `PRTC` ...

```
mov    fore_color, 7    ; make foreground white
mov    back_color, 0    ; make background black
mov    text_line, 10    ; position will be line 10, column 1
mov    text_column, 1   ;
mov    al, 'A'          ; will display letter 'A'
push   ax               ; push argument onto stack
call   __prtc           ; display the character
add    sp, 2            ; adjust stack
```

## The Font Table

There is one thing you'll need to note before trying the `__prtc` routine in your own program. The start of the font table (represented by the variable `FONT` in our listing) is not defined. You will need to define your own font table of 95 printable ASCII characters (nine bytes per font character). The start of this table in your data segment should be indicated with the label `"FONT"`.

An alternative to creating your own font table is to use the MTR-100 table that already exists in memory. The segment: offset address of this table is at offset 06FH in the MTR-100 data segment. Here's how you find it ...

```
mov ax, 0                ;
mov ds, ax                ; get interrupt page segment
mov si, 0FEH              ;
mov ds [si]               ; get MTR-100 data segment (stored at 0:0FEH)
mov si, 06FH              ;
mov bx, [si+2]            ; get segment:offset of table
mov ds, [si]              ; DS:BX is now table start address
```

If you elect to use the MTR-100 font table, you'll need to modify the `__prtc` routine so that `SI` is initialized to point to the table, and you'll need to set `DS` to the table segment.

## Wrap-Up

The `__prtc` routine described here isn't quite as fast as simply calling the MTR-100 monitor ROM routine. The reason is because we haven't used any tricks to do multiple bank accessing, and we have kept the code as short and understandable as possible. More experienced programmers who are still interested may want to study the listing of the DFC (Display Font Character) routine in the MTR-100 ROM listings (available as a part of the Z-100 technical manual set). This routine demonstrates how to use the multiple access capability of the video RAM port to crank the last bit of speed from the Z-100 video.

The question might arise ... "If the MTR-100 routines are faster (or just as fast) as my own routine, why bother?" The answer lies in the flexibility of having your own code to write characters to the

screen. The routine given in Listing 1 is just a starting point for other variations you might want to develop for displaying odd size fonts, or writing text in non-standard graphics modes. If your program can live with the standard 25 line by 80 column display, your right — it probably would make a lot more sense to stick with DOS functions or BIOS calls for screen output.

## Customized Fonts

For the ambitious programmer, the possibility exists to program the Z-100 to display any type, or shape, of font. I'm not talking about simply changing the design of the standard font here. I mean using fonts with different sizes or characteristics than the standard text font. For example, you might develop a font with a 12 × 16 matrix size (instead of the standard 8 × 9). Or you might want to display italic characters on the screen. Most graphics programs give you the ability to have these types of fancy fonts.

Generally speaking, any font that doesn't fall neatly into byte boundaries will have to be displayed on the screen

one pixel at a time. In other words, your program will have to scan the font character matrix from left to write, top to bottom, and set the pixels one at a time. This will be slow, but can reasonably be done in assembly language.

## Fan Mail

In the first installment of Survival Kit, I asked you to write and tell me about your interests. And I asked for your help in determining a direction for this column. Well, the mail has started to flow from that first column in January 1989. The response has been very enthusiastic (and very voluminous). Right at this moment, I'm not too sure how I'm going to keep up with it all. But rest assured that I am formulating a plan, even at this very instant. Part of the plan is that I hope this mail subsides to reasonable levels soon!

About one fourth of the mail I have been getting is simply expressing the writer's gratification at having a Z-100 specific column to read. I appreciate this kind of mail, but you really should be sending it to Jim Buszkiewicz. If you think this col-

umn is great, don't tell me (I already know!), tell the editor.

## Q & A

The balance of the mail (about three pieces per day at present) is composed of letters asking questions about the Z-100. Most of them are very technical questions. Some of them I don't know the answers to. But I got myself into this thing ... I guess I can make it through.

As stated in installment #1 of Survival Kit, I'll try to get you a personal reply if you write. I think it's too much to ask people to wait several months to see the answer to a question in the magazine. If things just get too crazy, you'll at least get a note acknowledging that I received your letter. As for including a SASE with your letter — I'm not real picky, but that would be nice. I'd much rather have a self-addressed, stamped fifty dollar bill (just kidding — don't send money!).

At any rate, it would appear that the first directional push my readers are giving me is for a question and answer section in this column. I'll try to be accommodating. Here goes ...

**Q.** Running under Z-DOS, I was able to patch the BIOS to speed up the step rate of my floppy drives. Do you know the patches to do this with MS-DOS versions 2 or 3?

**A.** It's much easier with DOS 2.x or up. You configure the step rates of all your floppy drives by using the `CONFIGUR.COM` program (option D on the main menu) supplied with MS-DOS. Keep in mind that the maximum step rates are typically 6 milliseconds for 5-1/4 inch drives, and 3 milliseconds for 8 inch drives. Individual drives may not be quite this fast.

**Q.** I'm looking for a good memory diagnostic program. I know I have bad memory, because one of my larger programs consistently causes a Parity/Buss error. But my present RAM diagnostic doesn't find the problem.

**A.** The best memory diagnostic I've seen is the one included with the Z-100 disk-based diagnostics, which I assume are available through Heath/Zenith. I got my copy of the diagnostics because my company used to be a Zenith service center. The disk-based diagnostics contain a RAM test (for system and video RAM) which will not only do a thorough check, but will also tell you which chip is causing the problem.

**Q.** I would like to install a 60 megabyte hard disk in my Z-100, but it is my understanding that only later versions of MS-DOS (not available on the Z-100) allow one to break the 32 megabyte barrier. How can this be done?

**A.** MS-DOS doesn't really care about the megabyte capacity of a hard disk. The real limitation is the number of sectors (or



## Listing 1

```

; _prtc - Display a character on the screen
;
; void prtc(char c);
;
; Entry:
;     c = character to display
;     fore_color, back_color should be set to desired colors.
;     text_line, text_column should be set to desired position.
;     no range checking is done on text_line and text_column.
;
; Returns:
;     nothing
;
; Action:
;     Displays character on screen.
;     text_column is incremented by one. Line feeds and wrap are
;     not handled.
;
;     extrn fore_color:byte, back_color:byte
;     extrn text_line:word, text_column:word
;
; _prtc
;     public _prtc
;     proc      near
;
;         push    bp
;         mov     bp, sp
;         mov     al, [bp+4]
;         cmp     al, 32
;         jge     PC2
; PC1:      pop     bp
;         ret
;
; PC2:      cmp     al, 127
;         jnc     PC1
;         push    di
;         push    si
;         mov     bx, _text_line
;         dec     bx
;         xchg    bl, bh
;         xor     bl, bl
;         shl     bh, 1
;         shl     bh, 1
;         shl     bh, 1
;         mov     di, bx
;         add     di, _text_column
;         dec     di
;         mov     ah, 0
;         sub     ax, 32
;         mov     si, ax
;         shl     ax, 1
;         shl     ax, 1
;         shl     ax, 1
;         add     si, ax
;         add     si, offset FONT
;         mov     dx, 0E000H
;         mov     ah, 4
;
; PC4:      push    ax
;         push    si
;         push    di
;         mov     es, dx
;         xor     bx, bx
;         test    byte ptr _fore_color, ah; make color masks
;         jz      PC7
;         mov     bl, 0FFH
;
; PC7:      test    byte ptr _back_color, ah;
;         jz      PC8
;         mov     bh, 0FFH
;
; PC8:      mov     cx, 9
;
; PC9:      lodsb
;         mov     ah, al
;         and     ah, bl
;         not     al
;         and     al, bh
;         or      al, ah
;         mov     es:[di], al
;         add     di, 80H
;         loop    PC9
;         pop     di
;
;         ; get character to display
;         ; test for legal ASCII value
;         ; if not legal, simply return
;         ; get text line for character
;         ; this is the same as multiplying
;         ; BL by 256
;         ; AX has now been multiplied by 2048
;         ; DI now holds video RAM offset
;         ; calculate font offset
;         ; by multiplying font index by 9
;         ; SI is font offset
;         ; get video RAM segment
;         ; BL is foreground color mask
;         ; BH is background color mask
;         ; will write 9 bytes
;         ; get byte from font table
;         ; AND with foreground mask
;         ; reverse font image
;         ; AND with background mask
;         ; Or them together
;         ; write byte to video RAM
;         ; next font byte
;
;     endp
; end

```

```

pop     si
pop     ax
shr     ah, 1
sub     dx, 1000H
cmp     dx, 0C000H
jge     PC4
inc     byte ptr _text_column
pop     si
pop     di
pop     bp
ret
;
; _prtc      endp

```

clusters) that can be accommodated. Heath/Zenith gave Z-100 owners a way around this limitation (not previously available to PC clone owners) by allowing the hard disk to be PREP'ed using 1024 byte sectors, instead of the standard 512 byte size. To use a hard disk larger than 32 megabytes, simply run the PREP program using the /K switch. This will tell PREP to use 1024 byte sectors. One disadvantage to using 1024 byte sectors is that the minimum amount of space consumed by any file will be increased to 2048 bytes (assuming a smallest cluster size of two sectors). If you have a 40 meg hard disk that formats out to say 35 megabytes, you might be better off just using 512 byte sectors, and living with a 32 meg capacity disk. Especially if most of your files are small. \*

## EGAD

Graphics and Text Screen Print Package for the VGA, EGA, and CGA displays.

- Print any part of the screen ('crop box' lets you use the cursor keys to select any rectangular area)
- Enlarge for emphasis (1 to 4 times in graphics modes).
- Color graphics and text print in color (on color printers) or in black and two shades of gray (for black-only printers).
- Set program configures which screen colors map to which printer colors (or gray and black tones).

Supported printers: Epson and compatibles (Black + 2 grays); Star NX-1000 Rainbow (30+ colors); Xerox 4020 (64+ colors); Dataproducts 8020 (Black, 6 colors); NEC-8023/C.Itoh 8510 (Black, 2 grays). EGAD, \$25.00 Postpaid.

Lindley Systems 4257 Berwick Place,  
Woodbridge, VA 22192  
(703) 590-8890. Call for Free Catalog

Reader Service #136

### \*\*\* Z-100 SERIES SOFTWARE \*\*\*

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-463-1	Z-Basic (16 bit)	\$175.00	\$12.00
MS-463-7	Multiplan	\$195.00	\$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	All 4 Listed Above	\$819.00	\$40.00

### \*\*\* IBM COMPATIBLE SOFTWARE \*\*\*

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-5063-30	Microsoft Windows	\$ 99.00	\$ 24.00
NU-413	Norton Utilities Adv.	\$150.00	\$ 99.00
WP-528	WORDPERFECT 5.0	\$495.00	\$269.00
BO-290	QUATTRO	\$239.00	\$179.00
CP-311	PC TOOLS DELUXE	79.00	\$ 68.00

### \*\*\* ZENITH LAPTOP COMPUTERS \*\*\*

<b>SUPERSPORT 184-1</b>	2 3 1/2" Floppy Drives, 640K RAM	\$1587.00
<b>SUPERSPORT 184-2</b>	1 Floppy, 20 Meg Hard Disk, 640K RAM	\$2373.00
<b>SUPERSPORT 286-20</b>	12/6 MHz, 80286 CPU, 3 1/2" Floppy, 20 MEG Hard Disk	\$3292.00

### \*\*\* VIDEO MONITORS \*\*\*

ZCM-1490	ZENITH Color Flat Screen VGA	\$718.00
MA2565	SAMSUNG Amber TTL 720x350	\$89.00
CW4644	SAMSUNG Color RGB 640x200	\$274.00
CM4531	SAMSUNG Color EGA 640x350	\$389.00
CN4551	SAMSUNG Multi-sync VGA 800x560	\$489.00
NC800	NEC Multi-sync II 800x560	\$639.00

### \*\*\* 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 (28 MS) 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. 32 MEG ST-138/150 Kit . . . . . \$383.00

**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 required to replace one floppy drive with a high speed low power Hard Disk Drive. 32 MEG ST-138/Z-148 Kit . . . . . \$459.00

**ST-138/Z-148 Kit With SmartWatch** . . . . . \$489.00

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

**INTERNAL MODEM** Fully Hayes compatible (software included)  
1200/300 baud . . . . . \$94.00  
2400/1200/300 baud . . . . . \$159.00

**EXTERNAL MODEM** Fully Hayes compatible (software included)  
1200/300 baud . . . . . \$137.00  
2400/1200/300 baud . . . . . \$199.00

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

**V-20 Chips** High Speed NEC V-20-8 8088 replacement. These run at up to 8 MEG and are said to increase CPU speed 10-30% . . . . . \$14.75

### \*\*\* 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 234.00 . . . . . \$351.00  
Bare Drive and Cable for internal mount . . . . . \$136.00

**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

**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 Easy87.** Add an 8087 Numeric Coprocessor. \$69.00 for the board without an 8087 Chip. With 5 MEG 8087 \$188.00 or with 8 MEG 8087 installed . . . \$234.00

**ZMF100A by FBE Research.** A modification package which allows 256K chips to be used on the old-style motherboard to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. . . \$60.00  
Requires 27 256K RAM chips to complete the kit.

**UCI Memory Upgrade Pal Chip Set** For the Z-100's with the newer motherboard part number 181-4818 or greater. This chip set allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K RAM chips, a total memory of 768K is obtained. PAL Chip Set . . . . . \$64.00

**UCI Memory Upgrade Card** We recommend this one highly. The board has sockets for up to 2 MEG of RAM. 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 to complete this kit.

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

**UCI EasyWin** Winchester Drive Systems at reasonable prices. Complete Hard Disk Systems for mounting inside your Z-100. Systems complete with Seagate Drives, 21 MEG \$598.00, 31 MEG \$634. System without Drive . . . . . \$317.00

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

### \*\*\* FLOPPY DISK DRIVES \*\*\*

MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	\$ 89.00
MITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	\$106.00
MITSUBISHI M-353	3.5" in 5.25" frame 720K	\$98.00
MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	\$129.00
	M-355 Software Driver	\$ 19.00

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

### \*\* SEAGATE HARD DISK DRIVES \*\*

ST-125	21 MEG, 28 MS, Auto Park Heads With Controller & Cables	\$ 275.00 \$ 329.00
ST-138	31 MEG, 28 MS, Auto Park Heads With Controller & Cables	\$ 329.00 \$ 383.00
ST-238	31 MEG, 65 MS, RLL With RLL Controller & Cables	\$ 258.00 \$ 309.00
ST-251	42 MEG, 40 MS, Auto Park, Software With Controller & Cables	\$ 384.00 \$ 438.00
ST-251-1	42 MEG, 28 MS, Auto Park, Software With Controller & Cables	\$ 465.00 \$ 519.00
ST-4096	82 MEG, 28 MS, Auto Park, Software.	\$ 647.00

**V-20 Chips** High Speed NEC V-20-8 8088 replacement. These run at up to 8 MEG and are said to increase CPU speed 10- 30%..... \$14.75

### \* PAYLOAD CUSTOM ASSEMBLED COMPUTERS \*

We can assemble 8088 XT, 80286 AT or 80386 IBM compatible computers to your specifications. We use high speed turbo motherboards with Phoenix BIOS. Other features include clock/calendar, Keytronics 101 key keyboard, Mitsubishi floppy drives, Seagate hard disks, front panel mounted turbo and reset buttons, memory and I/O ports, all to your specification. Please write or call for price information. We can send you a work-up sheet showing items available and prices.

**Example #1**  
IBM XT Compatible 8088 10 Meg CPU, two 360K floppy drives, 640K RAM, one serial port, one game port, two parallel ports, 101 keyboard, clock/calendar, 10 Meg CPU and amber TTL monitor . . . . . \$831.00

**Example #2**  
Same as above with the two floppy drives plus a Seagate ST-238 30 Meg RLL hard disk drive . . . . . \$1126.00

**Example #3**  
IBM AT Compatible 80286 12 Meg CPU, 2 Meg RAM, clock/calendar, one each serial, game and parallel ports, one 360K floppy, one 1.2 Meg floppy, 101 keyboard, Seagate ST-251 40 Meg hard disk, EGA color monitor . . . . . \$2234.00

MS-DOS 3.3 Operating System . . . . . \$85.00

## PAYLOAD COMPUTER SERVICES



15718 SYLVAN LAKE  
HOUSTON, TEXAS 77062  
PHONE (713) 486-0687



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.



# Turbo Pascal(tm) Versions 4.0 & 5.0 Common Ground for The Z-100 and PC Compatible

Keith Greer  
1405 Bills Drive  
Beavercreek, OH 45385

## Background

When I first started programming, BASIC was just about the only game in town for the microcomputer. BASIC is still a very viable language, especially with today's new BASICs which offer true subroutines, modular code, etc. One of the nice things about BASIC was that you could write a program whose source could most likely be run on either the Z-100 or PC compatibles (hereinafter referred to simply as PC) unless you did a lot of PEEKs and POKEs. At a time when BASIC was suffering from a lot of deficiencies, a new kid on the block appeared called Turbo Pascal™ from Borland International. Turbo quickly endeared itself to hosts of new programmers and even converted old BASIC programmers, like me, to its new structured ways. Borland was even far-sighted enough to build several versions. There was a CPM version, an "MS-DOS Generic" version which could be trained to run on nearly anything, including the Z-100. There was, of course, the PC version which had built-in graphics and speaker control. Turbo shared with BASIC the very nice feature that if you stayed within the bounds of the language, it was possible to write code which could be ported to and from the Z-100 and PC.

There were complaints about Turbo version 3.0 though, the main one being that it used the small memory model, limiting the size of code and data you could manipulate. There were work-arounds for these limitations, but the critics would not relent. So Turbo Pascal 4.0 was born, and Borland support for the Z-100 died. Less than a year after 4.0 was released, 5.0 made its debut. Resurrecting the Z-100 to the world of Turbo Pascal is the subject of this article.

I will not attempt to teach you Turbo Pascal in this article. There are many articles on Turbo in REMark and other publications which address some of its capabilities and features. Instead, I will assume that you have some familiarity with Turbo Pascal 4.0 or 5.0, particularly the concept of units.

## Implementing Turbo Pascal on the Z-100

I wanted to have my cake and eat it, too. I didn't want to give up my dear old Z-100, but I wanted to continue programming in my favorite language, too. Practicality and economics dictated that my programming efforts had better be able to run on PC compatibles also. My goal, therefore, was to design a set of library routines specifically for the Z-100 which would look like Borland's PC routines both in syntax and in behavior. My intent in the rest of this article is to make you aware of the existence and capabilities of the result, TURBOZ.

## What's TURBOZ?

TURBOZ is a library of machine-dependent routines for Borland's Turbo Pascal Versions 4.0 & 5.0, which enables the Zenith Z-100 to compile and run Turbo Pascal programs in Z-100 mode. TURBOZ provides a platform for source code portability to and from the Z-100 and PC compatible computers. To use this product, you will need Borland's command line version of the compiler, TPC.EXE, and the SYSTEM and DOS units in their TURBO.TPL file. You may write Turbo programs in your favorite editor, and compile them using TPC.EXE. The resulting programs will run on the Z-100 without assistance from a PC emulator.

## Why You Need TURBOZ

Most of the units in Borland's TURBO.TPL library file contain code which is very PC-dependent. Any code which must control the screen display, cursor, keyboard, and printer is usually unique to the particular computer for which it is designed. In the case of the keyboard and printer, the code CAN be designed to be MS-DOS generic, but in their quest for speed, Borland designed even this code to be PC-dependent. They totally bypassed MS-DOS for nearly every function, except in the DOS and System units. For you advanced programmers, there is one other benefit to avoiding DOS's built-in services besides speed — you can use many of Turbo's I/O (Input/Output) procedures in interrupt service routines. TURBOZ's units contain Z-100 unique code which performs nearly all the same functions as Borland's code for the PC.

## Requirements

To use TURBOZ, you will need a Z-100 computer equipped with version 2.xx of the monitor ROM, Turbo Pascal Version 4 or 5 from Borland International, MS-DOS ver 2 or above, at least 256K of memory for the command-line compiler.

