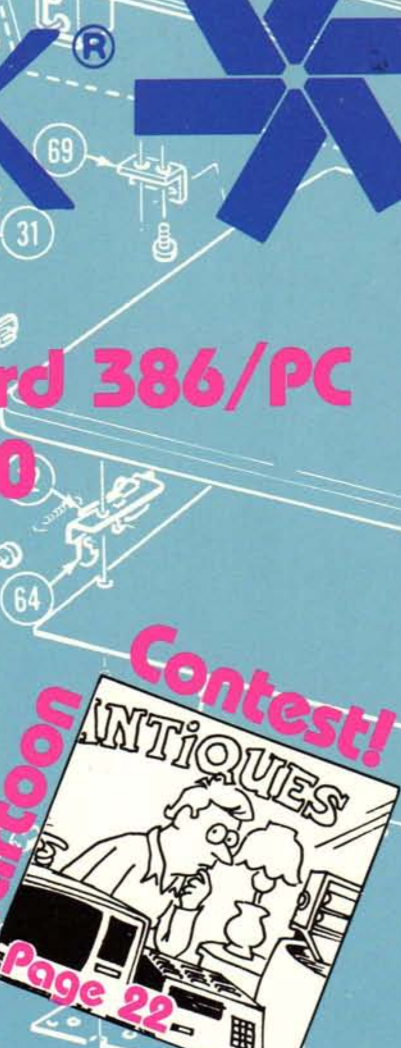
The Official Zenith/Heath Computer Users Magazine

REMark®

November 1989

Installing the Intel Inboard 386/PC into the Heath/Zenith-150

Page 5



Cartoon Contest!

Page 22

Change the ROM in Your SupersPort 286

Page 23

Interfacing Using the A-Bus System

Page 45

ZWL-200-2 and ZWL-200-4 Exploded View

"How Can You Take Advantage of Me"

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

The Official Zenith/Heath Computer Users Magazine

REMark®

Volume 10, Issue 11 • November 1989

PC Compatible

PC Compatible & H/Z-100

H/Z-100 Only (Not PC)

General

PC Compatibles

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

Resources

Reader Service No.		*Page No.*
196	Covox, Inc.	33
104	FBE Research Co., Inc.	38
137	Jay Gold Software, Inc.	21
107	Paul Herman, Inc.	18
136	Lindley Systems	44
114	Micronics Technology	28
117	Payload Computers	16
130	QuikData, Inc.	4
***	REMark Authors	34
188	Serendipity Associates	15

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

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

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

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

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

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

Contributing Editor...... William M. Adney

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

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

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

*U.S. Funds

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

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

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.

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	MSODS	PRINTER PLOTTING UTILITY	25.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHICS	885-3031-37	MSDOS	ENTERTAINMENT	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BACKGROUND PRINT SPOOLER	885-1247-37	CPM	UTILITY	20.00
KEYMAC	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CPM-85	885-1245-37	CPM	UTILITY	20.00
MAPLE	885-8023-37	CPM	COMMUNICATION	35.00
MATHFLASH	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PARTY	885-8042-37	MSDOS	ENTERTAINMENT	20.00
SCICALC	885-8028-37	MSDOS	UTILITY	20.00
SKYVIEWS	885-3015-37	MSDOS	ASTRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PACKAGE	885-3018-37	MSDOS	VARIOUS SPREADSHEETS	25.00
TREE-ID	885-3036-37	MSDOS	TREE IDENTIFIER	20.00
USEFUL PROGRAMS I	885-3022-37	MSDOS	UTILITIES	30.00
UTILITIES	885-3008-37	MSDOS	UTILITY	20.00
ZBASIC DUNGEONS & DRAGONS	885-3009-37	MSDOS	GAME	20.00
ZBASIC GRAPHIC GAMES	885-3004-37	MSDOS	GAMES	20.00
ZBASIC GAMES	885-3011-37	MSDOS	GAMES	20.00
ZPC II	885-3037-37	MSDOS	PC EMULATOR	60.00
ZPC UPGRADE DISK	885-3042-37	MSDOS	UTILITY	20.00
H/Z-100 and PC Compatibles				
ADVENTURE	885-3016	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048	MSDOS	UTILITY	20.00
CXREF	885-3051	MSDOS	UTILITY	17.00
DEBUG SUPPORT UTILITIES	885-3038	MSDOS	UTILITY	20.00
DPATH	885-8039	MSDOS	UTILITY	20.00
HADES	885-3040	MSDOS	UTILITY	40.00
HELP	885-8040	MSDOS	CAI	25.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
HUG BACKGROUND PRINT SPOOLER	885-3029	MSDOS	UTILITY	20.00
HUG EDITOR	885-3012	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THRU 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
HUGPBBS SOURCE LISTING	885-3028	MSDOS	COMMUNICATION	60.00
HUGPBBS	885-3027	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAGAZINE DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS's PC & Z100 UTILITIES	885-3052	MSDOS	UTILITY	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
SCREEN DUMP	885-3043	MSDOS	UTILITY	30.00
UTILITIES II	885-3014	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047	MSDOS	UTILITY	20.00
PC Compatibles				
ACCOUNTING SYSTEM	885-8049	MSDOS	BUSINESS	20.00
CARDCAT	885-6006	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOTTING UTILITY	25.00
GRADE	885-8037	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001	MSDOS	UTILITY	20.00
LAPTOP UTILITIES	885-6014	MSDOS	UTILITY	20.00
PS's PC UTILITIES	885-6011	MSDOS	UTILITIES	20.00
POWERING UP	885-4604	N/A	GUIDE TO USING PCS	12.00
SCREEN SAVER PLUS	885-6009	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00

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

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

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

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

ORDERING INFORMATION

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

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

QUIKDATA - 12 YEARS OF H/Z SUPPORT!

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

ACCELERATE YOUR PC!

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

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

MEMORY UPGRADES

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

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

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

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

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

WINCHESTER UPGRADE KITS

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

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

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

BARE WINCHESTER DRIVES

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

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

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

ANY DRIVE IN YOUR PC/XT/AT

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

DOES YOUR COMPUTER KNOW THE TIME?

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

ZENITH COMPUTER SAMPLE

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

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

Zenith MinisPort notebook computer with 1MB RAM - $1395

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

OTHER STUFF

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

8-BIT

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

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

H-SCOOP

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

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

QUIKDATA, INC.

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

Reader Service #130

INSTALLING THE INTEL INBOARD 386/PC INTO THE HEATH/ZENITH-15O

Copyright (c) 1988 by

David M. Caranci
Meadow Street Box 39
Litchfield, Ct 06759

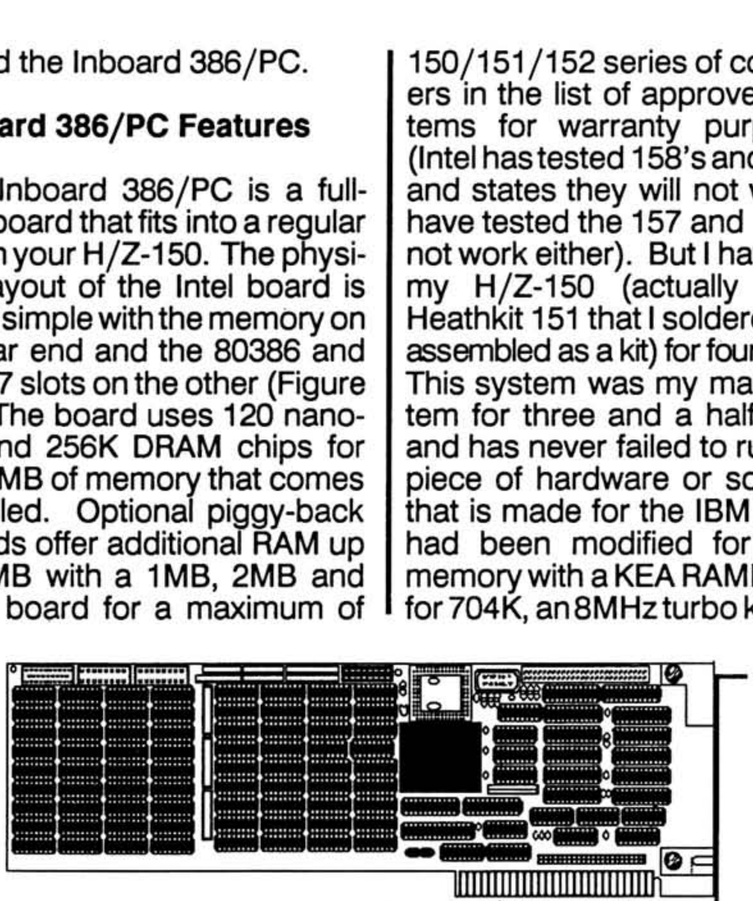
By all accounts, except IBM, the trade-in program sponsored by IBM to encourage owners of PC's, XT's and AT's to turn in their older systems for new PS/2's has not met with much success. There could be a lot of reasons why, but the one that intrigues me most is that most people are probably still getting very good use out of their older systems.

What if it were possible to tap into this installed base of personal computers and help extend the life of these older systems even further? What if it were possible to have owners and users of PC and XT machines skip a whole generation of 80286 AT type computers and move into the 80386 class of machine? What if it could be done for a fraction of the cost of a new 80386 machine? **What if it could be done for less than $900.00?**

Several companies have asked those very same questions and have come up with answers. The answers take the form of add-in accelerator boards that replace your old 8088 CPU or complete motherboard replacements. This is the story of one such answer from Intel Corporation called the Inboard 386/PC.

Inboard 386/PC Features

The Inboard 386/PC is a full-size board that fits into a regular slot in your H/Z-150. The physical layout of the Intel board is quite simple with the memory on the far end and the 80386 and 80387 slots on the other (Figure 1). The board uses 120 nanosecond 256K DRAM chips for the 1MB of memory that comes installed. Optional piggy-back boards offer additional RAM up to 4MB with a 1MB, 2MB and 4MB board for a maximum of 5MB. The Inboard's 80386 runs at 16MHz and no wait states. There is a socket for an 80387 math co-processor chip that also runs at 16MHz.