## Files

### Disk 1: (Distributed as Shareware)

TURBOZ.DOC — This documentation  
Z100CRT.TPU — Console interface unit with windowing capabilities  
MINICRT.TPU — Alternate console unit with less overhead, capabilities  
ZGRAPH.TPU — Graphics unit with View-Ports & clipping  
ZGRAPH2.TPU — Auxiliary Graphics unit with Doodler support  
ZLST.TPU — Printer unit

# TURBOZ

by Keith Greer

## A Complete Library for the Z-100

using Turbo Pascal Versions 4/5

from Borland International

Figure 1  
Special text is easily generated using Borland's BGI fonts.  
See Listing 1 for the code which generated this screen.

CRT2.TPU — Supplies features found in TURBOZ to Borland's CRT unit

GRAPH2.TPU — Doodler support for the PC

FONT2.TPU — Borland stroked font support unit

\*.LST — Interface listings for the units

README.DOC — Latest changes

DEMO.ARC — Demo of TURBOZ power and stroked font capability

XYPLOT.ARC — A fully functional XY data plotter demonstrating windows.

ARCE.COM — Used to extract the ".ARC" files

### Disk 2: (Available from the author)

\*.ASM, \*.OBJ, \*.PAS — Complete source code to everything except the DOS and System units.

### Setting Up

To get started, simply copy TPC.EXE and TURBO.TPL from the Borland distribution disks, and the contents of TURBOZ disk #1 onto a working disk or into your Turbo directory on your hard disk. Previous versions of TURBOZ were delivered with the units already in a library file to replace Borland's TURBO.TPL in entirety. Most users seemed to want the capability to manipulate the units individually, however, so TURBOZ is now delivered as a collection of ".TPU" files. You may decide to place the units you use most into TURBO.TPL. To do this, you will probably need access to a PC compatible machine in order to run Borland's TPUMOVER utility. If your memory is limited (the computer's, that is) I recommend that you extract (not delete) the unused PC units from TURBO.TPL, leaving the DOS and SYSTEM units. You may want these unused units for compiling a PC program later.

Now insert the desired TURBOZ units and save TURBO.TPL to disk. Since TPC will always find and load TURBO.TPL, your TURBOZ units will now always be available. Remember, do these changes on a working copy only!

Alternatively, you can create a TPC.CFG file in your Turbo directory containing the line:

/Ud:\name

where d:\name is the drive/path to your TURBOZ unit files directory.

### Unit Analogies

The following TURBOZ units are functionally analogous to the indicated Borland units.

TURBOZ Unit Name	Borland Unit Name
Z100Crt	Crt
ZLst	Printer
ZGraph	Graph

TURBOZ does not include the Turbo3 and Graph3 units. Since the generic version of Turbo3 which we Z-100 programmers had to endure did not support graphics, the Graph3 unit is irrelevant. I do not advocate using the Turbo3 unit, since it is so easy to upgrade your programs using UPGRADE.

### TURBOZ Unit Capabilities

TURBOZ provides nearly complete compatibility with Borland's IBM version to the extent that the Z-100 hardware will allow it. This allows software to be developed on the Z-100, and then recompiled using Borland's PC units with little need to change the source code. Conversely, if Turbo source code is available, it could be recompiled using TURBOZ for use on the Z-100.

Z100Crt features built-in line editing capability for any Read or ReadLn, and a super-fast screen-output routine for Write

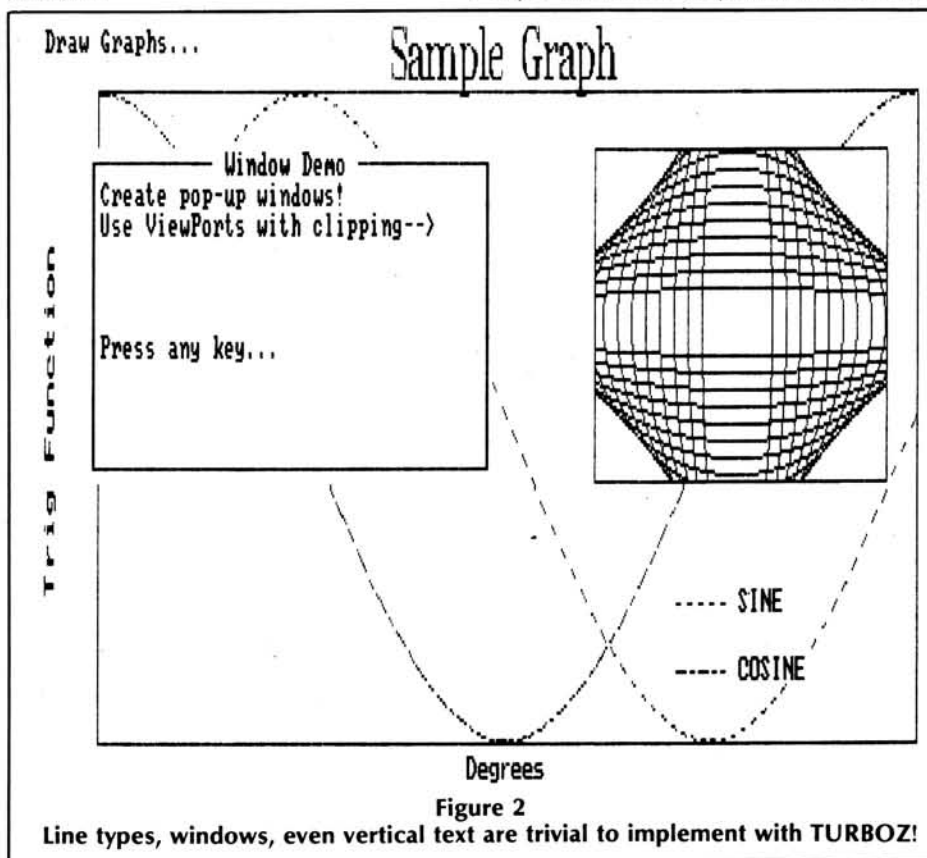


Figure 2  
Line types, windows, even vertical text are trivial to implement with TURBOZ!



Using Z100Crt, you have access to the entire rich set of IBM (tm) graphics characters:

0	24	44	64	84	T	104	h	124	!	144	ë	164	ñ	184	¿	204	¡	224	α	244	↑		
1	25	45	-	65	A	85	U	105	i	125	ˆ	145	æ	165	ā	185		205	=	225	β	245	↓
2	26	46	.	66	B	86	V	106	j	126	~	146	ā	166	ā	186		206		226	γ	246	÷
3	27	47	/	67	C	87	W	107	k	127		147	ō	167	ō	187		207		227	π	247	≈
4	28	48	0	68	D	88	X	108	l	128	ç	148	ö	168	ö	188		208		228	Σ	248	°
5	29	49	1	69	E	89	Y	109	m	129	ü	149	ö	169	ö	189		209		229	σ	249	·
6	30	50	2	70	F	90	Z	110	n	130	ë	150	û	170	û	190		210		230	μ	250	·
9	31	51	3	71	G	91	[	111	o	131	ä	151	ü	171	¼	191		211		231	γ	251	√
11	32	52	4	72	H	92	\	112	p	132	ä	152	ü	172	¼	192		212		232	δ	252	∞
12	33	53	5	73	I	93	]	113	q	133	ä	153	ü	173	¼	193		213		233	θ	253	²
14	34	54	6	74	J	94	^	114	r	134	ä	154	ü	174	¼	194		214		234	Ω	254	■
15	35	55	7	75	K	95	_	115	s	135	ç	155	ç	175	»	195		215		235	δ	255	
16	36	56	8	76	L	96	'	116	t	136	ç	156	ç	176	»	196		216		236	ω		
17	37	57	9	77	M	97	a	117	u	137	ë	157	ü	177	»	197		217		237	φ		
18	38	58	:	78	N	98	b	118	v	138	ë	158	ü	178	»	198		218		238	€		
19	39	59	;	79	O	99	c	119	w	139	ï	159	ü	179	»	199		219		239	∞		
20	40	60	<	80	P	100	d	120	x	140	ï	160	ü	180	»	200		220		240	≡		
21	41	61	=	81	Q	101	e	121	y	141	ï	161	ü	181	»	201		221		241	±		
22	42	62	>	82	R	102	f	122	z	142	ï	162	ü	182	»	202		222		242	≥		
23	43	63	?	83	S	103	g	123	€	143	ï	163	ü	183	»	203		223		243	≤		

Press any key...

**Figure 3**  
The complete IBM character set can be easily output with a simple Write or WriteLn statement.

and WriteLn. Every attempt has been made to duplicate the behavior of the routines in Borland's Crt unit wherever the Z-100 will allow it, even Windows are implemented! (See Figure 2) This unit makes writing programs with a professional touch easy. Sophisticated user interfaces are simple to implement using these basic routines. As Figure 3 illustrates, you have the capability to Write or WriteLn the entire IBM™ character set, including the rich graphics character set. I should emphasize that this does NOT require the graphics characters in ALTCHAR.SYS! The IBM set is a standalone capability in Z100Crt. This makes back-porting PC source to the Z-100 nearly trivial.

MiniCrt is a scaled down version of Z100Crt which offers most of the standard Crt functions, except windowing and input editing. If you just want to knock out a "quick and dirty" program without a lot of overhead, this is your unit.

The ZGraph unit likewise implements all the pertinent features of Borland's Graph unit except patterned area fills. Even ViewPorts with clipping, line types, and stroked fonts are implemented! Figures 1 and 2 illustrate some of the graphics power in ZGraph. Listing 1 is included to give you an idea of the code required to achieve the effects in Figure 1. Solid-fill (paint) is provided, and the GetImage and PutImage routines provide support for the BitBlt boolean operators allowing special effects, such as animation and negative images. All the fundamental

graphics routines are written in carefully crafted assembly language for speed and compactness. If you are using Turbo Pascal 5.0, Arcs and Sectors are now drawn using the 8087 if you have one. If not, the real number math is emulated. All other graphics require only LongInt math, at most. ZGraph also supports the Turbo 5.0 features: FillEllipse, Sector, SetAspectRatio, and SetWriteMode. These are documented in the Borland manual.

The ZGraph2 unit supplements ZGraph by providing your choice of two bit image routines which support Paul Herman's Doodler V ".PIC" format. Get50Pic/Put50Pic works with Doodler 5.0x, and GetPic/PutPic works with Doodler 5.1x. The Graph2 unit provides these same routines for the PC. The Z-100 routines support the boolean operators, but the BitBlt parameter is currently ignored by the PC routines. You can exchange screen images both ways between your programs and Doodler V!

ZLst provides BIOS output to the PRN device via the familiar Lst device. Your programs will not hang if the printer is not available; this unit will time out after a few seconds of trying to access the printer, and return to your program. The boolean function LstReady returns True if the printer is online and may be called prior to attempting printer output.

#### The Units in Detail

Z100Crt features built-in line editing. The user has access to the following editing keys during any Read or ReadLn:

Left & Right Arrow — moves the cursor non-destructively over entered text.

ICHR — Toggles insert mode on/off. Insert mode is indicated by a block cursor, overwrite by an underline cursor.

DCHR — Deletes the character at the cursor, "gobbles" text to right.

Backspace — Normal destructive backspace.

ESC — Abort current input, return null.

RETURN — Terminate input and return string. May be pressed even if cursor is in middle of text.

#### IBM Return Codes

The ReadKey function returns the same codes for special keys as the PC compatible. This was done to make the function behave as closely as possible to the Borland version, further reducing the need for source code modification.

#### Color Constants

These have been re-mapped to Z-100 color numbers. When a color name unique to the IBM is used in a program, it is mapped to a corresponding Z-100 color. Note that if a program uses actual numbers (e.g., TextColor(4)), this unit will not display proper colors. Use the constants!

#### Windows

This unit supports windows on the Z-100 as implemented in Borland's Crt unit. This major development requires some cooperation from you, however. When you program, avoid directly sending ANY Z-100 escape sequence such as: Write(#27'E') to clear the screen (use ClrScr!) or perform any other screen-related function for which a defined function exists in this unit. The routines in this unit contain the code necessary to handle windows, including wrapping and scrolling, but they must not be circumvented by an external escape sequence. Most of the screen routines are window-relative. GotoXY, ClrScr, ClrEol, ClrEop, ClrTop, Center, etc. all operate as though the current window were the whole screen with 1,1 at the upper left. The rest of the screen outside the window is inaccessible. A couple of very important exceptions occur. They are:

Window  
ClrBox  
GetWindow  
PutWindow

These routines are available for you to use as fundamental tools in implementing more elaborate windowing systems, and are also used internally by windowing routines in this unit. As such, they require that their parameters be ABSOLUTE coordinates, with 1,1 at the upper left corner of the SCREEN, not window. Be aware that ClrBox & Get/Put Window DO NOT respect current window boundaries when you invoke them directly in your

programs. When you USE this unit, the whole Z-100 screen is initially set as the current window. In this state, absolute and window-relative coordinates are, of course, synonymous. You may return to this state at any point by executing Window(1,1,80,25) in your program. The use of a window slows screen output slightly, but the difference is a very small price to pay for the greatly facilitated screen-handling. Whenever a Window is in effect, Write and WriteLn disable escape sequences. That is, the escape character (#27) is simply skipped and not output. This keeps the windowing code from "losing control" to an escape sequence. Built-in procedures are provided to do most of the functions you would ever want to do with an escape sequence, and directly writing escape sequences makes your code non-portable to a PC anyway.

The global variables WindMin and WindMax contain the upper left and lower right 0-based corners of the current window as defined in the Borland manual. You will find these useful in developing more sophisticated windowing systems and even pull-down menus. DO NOT change them directly! The Window procedure sets these variables. Consider them read-only. They are global here only because that's the way they are in Borland's Crt unit.

Since the Z-100 BIOS contains no support for windowing, I had to write all the code to implement scrolling, wrapping, etc. Borland's custom text device driver capability made this possible (but not easy).

### Program Overhead — An Alternative

The simple act of "Using" this unit will incorporate all of the windowing code, since windowing by nature had to be an inherent capability of the output driver. This implies that if you use this unit at all, you might as well avail yourself of the window power in your program. No code savings is realized by avoiding calling the window routines. By the same token, very little additional code is required to implement some beautiful effects in your programs. The overhead introduced into your program by using this unit is approximately 6K. If this is objectionable to you for any reason, and you don't need windowing or input editing capability, use the unit MiniCrt instead. Its overhead is only about 2K, and you will still have access to most of the Crt functions.

ZGraph is a Z-100 replacement for the Graph unit which is a hardware dependent module provided with Turbo Pascal™ to perform graphics on the IBM PC and compatibles. Since the video hardware and memory is so vastly different on the Z-100, this unit's mimicry of Borland's unit is not as good as Z100Crt, which has only text to worry about. Also, with such an array of different display

adapters for PC's, each with their own resolution, porting a graphics program between them and the Z-100 is very difficult. It can be done with care, but generally requires more modification to the source code to account for differences in resolution, aspect ratio, etc. Also, I have yet to develop the patterned fill capability present in Graph. Having qualified myself, here are some characteristics of the routines herein which are unique to this implementation. Unless mentioned, other routines behave as described in the Turbo manual, including stroked fonts, ViewPorts, clipping, and the current pointer (CP) features.

### Unimplemented Routines

First of all, since there are no display adapters for the Z-100, none of the routines pertaining to graphics drivers are implemented here. Also, since there are no graphics modes, (the Z-100 is always in graphics) there are no routines pertaining to modes or palettes. The following routines are not present in this unit:

- GetPalette
- SetAllPalette
- SetPalette
- RegisterBGIdriver

Unimplemented Turbo 5.0 routines:

- SetRGBPalette
- GetDriverName
- GetMaxMode
- GetPaletteSize
- InstallUserDriver
- InstallUserFont

Since I have not yet implemented fill patterns, the SetFillPattern routine is absent.

All other procedures and functions in Graph are implemented in ZGraph and function as described in the Turbo manual except as noted below.

### Deviations

FillPoly and FloodFill do not use fill patterns. The SetFillStyle procedure may be used, but only EmptyFill and SolidFill are defined as fill styles. The former causes the background color (not the same as the TextBackground color) to be used as the fill color. The latter causes the Color passed to SetFillStyle to be used. The fill color used may be the same as the border color. FillPoly may not work on areas which are highly non-convex for the following reason. After the enclosing line segments are drawn, FillPoly calculates the "center of gravity" (centroid) of the area. It then uses this point as the "seed" point to begin filling the area. If the centroid lies outside the area, the paint will spill all over the screen. Use DrawPoly first to test draw the area, if it looks Ok (closed, mostly convex, etc.) go ahead and try FillPoly. If you MUST draw and fill a very concave area, draw it using DrawPoly, then calculate a seed point yourself inside the area and pass it to FloodFill. If

anyone comes up with an algorithm to unambiguously calculate a point inside any area, given a set of vertex coordinates, send it to me, please.

A supplementary fill routine has been included: FillReg. It is much faster than FloodFill, but is undependable for non-convex areas. Use it if you can, and don't care about PC porting. If it fails for your application, FloodFill is quite adequate for most purposes.

The procedure SetGraphBufSize is not required and unimplemented in ZGraph.

### Viewports

As described in the Turbo manual, all drawing routines are ViewPort-relative. For fill routines, the seed point must lie within the ViewPort. Fills are also clipped at ViewPort boundaries. Be careful about turning clipping off. Without clipping, you must ensure that all parameters passed to drawing routines will resolve to valid on-screen locations. They will still be ViewPort-relative, just not checked to prevent writing to invalid video RAM locations. As noted above, ViewPorts and Windows operate independently. If you desire both graphics and text confined to the same rectangular area, simply overlay a ViewPort on top of the defined Window. Note that the Window procedure takes CHARACTER (row & column, 1,1 upper left) parameters, and that ViewPort takes PIXEL (0,0 upper left) parameters. A little math using the fact that on a Z-100, characters are 8 pixels wide by 9 pixels tall, and you're home free.

### The DOODLER Routines

There are two pairs of routines for use in interfacing with Paul Herman's Doodler V ".PIC" format. Get50Pic/Put50Pic is for use with Doodler 5.0x and GetPic/PutPic is for Doodler 5.1x. For the Z-100, these routines are in the unit ZGraph2. For the PC, they are in Graph2. You will need to have the ".TPU" file for the appropriate unit available in your units path, since they are not present in the TURBOZ library file. In the discussion below, GetPic/PutPic generically refers to these two routines.

A screen image gotten into a buffer with GetPic and written to disk with a file name ending in ".PIC" may be edited with Doodler V. Use the size returned by GetPic to tell how many bytes from the buffer to write to disk. Conversely, PutPic may be used to place a Doodler image on screen from within your program. For the Z-100, including one of the BitBlt operators (XorPut, NormalPut, NotPut, etc.) when calling PutPic will cause different effects when placing the image on screen. A NotPut, for example, will cause a "negative" image to appear. Experiment with these operators to see their effect. The BitBlt operator is currently ignored by the PC routines. This is due to the fact that



Borland did not provide me a way to do a PutPixel using the boolean operators. The parameter is included anyway for declaration compatibility with Z-100 code. These routines are slow compared to GetImage and PutImage due to the compression algorithm and should only be used for Doodler V work, or to save disk storage space.

### On Writing Portable Code

When writing code with Z-100-to-PC portability in mind, here are a few useful hints. For the "uses" clause, I use code like this:

```
{IFDEF Z100}
uses Dos,Z100Crt,ZGraph;
{$ELSE}
uses Dos,Crt,Crt2,Graph;
{$ENDIF}
```

Then, when compiling the program on my Z-100, I give the command:

```
TPC /DZ100 myprog
where myprog is the file name of the program, of course.
```

This defines the symbol Z100 so I can compile selected segments of code depending on the target machine. The program can be compiled on a Z-100 for a PC target. Just log to another drive (or sub-directory) where you have TPC and the Borland version of TURBO.TPL available, and recompile the program without defining the Z100 symbol. The conditional compilation construct above may be used in any number of places in your code.

As mentioned above, avoid "hard-wiring" machine-specific numbers into your code. Use constant declarations instead, in conjunction with the conditional compilation directive above to compile the appropriate constants for the target machine. In the case of color numbers, use the color NAMES instead of numbers. Both Z100Crt and Borland's Crt have these names already declared, so your colors will come out right.