Accelerator boards have a nasty habit of not running in a few close compatibles and many clones. In fact at this writing Intel does not currently include the Zenith 150/151/152 series of computers in the list of approved systems for warranty purposes. (Intel has tested 158's and 159's and states they will not work, I have tested the 157 and it does not work either). But I have had my H/Z-150 (actually it's a Heathkit 151 that I soldered and assembled as a kit) for four years. This system was my main system for three and a half years and has never failed to run one piece of hardware or software that is made for the IBM XT. It had been modified for more memory with a KEA RAMPAL kit for 704K, an 8MHz turbo kit from Dante Bencivengo, and the video board has the Heathkit modifications so I can run a Hercules video graphics board. The 150 has a 30MB Seagate hard disk with a Western Digital controller. It runs great, but it is old and lags behind even some "turbo" type 8088 clones. A much needed shot in the arm is required for it to keep up with todays high-powered systems.

The Intel Inboard 386/PC is just such an injection. This board is designed to be a drop-in affair with no switches or jumpers to set. You simply pull the 8088 out, run a cable from the 8088 socket to the board, and turn on the computer. But it really is not as simple as all that. This product is actually a complete kit that has all of the components that you will need to do the installation. Intel even supplies a chip puller and a small plastic tube to be used when you remove the 8088 CPU chip.(Figure 1).

INSTALLATION

STEP ONE in installing this board is to remove the cover of the H/Z-150. (NOTE: Before any procedures are followed, make sure the system has been turned off and the power cord removed from the system!!) Next remove the keyboard plug from the back of the computer. Finally, remove the CPU board and the RAM board from their respective slots.(Figure 2)

STEP TWO is to configure the RAM board with the proper amount of memory. The Inboard requires that 256K be the only amount of memory installed in your H/Z-150 system. If you were installing the Inboard in an IBM XT you would disable all memory above 256K on the motherboard by changing Switch 2. But on the H/Z-150 that won't work. Not only don't you have a motherboard but if you change the switches on the CPU board to indicate 256K and you actually have more or less RAM installed than that, the Inboard will not run.

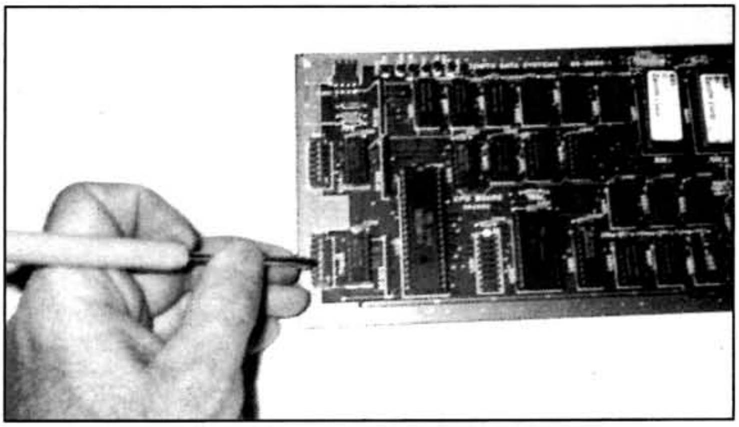


Figure 3- Setting Switch 2 on CPU board to 640K

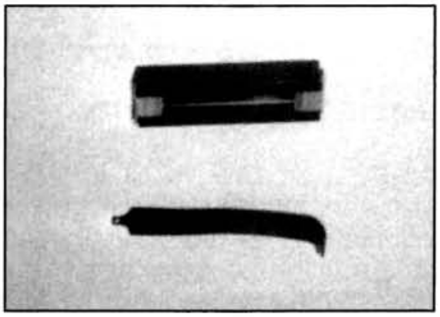


Figure 1 - Inboard 386/PC chip puller and tube

For the Inboard to work properly on the H/Z-150, you must remove all memory down to 256K on the RAM board and **then set the switches on the CPU board to 640K, not 256K like the manual says** (Figure 3).

If you have a RAMPAL kit that lets you combine 256K chips and 64K memory chips you don't need to remove it. If your RAMPAL uses 256K chips you must physically remove all RAM chips except for the first bank or row of chips. If you use only 64K chips you must have the first 4 banks of chips installed. Once the memory board has had the chips removed, **set Switch 2 on the CPU board for 640K!! THIS IS IMPORTANT!! No other switch setting will work!!** If you get an S O S (three short, three long, three short) beeping from the system when you boot up you will know that there is some problem between the memory on the Inboard and the RAM board.