Stay within the bounds of the language. Avoid using Escape Sequences on the Z-100. Use the routines I have provided. Do not write to some of the global system variables like WindMin and WindMax, or TextAttr. These variables are intended to be available for reading only so you can write other routines using their information. To change colors, use TextColor and TextBackground instead of directly changing TextAttr. Z100Crt doesn't use TextAttr anyway, although it is declared for compatibility. Do not use Direct Memory Access (DMA) in your main programs unless it is part of a procedure whose function can be duplicated on the other machine in some way.

Some hardware differences between the machines get in the way. Line 25 on the Z-100, for instance, is a strange animal and should be avoided for purposes of Input. You MUST do a ReadLn in Turbo to

get a string, and doing ReadLn on line 25 of a Z-100 will not change the rest of the screen since line 25 does not scroll. On a PC, however, a ReadLn on line 25 will scroll the entire screen up one line. This situation is, to my knowledge, unavoidable. In graphics, the differences in resolutions should be handled using constants as mentioned above. GetMaxX and GetMaxY may also be used to advantage here. Graphics is the the most difficult to port from the Z-100 to the PCs due to the many different display adapters, but a surprising amount of graphics can be made portable.

Suppose you need to write a program which uses only MS-DOS for I/O, such as a filter which uses I/O redirection. Here's what you do. Before doing any I/O in your program, set DirectVideo:=False and CheckBreak:=True. It's that simple! This action will also enable Ctrl-C checking for both Input and Output. Conversely, if you plan to write your own interrupt service routines using any of Turbo's I/O statements, make sure you set DirectVideo:=True and CheckBreak:=False before you do I/O in your routine. This will force TURBOZ to use the BIOS/Monitor routines for I/O. You will not be able to abort a program with Ctrl-C in this case. The defaults at start-up are: DirectVideo:=True and CheckBreak:=True.

### How to Get TURBOZ

I am distributing only the TURBOZ units as Shareware. For those that are unfamiliar with this term, it means that the library is NOT free, but that you DO get to try before you buy. I have made Disk 1 available to HUG for distribution on the HUG BBS. I also run a BBS for our local HUG, the Dayton (Ohio) HUG, DAYHUG. TURBOZ will be available there at (513) 429-5818. I emphasize that only Disk 1 is available as shareware, and you MUST obtain the correct units for your version of Turbo since Turbo 4.0 units no workee with Turbo 5.0, and vice versa. Check the BBS directory for the correct file names. If you don't have a modem, send a buck and a disk to the address at the beginning of this article. Again, specify which version of Turbo Pascal you're using. If you keep and use the library, you pay me according to the terms in the documentation.

### Source Code (Disk 2)

The source code to all the units is a combination of Turbo and assembly language. Most of the assembly code is written for MASM 5.1 to take advantage of the high level language support features. The source is unauthorized for distribution by anyone except me, even if modified. It is sold as is. If you modify it, you're on your own as far as support from me. Send \$25.00 and 2 disks to me, and I'll send you the compiled units, and all the source code to everything.

### Credits

Thanks to Randy DeBey for his article in REMARK, Dec 86, pp 15- 19, and to D.J. Delorie of Clarkson U. Their ideas and examples were the seeds from which this package has grown. I have taken the liberty of modifying all routines (some heavily) to fold them into this package, to correct some minor problem, or to improve their speed somewhat.

A very, very big thank-you goes to Harry Landau of Lorton, VA for figuring out the Borland font definitions and supplying the code to implement stroked fonts. It's nice to know the HUG spirit is still alive in some folks.

Also, to Dave Frailey at DAC Micro Systems for supplying me with the IBM font definitions used in his fine product, dCOM. If you own a Z-100, get dCOM! This is the best product I've ever purchased for my Z-100. It has significantly reduced TURBOZ development time. Thanks a bunch, Dave!

### Products Mentioned

Turbo Pascal  
Borland International, Inc.  
4585 Scotts Valley Drive  
Scotts Valley, CA 95066

Scottie Board  
Scottie Systems, Inc.  
2667 Cropley Avenue #123  
San Jose, CA 95132

ZPC IBM Emulator  
Heath Users' Group  
P.O. Box 217  
Benton Harbor, MI 49022-9974

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

dCOM - Directory Commander \$49.00  
Author, Dave Frailey  
DAC Micro Systems  
40941 176th Street E.  
Lancaster, CA 93535

### About the Author

Keith Greer is a Captain in the U.S. Air Force currently serving as Chief of a Computer Integrated Manufacturing Branch at Wright Patterson AFB, OH. He has been programming since the mid 1970's, and is fluent in BASIC, FORTRAN, C, Pascal, and Assembly Language. Keith began learning Turbo Pascal in mid 1985. Shortly thereafter, he learned to use assembly language to improve the performance of his engineering applications, and then extended his assembly programming efforts to include Z-100 graphics. Keith is also President of the Dayton Heath Users' Group (DAYHUG).

## Prime H/Z Enhancements!

### Clock Uses no Slot

**FBE SmartWatch:** On-line date/time. Installs under BIOS/Monitor ROM. Ten year battery. Software included. Works with all Heath/Zenith MSDOS computers. For PC's \$35; Z-100 \$36.50. Module \$27.50

### H/Z-148 Expansions

**ZEX-148:** Adds 1-1/2 card slots. \$79.95. ZEX-148 + SmartWatch \$109.95  
**ZP-148:** PAL chip expands existing 640K memory to 704K. \$19.95

### H/Z-150 Stuff

(Not for '157, '158 or '159)

**VCE-150:** Eliminate video card. Install EGA/VGA card. All plug in. Includes VEM-150, RM-150. Requires SRAM chip. VCE-150 \$39.95, SRAM Chip \$15  
**VEM-150:** Card combines existing two BIOS ROM's into one socket. \$34.95  
**RM-150:** Decoder PROM used in removing video card. With detailed instructions. \$9.95

**ZP640 PLUS:** Expand to 640K/704K by adding 2 banks of 256K RAM chips (not included). ZP640 PLUS \$19.95 (first one); \$9.50 thereafter.

**LIM 150:** 640K RAM plus 512K of simulated Lotus/Intel/Microsoft EMS v3.2 expanded memory. Installs on H/Z-150/160 memory card. No soldering. Requires forty-five 256K RAM chips (not included). LIM150 \$39.95

**Mega RAM-150:** Get 640K/704K main memory plus 512K RAM disk on H/Z-150/160 memory card. No soldering. Without RAM chips \$39.95

**COM3:** Change existing COM2 port address. Internal MODEM at COM2. Don't lose serial port. COM3 \$29.95

### Maximize Your Z-100

**ZMF100A:** Put 256K RAM chips on "old" motherboard (p/n 181-4917 or less). Expand to 768K. No soldering. Without RAM chips. \$65.00

**ZRAM-205:** Put 256K RAM chips on Z-205 board. Get 256K memory plus 768K RAM disk. Contact us for data sheet before ordering. Without RAM chips. \$49.00

### Z-171 Memory Expansion

**MegaRAM-171:** Put 256K RAM chips on memory card. Get 640K memory plus 384K RAM disk. \$59.95

### H/Z-89 Corner

**H89PIP:** Two port parallel printer interface card. With software.

H89PIP \$50.00; Cable \$24.00

**SPOOLDISK 89 and SLOT 4:** Cards still available. Contact us for information.

Order by mail, phone or see a Heath/Zenith Dealer. UPS/APO/FPO shipping included. VISA or MC. WA residents add 8.1% tax. Hours: M-F 9-5 PST. We return all calls left on answering machine!

# FBE

**FBE Research Co., Inc.**

P.O. Box 68234, Seattle, WA 98168

**206-246-9815**

Reader Service #104

## Listing 1

Program Sample;  
uses Z100Crt, ZGraph, Fonts;  
{

Listing 1

A simple illustration of the code required using TURBOZ to produce the screen in figure 1

The unit Fonts is constructed from the .CHR files in the Borland package  
}

const

GraphDriver : integer = Detect;  
GraphMode : integer = 0;

TZ : string[6] = 'TURBOZ';  
Msgs : array [1..4] of string[50] = (  
'by Keith Greer'  
'A Complete Library for the Z-100',  
'using Turbo Pascal Versions 4/5',  
'from Borland International');

var

i : word;

begin

InitGraph(GraphDriver, GraphMode, ''); {Initialize the graphics}  
SetColors(Green, Black); {Set colors to Green on Black}  
ClearDevice; {Clear the screen}

{Load the Triplex font and make it an integral part of the final EXE...}  
if RegisterBGIFont(pointer(@TriplexFontProc)) < 0 then halt(1);

SetTextStyle(TriplexFont, HorizDir, 8); {Use the Triplex font, size 8}  
MoveTo((GetMaxX-TextWidth('TURBOZ')) div 2, 25); {Text goes here...}  
for i:=1 to 6 do begin {Draw each character in a different color}  
SetColor(i);  
OutText(TZ[i]);

end;

SetColor(Yellow); {Draw in Yellow}  
Line(450, 35, 575, 35); {Put in the "speed" lines}  
Line(475, 45, 600, 45);  
Line(500, 55, 625, 55);  
SetTextStyle(TriplexFont, HorizDir, 4); SetColor(Cyan); {Change size & color}

{Write out the rest of the screen}

MoveTo((GetMaxX-TextWidth(Msgs[1])) div 2, 80);  
OutText(Msgs[1]);  
SetTextStyle(TriplexFont, HorizDir, 3);  
MoveTo((GetMaxX-TextWidth(Msgs[2])) div 2, 115);  
OutText(Msgs[2]);

{Switch fonts, color. DefaultFont is always available. No need to load or register it.}

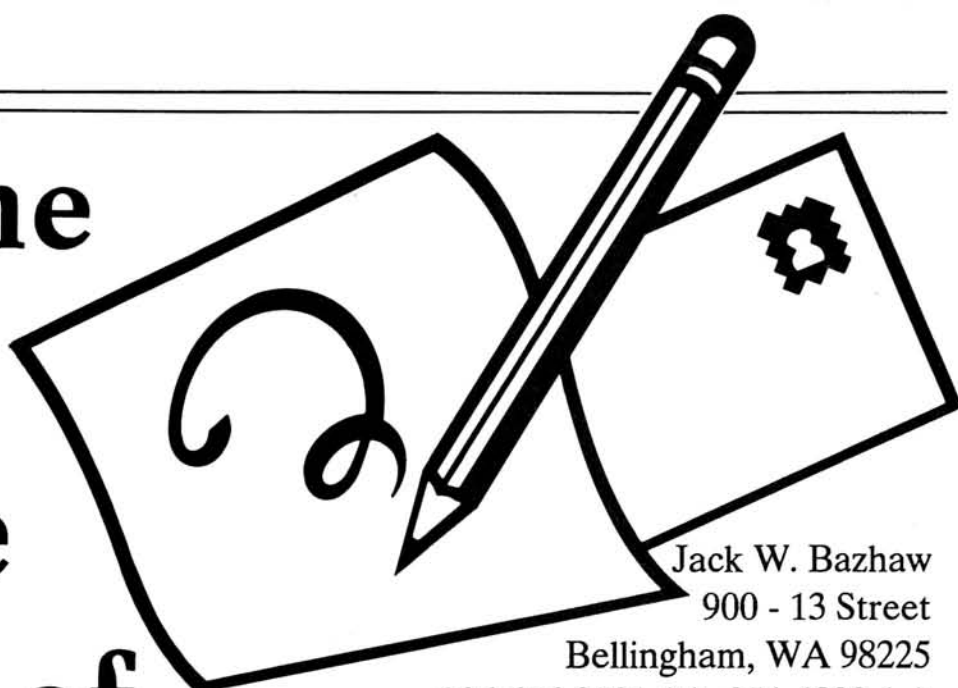
SetTextStyle(DefaultFont, HorizDir, 1); SetColor(Yellow);

MoveTo((GetMaxX-TextWidth(Msgs[3])) div 2, 140);  
OutText(Msgs[3]);  
MoveTo((GetMaxX-TextWidth(Msgs[4])) div 2, 155);  
OutText(Msgs[4]);  
repeat until KeyPressed;  
end. {Sample}

\*



# Using the Mail Merge Feature of MicroSoft Word



Jack W. Bazhaw  
900 - 13 Street  
Bellingham, WA 98225  
206 676-9531 (h) 354-4892 (w)

Mail merge. Sort of conjures up images of huge line printers throwing out reams of paper a minute and a mail room toting sacks of mail out the door. Also sounded to me like something I would have little use for in a word processor. That is until recently.

Perhaps you have children that are graduating from school. You'd like to avoid the hassle of getting thank-you notes written for graduation presents. If you have a script-like font for your printer you might find computer-generated notes a viable alternative to none at all.

My first need for mail merge required using a word processor system that was as flexible as a concrete fishing pole. Put the current date in your letter? Yes sir, as long as you want it smack against the left margin and formatted in military style (25 Aug 88). Have some addresses that use three lines and others that need four? No problem. We'll just throw a blank line in to make them all the same. Sorry, no changes allowed in the text of the letter; this is a form letter you know.

If you have any need for mail merge capabilities you owe it to yourself to investigate using Word to fill that need. It is a very powerful system without being overpowering to learn.

Form letters can be created that don't look or read like they were cranked out en masse. All the formatting capabilities of your printer are available for such letters and even labels. Decision making capability can be built in to change the text based upon the contents of the data document. Plus, you can perform calculations on numerical data and enter the result.

The basic idea behind Word's approach requires you to create two documents. The Main document or file contains text that does not change and the instructions for handling the text that does change. The Data document contains the text that changes. Then with the Print Merge command the two files are brought together to create an individualized document. You are not limited to merging two documents; as many as are needed may be combined.

## Creating the Main Document

I suggest you create your Main document first. That will reduce the chance you omit a data element from the Data document. If you have a data element, or field in the Main document that does not exist in the Data document Word will give you an error message. The reverse is not true because it is not necessary to use all data fields in the Main documents. This gives

more flexibility as you can keep the Data document crammed with everything you might ever need, yet use only a portion of it for any one task.

We'll create a Main document to be used as a thank-you note with variety. Start Word with a clean screen and save it as **THANKS.DOC**. Our first step is to tell Word the name of the Data document (which we will create later). Let's call it **LOOT.DOC**. Type that as the very first word. So Word will understand that **LOOT.DOC** is not just plain ordinary text, we need to do two things. First, we will enclose **LOOT.DOC** inside chevrons (« »). To make them, press **CONTROL-[** for the left one and **CONTROL-]** for the right side. Now we tell Word that **LOOT.DOC** is our Data file and not a data field by adding the word **DATA** before the file name. The line has to end with a paragraph mark (¶), so press **END** then press **RETURN**. The first line should look like this:  
«DATA LOOT.DOC»¶

If you do not see the paragraph mark, set Options to Partial (press **ESC O P** and **RETURN**).

The date will be our next entry. Instead of typing in the current date or some future date you hope to get the notes mailed by, use Word's dateprint function to add the

```
1 «DATA LOOT.DOC»¶
    (dateprint)¶

    «FName» «LName»¶
    «address2»¶
    «address»¶
    «city», «state» «zip»¶

    Dear «relation» «FName»,¶
    ♦
```

**Figure 1**

date the day you print the notes. Just type `dateprint` and press `F3`. To center the paragraph press `ALT X C` or select the appropriate style from your style sheet.

Keeping with the spirit of things, our thank-you notes will be mailed in window envelopes to allow us to put the mailing address right in the note. Here we have some serious thinking to do. How we arrange the data in the fields will determine how flexible our letter can be made. For example, if we combine the first name, middle initial and last name in one field and call it "addressee" we can't use a salutation of "Dear Aunt Judy". It would come out "Dear Aunt Judy R. Jones".

Down on the page where you would start out with the mailing address type in `FName` and `LName`. Tell Word these are data fields by enclosing each word in chevrons. Make sure there is a space between the two words so the line looks like this:

```
«FName» «LName»¶
```

The next line will be the street address. We will allow two lines in case someone has a company name or such. Later we will handle eliminating a blank line for the majority that need only one address line. Type `address`, press `RETURN`, then `address2`, and add the chevrons so these two lines look like this:

```
«address2»¶
«address»¶
```

The final line of the paragraph for address entries will contain the city, a comma, state and the zip code. Type the following and note there is a space after the comma and between the state and zip entries:

```
«city», «state» «zip»¶
```

For the salutation, we'd like to include the family relationship (aunt, cousin, etc.). Being an informal note, we will use the first name. Start it as a new paragraph by pressing `RETURN`. Now type:

```
Dear «relation» «FName»,¶
```

Note that the word "Dear" is not enclosed within chevrons and will always print. Figure 1 is what our note should look like so far. The spacing between paragraphs is set by the style sheet.

Now we will add the body of the note. It will be individualized by calling the gift by name and including a remark about its usefulness. Type in the following, which includes the complementary close:

```
Thank you for the «gift»
for my graduation. «remark»¶
```

```
I really appreciate your
thoughtfulness.¶
```

```
Sincerely,¶
```

If we left `THANKS.DOC` the way it is it would work. However, an extra blank line would appear in most of the addresses. That blank line can be eliminated by telling Word to print only if there is an entry for `address2`.

Change the `address2` line to read:

```
«IF address2»«address2»¶
«ENDIF»¶
```

```
1 «DATA LOOT.DOC»¶
    (dateprint)¶

    «FName» «LName»¶
    «IF address2»«address2»¶
    «ENDIF»¶
    «address»¶
    «city», «state» «zip»¶

    Dear «relation» «FName»,¶

        Thank you for the «gift» for my graduation.
        «remark»¶

        I really appreciate your thoughtfulness.¶

    Sincerely,¶
    ♦
```

**Figure 2**

The IF statement says if `address2` has an entry print `address2` and go to the next line. If the field is empty nothing will be printed. Note the closing chevron after the `ENDIF` has been replaced with a paragraph mark which prevents a blank line being printed.

The completed Main document is shown in Figure 2.

Character formatting is applied to the note just as you would any other document. The manual states you can format the data fields by formatting the first character in the field name in the Main document. That is true, however, it is difficult to see a single character that is in italics or bold, particularly with a 43-line screen. In addition, I am more nimble clicking the mouse on a complete word instead of one character. Fortunately Word also accepts formatting applied to the complete word.

## Creating the Data Document

The next step is to create the Data document. Open a new window by clicking the right mouse button on the right border. Save it as `LOOT.DOC`.

Our first line needs to contain the field names and they must be spelled *exactly* the same as in the Main document. The easiest way for me is to have each document in a window, click on the field names in the Main document, copy them to scrap, then insert them into the other window. They can be entered in any order as far as Word is concerned but you will find a certain natural order, like city, state, zip, easier to deal with at data entry time. Do



```
FName,LName,address2,address,city,state,
zip,relation,gift,remark¶
```

Figure 3

LOOT.DOC

not copy the chevrons into the Data document; instead separate each field name with a comma. Press ENTER to end the line with a paragraph mark. You should see something like figure 3 on screen.

Now comes the fun part, entering the data. In most cases you will have no entry for address2. A blank field still requires the comma to separate it from the other fields which means you will have two commas right together. If you have a field that contains a comma put the entire field in quotation marks. A field that contains quotation marks should be enclosed in quotation marks and the ones in the field should be doubled. Your paragraph formatting for the Data document should not add any blank lines between paragraphs. Select **FORMAT PARAGRAPH** (press ESCAPE F P) and make sure the "space before" and "space after" are set to zero and "line spacing" is one line.

Unless you never make mistakes it would be a good idea to enter only two or three data records and then make a test print. Try entering this sample data, which illustrates some of the points to watch for in the field entries:

```
Billy Bob Doe;1734 South
Main St., Apt 17;Tulsa, OK
77777;uncle sent Mercedes
convertible;Gee, Uncle
Billy Bob, the car is just
what I wanted. I love the
color.
```

```
Sahara Jane;c/o Timber-
lawn;222 S. Forest;Atlanta,
GA 55555;aunt sent dozen
compact disks;Those origi-
nal recordings of Elvis are
fantastic Aunt Sahara.
```

```
Vicki Hunt;3333 St. Cloud
Cr.;Costa Mesa, CA
99999;sister sent 22 tick-
ets to "Greatful Heap" con-
cert;What a great gift,
sis.
```

## Printing

The moment of truth is at hand. Go back to window 1 which contains the Main document by pressing F1 or click in the window. Make sure your printer is on and press ESCAPE P M P RETURN to select the PRINT MERGE PRINTER command. If everything went just right you should now have three thank-you notes printed. If you are getting error messages check for missing commas for blank fields and fields with commas that should be enclosed in quotes. Make sure the active window is the one containing THANKS.DOC. (The active window has its number highlighted.) If all else fails, compare your entries with figure 4. The finished letters are shown in figure 5.

If you need to print only certain of the records, use PRINT MERGE OPTIONS

to list just the desired records. Record number 1 is the line right after the field names. For a file with too many records to count the numbers off of the screen, you can add a new first field to hold the record number. Use **LIBRARY NUMBER** to keep the numbers in order after sorting or making additions or deletions.

## Sorting

If your data is such that you will need to sort it you may wish to modify the Data document. If you have only one field to sort on, like the zip code or name, the easiest method is to make that field the first entry and use Word's **LIBRARY AUTOSORT** command. Choose **NUMERIC** to sort zip codes and **ALPHANUMERIC** to sort names.

A bit more work is required if you need to sort more than one field. The Data document should use tabs and not commas to separate the fields. None of the lines should wrap. If necessary increase the "width" in **FORMAT DIVISION MARGINS** (22-inch maximum) to allow each record to fit on just one line. Then set your tab stops so that you have everything lined up in columns.

To set the tab stops, display the ruler (if not already) by pointing the mouse to the

```
1 «DATA LOOT.DOC»¶
      (dateprint)¶

«FName» «LName»¶
«IF address2»«address2»¶
«ENDIF¶
«address»¶
«city», «state» «zip»¶

Dear «relation» «FName»,¶

      Thank you for the «gift» for my graduation.
«remark»¶

      I really appreciate your thoughtfulness.¶

Sincerely,¶
```

Figure 4

THANKS.DOC

```
2 FName,LName,address2,address,city,state,
zip,relation,gift,remark¶
Billy Bob,Doe,, "1734 South Main St., Apt 17",Tulsa,
OK,77777,Uncle,Mercedes convertible,"Gee, Uncle
Billy Bob, the car is just what I wanted. I love
the color."¶
Sahara,Jane,c/o Timberlawn,222 S. Forest,Atlanta,
GA,55555,Aunt,dozen compact disks,Those original
recordings of Elvis are fantastic Aunt Sahara.¶
Vicki,Hunt,,3333 St. Cloud Cr.,Costa Mesa,CA,99999,
Sister,"22 tickets to ""Greatful Heap"" concert",
"What a great gift, sis."¶
```

LOOT.DOC

upper-right corner of the window and click the left button. Select the entire Data document (SHIFT F10), then choose **FORMAT TAB SET** (or press ALT F1). Point on the ruler where you want to set a tab then click the left button (click both to remove a tab).

Once you have the tabs set so the data fields line up into columns you are ready to sort. Select the **least** important column to sort first. Position the cursor in the upper-left corner of the column and press SHIFT F6 to turn on column selection (CS should appear on the status line of the screen). Move to the bottom-right corner of the column with the mouse or arrow keys and the entire column should be highlighted. Choose **LIBRARY AUTOSORT**, making sure the "column only" field is set to NO.

Continue sorting the columns from the least important up to the most important one last. If you have a problem during the sort you must use **UNDO** immediately after the sort for it to work.

## Conclusion

By now you should have a feel for using mail merge and how just flexible Word is. For example, the order of the entries in the Data document is not important nor must there be a one-for-one match with fields used in the Main document. The same Data document may be used to print different documents (letters and labels, for example) by using different Main documents. With the **ASK** and **SET** instructions you can provide the field entries yourself at the time of printing.

We used window envelopes in the example and didn't need mailing labels, but Word can print them in single-column or multi-column with continuous-feed or sheet-feed label stock.

If your data information already exists in a spreadsheet or data base, you can import the data into Word with an ASCII file.

(The social implications of computer generated thank-you notes are not a part of this discussion but is left as an exercise for the reader. They were used only as an example of the mail merge function.)

September 15, 1988

Billy Bob Doe  
1734 South Main St., Apt 17  
Tulsa, OK 77777

Dear Uncle Billy Bob,

Thank you for the Mercedes convertible for my graduation. Gee, Uncle Billy Bob, the car is just what I wanted. I love the color.

I really appreciate your thoughtfulness.

Sincerely,

September 15, 1988

Sahara Jane  
c/o Timberlawn  
222 S. Forest  
Atlanta, GA 55555

Dear Aunt Sahara,

Thank you for the dozen compact disks for my graduation. Those original recordings of Elvis are fantastic Aunt Sahara.

I really appreciate your thoughtfulness.

Sincerely,

September 15, 1988

Vicki Hunt  
3333 Ft. Cloud Cr.  
Costa Mesa, CA 99999

Dear Sister Vicki,

Thank you for the 22 tickets to "Greatful Heap" concert for my graduation. What a great gift, sis.

I really appreciate your thoughtfulness.

Sincerely,

## Figure 5



Did you know that HUG has a small business accounting package? Its unique name is **Accounting System**. As with most HUG software, it is user-friendly, double entry, can handle up to 999 separate accounts during any calendar year, and is available for ANY Heath/Zenith computer with a double density disk drive. The different versions available are as follows: **CP/M — P/N 885-8047-37, Z-DOS/MS-DOS — P/N 885-8048-37, MS-DOS — P/N 885-8049-37.**



# Protect Your Setup

**Pat Swayne**  
HUG Software Engineer

One of the nice features of Heath and Zenith computers is the ability to enter the ROM monitor by pressing certain key combinations for the purpose of changing the system setup or performing tests. You can hold down the Ctrl and Alt keys, and then press Enter to break from whatever program you are running and enter the built-in debugger. Or, you can hold down Ctrl and Alt, and press the Ins key to restart the computer, and enter the monitor at the reset point. After you enter the ROM monitor using either of these methods, you can execute any of its commands. If you have any of the computer models that have an 80286 or 80386 processor, one of the commands you can execute is the Setup command.

Unfortunately, there are times when you may not want to allow the user to access the Setup command or any other of the monitor commands. The computer coordinator of a school wrote me about a problem that access to the Setup command was causing with his Z-248 computers. Some of the students were altering the way the computers were set up to the extent that they would no longer function. He asked me if I could write a program

that would disable access to the ROM monitor. I could, and did, and the result is contained in this article.

My setup protection program is called NOMON.COM. If you type in and run the following BASIC program, it will create a copy of NOMON.COM on the default drive and directory.

```
10 PRINT "CREATING NOMON.COM"
20 OPEN "O",1,"NOMON.COM":L=100
30 FOR I=1 TO 16 :C=0:FOR J=1 TO 8
40 READ B:C=C+B:PRINT #1,CHR$(B):NEXT J:READ S
50 IF S<>C THEN PRINT "TYPING ERROR IN LINE":L:STOP
60 L=L+10:NEXT I:CLOSE #1:SYSTEM
100 DATA 235,96,0,0,0,0,0,80,411
110 DATA 228,96,60,29,117,8,46,128,712
120 DATA 14,6,1,1,235,54,60,157,528
130 DATA 117,8,46,128,38,6,1,254,598
140 DATA 235,42,60,56,117,8,46,128,692
150 DATA 14,6,1,2,235,30,60,184,532
160 DATA 117,8,46,128,38,6,1,253,597
170 DATA 235,18,60,28,117,2,235,4,699
180 DATA 60,82,117,8,46,128,62,6,509
190 DATA 1,3,116,6,88,46,255,46,561
200 DATA 2,1,228,97,12,128,230,97,795
210 DATA 36,127,230,97,176,32,230,32,960
220 DATA 88,207,51,192,142,216,190,36,1122
230 DATA 0,191,2,1,252,86,165,165,862
240 DATA 94,199,4,7,1,140,76,2,523
250 DATA 14,31,186,98,1,205,39,0,574
```

To use NOMON.COM, place a copy of it in your root directory or a directory pointed to by your PATH command. Then add a line with the command NOMON to your AUTOEXEC.BAT file. Then, when you reset your computer, NOMON will install itself into memory, and disable the Ctrl-Alt-Ins and Ctrl-Alt-Enter key combinations. You will still be able to reset your computer with Ctrl-Alt-Del. It should be

set up to auto boot a disk rather than enter the monitor when you use that key combination. If you want to restore the ability to enter the monitor, you can rename NOMON.COM to something else and then re-boot. (But don't let your students know that, and don't let them see this article!)

Below is the assembly source listing for NOMON.COM.

```
; NOMON - A PROGRAM TO DISABLE ZENITH MONITOR ACCESS
; BY P. SWAYNE, 06-MAR-89
; COPYRIGHT (C) HEATH/ZENITH USERS' GROUP 1989.
; ALL RIGHTS RESERVED.
```

```
CODE SEGMENT
ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE
ORG 100H

START: JMP SETUP
```

```
INT09V DW 0,0 ;INT 9 VECTOR
KEYFLG DB 0 ;KEY FLAG
```

```
; KEYBOARD INTERRUPT PROCESSOR
```

```
INT09: PUSH AX
IN AL,60H ;READ KEY
CMP AL,1DH ;CTRL?
JNZ NOTCD ;NO
OR CS:KEYFLG,1 ;ELSE, FLAG IT
```

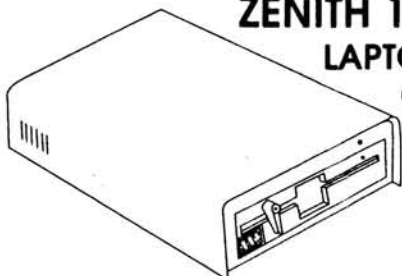


# WELTEC

## W 525 Subsystem

### 5 1/4" EXTERNAL DRIVE FOR ZENITH 181 & 183

### LAPTOP COMPUTERS



- EXTERNAL 5 1/4" DRIVE FOR ZENITH 181 & 183
- ACTS AS ADDITIONAL SYSTEM DRIVE
- TRANSPORT AND SHARE DATA WITH PCs
- POWER SUPPLY & CABLE INCLUDED
- EXTERNAL DRIVES FOR OTHER LAPTOPS AVAILABLE

**WELTEC**

17981 Sky Park Circle, Suite M

Irvine, CA 92714

(714) 250-1959

Reader Service #194

## VOICE MASTER KEY<sup>®</sup> VOICE RECOGNITION SYSTEM FOR PC/COMPATIBLES & TANDY 1000 SERIES A FULL FEATURED VOICE I/O SYSTEM

GIVE A NEW DIMENSION TO PERSONAL COMPUTING. . . The amazing **Voice Master Key System** adds recognition to just about any program or application. Voice command up to 256 keyboard macros from within CAD, desktop publishing, word processing, spread sheet, or game programs. Fully TSR and occupies less than 64K. Instant response time and high recognition accuracy. Voice recognition tool-box utilities are included. **A genuine productivity enhancer!**

**SPEECH RECORDING SOFTWARE.** . . Digitally record your own speech, sound, or music to put into your own software programs. Software provides sampling rate variations, graphics-based editing, and data compression utilities. Create software sound files you can add to macros for voice recognition verification response. **a complete, superior speech and sound development tool.**

**SOFTWARE CONVERSION CODES.** . . The **Voice Master Key System** operates a growing list of third party talking software titles using synthesized phonetics (text - to - speech) or digitized PCM, ADPCM, and CVSDM encoded sound files. **Voice Master Key system does it all!**

**EVERYTHING INCLUDED.** **Voice Master Key System** consists of a plug-in card, durable lightweight microphone headset, software, and manual. Card fits any available slot. External ports consist of mic inputs and volume controlled output sockets. **High quality throughout, easy and fun to use.**

**ONLY \$149.95  
COMPLETE**

**ONLY \$89.95 FOR TANDY 1000 SL/TL MODELS  
SOFTWARE ONLY—REQUIRES TANDY BRAND  
ELECTRET MICROPHONE.**

**ORDER HOTLINE: (503) 342-1271 Monday-Friday, 8AM to 5 PM Pacific Time**

Visa/MasterCard, company checks, money orders, CODs (with prior approval) accepted. Personal checks subject to 3 week shipping delay. Specify computer type and disk format (3 1/2" or 5 1/4") when ordering. Add \$5 shipping charge for delivery in USA and Canada. Foreign inquiries contact Covox for C & F quotes. **30 DAY MONEY BACK GUARANTEE IF NOT COMPLETELY SATISFIED. ONE YEAR WARRANTY ON HARDWARE.**

CALL OR WRITE FOR FREE PRODUCT CATALOG



**COVOX INC.**

675 Conger Street, Eugene, OR 97402

TEL: 503-342-1271 • FAX: 503-342-1283

Reader Service #196

```

      JMP      INT09X
NOTCD: CMP     AL,9DH          ;CTRL UP?
      JNZ     NOTCU           ;NO
      AND     CS:KEYFLG,0FFH-1 ;ELSE, FLAG IT
      JMP     INT09X
NOTCU: CMP     AL,38H          ;ALT?
      JNZ     NOTAD
      OR      CS:KEYFLG,2      ;FLAG IT
      JMP     INT09X
NOTAD: CMP     AL,0B8H          ;ALT UP?
      JNZ     NOTAU
      AND     CS:KEYFLG,0FFH-2 ;FLAG IT
      JMP     INT09X
NOTAU: CMP     AL,1CH          ;RETURN?
      JNZ     NOTRET
      JMP     CHKCA           ;CHECK FOR CTRL-ALT
NOTRET: CMP     AL,52H          ;INS?
      JNZ     INT09X
CHKCA: CMP     CS:KEYFLG,3      ;CTRL, ALT DOWN?
      JZ      ABSORB          ;IF SO, ABSORB KEY
INT09X: POP     AX
      JMP     CS:DWORD PTR INT09V
ABSORB: IN      AL,61H          ;PREPARE TO ABSORB KEY
      OR      AL,80H
      OUT     61H,AL          ;ABSORB KEY
      AND     AL,7FH
      OUT     61H,AL
      MOV     AL,20H
      OUT     20H,AL          ;RESET INT. CONTROLLER
      POP     AX
      IRET
ENDRES:

;      SET PROGRAM UP

SETUP: XOR      AX,AX
      MOV     DS,AX          ;POINT TO INT. SEGMENT
      MOV     SI,9*4          ;POINT TO INT 9 VECTOR
      OV      DI,OFFSET INT09V ;PUT IT HERE
      CLD
      PUSH    SI
      MOVSW
      MOVSW
      POP     SI
      MOV     WORD PTR [SI],OFFSET INT09 ;INSTALL NEW ONE
      MOV     2[SI],CS
      PUSH    CS
      POP     DS          ;FIX DS
      MOV     DX,OFFSET ENDRES
      INT     27H          ;INSTALL PROGRAM

CODE  ENDS
      END      START

```

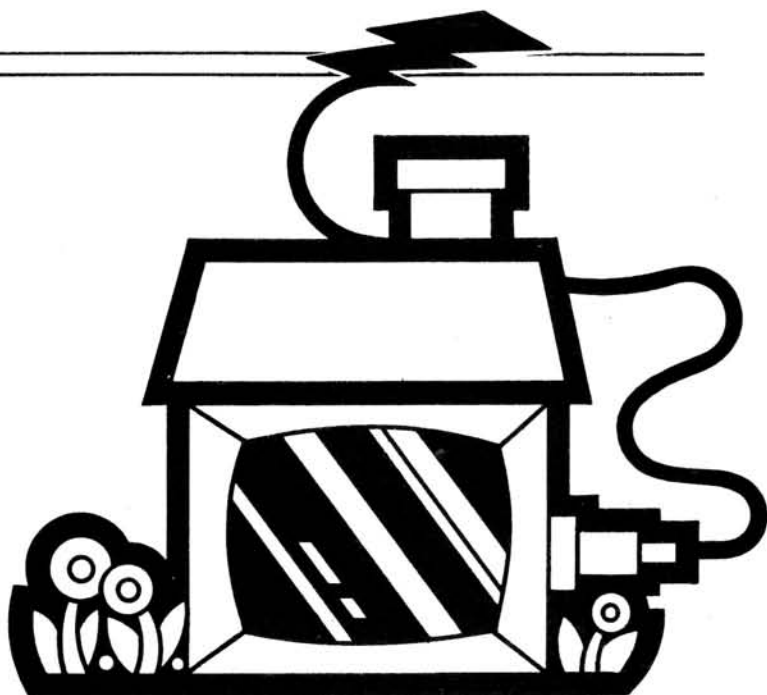
\*

# Glitches

On page 51 of the March 1989 issue of REMark, the program in the article "The Most Important Peripheral: Printers" has a couple of typos. In Line 380, the word should be MODE, not MOD. In Line 110, 2 letters are transposed in filename. We, at HUG, are sorry for any inconvenience this might have caused.

# The Use of Computers In The Home

Thomas W. Putnam  
233 Colmar Road  
Ft. Ord, CA 93941-1651



"What are we going to use a computer for?" That is probably one of the most common questions asked by a wife when the husband comes home one day and announces that "We ought to have a computer!". At least that was one of the reactions that I was greeted with when I made my announcement back in 1981. (In fact, there are also probably some husbands out there who asked the same question when the wife came home with the same statement!) I'd been looking at Heath computers for about 4 years by that time but had never bought one since I had never really decided what I NEEDED one for. Finally, though, the bug had bitten hard enough and I had decided that the time had come.

Having done my homework, I knew that for the venture to be successful I had to convince my better half that the investment was a good one and would be useful around the house. As a result, I had written down the standard list of reasons why we could use a computer in the family. The reasons were many. We could use it for word processing since I am about the world's worst letter writer, we could also computerize the family's favorite recipes, we could keep track of the contents of the family freezer so that we wouldn't end up with year-old ground beef packages hidden under the frozen vegetables and I could maybe even write a program to keep track of our household belongings and another one to help us keep track of our finances. Sound familiar? Well, it took me about three years to get most of those missions accomplished and I'm going to attempt to show how I set my financial and household inventory programs up and how they interface with one another.

## Areas

The article will deal with two basic areas: keeping track of household property and keeping track of household finances. Hopefully, the manner in which I set the areas up will help you in setting up your own personal programs and will save you the wasted efforts that I made due to having to backtrack and modify the programs that I have utilized.

Although I will be explaining the procedures I've used on the basis of the particular software programs that I'm presently using, I will attempt to do it in a manner which will lend itself to any software presently on the market. As a result, although my descriptions will reflect the use of certain programs, I will attempt to keep the principals as 'generic' as possible. As a point of interest, however, the particular software programs and hardware that I am using at present are Lotus 1-2-3, Jay Gold's Home Finance System, and Condor rDBMS and an H120 although I originally began on an H89 using Zencalc, Query!2, and Jay Gold's program.

## Household Property

The first thing that I did as far as utilizing the computer for home use was to begin cataloging all of the items that my family possessed. I originally utilized Hoyle & Hoyle's Query!2 but moved up to Condor when I had the chance to pick up a copy at a bargain price. Query!2 has to be one of the best bargains on the market in the data base field. (Yes, I know that Query!3 is the current version, but I have never worked with it and don't want to talk about something that I'm not familiar with.) It has some features that I wish Condor had, such as searching for a subfield of a record (actually Condor 3

does have the ability to do that by using an asterisk on either end of the subfield e.g. if you're looking for a book titled "Arabian Peninsula" but you can't remember if it's "The Arabian Peninsula" or "Arab Peninsula Land" or something else, you can search for it under Condor 3 by using the form "\*Peninsula\*" as your search criteria. Condor 20, unfortunately, does not have that ability, or the ability to search with ANDs connected by ORs or vice versa. This is especially handy when you can remember part of what is in a field but not the whole thing (Condor 20 can search on a subfield if the subfield is the first part of the field, but not if it's in the middle or end) or when you have a number of different conditions that you are trying to satisfy with only one search.