The Intel board uses 640K of its own 32-bit RAM as the main system RAM. The 256K on the H/Z-150 RAM board is lost to

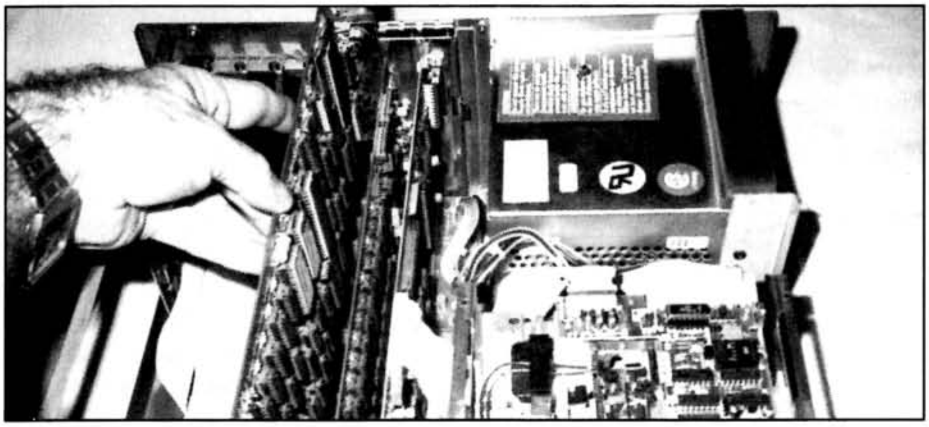


Figure 2 - Removing H/Z -150 CPU board

there is some problem between the memory on the Inboard and the RAM board.

The Intel board uses 640K of its own 32-bit RAM as the main system RAM. The 256K on the H/Z RAM board is lost to you and cannot be accessed. However the Inboard 386/PC uses 256K of its own 32-bit RAM as extended or expanded memory.

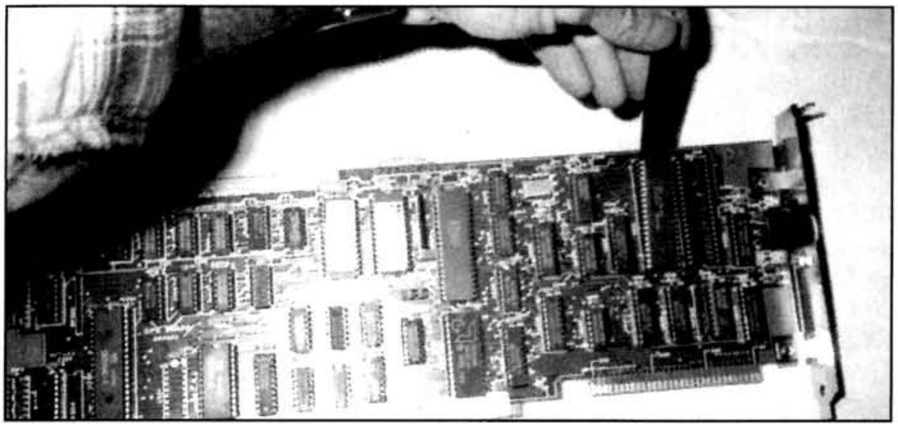


Figure 4 - Removing the 8088 CPU chip

STEP THREE, after these preliminaries are done, is to remove the 8088 CPU from the CPU board. Place the CPU board that you removed from the computer on a flat surface in front of you with the top of the board away from you. Ground yourself to the computer by touching the metal frame. This discharges any stored up static electricity in your body. Then, using the chip puller that comes with the kit, gently pry up and remove the 8088 chip from it's socket (Figure 4).

If you don't know where the 8088 chip is, it is the largest chip in the upper right of the CPU board. If you have an 8087 math co-processor chip installed you must remove it too, at this time. Avoid touching the metal pins of the 8088, and place the it in the black plastic tube that came with the Inboard. Place the tube in a safe place. You won't be needing the 8088 anymore if all goes well.

STEP FOUR requires that you install the cable that connects the 8088 socket to the board. The cable is a 40 pin cable that is heavily shielded on one side. The length of the cable with the Inboard 386/PC is just rightfor the H/Z-150 and a longer one is available from Intel at no cost (except time). Place the forty pin male end of the cable into the empty 8088 socket. Make sure the #1 sticker on the 40-pin cable end is at the top of the Inboard (Figure 5).