## Setting Up The Data Base

In setting up the fields, determine exactly what kind of information will be kept in each field. Originally, I defined the serial number field as being a numeric field. That worked okay until I tried to enter a serial number with a letter in it. Then it was back to the drawing board to redefine the field. Make sure that you leave enough space in each of the fields and give the fields a default value if your data base permits you to and if you have a particular entry that will be made more often than not. That will save you many keystrokes over the period of several hundred entries. I originally included only 5 spaces in the zip code field and had to go back and increase the size to 10 spaces when the zip plus program got started. One tip that I learned also is to make the zip code field an alphanumeric field. I made mine only numeric at first thinking that since only numbers would be en-



tered that no letters would be applicable. Then I noticed that when my data base was printed out that the zip codes that started with a 0 had them deleted so that "09710" would print out as "9710". By making the field an alphanumeric, I eliminated the problem so that now the leading zeros print out as they should. If the data base you use doesn't delete leading zeros in numeric fields, this will be unnecessary for you.

When defining your data base, take a few moments and define another one with the same definitions and format as your property data base. Name this one newprop or something similar. Enter new property items into this data base instead of the property data base. The reason being, if you need to go back and make any corrections, it's much quicker and easier to search through a relatively small data base than a large one. Also, if your data base has the ability, use the Auto Repeat function if you are entering a number of items that have repetitive fields, such as a number of books by the same company or a number of items of furniture by the same manufacturer. This becomes even more important when your property data base becomes so large that it's taking up 200K+ space. You can then just APPEND the entry data base to the basic one (Append property newprop) and you're all done. Don't forget to empty the entry data base after you append it to the basic data base (Empty newprop ok) or you'll have double entries the next time you append the 2 data bases. If your data base does not have an append-type feature, then you will probably not want to do this as it would serve no purpose.

If you index the data base, your searches on the indexed field(s) will be quite a bit faster, HOWEVER, your data base will take up quite a bit more room. For example, with Condor when you have a 258K .DAT file and two fields indexed, your .IDX file will be about 148K so that with 48TPI drives you will be out of disk space! Your problem then will be to decide whether to index or not, or to move up to 96TPI drives. Of course, if you're lucky enough to have a hard disk, that problem becomes moot.

With good planning, you can avoid all, or at least many, of the problems that I encountered. The first problem raised its head when after putting in a number of records, I decided that the size of some of the fields in the records weren't large enough and that I needed more fields than I had. I originally included only a small number of fields such as item, date purchased, cost, brand, where purchased, and comments. The more I worked entering items into the data base, the more I realized that the 'planning' that I had done was less than adequate and that I had better sit down and figure out exactly what data I was going to need to keep on each

PROPERTY INVENTORY

---

[ITEM]: \_\_\_\_\_ [SERIAL.NO]: \_\_\_\_\_ [BRAND]: \_\_\_\_\_

[COST]:\$ \_\_\_\_\_ [PURCH.DATE]: \_\_\_\_\_ [PURCH.FROM]: \_\_\_\_\_

[ADDRESS]: \_\_\_\_\_ [CITY]: \_\_\_\_\_ [STATE]: \_\_\_\_\_

[ZIP.CODE]: \_\_\_\_\_ [TYPE.PROPERTY]: \_\_\_\_\_ [NO.IN.SET]: \_\_\_\_\_

[COMMENTS]: \_\_\_\_\_

**Figure 1**

Attributes of the PROPERTY Data Base

<u>Item</u>	<u>No. of Spaces</u>	<u>Type of Data</u>	<u>Description</u>
ITEM	40	Required, alphanumeric	Names the item to be stored.
SERIAL.NO	15	Alphanumeric	Serial number of the stored item.
BRAND	15	Alphanumeric	For identification and insurance.
COST	8	Required numeric	Self-explanatory.
PURCH.DATE	8	Date	Self-explanatory.
PURCH.FROM	16	Alphanumeric	Self-explanatory.
ADDRESS	25	Alphanumeric	Self-explanatory.
CITY	15	Alphabetic	Self-explanatory.
STATE	2	Alphabetic	Self-explanatory.
ZIP.CODE	10	Alphabetic	Self-explanatory.
TYPE.PROPERTY	15	Required alphabetic	Allows the classification into general groupings.
NO.IN.SET	2	Numeric	For silverware, china, etc.
COMMENTS	60	Alphanumeric	Short description of the items.

**Figure 2**

Types of Property

Appliance	Hobby
Automobile	Jewelry
Book	Kitchenware
Camping	Laser disk
Ceramic	Misc
Clothing	Pewter
Coin	Photography
Collection	Porcelain
Computer	Professional
Crystal	Record
Dishes	Silver
Electronics	Software
Fishing	Sport
Furnishing	Stamp
Furniture	Tool
Gun	Xmas

**Figure 3**

item before I went any further. So, I sat down one night with a pencil and piece of paper (Hey, wasn't the computer supposed to keep me from having to do this?) and decided what was necessary.

Once I had that decided, I went back to the computer and redesigned the format of the data base and the definitions for the fields. The result (See Figure 1) has been almost completely satisfactory. Al-

though looking at the format it may seem that I have decided to keep too much information but I would rather keep too much than not enough.

Notice that everything that is kept has a purpose. Most of the items are self explanatory, but a brief description of each individual entry is shown in Figure 2. Item, serial number, cost and brand are useful in case of loss through theft or fire. Purchased From, Purchase Date, Address, City, State and Zip Code are useful for insurance claims, should they be necessary. They are also useful in that data can be pulled from the files to be used in addressing letters to stores where purchases were made for warranty information, complaints or whatever other reasons you might have for writing. Sometimes it's awfully hard to recall exactly where or when something was purchased and it is very helpful to be able to quote a date and/or voucher number when writing to a business or company. Thus one of the things you might want to do is enter the vendor's receipt number in the comments block or even create another field in your data base to store the information. If you do that, however, make the block alphanumeric as many businesses have letters in their voucher numbers.

The Type Property entry is to define large, general categories under which you can group the individual possessions (see Figure 3 for a suggested list of property types) and No. In Set is handy for recording such things as dishes, silverware, sets of items, etc. where it would be impractical or impossible to list each individual piece. You can imagine the headache it would be to print out every piece of your china or silverware individually, to say nothing of the labor required to input the data. Conversely, to only make an entry of China would be of only limited use. How many total pieces would need to be known for an insurance claim and you would probably forget the serving pieces that you had in the stress filled time frame of a post-fire or post-burglary insurance claim. You can even use a code such as '999' to indicate an indeterminate number of items in a set since it would be a rarity to have that many of an item. This might be of use should you ever get down to where you are keeping track of how many sets of underwear you have but you don't really want to count them. (Really, I'm not crazy! Remember, we can keep a record of every little thing we own now and someone might want to keep that close track of their possessions.)

I've often found that when I sit down to enter data there is invariably some piece of information that I can't find but which I know I have somewhere. Because of this, there is one item that I would include in the data base if I were starting over again and that is a one-space entry which I would title Complete. The default

STORAGE

---

[ITEM]: \_\_\_\_\_ [TYPE-PROPERTY]: \_\_\_\_\_

[KEYWORD1]: \_\_\_\_\_ [KEYWORD2]: \_\_\_\_\_

[KEYWORD3]: \_\_\_\_\_ [KEYWORD4]: \_\_\_\_\_

Figure 4

* * * Codes * * *	
Expense Codes:	
1. Rent	16. Fabric/crafts
2. Utilities	17. Genealogy
3. Food-bulk	18. Presents
4. Food-other	20. Insurance-car
5. Laundry/cleaning	21. Insurance-home
6. Home expenses	30. Car-gasoline
7. Personal care	31. Car-maintenance
8. Juv-Clothing	40. Clothing
9. Juv-Child Care	50. Rental Expense
10. Juv-Education	52. Land
11. Juv-Toys	53. Tax
12. Books/records/mags	61. Investments/savings
13. Entertainment	79. Interest
14. Computer	80. Miscellaneous
15. Juv-Activ. & Misc	
Deposit codes:	
1. Salary-Dianna	4. Investment Income
2. Salary-Tom	5.
3. Rental Income	6.

Figure 5

setting for the field would be an N and that would indicate that all of the information on that record was not present. Any time I had completed all of the entries for a particular record I would enter a Y in the field. I would then be able to search the entire data base and look only at the records that might not be complete in any way. This is probably not very desirable to most people but I like to have everything as complete as possible.

By printing out the data base using Type Property as the grouping item and using only the Item, Cost, Serial No. and Brand fields I am able to get a printout on normal paper that I can use as a ready reference to what I have and what I don't have on the data base. I could use more fields and compressed print but this provides me with what I usually need to work with. The width of your paper and the capability of your printer to do the compressed printing along with the number of characters in each field will determine what you can put on your printout. Of course you can always make as many different reports as you want and include whatever items you want in each different one, but if the data base you use has a report generator of any sort, then you may want to come up with a 'standard' setup

for your most common report that will include only that information that you will need or want on a recurring basis.

It is absolutely necessary that you keep a copy of your data disk in a separate location from your home. There is nothing more useless than a back-up disk that resides in the same, or nearby, disk storage box and is stolen, burned or otherwise damaged right along with the original. I would suggest keeping the back-up disk in either a safety deposit box or in your office at work. If you have close friends or relatives that live near you, then it might be best for you to have them keep your backup disks. Don't do what one acquaintance of mine did. He was driving his camper through the mountains back east while on a genealogical fact-finding trip when his brakes failed on a down grade. The truck, the camper, the computer that was in the camper along with a large number of disks all went into a lake along with my friend. As he says "But I was smart - I had backups of my important disks. My original disks were in the front of the camper while the backup disks were all the way, safe and sound, in the back of the camper!"

### Storing Belongings

If your family is like mine, you always have more belongings than will comfortably fit in your home so that many things end up stored in boxes in the garage, the attic, the storeroom, or wherever. Of course, you can never find where the particular item is that you need until after don't need it any more. I finally got tired of looking through box after box for things so that when we moved back from overseas I took some extra time and sat down and listed everything that went into each box of stored goods. Then I placed a number on each of the four sides of the cardboard box using a large tipped black felt marker. I made up a 'quick and dirty' data base for the items in storage which I named Storage. Original, huh? This data base contained the Box No., the Item, and up to 4 keywords upon which I can search (Figure 4). Since Condor doesn't have the ability to search a data base for a subfield as some others (such as Query!2) do, I had to set it up this way. When I search I just do a multiple search (Display storage where keyword1 is Ceramic or keyword2 is Ceramic etc). Not the fanciest way to do it, but effective. I do

the same thing with Storage that I do with Property in that I have a second data base labeled Newstore where I initially enter all new entries. If you have your storage boxes located throughout your house, then you would probably want to add a field to your storage data base with a name such as Location to allow you to show where the box is stored.

## Finances

The setting up of financial records is a bit more involved than setting up the data bases. Here, the problem is determining what different expense codes you want to utilize. I must have added, deleted and changed my expense codes a half dozen times, and even now I find changes that I want to, or need to, make. My present codes (Figure 5) have pretty well covered most of my needs, especially when considering that there are presently only 30 expense codes to work with. The Home Finance System also allows you to group your codes into as many as 6 different groupings (Figure 6). Thus, you can analyze your expenditures in more general terms than the individual codes allow. For someone who is self-employed this would allow you to place all of your employment expenses into one category which should make your accounting easier at tax time. Right now you're probably thinking "He must be crazy, 30 expense categories should be plenty!" That's what I thought too, at the start. However, when Miscellaneous becomes the single largest expense category you have, that's a pretty good sign that you need to utilize some more categories in your expense tracking or at least look into consolidating the ones you have.

One of the things that we wanted to do was to see exactly how much it costs to raise a child today. After hearing all of these horrendous claims about how much it was costing the average family to raise children, we were curious. This explains why there are so many categories labeled Juvenile. You notice that one of them is separated from the others. This shows how you have to plan your codes in advance so that you don't end up spread all over within the various code groupings.

All this should clearly indicate that the first thing you must do after you decide that you're serious about tracking your expenditures is to sit down and write down every expense code that you can come up with. Don't worry about how many you have at this point, right now you're more concerned with coming up with a thorough listing of as many as possible of the expense categories that your yearly purchases will fall into. After you come up with your list, THEN, you worry about paring down the list to fit into the number of categories that can be handled by your particular program. If you end up

* * * Code Groups * * *			26-Mar-87
=====			
Living			
=====			
3. Food-bulk	4. Food-other	5. Laundry/cleaning	
6. Home expenses	7. Personal care	40. Clothing	
Juvenile			
=====			
8. Juv-Clothing	9. Juv-Child Care	10. Juv-Education	
11. Juv-Toys	15. Juv-Activ. Misc		
Entertain.			
=====			
12. Books/records/mags.	13. Entertainment	14. Computer	
16. Fabric/crafts	17. Genealogy		
Fixed			
=====			
1. Rent	2. Utilities	20. Insurance-car	
21. Insurance-home	30. Car-gasoline	31. Car-maintenance	
Inves/save			
=====			
50. Rental Expense	52. Land	61. Investments/savings	
Other			
=====			
18. Presents	53. Tax	79. Interest	
80. Miscellaneous	96. Instcash	97. Bank charge	

Figure 6

having to group a number of categories into one then you may still be able to track the 'sub-categories' through your comments on any expenditures entry. For example, suppose that your hobby is computers (naturally!) and your wife's is stamp collecting but you don't have enough categories to list both of them. You could then combine them under the single category of Hobbies and place a (C) or an (S) in the comments section of your check register to differentiate between the two. It's not the best solution but it's one that is workable.

Once you have decided on your expense categories and have grouped them into similar areas such as Food, Insurance, Business, etc. you must then assign numbers to the categories. Here we take a lesson from what we learned when writing Basic programs where we numbered our line numbers in multiples of tens. Begin by numbering the areas in multiples of ten (or some other number should you desire). Insure that there are enough numbers between the areas to allow for all of the categories. This should be no problem as I don't imagine that anyone will have over 10 codes within one area. Next, allow for 1 or 2 empty numbers at the end of every area to allow for future expansion. For example, if we number our Juvenile expenses from 20 to 25, then we wouldn't want to begin numbering our next major grouping at 26, as this would not allow us to add another Juvenile code later and maintain our continuity. Instead, we would begin the next grouping at maybe 28, or even 30 if we had the space. You may want to leave one or two categories vacant in case something

comes up after you've started record keeping that you really feel the need to keep track of. This is especially true if your program has a large number of categories available.

Some explanation of some of the codes is in order since there may be some confusion on them. Basically, the differentiation that we made between Food-bulk and Food-other is where they are purchased and the intent. If we purchase food in a grocery store, market or convenience store for home consumption it's a bulk food purchase. If we buy a coke, eat out or something on that order it's another food purchase. How close you want to call things depends entirely upon your desires and interest.

Many times, things will fall into 2 or more categories and it's going to be strictly a decision on your part where you will list them. For example, if you and your wife eat out and the meal comes to \$24.90, and \$1.49 tax and a \$5.00 tip, there are a couple of ways you can list them. You can do it the easy way and list the whole \$31.39 against Other food and be done with it. Or you can list it as \$24.90 - Other food, \$1.49 - Tax, and \$5.00 - Miscellaneous. I prefer the latter for a couple of reasons. First it's more accurate and secondly, it breaks a taxable item (sales tax) out where I can keep track of it. Of course with the new tax laws, the relevancy of keeping track of sales tax becomes somewhat moot. In the end though, you are going to be the determining factor. The prime concern should be, whatever your decision, to keep your method constant throughout your record keeping. A second 'problem' arises when



you make a purchase where the item can be charged against two different codes. For example, if you go out and renew your subscription to *REMark* what do you charge it against here - Computer or Magazine? Here I try and determine what the primary purpose of the item purchases was or is going to be. In this example, the primary purpose the magazine was purchased for was due to the interest in computers so that the expense would be charged against Computers. *Life* magazine would be charged to Magazines and *Knitting Today* would be charged to Crafts. Again, consistency is the key here. Once you make your determinations as to how you want to work things, it would be advisable to write down the 'rules' that you've established and keep a copy near the computer so that when you're making entries you have a ready reference. Otherwise, you'll find yourself charging something against one expense code one time and another code the next.

### Daily Expenditures

One of the problems that exists with HFS in its present configuration is that there is really no good way of tracking cash purchases. One way to do it is to initiate an Asset account labeled Cash. Then whenever you write a check for cash you deposit that amount to the asset account and everytime you make a cash purchase you deduct the appropriate amount from the Cash account and charge the appropriate expense code. At first I didn't think of doing it that way and came up with a method using a spreadsheet (Figure 7). When I finally figured out that I could account for our cash expenditures with the Cash account method, I noted that that would entail a large number of entries to the Home Finance System on a daily basis and I didn't want to do that. Of course, looking back, it's probably six of one and a half dozen of the other. Anyway, I decided that I would stick with the spreadsheet method.

When I set up my method I included all of the expense codes in my spreadsheet that I thought I would need. Thus, all of the expense codes are not included in the spreadsheet (Most people don't pay rent or utilities in cash). By entering each daily expense in a small pocket notebook during the day and then entering them into the computer every day or two, an accurate running total is maintained. At the end of the month comes the second part of the problem, however. Now that we've written these 4 or 5 (or however many) checks for cash during the month (all of which showed an expense code of 80-Miscellaneous), how are we going to divy up the various expense codes? Here, unfortunately, I have to revert to the old pencil and paper routine. I go through each expense code and break down each check into 3 expense codes,

Jan Code:	Food-blk (3)	Food-oth (4)	Pers-care (7)	Juv-cloth (8)	Misc (80)	Totals	Jan ←Codes
1	0.86	0.00	0.00	0.00	2.15	3.01	1
2	1.65	0.00	0.00	0.00	0.00	1.65	2
3	10.46	0.00	8.25	0.00	2.64	21.35	3
4	0.00	0.00	0.00	0.00	0.33	.33	4
5	0.00	1.05	0.00	0.00	0.00	1.05	5
6	0.00	6.16	0.00	0.00	0.00	6.76	6
7	0.00	0.86	0.00	0.00	0.00	0.86	7
8	0.00	19.88	0.00	0.00	0.00	19.88	8
9	0.00	5.02	0.00	0.00	3.53	8.55	9
10	3.40	2.84	5.60	0.00	2.00	13.84	10
11	0.00	1.00	0.00	0.00	0.00	1.00	11
12	0.76	4.85	0.00	0.00	7.50	13.11	12
Totals: Code:	17.13 (3)	42.26 (4)	13.85 (7)	0.00 (8)	18.15 (17)	91.39	Totals ←Codes

Figure 7

1987 Code: Month	Food-bulk (3)	Food-other (4)	Pers care (7)	Juv-cloth. (8)	Misc (80)	Totals
Jan	39.89	113.62	30.60	43.61	87.15	314.87
Feb	97.32	105.56	52.23	83.15	66.49	404.75
Mar	57.82	97.70	40.77	22.62	78.21	297.12
Apr	69.31	202.41	92.75	40.48	132.71	537.66
May	87.41	166.71	16.85	45.45	127.34	443.76
Jun	0.00	0.00	0.00	0.00	0.00	0.00
Jul	0.00	0.00	0.00	0.00	0.00	0.00
Aug	0.00	0.00	0.00	0.00	0.00	0.00
Sep	0.00	0.00	0.00	0.00	0.00	0.00
Oct	0.00	0.00	0.00	0.00	0.00	0.00
Nov	0.00	0.00	0.00	0.00	0.00	0.00
Dec	0.00	0.00	0.00	0.00	0.00	0.00
Totals: Code:	351.75 (3)	686.00 (4)	233.20 (7)	235.31 (8)	491.90 (80)	1,998.16
Mo. avg	70.35	137.20	46.64	47.06	98.38	399.63
Mo. max	97.32	202.41	92.75	83.15	132.71	537.66
Mo. min	39.89	97.70	16.85	22.62	66.49	297.12

Figure 8

trying to get rid of one or two small expenditures and using the rest of the total to decrease one of the larger codes (Figure 5). I do this for each check for cash until all cash expenditures are accounted for. Hopefully, the number of checks for cash and the cash expenditures come out pretty close at the end of the month. If I had to start all over again, I believe that I would go with the Cash asset account method, since it would end up being easier in the long run.

The last thing I do is to transfer the data from the monthly spreadsheet to the annual one (Figure 8). Care should be taken here if you try and make a transfer by extracting the totals row from the monthly file, saving it to another file and then reading it into the annual field. When I get ready to print out my monthly cash expenditures spreadsheet, I delete any columns for which I had no expenditures during the month so as not to print out any unnecessary items. If you have deleted any column prior to saving the range, the correct columns in the saved range will not print to the corresponding expense column and will really mess up your figures. Actually, I keep two annual accountings. The first is strictly concerning cash expenditures, whereas the second includes both cash and check expenditures. The latter is actually provided internally by HFS also in its Print Using Codes program.

Although probably very rudimentary

in some ways, the system I have developed to keep track of my family's finances has served me very well during the past two and a half years. With the impending release of the new MSDOS version of HFS (tentatively planned now for around the middle of the summer) I look forward to even more success.

### The New HFS

As I mentioned earlier, a new Home Finance system - Version III - is due for release in the very near future (hopefully by the time this article is in print). A number of improvements and program enhancements have been included in this new version which will not only make the program more efficient but will also add new features.

Some of the improvements in the new version include:

(1) An increase from 3 to 14 in the number of expense codes which can be charged against each check. This will also be extremely helpful in paying bills which cover a vast area and a long period of time, such as Visa or MasterCard.

(2) Withdrawals from asset accounts will see a much more dramatic increase in the number of chargeable expense codes, going from the present 1 to 14, and you will also be able to credit any deposits to as many as 14 different deposit codes. This will be a big help to those people having

income from a number of different sources.

(3) The method of handling banks under HFS-III will change drastically. In fact it is completely different. Under the new system, there will be 100 asset accounts and 100 credit accounts. Asset accounts will be of four types: checking, regular, parent and reserve. The user will be able to allocate them in any fashion that he/she wants. Thus, you could have 0 bank accounts on up to 100 bank accounts.

A few words of explanation on the types of asset accounts may be in order here. A Checking account is pretty self-explanatory. A Regular asset account could be a savings account, cash account, Treasury Bill, Certificate of Deposit or any other type of asset.

The Parent account is a special account that has no transactions but serves to keep track of the balance of its children. For example, you might designate an IRA at a certain bank as a Parent account and assign individual CDs in that IRA to separate regular asset accounts. The Parent account will keep track of the total balance of all of the separate Regular (and Checking) asset accounts that you have assigned to it.

A Reserve account is an imaginary account that is best explained by an example. You need to make a purchase in 2 months. To assure that you keep enough money set aside, you can establish a reserve accounts *within* your checking account and transfer money into it from time to time. Your free checking account balance will decrease accordingly, although the total amount for purposes of balancing your account will be unchanged. (You can see your *free* or *total* balance by pressing a special function key.) When you want to make your purchase, you transfer the accumulated funds back into the checking account and write your check.

(4) Once you select the display of the current bank balance, it will remain on the screen as you move from menu to menu instead of disappearing everytime you change menus and having to be recalled as is now the case.

(5) You will be able to pay credit accounts from asset accounts other than checking. This will allow payroll deductions to be recorded quite easily. Often used transactions can be saved for easy recall.

(6) More functions will be included in each program so that moving from program to program will be less frequent.

(7) MSDOS directories will be supported. Programs and data essentially can be on

any valid disk rather than on specific disks on specific drives as is now the case.

(8) On the cosmetic side of the picture there will be a number of changes also. Menus and commands will become more uniform with no more seeing "1 for Display or 2 for Print" on one screen and "D for Display or P for Print" on others. The various modules are being written in assembly language for increased speed and efficiency. The speed and power of assembly language will also allow numerous enhancements in screen displays including context sensitive help windows. Pop-up menus will be available for selecting accounts, codes and individual transactions for editing if desired and codes and accounts may also be referenced by name rather than number. All in all, the program will include a number of enhancements to an already good program.

#### General

I'm sure that any number of readers can or will come up with more elegant solutions to the problems of property accountability and home financial management and I would enjoy hearing from any of you with your suggestions, comments, or critiques on this article or ideas for future articles. You can never tell, there may be enough fresh, new ideas out here on subjects in this area that we may be able to make this a continuing set of articles. I can be reached either at my home address which is listed at the start of this article or through Compuserve at (76347,2310).

#### Ordering Information

Home Finance System, Version 2

Jay H. Gold, M.D.

Jay Gold Software

Box 2024

Des Moines, IA 50310

(525) 279-9821

\$89.00 (\$21.00 for manual only) - MSDOS or CP/M-85/86 for Z-100; CP/M for H8/H89 (HDOS no longer available.)

Zencalc (Mycalc)

The Software Toolworks

14478 Glorietta Drive

Sherman Oaks, CA 91423

\$59.95 - HDOS, CP/M, MSDOS

Query III

Hoyle & Hoyle Software, Inc.

716 S. Elam Avenue

Greensboro, NC 27403

(919) 378-1050

\$99.95 - HDOS, CP/M, MSDOS, ZDOS, PCDOS

Condor III rDBMS (CD-5063-3)

Heath Company

Benton Harbor, MI 49022

(800) 253-0570

\$650.00 - MSDOS

Lotus 1-2-3

(Version 1.A for Z-100 LS 463-1)

(Version 2 for PC compatibles LS-5063-2)

Heath Company

Benton Harbor, MI 49022

(800) 253-0570

\$299 - Version 1.A

\$495 - Version 2



Continued from Page 9

can find time to do that. If anyone else out there has solved the cheap back-up problem for a complex system, I'd sure like to read about it!

The bottom line, for me, is a very good disk upgrade at a reasonable cost. I made several compromises in this project.

Since I didn't buy the disk from Heath they couldn't help me with the hardware installation, but they did help with my software questions. I only got a thirty-day warranty. I still don't have a really fast hard disk, but it is a noticeable improvement over the old one and I have a renewed appreciation for any hard disk compared to floppies. I can't use all of the space available on the new disk, but I got as much as I could without spending any more than the cost of the disk itself, and I may one day get MS-DOS 3.3 and get that extra 3.7 MB. I learned a lot about back-up requirements, and I will work on a good back-up/restore system I can depend on. I got a lot of help over the telephone from both the Denver Heath store and Priority One Electronics. The cost of the phone calls was well worth the information I got. But my primary source of information was the manuals I got with the system, and I can't imagine how to approach a job like this without them. \*



"Oh, please, don't let this happen to me!"

# Programmers

# Learning

## Real Fractious Numbers

Don Keller  
1330 Eden Valley Road  
Port Angeles, WA 98362

Years ago, when first introduced to fractions in school, I wasn't overjoyed. They didn't look like they'd be anywhere near as much fun as baseball. Their utility eventually grew to be obvious and my reflexes finally slowed enough to keep me off the baseball diamond anyway. But at the beginning, fractions looked peevish, irritable and unruly to me. In other words — fractious.

Fractions are real numbers, plainly not integers. They can be represented with a decimal point somewhere in a number or with the familiar bar or slash.  $1/4$  and .25 mean the same thing. It's a little less obvious that  $11/12$  and .916666666666..... are approximately equivalent to each other. And therein lies a rub. It's easy enough to give a computer a fraction like  $11/12$ . That's just eleven divided by twelve. But when the computer gives you a value like  $11/12$ , it always wants to say .916666666666..... So now real numbers have turned fractious again.

Sometimes fractions, with a bar or slash, are more meaningful in a display than decimal notation. And sometimes a string like  $5/4\_pi$  is a lot more meaningful than 3.926990817. The program listing shown here is for a C language function that converts decimal fractions to ASCII strings containing slash type fractions, including those that involve pi, the square root of two and the square root of three. If a match for one of those can't be found, the string will contain the ASCII representation of the original decimal value.

The function that does the job is called frax. It's preceded here by a main() function whose only purpose is to demonstrate frax without turning what ought to be a library function into an executable program. The entire package, when compiled (and assembled and probably linked) accepts a real number as a command line argument and writes out the same number in whatever condition it might be after frax has worked it over. If you took your calculator and discovered that (the square root of 2) over 2 is about .707107, you could type

```
FRAX .707107 <ENTER>
```

and the program would write out

```
1/2_sqrt2
```

that is, one half of the square root of two.

The compiler directive

```
#include <printf.h>
```

at the beginning of the source file may or may not be right for your system. Do this or whatever is required to make a floating point version of the printf function from the C library available to the program.

```
main(argc, argv)
```

```
int argc;
```

```
char *argv[];
```

```
{
```

begins the program; execution starts here. argc is the number of arguments typed by the user on the command line, and argv is an array of pointers to the character strings containing those arguments.

```
double x;
```

x is declared to be a variable of the type double to store a double precision real number.

```
double atof();
```

Atof is a function from the run time library that reads a character string, assumes it represents a floating point number and returns the actual floating point value the string of printable digits represents. Since a floating point value is returned, and a double precision value might be encountered, atof must be declared double in the program function — in this case, the main function — that expects to call it.

```
char fs[100];
```

When a program calls frax, it has to supply storage for the character string frax generates. That's what the char array, fs does. It has room for 99 characters, plus a terminating null character ('\0'). Pretty lavish, but you should never have to worry about over filling that array.

```
if (argc != 2) exit();
```

If the argument count is not 2 (the program name is argument 0, the actual argument you type on the command line is argument 1 and a terminating null argument is argument 2), then the program doesn't like your style and bails out back to the operating system. This executable program is intended to demonstrate frax as simply as possible and no further error



checking is attempted.

```
x = atof(argv[1]);
```

Atof converts the string of numeric digits you typed as a command line argument to the floating point value the string represents and assigns the value to x.

```
frax(x, fs);
```

The call to frax supplies the floating point value to be converted in the argument x and the address of the character string storage in the argument fs. When the frax function returns control to the main function, the fs array will contain the number you typed on the command line — in fraction form if frax was able to find one that was appropriate — or in decimal notation if it couldn't.

```
printf("\n%s\n", fs);
```

The string in the fs array is written to the standard output (usually the terminal screen, but it can be redirected) surrounded by newlines to make it a little easier to see.

Following the right curly brace that ends the main function, the rest of the source code is for the actual frax function. This is the code to save in a library of utilities.

```
#define MAXNUM 12
```

```
#define MAXDEN 12
```

These are the maximum values frax will attempt to use as numerator and denominator when it's looking for a match. You can change these values in the source code if you like. Increasing them will make for slower response.

```
#define TOLER .0001
```

You could change this one, too. Frax makes a conversion if it can find one whose value is in the range from (1.0 - TOLER) times the original value, to (1.0 + TOLER) times the original value. By the way, in a C program, a floating point constant has a decimal point and an integer constant does not. Sometimes you can get away with mixing them up and sometimes you can't. Habitually writing whole numbers with a trailing '.0' when you expect them to participate in floating point expressions tends to reduce excess digestive acids in programmers.

```
frax(x, s)
```

```
double x;
```

```
char *s;
```

```
{
    Some compilers may require you to declare the frax function to be of the type void since it returns no useful value. Like this:
```

```
void frax(x, s)
```

```
{
    The x argument is the double precision original number to be converted and s is a pointer to characters, the address of the string storage provided by the caller.
```

```
static struct {
```

```
    double cof;      char *label;
```

```
} const[] = {
```

```
    1.0,              "",
    3.141592654,      "pi",
    1.414213562,      "sqrt2",
    1.732050808,      "sqrt3"
```

```
};
```

This is a pretty dense chunk of code, but bear with me. An array of structures is declared and initialized. Each element of the array is a structure which contains a double precision floating point variable and a pointer to characters. The word, 'static,' is required; without that, the compiler will assume any data object inside a function is automatic and won't allow it to be initialized at the same time the declaration is made. A bit further along in the program we'll see how to access the data in a structure array.

```
int i, num, den;
```

We'll be using as many as three 'for' loops at a time in this function and i is the index for the outermost loop. Num is the variable that stores the numerators frax will test and den does the same for denominators. Num and den will also index 'for' loops.

```
double hi, lo, try;
```

These double precision real variables are explained in the code segments where they acquire values.

```
char sep[2];
```

This is a kind of ephemeral little character array declared to be large enough to hold one character and the terminating null byte ('\0') about as short as a visible string can be. When the function executes, sometimes that string is there, and sometimes it isn't. If a fraction is found associated with pi or the square root of two or the square root of three, sep[0] will contain the underscore character to separate the fraction from the rest of the string. Otherwise, sep is not only blank; it has no length at all.

```
if ((int)x == x) {
    sprintf(s, "%g", x);
    return;
}
```

If an integer converted version of x, (int)x, is equal to x, then the original number must be an integer and there's no point in trying to find an equivalent fraction. Frax generates the string representing the original number and returns. Notice that frax needs access to the printf function, in this case, the part of printf that's called sprintf. Sprintf writes formatted output to a memory location identified by the first argument. Here, the first argument is s so the string gets written into the character array supplied by frax's caller. The control string, "%g", in the second argument, means print a real number in the shortest possible format that will fully represent the number, without trailing zeros, for instance. The x in the third argument is the number to be written.

```
sep[0] = '\0';
```

The first thing that happens to that undersized array is its total disappearance. If its first character is a null byte, any function that tries to print it has to believe it's over before it starts. But it will be back ... maybe.

```
hi = 1.0 + TOLER;
```

```
hi *= x;
```

This establishes the upper size limit of the number frax believes it's trying to convert. (1.0 plus the tolerance, multiplied by the original exact number in x.)

```
lo = 1.0 - TOLER;
```

```
lo *= x;
```

And this fragment of code puts together the lower limit. (1.0 minus the tolerance, multiplied by x.) Frax will accept a match anywhere from the generated lo to the generated hi, inclusive. First thing, we'll see if the decimal fraction supplied as an argument to frax is one of our predefined constants.

```
for (i = 1; i <= 3; ++i) {
```

The loop started here is going to make some tests and conditionally copy a string from the structure array, const[i].

```
if (const[i].cof >= lo && const[i].cof <= hi) {
```

const is an array of structures. const[i] refers to the structure at element number i of the array. const[i].cof identifies the variable named cof in that structure. If the constant stored in const[i].cof is in the range from lo to hi that we just established, sure enough it's a match and ...

```
sprintf(s, "%s", const[i].label);
```

```
return;
```

copies its representative string from const[i].label to the string storage supplied by the caller in the argument s and control returns to the calling routine.

```
}
```

The first curly brace terminates the compound statement following the 'if' test and the second one ends the 'for' loop.

```
for (i = 0; i <= 3; ++i) {
```

Just like the last time we started a loop with an i index, the purpose is to test the x argument against a set of constants that are stored in the structure array. This time though, a constant value of 1.0 is included in the test; it was skipped over the last time around by initially setting the index of the 'for' loop to 1 instead of 0. Multiplying by a constant of 1.0 sounds like sprinkling water on a lake, but if the match test does that, it can distinguish between a fraction all by itself and a fraction times one of the other constants.

```
for (num = 1; num <= MAXNUM; ++num) {
```

This starts a loop that looks at some possible numerators, from 1 to whatever value was defined for MAXNUM.

```
for (den = 2; den <= MAXDEN; ++den) {
```

And the same thing for denominators and MAXDEN. Notice, it begins at 2. Anything with a denominator of 1 would never get this far; the integer test at the beginning of the frax function would already have written the representative string and returned.

```
try = (double)num / (double)den *
      const[i].cof;
```

Now here is the construction of the standard against which the test will be made: A trial numerator divided by a trial denominator multiplied by a trial con-

stant. Num and den need to be integers to index their respective loops, but the expression above must have real data types, hence the casts (double)num and (double)den. That doesn't change the type of num or den; it just provides the expression with real equivalents of the integers.

```
if (try >= lo && try <= hi) {
    If the standard we just constructed falls within the range of tolerance we constructed earlier, there is a match.
```

```
if (i) sep[0] = '-';
```

If the match occurred when i was non-zero, the fraction must be multiplied by one of the constants — other than 1.0 — in the structure array so it will need a separator between the fraction and the constant in the ASCII string. When a match is found with i at 0, the sep array keeps it zero length with a terminating null byte in its first cell.

```
sprintf(s, "%d/%d%s%s", num, den, sep,
        const[i].label);
```

```
return;
```

In the format string for sprintf, the first %d is the numerator, then the format string literally prints a slash followed by a second %d for the denominator. The first %s invokes the character array containing the separator character and the second %s calls for the name of the constant that was matched. The remaining arguments supply the values needed by the format string: num for the numerator, den for the denominator, sep for the separator char-

acter array, and const[i].label for the name of the constant. When the match was for a fraction by itself, i will still be set to 0 so sep[] will still have zero length and, if you take a look at the structure array, you'll see that the label in the 0 element of the array also has a zero length. In that case, no writing at all is done for those segments of the string under construction for the caller and there is no padding with white space. The constructed string is only as long as it needs to be. Once the string has been built, frax returns.

```
}
}
}
}

These four curly braces close the match test, the denominator loop, the numerator loop and the outer i loop which indexed access to the structure array.
```

Not every number will find a match and when one doesn't, this line writes the original number to the supplied string and ...

```
}
returns to the caller by running off the end of the function.
```

### Using Frax in a Program

Keep the demonstration program (the main() function in the source file that accompanies the article) for a guide. Remember the caller has to provide a character array to store the string generated by frax. The caller obviously has to supply

the floating point number to be converted. Those are the only arguments to frax.

```
frax(x, s)
double x;
char *s;
```

That's about it. Oh, don't forget ... after conversion, you have a string representation of the fraction to print, not a floating point value that still needs formatting.

```
/*      frax.c
*/

#include <printf.h>

main(argc, argv)
int argc;
char *argv[];
{
    double x;
    double atof();
    char fs[100];

    if (argc != 2) exit();
    x = atof(argv[1]);
    frax(x, fs);
    printf("\n%s\n", fs);
}
```

```
#define MAXNUM 12
#define MAXDEN 12
#define TOLER .0001

frax(x, s)
double x;
char *s;
{
    static struct {
        double cof;
        char *label;
    } const[] = {
        { 1.0, "" },
        { 3.141592654, "pi" },
        { 1.414213562, "sqrt2" },
        { 1.732050808, "sqrt3" },
    };

    int i, num, den;
    double hi, lo, try;
    char sep[2];

    if ((int)x == x) {
        sprintf(s, "%g", x);
        return;
    }

    sep[0] = '\0';
    hi = 1.0 + TOLER;
    lo = 1.0 - TOLER;
    for (i = 1; i <= 3; ++i) {
        if (const[i].cof >= lo && const[i].cof <= hi) {
            sprintf(s, "%s", const[i].label);
            return;
        }
    }

    for (i = 0; i <= 3; ++i) {
        for (num = 1; num <= MAXNUM; ++num) {
            for (den = 2; den <= MAXDEN; ++den) {
                try = (double)num / (double)den * const[i].cof;
                if (try >= lo && try <= hi) {
                    if (i) sep[0] = '-';
                    sprintf(s, "%d/%d%s%s", num, den, sep, const[i].label);
                    return;
                }
            }
        }
    }

    sprintf(s, "%g", x);
}
```

\*

## Most home finance programs have serious limitations.

Like the dodo bird, they're slow, awkward and not very smart. You'll waste eons of time, only to realize they're dead ends.

HFS-III, the Home Finance System, is fast, loaded with features and intuitively smart.

- Easily manage up to 100 checking, saving, CD, IRA and other accounts.
- Print checks on any form.
- Organize financial summaries and activity reports based on 100 expense and 15 deposit codes you define.
- Flag tax-deductible or medical expenses.

In short, control your finances to save money, save time and simplify taxes.

Find out why so many leading computer authorities and users are calling HFS-III the "superior species." See it today at participating Heath/Zenith Computers and Electronics Centers, or call toll free for more info.

**Only \$49<sup>95</sup>**

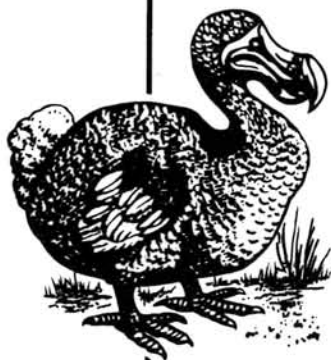
Add \$3.50 handling & shipping.  
Iowa residents add 4% sales tax.