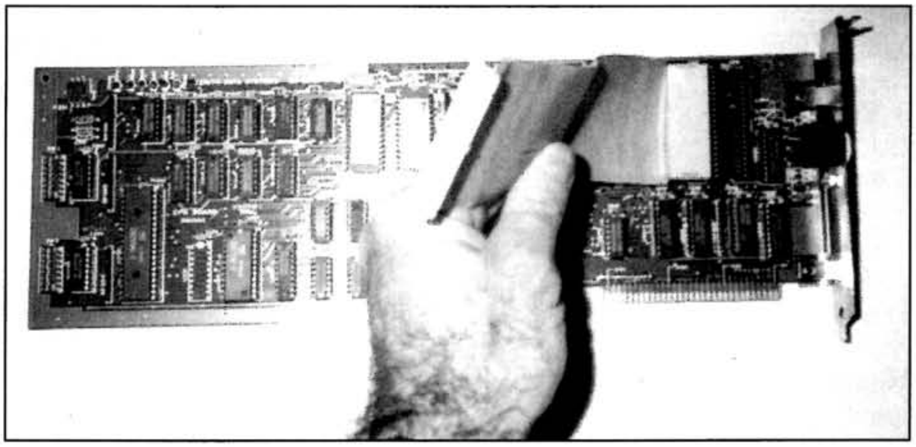


Figure 5 - Installing the Inboard Cable

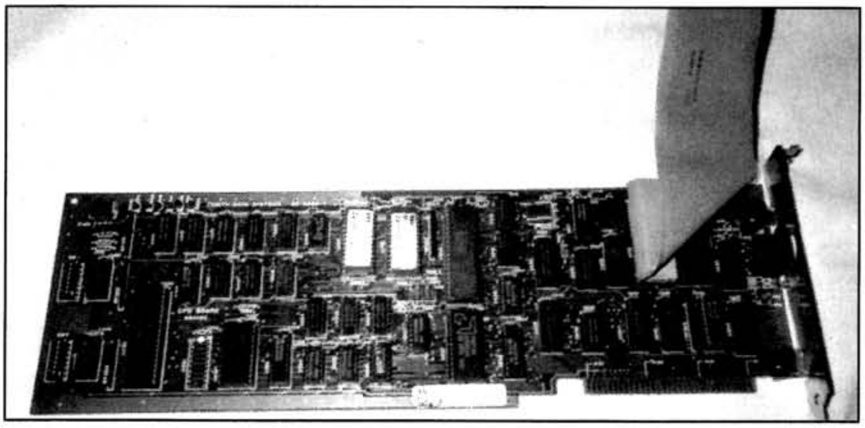


Figure 6- Bending the Inboard 386/PC cable

Be very careful so you do not bend any of the gold plated pins.

Pins that bend outward can be seen quickly, but not those that bend inward. **Take your time as this is the most critical step in the installation process.**

Next you need to bend the cable (Figure 6) so it will be in the proper position to run over the top of the other boards in the computer.

STEP FIVE is to install the CPU board back into its slot and screw down the tightening screw. There is just enough room between the CPU board and the video board for the cable to fit if you have bent it correctly. Don't manhandle the cable. Even though it is sturdy, it will break if too much pressure is applied, especially at the connection ends.

Next install the Inboard 386/PC in any empty slot next to the RAM card. Which slot you use is up to you, it doesn't matter to the Inboard. I installed mine right next to the RAM card. If you move it farther away there is less cable to have to pack down when you put the cover on, but too far away and you may not have enough cable.

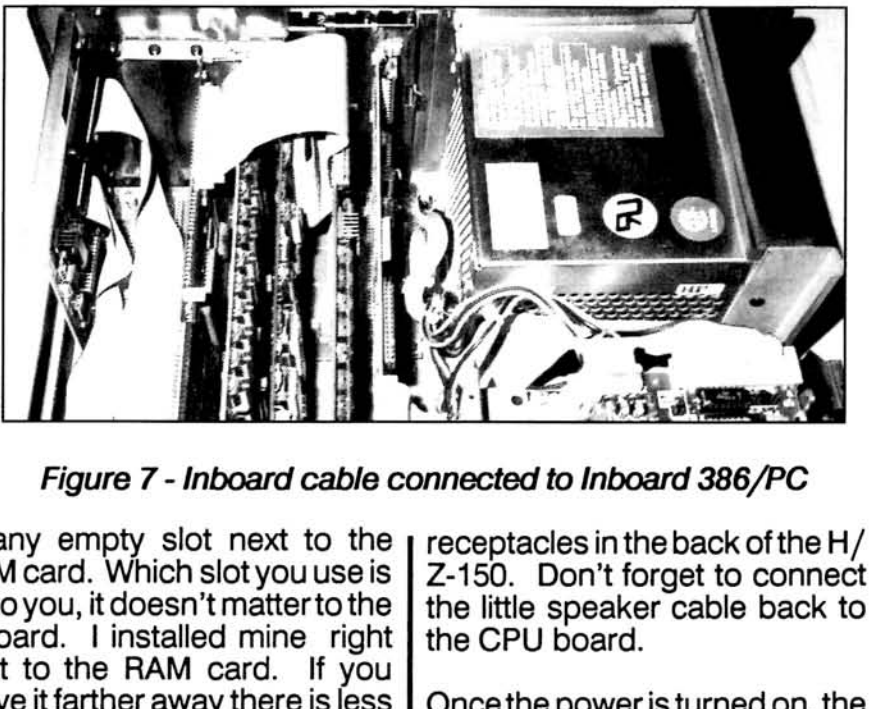


Figure 7 - Inboard cable connected to Inboard 386/PC

STEP SIX will be to connect the other end of the Inboard cable to the Inboard 386/PC board (Figure 7).