**Requires:** DOS 2.0 or higher, IBM\*  
PC/ XT/AT or compatible or Zenith\*  
Z100 computer; 2 disk drives or hard  
disk, 256K RAM.

Reader Service #137



Jay Gold Software, Inc.  
P.O. Box 2024  
Des Moines, IA 50310  
(800) 541-0173



## Back to the Books

Let's face it, sooner or later you're gonna have to try and read those computer USER manuals! But, before you do, read "POWERING UP". This book was written especially for you in a non-technical, easy-to-understand style. Who knows, with "POWERING UP", you may NEVER have to read your user's manuals again! Order HUG P/N 885-4604 today!

## Attention Zenith Z-248™ Users!

**Convert your 286 to a 386 with our one Board Solution!**

**The Aox Z-MASTER 386**

**CALL US FOR OUR GSA SCHEDULE NUMBER!**

**800-822-0386**

**SAI**

Strategic Alliance, Inc.  
477 Washington Street (617) 762-7485  
Norwood, MA 02062

Since 1979 Aox has been a leading designer of accelerator, coprocessor and compatibility cards providing superior technological solutions.

Reader Service #195

## WANTED:

### Quality Assurance Engineer

The Zenith Computer Group is continuing to experience unprecedented growth. We currently are seeking a Quality Assurance Engineer with a BSCE degree and strong background with assembly code or "C" on Zenith machines. A BSCS with strong hardware background will be considered.

For immediate consideration, send your resume to: Ms. Pat Kauffman, Zenith Computer Group, Hilltop Road, St. Joseph, MI 49085. The Zenith Computer Group is a group of wholly-owned subsidiaries of Zenith Electronics Corporation. An Affirmative Action Employer.



---

# ENABLE

---

## Part 17

---

# Advanced Data Base

---

**George D. Elwood**  
**1670 N. Laddie Court**  
**Beavercreek, OH 45432**

This is part 17 in a series on ENABLE, an integrated software package from ENABLE SOFTWARE. The company changed its name in March from The Software Group. This article will cover advanced data base management. I will go over an application I developed last year for my office to show what is possible with ENABLE.

I would like to thank those of you who have written me about the series. If you need copies of back articles, please contact HUG and they will be able to help you. I have also received several questions on where to buy the Z-100 version of ENABLE. W. S. Electronic in Xenia, OH 45324 is one source. The complete address is at the end of the article.

I have been thinking about putting together all of these articles in a book. It would be spiral bound and on 8.5 X 11 inch paper. If interested, please let me know. The format would be all of the word processing articles followed by the spreadsheet and then data base. It would include better graphics of the screen and possibly a keyboard overlay for the PC and Z-100 function keys.

ENABLE's data base is easy to use for most applications. The built-in error checking capability makes it easier to use than dBase III. The data base definitions permit you to define acceptable entries or show relationships between State codes, Area codes, and ZIP codes. This is accomplished without having to write a separate program. Our company is developing an application to track dollars in a budget system using ENABLE. It will use the LAN version so sharing information will be possible. Like this application, it will be completely menu driven.

The ENABLE data base management system is very easy to use and using the

menus makes it friendlier. The application that will be covered in this article is a payroll system. I developed it while learning the capabilities of data base. I would change it now that I have used ENABLE for several other data base applications. This application took about three days to write with some troubleshooting later due to changes in company policy. Before I developed this program, the office manager would spend two days manually completing the payroll. Employee checks were done one at a time. The checks were typed individually and they did not contain year-to-date information. The ENABLE data base application permits the office manager to complete the payroll in about two hours, including printing the checks and the required company records.

As part of the courses we teach, our instructors stress spending time with pencil and paper. This is an important step before you start to key in the data base. It is easier to change something before entering data than after adding a thousand records. Before I started to write this application, I talked with the office manager to see what was needed. I then went back to my office and wrote down what was said. I took this back to the manager to ensure that everything was correct. The day I spent doing this permitted me to develop the application in the three days with the program meeting the stated objectives.

ENABLE is a relational data base. It is an easy-to-use information spread between several data bases without having to go into great amounts of coding. The fields in these data bases are called by simply indicating the data base and field. The experience I have had with ENABLE permitted me to rapidly develop a capa-

bility with ORACLE, a major SQL data base. I have written a manual and instructed ORACLE on a mainframe computer as one of my tasks.

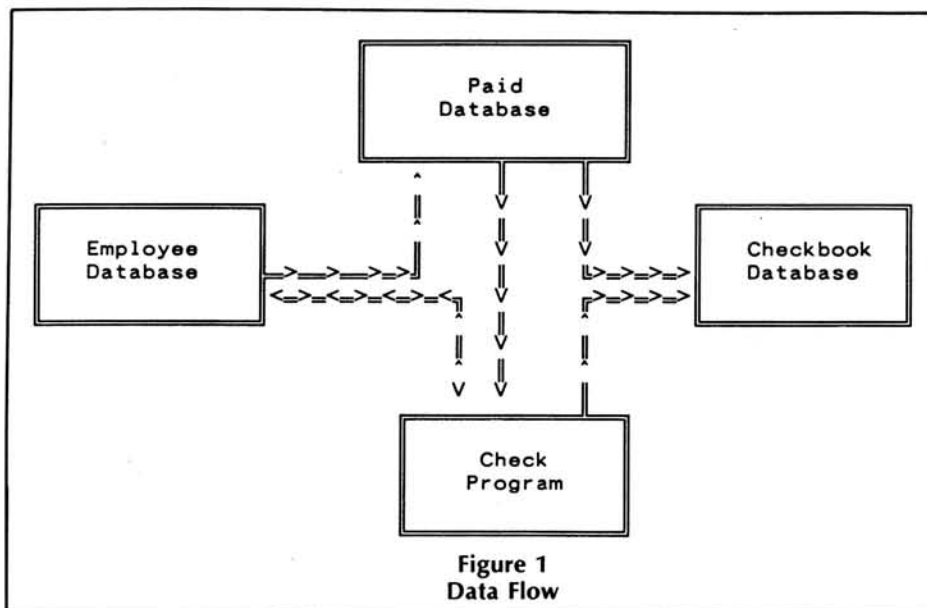
With the application firmly in mind, I set out to develop the data base definitions and then the coding to make it work. I decided that three data bases were required, one containing the employee data, the data base which contains the bi-weekly hours statement, and a checkbook ledger where all checks were recorded.

The EMPLOYEE data base contains personnel employee data. Information required to calculate the Federal and State income taxes are inserted here. This information includes married status and number of dependents. I wrote the program so these numbers could be different for both the state and federal calculations. Because some employees were having extra money withheld from either the state and/or federal income taxes, this also had to be accommodated. The year-to-date information was not printed on the checks before the program was developed. This was added to the check without problem as an aid to the employees.

The main data base is PAID. This is where the office manager enters the bi-weekly payroll information. Each employee fills out a time sheet based on the contract he/she is working on. The contract number and number of hours worked is entered. The office manager, using this sheet, keyed in the required information.

As part of the audit tracking of the systems, all checks are entered into a third data base called CHECKBK. The checks were not written from this data base, but recorded here. This data base is the source of company required reports.

The basic working of the program re-



quires the three data bases and one main program. During the data input into the PAID data base, some data comes from the EMPLOYEE data base. The program uses all three data bases when running.

Figure one shows the data flow in the program. For the most part, data moves only between one module. The exception is the actual Check program. This program receives data from both the Employee and Paid data bases. It sends data to the Employee and Checkbook data bases, along with outputting the checks.

As part of the paper layout of the program, a detailed list of output components is necessary. You must be able to get these components from the system so they must be included in one of the data bases or be calculated. When developing a data base, it is easier to work backwards to get the required input information. I knew what was required and what was nice to have on the check. I also knew what was required on the reports to the corporate office.

Once I had the output requirements in hand, I started building the data bases. The first one, PAID, consisted of the information that the office manager took off the time sheets. The main data elements are employee number, contract number, and hours. Information on leave, both normal and sick, holiday, and overtime for those employees who received it, are also used.

Once I had the data bases completed, I built the input screens for the EMPLOYEE and PAID data base. With ENABLE this takes only a few minutes. Because I was designing this system for somebody with little computer experience, other than word processing, I planned to make it run with menus and macros. This would hide the underlying ENABLE program. All selections were to be made off a menu, and quitting an input

screen would save the data and return the user to a menu. The input screens were designed to provide some error checking for the operator.

Once the data bases were developed, the major program was started. The Check program was to take the PAID and EMPLOYEE data and output to the CHECKBK data base and output checks, either to a word processing file or directly to a printer.

The ENABLE data base uses a word processing file for the program. This permits you to develop the program and make changes rapidly. During the development, I would have the program in one window and the data base in a second window. This way, I could rapidly move between the program and the application, making changes as necessary. I also used the document title to make a record of changes to the program. This way, I would know what changes were made and when.

The program states as all ENABLE data base applications with the statement .REPORT DIVISION

This is always the first line of a program and tells ENABLE's data base it is OK to use. All lines in the program start with a dot "." like dBase programs.

The second line defines the next section of the programs as the definitions section. It is also defined with a dot command.

.DEFINITION

This section defines the local fields for the program. These are fields that do not exist in your data base. You can also use this section to change or create a picture of the data that will be used in the report. You must also define the

.LET  
.INPUT

statements at this time. You must also establish the links between the various data

bases that will be used in the report. I had a lot of definition statements in this section. If I was to do it over again, some could be removed without causing problems.

Note: I used double semicolons to add comments. ENABLE will not process a line that starts in that manner. Also note that I ask for the last check number. Our printer will not print on the first check so this check number is input and one is added to it for all printed checks.

The next section is the

.INTRO

section. This section will setup certain values. Anything in the INTRO section is run only one time, at the start of the program. I feel as though this line could be left off the program without harm. You would use this section for the header or things that are printed at the start of a report.

The next section is where the work is done, the

.BODY

Here is where all of the calculations are completed. Each entry in the PAID data base is processed through this list of calculations. The order that they are inserted in the program makes a difference. This is where the pencil and paper work pays off. Deductions must be removed before certain taxes are computed and this all needs to be laid out first.

The first thing I do is to

.OPEN CHECKBK

This is the data base where all calculations for the check will be stored. This is an audit trail I found necessary to install. Next, I totalled all of the hours for the employee in the pay period. Because time can be charged to several different projects, all of these have to be added together. You must also include vacation, personnel leave, and holiday time in the calculations.

```

let hours = (hc1)+(hc2)+(hc3)+(hc4)+
(hc5)+(hc6)+(vused)+(plused)+(holiday)
+(other)
  
```

Even though the above line has been broken for this article, you must not break them in the program. One of the keys to a successful data base is to make the name of fields mean something. Don't make the field names cryptic.

Next, I calculated the pay for the employee.

```

let npay = (hours) * (prate)
let opay = (ot) * (prate)*1.5
let gpay = (npay)+(opay)+(other)
  
```

The next section calculated the State of Ohio tax. State tax like the Federal tax accounts for dependents and married status. These calculations can be obtained from the State tax agency. I wrote the program so that a person could have a different number of depends for the state tax (SDEPEND) and federal calculations. I also included the capability to have additional

I use the hours, a local field created above, times the pay rate for the employ-

### Figure 2 Credit Union Report

```
.if (tcity) = "D" AND (wcity) = "Dayton"
.let city = (gpay) * .0225
.let city_d = city
.elseif (tcity) = "D" AND (wcity) = " "
.let city = (gpay) * .0225
.let city_d = city
```

Then next section calculated the State of Ohio tax. State tax like the Federal tax accounts for dependents and married status. These calculation can be obtained from the State tax agency. I wrote the program so that a person could have a different number of depends for the state tax (SDEPEND) and federal calculations. I also included the capability to have additional

Once all of the taxes have been calculated, everything is put back in to get the total pay. Next, the program checks to

```

[ name ] [ en ] [ ssn ] [ pdate ] [ cn ]
Normal Time [ hours(4) ] [ npay ] [ ytdnpay ] FICA [ fica ] [ ytdfica ]
Overtime [ ot (4) ] [ opay ] [ ytdotpay ] Federal [ fed ] [ ytdfed ]
Total Hours [ totalhrs(4) ] State [ state ] [ ytdstate ]

City [ city ] [ ytdcity ]
Fairborn [ fbn ] [ ytdfbn ]
Other [ otherd ] [ ydotherd ]
401k [ e401k ] [ y401 ]
WPCU [ ctowpcu ] [ ytdwpcu ]
Flex Deduction [ flexded ] [ yflexded ]

Other Pay [ other ] [ ydotherp ]
Gross Pay [ gpay ] [ ytdgpay ]
Net Pay [ ckamt ] [ ydnetpay ]

Personal [ plbb ][plused(6)][pladd ] [plend ]
Vacation [ vlbb ][vlused(6)][vladd ] [vlend ] [dtotal ]

[ ckdate ] [ ckamt ]

*****[dollars] Dollars and [cents(2)] Cents *****

[ cname ]

```

**Figure 3**  
**Check Form**



see if the check is to go to a credit union. If this is the case, the check is written for a zero amount and the money is placed into the credit union field in the CHECKBK data base. Because some employees put only part of their check into the credit union, this also had to be accommodated.

```
.let gpay = gpay + e401k+flexded
.let stotal =
(fica)+(fed)+(state)+(city)+(fbn)+(otherd)+(e401k)+(flexded)
.if awpcu = Yes
.let ctowpcu = fwpcu
.endif
.if pwpcu = Yes
.let ctowpcu = (gpay) - (stotal)
.endif
.let dtotal = (stotal) + (ctowpcu)
```

I can not stress enough how important it is for anybody designing a data base to spend some time with pencil and paper. It is easier to think through these things than to try and go back and fix problems later.

The next part of the program starts to create parts of the check. The first and last names of the employee are put together as one name using the concatenate feature of the ENABLE to remove the spaces between the names.

```
.let ckamt = (gpay)-(dtotal)
.let name = (lname)&-(fname)
.let cname = (fname)&-(lname)
```

The office manager wanted the checks to reflect both dollars and cents in words on the check. A check normally has the amount printed out in words and also the numbers. Trying to work through the cents was no problem until I came to the even cents. Even though I wanted two positions in the number, the program continued to leave off the zero as it was not necessary. I had to add a short section to take care of this problem by converting the number to a string.

```
.let dollars = @int(ckamt)
.let ck = @string(@round(ckamt,2))
.let c = @len(ck)
.let cents = @substr(ck,(c-1),2)
.if cents = ".1"
.let cents = "10"
.elseif cents = ".2"
.let cents = "20"

.elseif cents = ".9"
.let cents = "90"
.endif
```

Because the office could have part time employees, they needed to be addressed. Part time employees do not receive certain benefits in the area of vacation time earned. The next part of the program took care of these people. The regular employees were also credited with the correct leave at this point.

```
.let plbb = yperl v
.let vlbb = yvacation
.if partime= @yes
.let pladd = (plmulti) * hours
.let vladd = (vlmulti) * hours
.let pladd = @round(pladd,2)
.let vladd = @round(vladd,2)
.let plend = (plbb + pladd)- (plused)
```

```
.let vlend = (vlbb + vladd)- (vlused)
.else
.let vladd = vac_pp
.if exempt = @yes
.let pladd = 3.1
.else
.let pladd = 2.15
.endif
.let plend = (plbb + pladd)- (plused)
.let vlend = (vlbb + vladd)- (vlused)
```

```
.endif
.let holpay = holiday * prate
.let perpay = plused * prate
.let vacpay = vlused * prate
.let othertp = other * prate
```

The final part of the program before actual outputting the check is to get the year-to-date information from the EMPLOYEE data base. This information is updated with the current pay period, printed on the check and sent back to the EMPLOYEE data base.

```
:: ytd pay
.let ytdnpay = (yregpay) + npay
.let ytdotpay = (yotpay) + otpay
.let ytdfica = (yfica) + fica
.let yflexded = (yflexded)+ flexded
.let cn= cn+1
.let regpay = npay
.let otpay = ot
.let otherpay = other
.let tdeduct = dtotal
.let ckno = cn
```

```
.let towpcu = ctowpcu
.let chkdate = ckdate
.let flexdedp = flexded
.let cen = en
```

Now comes the check. All of the information you need is now brought into the check. Since each check is different, you will need to change these settings. Before you start the checks you should turn off the reformat. This will permit ENABLE to line up the information exactly where needed. All fields are put in the location they are to be printed in by using the brackets. You place the field name in the location enclosed in brackets. The output form will use as much space as you enclose in the brackets. Note that I use the braces to specify the length of certain fields like TOTALHRS. These field names are long and ENABLE will try to use that amount. Using the braces to control the spaces for field will require that you work to establish the exact location of the next set of fields on the line.

```
.reformat off
```

After completing the check for the employee, the program then updates the EMPLOYEE data base. The statements used here are the same as used in ORACLE to use information from another data base. The statement shows the data base, a period, and then the field you wish to update.

```
.definitions
.define cn as integer 6
.define pp as text 8
.input "What is the last check number? " cn
.define tw as decimal 8.2
.define ow as decimal 8.2
.define fpay as decimal 7.2
.define stotal as decimal 7.2
.define dtotal as decimal 7.2 using "($nn,nnn.nn)"
.define hours as decimal 3.1
.define npay as decimal 7.2 using "($nn,nnn.nn)"
.define cfname = fname in employee link en
.define clname = lname in employee link en
.define depend = depend in employee link en
.define wcity = wcity in employee link en
.define fwpcu = fwpcu in employee link en
.define afedtx = addfedtx in employee link en
:: ytd definitions
.define ytdnpay as decimal 7.2 using "$nn,nnn.nn"
.define ytdotpay as decimal 7.2 using "$nn,nnn.nn"
.define ydnetpay as decimal 7.2 using "$nn,nnn.nn"
.define ytdgpay as decimal 7.2 using "$nn,nnn.nn"
.define y401 = y401 in employee link en
.define p401 = p401 in employee link en
.define partime = partime in employee link en
.define exempt = exempt in employee link en
.define name as text 25
.define cname as text 25
.define other using "$nn,nnn.nn"
.define dollars as integer 7 using "$nn,nnn."
.define cents as text 2
.define cent as integer 2
.define other using "$nn,nnn.nn"
.define c as integer 1
.define ck as text 7
::leave definitions
```

Figure 4  
Definition Section

```
.let employee.yperl = plend
.let employee.yvacation = vlend
.let employee.yregpay = ytdnpy
.let employee.yotpay = ytdotpay
.let employee.ynetpay = ydnetpay
.let employee.y401 = y401
.let employee.yflexded = yflexded
```

The next line directs ENABLE to send the information in the program to the CHECKBK data base. ENABLE will use all of the current local fields in the program to insert a new line in the other data base.

```
.add check
```

The next line updates the EMPLOYEE data base with the information shown above.

```
.update employee
```

The last section of the program resets all values to zero for the next employee.

```
.let ctowpcu = 0
.let city = 0
.let city_d = 0
.let city_t = 0
.let otherd = 0
```

The program will now recycle through the PAID data base for the next employee. All lines in the program are not included with this article due to its length. The lines included here are representative of the entire application.

The CHECK program is the key program in this application. It is one part of the entire system. Because the person using the system is not computer literate when it comes to a data base, I built a

menu system. The menu system permits the person to press a key from the main directory to change to the ENABLE directory and load the program with the subdirectory set for the payroll program. A menu is the first thing that is displayed. From here, selections can be made to edit or update records, do the reports, or other maintenance of the system. All selections are made from a menu and run macros. There are three different menus and 23 macros.

One of the other reports that is generated is the report to the credit union. This report uses information from the CHECKBK data base.

```
.report division
.definitions
.define wtotal as decimal 8.2 using "nn,nnn.nn"
.define name as text 25
.map section
.intro
```

This simple report shows how the INTRO section is used. The INTRO section contains the header for the report. After the header, the body contains the actual work. The dollar amount comes from the CHECKBK data base, but the EMPLOYEE data base contains the information as to who has money going to the credit union. The ENABLE data base capability to sum numbers is used to provide the total amount

```
[towpcu{s} ]
```

This is one of the capabilities that was discussed in an earlier article on the data base.

I hope that this has given you a brief overview of the capabilities of the ENABLE data base and its procedure language. This is a lot of capability in ENABLE and it is very easy to use.