You can't get it wrong because there is a key on the female end of the cable and a key slot on the receptacle of the board. Be careful again, it is very easy to bend or break pins.

STEP SEVEN. Once the cable is installed in both places, the 8088 socket and the accelerator board, the only thing left to do is to turn the machine on.

Before you do, make sure that everything is tight - the boards in their slots and the cable ends in their respective places. The 168 watt power supply in the H/Z-150 will have no problem supplying enough wattage to the added Inboard 386/PC. Connect the power supply cord and the video cable as well as the keyboard cable to their proper receptacles in the back of the H/Z-150. Don't forget to connect the little speaker cable back to the CPU board.

Once the power is turned on, the H/Z-150 goes through it's normal POST routine (Power On Self Test). If there is something wrong with the installation, now is when you will know it.

The LED's (red lights on the top of the CPU board) should turn off one by one like they normally would. If **any** of them stay on there is a problem (thank you Heath/Zenith for those lights, they have saved me more diagnostic time than you can imagine).

In a few seconds the DOS prompt should appear on the screen, indicating a successful boot. If the system doesn't boot, turn to the back of the Inboard 386/PC manual and find Appendix B, Troubleshooting, and follow what the manual says.

STEP EIGHT is to install the proper software. You can follow the procedures in the manual for this step. There is nothing different to do for the H/Z-150 here.

OPERATIONS

The Inboard 386/PC needs a driver to take full advantage of the 80386 technology. The Inboard will operate at 16MHz without any software from Intel installed, but the driver, called INBRD.SYS, provides additional features like changing speed with the keyboard, faster screen performance, and extended memory access. You can use some or all of the extended memory on the board as expanded memory. Expanded memory cards in sizes of 1MB, 2MB and 4MB are available from Intel. These cards are piggy-back designs and can also be used only with the Inboard 386/PC.

If you wish to use an expanded memory board also, you cannot use both the Inboard EMS driver (ILIM.SYS) and the driver that goes with your EMS board. You must use the memory on the Inboard only as extended memory if you use an EMS board.

The manual has a nice section that explains extended and expanded memory and I highly recommend each user who is unfamiliar with the terms read that section two or three times.

The setup program supplied by Intel does an excellent job at installing the software and configuring the CONFIG.SYS and AUTOEXEC.BAT files.

Blazing processor speed can be useless if your video is slow to get the information to the screen. The Intel board also comes with the ability to run an EGA ROM BIOS (if present) in the Inboard 386/PC's 32-bit memory. I have installed a Hercules video card in the H/Z-150 and this software did speed up my video.

The Intel board can also be configured to speed up or slow down from batch files. This is done by inserting processor wait states. The slowest speed, 1, is close to the 4.77MHz of the the original 8088. Speeds of 2, 3, and 4 reduce the number of wait states and thereby increase processing speed.

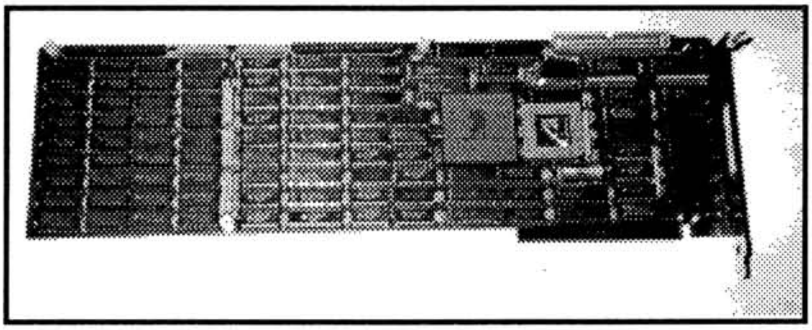


Figure 8 - The Inboard 386/PC

The Inboard 386/PC comes with an incredible hard disk caching program called ICACHE. In my own tests using PC Magazine's Benchmark series this piece of software out-performs every disk caching program that I have (4 or 5) and really complements the Inboard 386/PC.

COMPATIBILITY

My H/Z 150 has always run every piece of software that I have ever thrown at it. Adding the Inboard 386/PC hasn't changed that at all.

About the most finicky piece of hardware/software I have is a Microcom 9624x modem and QMODEM software. At the Inboard's highest speed the Microcom sometimes refused to dial a number. But it always worked on the second try.

PERFORMANCE

In PC Magazine's Benchmark tests the Inboard 386/PC consistently outperformed an 8MHz IBM AT. Actually the Inboard 386/PC runs about twice as fast as an 8MHz AT and just a shade slower than a 16MHz COMPAQ 386. There is no doubt that this board is a vast improvement over the 8088.

Recalculation of spreadsheets is cut down by a factor of 4 or 5 in my own tests, and bit mapped graphics type of programs really zip along.

A real plus for Intel in this category is that the Inboard will run a special version of Windows/386 made just for the Inboard 386/PC. The Windows environment has not been known for its speed and on 8088 systems it is a real slug, which makes it quite useless. But with the Inboard 386/PC installed, screens zip by and Windows/386 actually becomes usable.

Hard Disk Performance Boost

Another performance factor to take into consideration is the access speed of your hard disk. It is not uncommon for older PC's to have 5, 10 or 20MB hard disks with average access speeds of as much as 85-100ms.

Compared to newer AT type hard disk subsystems which can run in the 16-40ms access speed area, they are very slow. If you can't upgrade your hard disk to one of these newer type systems, then try two things. First make sure that your disk has been low-level formatted with the correct interleave. Some disks can experience a whopping 200% increase in throughput when the optimum interleave is used. Programs like Spinrite from Gibson Research Corporation are excellent for determining which interleave value is best for your combination of hard disk and controller. Once the hard disk is formatted with the optimum interleave, ICACHE, the disk caching program that comes with the Inboard 386/PC will make your hard disk seem to run faster by storing frequently used disk information in fast 32-bit memory.

Multitasking - on a PC?

Besides raw power, one of the main reasons someone should look into the purchase of these type of boards is the ability to multi-task or run multiple programs at the same time. The Intel Inboard ran a full system demo copy of PC-MOS 386 a multitasking operating system, without a problem.

I run multiple tasks (jobs) in Windows/386 and although the system does slow down as you add tasks, it is not as slow as my 8088. I wanted to see if DESKVIEW 2.0 would fly but Quarterdeck never sent me a copy to test, despite repeated calls to them and assurances they would.

If you wish to do multi-tasking you need memory and lots of it! Each task will want to have its own 640K of memory. Windows/386 requires 2MB to even start. Intel carries a full line of daughter-boards to supply 32-bit memory up to 5MB. These daughterboards come in 1MB, 2MB, and 4MB sizes. If you need 80387 co-processing power for those large spreadsheets or math intensive programs like CAD/CAM, Intel provides a 16MHz 80387.

As mentioned earlier, once booted up, there was not one piece of commonly used, off-the-shelf software that the Intel board refused to run. These included Lotus 1-2-3, MS Word 4.0, RBBS-PC bulletin board, caching programs, Windows 2.03, PC Paintbrush, the Microsoft Bus Mouse, Natural Microsystems Watson voice messaging/modem card and software, a Microcom 9624x modem and QMODEM communications software, and a few TSR programs.

OS/2 Support

The Inboard 386/PC will not run OS/2. Intel says that there are no plans to incorporate OS/2 compatibility.

The H/Z-150 is a great machine, and can be made better with this accelerator board, but it will never be an AT or a real 386. If OS/2 is going to be important to you, stick with a true 386 computer.

SUPPORT

Intel really shines here. Phone support for their products is offered two ways, a toll-free tech support number and a 24-hour BBS. The bulletin board reacts, in some cases, faster than Intel's own tech support. The sysop is Al Kinney, Senior Technical Marketing Engineer, and he does a great job, even answering technical questions and inquiries on some weekends. I dream of the day when every major vendor of software and hardware has a toll free number and a 24 HR BBS.

The manuals and associated documentation on disk that come with the Inboard 386/PC offer more than enough information to install and diagnose the board. The Intel manual is of a very high quality with great diagrams and separate chapters for each of the machines they support.

The H/Z-150 is not one of these machines, unfortunately. Intel says that this is because they are not sure how the board will react to a system with chips physically removed. However when pressed, Intel said it would help in support of the H/Z-150 because they don't want any dissatisfied customers. There's a refreshing outlook!! I have had the board installed in my H/Z-151 for over a year and have never had one problem! Hopefully my experience and that of other H/Z-150 owners will convince Intel to formally list it as a supported machine.

The Inboard 386/PC is warranted for 5 years. A fully illustrated, laser printed manual for installing the Inboard in the H/Z-150 can be sent to you first class by sending a $15 money order or cashiers check (sorry no personal checks) to me at the address listed on page one.

VALUE

In overall value the Intel board is a real bargain. For awhile the prices of 256K DRAM chips sent systems and peripheral prices through the roof. Prices have been declining since the summer of 1988 and at this writing the list price of the Intel board is $795. One well known mail-order establishment is selling the Inboard 386/PC for $695, including a copy of Windows/386.

CONCLUSIONS

It takes more than raw power to win over today's sophisticated user, the type of user that would most be attracted to this upgrade solution. It takes painless integration and operation. While no board is entirely painless to install, the Intel board comes close and offers many useable features.

The Inboard 386/PC is a finely tuned piece of equipment with software to match. I would not recommend a novice computer user to undertake this addition. But many H/Z people have intimate knowledge of their computers and this upgrade is a dream come true.

Running *Sprint* on the H/Z-100

Brad Rylander
328 Hanover Avenue #10
Oakland, CA 94606

Z-100 owners rejoice! At last, there is a modern full-blown word processor that runs on this venerable old machine. Sprint from Borland International will give you all the features that you've wanted and then some. Best of all, it runs on the Z-100 with or without Pat Swayne's marvelous ZPC.