The next article will cover the advanced graphics capability of ENABLE. I will cover the PERSPECTIVE package which is only for PC with EGA or better graphics.

Z-100 ENABLE  
W.S. Electronics  
1106 State, Route 380

Xenia, OH 45385  
(513) 376-4348



Promise her the sun and moon, but keep the "Word-STAR Connection" for yourself! Now you can run any generic copy of MicroPro's WordStar, Version 4 or 5, on your unmodified Z-100. Pat Swayne's "WordStar Connection" is P/N 885-3047-37, and sells for only \$20! An upgrade from the older version is also available.

## REMOVABLE HARD DISK DATA STORAGE



### FOR YOUR ZENITH Z-150, Z-248, Z-286, Z-386 MICROCOMPUTER SYSTEMS

- RANDOM ACCESS
- 10MB OR 20MB HARD DISK CARTRIDGES
- MAXIMUM DATA SECURITY
- IMAGE BACK-UP
- 100% MEDIA INTERCHANGEABILITY
- SCSI OR ST506 INTERFACE
- NOVELL COMPATIBLE
- EXTENDED 24 MONTH WARRANTY

For product information see your local Zenith dealer or call 1-805-964-0733.

In California call 1-800-654-8590.

FAX inquiries: 1-805-964-0734.

VISA & MASTERCARD accepted.

Also available on GSA contract GSOOK-88-AGS-5087.

**DMA**  
TECHNOLOGIES

P.O. Box 1236  
601 Pine Avenue  
Goleta, California 93116

Reader Service #197



**Note:** The following information was gathered from vendors' material. The products have not been tested nor are they endorsed by HUG. We are not responsible for errors in descriptions or prices.

Please be informed that **Quikdata, Inc.**, the sole remaining source of a complete line of products for the Heath/Zenith 8-bit machines, and the oldest independent H/Z vendor in the business, has lowered the price on the double-density controller boards for the H/Z-89/90 and H8 computers. The H-89 H-37 double-density disk controller board has been reduced to \$175 without ROMs and power supply upgrade (not needed except in early systems). The H8 WH8-37 board (which is both an H-37 double-density controller and a hard disk host adapter card) has been reduced to \$275. Both are completely assembled and tested and include board, HDOS diagnostics software and manuals. We also have some bare boards available.

If you have ever been thinking of adding a soft-sector controller board to your 8-bit machine, now is the time to act! We at Quikdata have decided that when our current, very limited quantity of these stock boards are sold out, they will be gone for good. Because of the slow

down in sales of these boards, we do not plan to make another production run. There is still plenty of life in these systems, and if you have been considering upgrading to a soft-sector controller board, now is the time, while there are still some left.

For further information, or to place an order (MC and VISA accepted) call or write:

Quikdata, Inc.  
P.O. Box 1242  
Sheboygan, WI 53082-1242  
(414) 452-4172

Beginning in April of this year, **Paul F. Herman Inc.** will begin publishing "Z-100 LifeLine", the only professional journal devoted exclusively to the Heath/Zenith Z-100 computer. Each issue will contain at least 16 pages of useful, practical information. Z-100 happenings, software and hardware reviews, how-to articles, tutorials for the novice, and in-depth programming tips for more advanced users, are some of the subjects that will be covered. A good deal of space in each issue will be devoted to program listings (Z-100 or generic MS-DOS programs).

"Z-100 LifeLine" is a one-stop source of valuable Z-100 information you won't find elsewhere. No need to cull out PC oriented articles — there aren't any. Everything in this publication is applicable to the Z-100. Paul Herman (author of Doodler, MousePack, ScreenPro, and other popular Z-100 software) will serve as editor, and chief contributor to "Z-100 LifeLine". In addition to Paul, a team of expert contributing authors is being put together to support the Z-100 through this publication. "Z-100 LifeLine" will also play an activist role to help maintain support for the Z-100 computer.

"Z-100 LifeLine" will be published six times per year. Subscription prices are as follows:

U.S. (U.S. zip code)	\$24.00
Canada & Mexico (Air Mail)	\$27.00
All others (Surface Mail)	\$28.00
All others (or Air Mail)	\$36.00

Florida residents must include \$1.44 sales tax. Make your check payable to "Paul F. Herman Inc." Send your subscription order to:

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



## Z-100 Graphics Software

DOODLER-V Graphics Package	\$99.00
with D-MOUSE Mouse Driver	\$109.00
with Logitech C7 Mouse & Driver	\$189.00
Font Library Disk for DOODLER-V	\$29.95
44 Additional Ready-to-use fonts	
Texture & Symbol Library Disk	\$29.95
Hundreds of pictures, symbols, textures	
ScreenPro Utility	\$59.00
The ultimate screen print/capture utility	

**FREE** Catalog available on request

From the Leader in Z-100 Graphics...



3620 Amazon Drive  
New Port Richey, FL 34655



Order toll-free...

**800-346-2152**

Or in Florida... 813-376-9347  
Unconditional money-back guarantee  
if your not satisfied!



# BUG

# ZAPPING

**BUG:** I have used the MS-DOS PART command to set up two partitions on my hard drive. Drive C: works fine. When I use the MS-DOS ASGNPART command to assign partition 2 to drive D:, I get the message "Invalid drive/drive not available". How do I access the second partition?

**ZAPPED:** Later versions of MS-DOS version 2 and all versions of MS-DOS version 3 have an Automatic Partition Assignment Flag. The Automatic Partition Assignment Flag gives the user two options:

1. Automatic Partition Assignment — MS-DOS recognizes 1 partition per hard drive.
2. Manual Partition Assignment — MS-DOS recognizes up to 4 partitions per hard drive.

The ASGNPART command (ASSIGN in MS-DOS version 2) will allow users to access more than one partition only if the partition assignment flag is set to manual.

## CHANGING THE AUTOMATIC PARTITION ASSIGNMENT FLAG

The procedure for changing the Automatic Partition Assignment Flag varies with the different versions of MS-DOS.

With MS-DOS version 2 and MS-DOS version 3.1, use the MS-DOS CONFIGUR command. With MS-DOS version 3.2, use the MS-DOS DSKSETUP command.

### MS-DOS Version 2

MS-DOS version 2 is distributed with the Automatic Partition Assignment Flag set to manual. Since manual partition assignment is the desired option, no changes are necessary. The Automatic Partition Assignment Flag can be changed (or looked at) by using the MS-DOS CONFIGUR command.

### MS-DOS Version 3.1

MS-DOS version 3.1 is distributed with the Automatic Partition Assignment Flag set to automatic (1 partition per hard drive is recognized). If more than 1 partition is being used, the Automatic Partition Assignment Flag needs to be changed to manual (up to 4 partitions per hard drive are recognized). The MS-DOS CONFIGUR command allows a user to change the Automatic Partition Assignment Flag.

### MS-DOS Version 3.2

MS-DOS version 3.2 is distributed with the Automatic Partition Assignment Flag set to automatic (1 partition per hard drive is recognized). If more than 1 partition is being used, the Automatic Partition Assignment Flag needs to be changed to manual (up to 4 partitions per hard drive are recognized). The MS-DOS DSKSETUP command allows a user to change the Automatic Partition Assignment Flag.

**Note:** The setting of the partition assignment flag is read by MS-DOS at boot time. After changing the flag in CONFIGUR (MS-DOS versions 2 and 3.1) or DSKSETUP (MS-DOS version 3.2), you will need to reboot in order for the change to occur. See your

MS-DOS manual for details on the CONFIGUR command (MS-DOS versions 2 and 3.1) and the DSKSETUP command (MS-DOS version 3.2).

**BUG:** My ZCM-1490 Flat Screen Monitor display flutters; it looks like a "fast ripple" running up the screen. Is there a known fix?

**ZAPPED:** On the power supply module (P/N 234-955), an open 47 ohm, 3-watt resistor at RX1 in the degaussing circuit causes the raster to be modulated by the AC field from the degaussing coil. Replace the power supply module to correct the problem.

**BUG:** How do I install an EGA video card in my Z-150 or Z-160 computer?

**ZAPPED:** Three steps are required before installing an EGA video card in a Z-150 or Z-160 computer:

1. Purchase a ZCA-6 video disable kit and follow the instructions to disable the existing video board in your computer.
2. Your computer's system ROM version must be 2.9 or higher. The most current ROM may be obtained from our Parts Replacement department at (616) 982-3571. When calling Parts Replacement ask for part numbers 444-229 and 444-260.
3. Positions 4 and 5 on the CPU switch SW1 must be set to 'ON'.

The EGA video board may now be installed. If you follow the above steps and the EGA video board will not operate properly, please contact our Technical Support line at (616) 982-3309.

**BUG:** How do I use the orange (Z-181-92/Z-181-93) or blue (Z-183-92/93) "+" and "-" keys on the Z-180 computer?

**ZAPPED:** In order to use the orange/blue "+" and "-" keys, press the FN key and while holding it down, press the Num-Lock key once, release the FN key and press Num-Lock once more. Next, press the desired "+" or "-" key and then press the Num-Lock key once to return to the Numeric Keypad Mode. Or press the FN key and then the SPACEBAR, and while holding the FN key choose the desired option, "+" or "-". To return to the Normal Alphanumeric Mode, press the FN key and while holding it down, press the NUM-LOCK key once.

**BUG:** Z-525 Cache Memory Card experiences intermittent failures.

**ZAPPED:** The main memory control PAL at U728 now has the state machine bit "STI" modified to correct an error which may occur when "HOLDA" is asserted early in a refresh or DMA cycle. On the #181-7320 cache board, the PAL is changed as follows: U728 from a #444-572 PAL to a #444-572-1 PAL. The #181-7320-1F and later modules have the new PAL installed.

**BUG:** The top 1" of raster, in the ZVM-1380, bends to the right in mode 1; mode 2 is okay.

**ZAPPED:** To correct the bending at the top of the raster when in Mode 1, replace the power supply (P/N 969-536).

**BUG:** The ZVM-1380 experiences excessive width in Mode 1; Mode 2 is okay.

**ZAPPED:** On the main board, check for a shorted 2SB856 transistor (P/N 969-522) at Z405.

**Bug:** When writing to either floppy disk drive on the H/Z-100, an occasional glitch would cause the file to become corrupt and unreadable. Formatting a disk with the same drives, would return a "bad sectors" message. This problem is intermittent and, in some cases, the drives operated flawlessly.

**Zapped:** The problem will follow the disk controller card, indicating that the card is at fault, and most likely an intermittent connection, since the actual problem is intermittent. Keeping the tip of a 25-watt soldering iron clean, resolder (re-heat) all the connections on the back of the floppy controller card. Also, carefully inspect every IC for proper insertion (no bent under pins).

(submitted by E. Howell)

**Bug:** Can I install 3-1/2" drives in my Z-150 series computer?

**Zapped:** Yes! Order the ZD-372 kit. This kit includes the following:

- 3-1/2" floppy disk drive
- Adapter bracket
- 4-pin power connector cable
- 34-conductor drive-to-controller cable assembly
- Packet of hardware items

The 3-1/2" drives included in this kit are qualified for use on the Z-150 series computers. These drives run at 6 milliseconds step rate used by the Z-150 systems. Step rate and cabling requirements prohibit the use of this kit on any other Zenith series computers.

**Bug:** I have a Z-183-92 with an internal 1200 baud modem, and DOS 3.2 (BIOS 3.31). When I follow the instructions for "Test Dialing a Number from DOS" in the modem manual, I get an error writing to COM2. How can I test dial the modem from DOS?

**Zapped:** When DOS 3.2 boots, the only serial port recognized by CONFIGUR is COM1. To turn on the modem, as well as make CONFIGUR recognize the second serial port, you must enter the "MODE MODEM ON" command. If you receive an "Invalid Parameter" error message after entering this command, download the MODE.COM file from section W of the Zenith Data Systems Information Data Base Bulletin Board (616) 982-3503, and use the new version of MODE.COM to enter the "MODE MODEM ON" command. Next, run CONFIGUR, and configure the COM2 port for a user-defined serial device, with 1200 baud, 8-bit word, 1 stop bit, no parity, and no handshaking.

The computer is now set up to recognize the internal modem as COM2. Follow the instructions in the modem manual for "Test Dialing a Number from DOS", and the number will be dialed. Communications software that can access COM2 is required to actually make a connection through the modem.

**Bug:** I had heard that if the Z-180 computer internal Nicad batteries were not used properly that the battery lifetime could be reduced to 10 minutes. Is this true?

**Zapped:** Yes. If the Nicad batteries are not fully charged and discharged, they may be damaged. To ensure that this does not happen, charge a discharged Nicad battery pack for 8 to 12 hours. Do not overcharge the battery! Eight hours will fully charge the battery

pack, but charging the battery pack for more than 12 hours will shorten the life of the batteries.

When the battery is fully charged, disconnect the external power source and allow the computer to run from the internal batteries until they are fully discharged. Then, when the low power light is displayed, plug in the external power source and fully charge the unit again.

**Bug:** What type of mouse device can be used on the Z-180 series computers?

**Zapped:** Z-181-92/93 and Z-183-92/93 computers supply five volts of power to the RS-232 serial port. Therefore, if you plan to use a serial mouse, select one that uses five volts or less, such as the Microsoft Serial Mouse.

The Z-184/SupersPort computer supplies 12 volts of power to the RS-232 serial port. Therefore, any serial mouse needing 12 volts or less may be used.

**Bug:** Your MS-DOS ver 3.2 software is unable to read some floppy disks that are readable under both MS-DOS ver 3.1, and MS-DOS ver 2. Does this represent an MS-DOS ver 3.2 BUG?

**Zapped:** MS-DOS ver 3.2 uses a method for determining media type (type of disk to be read) that differs from earlier MS-DOS versions. MS-DOS version 3.2 will read any disk formatted under compatible (with PC-DOS) MS-DOS operating systems regardless of version level. It will not read disks that have been formatted under some non-compatible MS-DOS operating systems. This change is not considered a bug.

Earlier versions of MS-DOS examined the first byte of the File Allocation Table (FAT) to determine media type. MS-DOS version 3.2 examines a media type byte located in a structure called a Bios Parameter Block (BPB) which is located in the very first sector (BOOT SECTOR) of MS-DOS formatted media. Zenith MS-DOS versions (and PC-DOS) locate the start of the BPB at the 12th byte (offset B) of the Boot Sector. Some systems locate the start of the BPB at the 4th byte (offset 3) of the Boot Sector. Attempts to read such disks under MS-DOS version 3.2 will fail. After attempting to read such a disk from a given drive, properly formatted disks cannot be read from that drive until the system is rebooted.

**Bug:** What is the procedure for preparing a system with two hard disks for transport? Does the SHIP utility prepare both drives.

**Zapped:** The SHIP utility moves the read/write heads of hard disk#0 and hard disk#1 (if installed) to the hub of the hard disk platters. In this position (the shipping zone), the heads and the platters are least susceptible to damage from vibration.

**Bug:** What's the latest Monitor ROM version for the Z-386?

**Zapped:** Order part number 444-549-8 from the Heath Parts Department (616) 982-3571. This ROM is version 2.6E.

✱



# Announcement!

## HUG MEMBERS ONLY!!

The HUG-386 and HUG-386-C upgrade kits will be available shortly. Wheelin' Dealin' Jim has managed a super-fantastic deal on these two products for Heath Users' Group members who originally purchased an H-241 or H-248; **one-thousand five-hundred dollars** off the regular purchase price! That's right! If you originally purchased an H-241 or H-248, and you're a HUG member, you can get \$1500.00 off the regular retail price of either of these two upgrade kits!

The HUG-386 and HUG-386-C are upgrade kits that let you upgrade your H-241 or H-248 series computers up to a full H-386. Now, how do you determine which upgrade kit to buy? The H-386-C includes a dual winchester/floppy controller, while the H-386 does not include any disk controller. Since the old H-241 controller is not '386 compatible, you'll probably want the "C" model if you're upgrading a '241. If you're upgrading a '248, your decision will depend on whether you need a new dual controller or not.

Here are the three ways you can order your upgrade:

### Write-In Orders

- Non-HUG members **can** order by including payment (with the upgrade kit order) for one year's membership in the Heath Users' Group.
- All orders should be submitted to the Heath Users' Group.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased for.
- Each order must have the persons HUG ID number written on it.

### Phone-In Orders

- Non-HUG members **can** order by first ordering a one year's membership in the Heath Users' Group.
- All orders must be phoned in to (616) 982-3838 from 8 AM to 4:30 PM EST.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased for.
- The person ordering must supply his/her current HUG ID number.

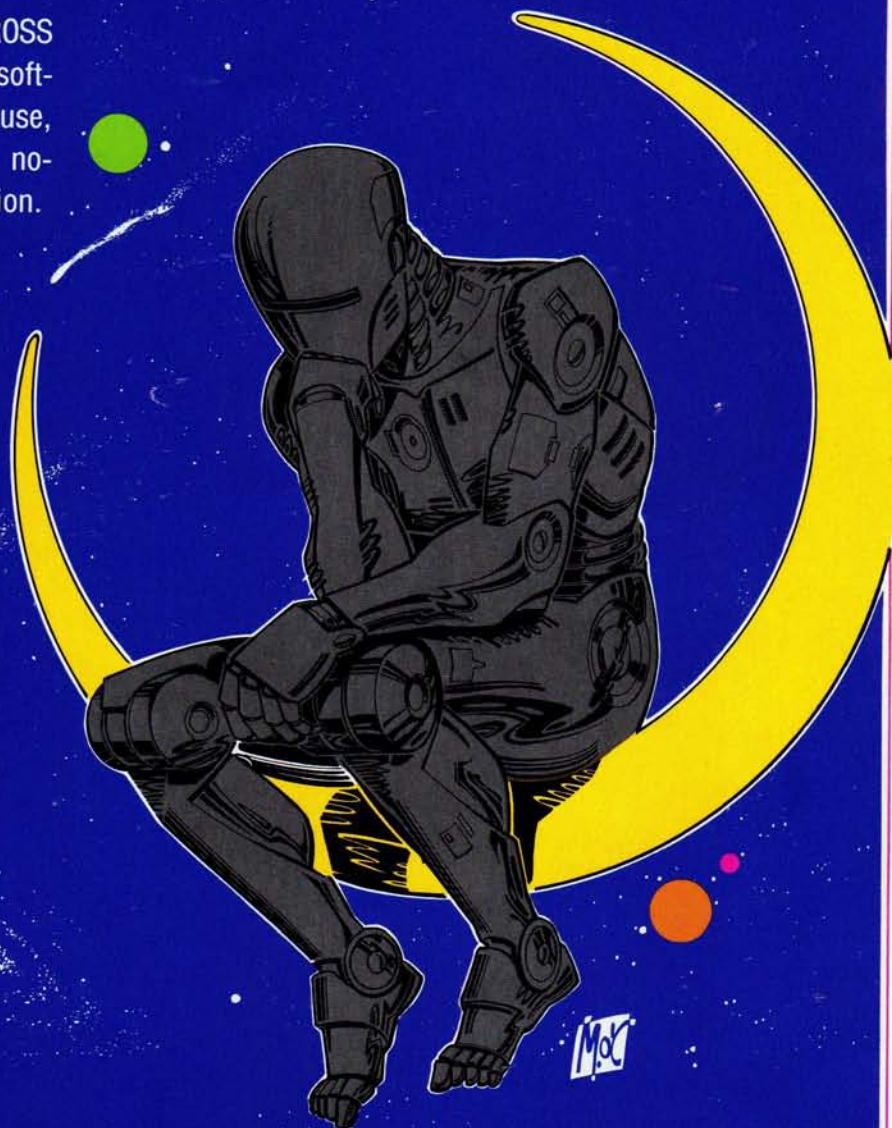
### Heath/Zenith Computer Store Sales

- Non-HUG members **can** purchase an upgrade kit by first purchasing a HUG membership from the store.
- Orders for the upgrade kit can be taken in the normal fashion.
- Each order must have the buyer's HUG ID number on it.
- Each order should indicate which computer kit the upgrade was purchased





Don't get HYPER or CROSS  
because your modem soft-  
ware is too complex to use,  
get *HUGMCP* for the no-  
hassle modem connection.



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 885-2112