To get the most from Sprint on the Z-100 in its "native" mode (without ZPC) you'll need 256K of RAM, MS-DOS version 2 or greater and HUG's KEYMAP program. There's also some simple tricks involved to install the program and I'll explain them to you. But first let me tell you a few things about Sprint.

Yet Another Word Processor?

No. Sprint is unique among word processors available for microcomputers. Unique is a pretty bold word, but read on and I think you'll agree that Sprint *is truly unique*. Sprint was once *The Final Word* from Mark of the Unicorn. I had long lusted after that program, a genuine cult classic, but its non-discounted $500 price tag moderated my passion.

Borland spiffed up the text formatting to bring it closer to a true WYSIWYG (what you see is what you get), added more printer drivers including support for laser printers and a Postscript driver, improved the documentation and support, changed the name and dropped the price.

Sprint is based on EMACS, the powerful UNIX editor written by the hacker's hacker, Richard Stallman. Actually, Sprint *is* EMACS combined with an on-screen text formatter that can handle everything including pictures and graphs if you have a Postscript printer.

Add to this a decent mailmerge, a well integrated spelling checker and thesaurus, both with huge dictionaries and both surprisingly quick even on my 8" floppy, and you have a pretty complete word processor. Complete but not yet unique — there's more.

Sprint can be installed with "user interfaces" that mimic the command set of Microsoft Word 4.0, WordStar 4.0, WordPerfect 4.2, SideKick and MultiMate II. It can also read and write files formatted for these other programs provided that you have at least 448K of RAM free.

If you're looking for real editing power, you'll stay away from these rudimentary command sets and choose one of the advanced interfaces; Final Word, EMACS, MINCE or Sprint's own Borland User Interface with its WordStar-like commands.

The program also has an automatic "incremental save" that's a pretty good imitation of UNIX's auto-save and mail. Sprint saves your work in a temporary file every time you stop typing for a period that's adjustable from one to sixty seconds. Should you lose power or crash, this temporary file can be recovered and only your most recent keystrokes will be lost.

If you live in an area of frequent thunder storms or unreliable power, or if you have less than one hundred percent faith in your local area network, this feature alone could make Sprint worth your while. You can turn the auto-save off, but it's so fast that you'll hardly notice it, even on a floppy system.

Sprint's place-markers are just another of the program's many pleasant surprises. Any decent editor supports place-markers, but Sprint takes the concept a step beyond the usual. Although the place-markers are not maintained across editing sessions, they can be placed in as many as ten different files. Two keystrokes let you bounce between files — and bounce is the word, the process takes place as quickly as your screen can be repainted.

The program has so many built-in niceties that I can't possibly list them, but here's one more to help you get an idea of the beauty of the program's design. Sprint automatically detects ASCII files and edits them in the ASCII mode. Never again will you have to filter out "garbage" characters from your source code, wonder why your CONFIG.SYS bombed or upload gibberish to your favorite bulletin board.

Sprint's many well thought out features put it in a class by itself, but what makes the program truly unique is that it's more than just a fine tool . . .

It's a Toy Too

I think many of you Huggies will appreciate a word processor that's infinitely extensible and modifiable. Sprint fills the bill. Sprint's text formatter can be customized — even completely rewritten with a macro language that has commands and structure much like English.

The real fun, at least from my point of view, is the ability to modify the editor itself. This can be done at several different levels of complexity. At the easiest level, you can assign your own keystrokes to commands and activate "unused" commands from within Sprint's menu structure. At the opposite extreme, you have access to a powerful macro programming language.

I'm not using "powerful" as a journalistic buzz word. Sprint's language can read or write directly to the CPU's registers (although writing to the base pointer and segment registers is forbidden — thank goodness). It can also perform MS-DOS software interrupts, read and write to any memory location and to the machine's I/O ports. That's getting pretty close to the bare metal. In other words, like C or assembler, Sprint's macro language allows you to make a fine mess of things.

If this sounds too daunting, don't worry. There's little reason to attack the machine's hardware. If you decide to do some extensive customization, you'll work with a language that looks like C and use built-in macros with meaningful names, such as ***scroll, insert*** and ***move.***

These built-in macros handle all the nitty-grity details so programming with this language is as easy as programming can be. Sprint's five hundred page "Advanced User's Guide" provides clear examples for using each of the two hundred or so macros. If you've never written a line of code in your life, but have wanted to try, this is a good place to start. It's easy and you get instant, useful results.

Each of the user interfaces I've mentioned comes in the form of complete well commented source code. Browsing through this code with a good ASCII editor such as Sprint, you're bound to find little changes to make — this program begs to be tinkered with.

```
;;screen non-IBM.Zenith Z100,
;; The bit-mapped screen is slow.
;; When Sprint is running, "key expansion" is disabled, so every key
;; returns a 1-byte code.  This makes it easy to write a macro set for
;; this machine.
rows 25,cols 80,
init ^[x1^[y?^[x4^[w,
reset^[y1^[x?^[v^[H^[J,
reinit^[E,cleol^[K,cur^[Y%32+c%32+c,
set^[%[q%:p%],
```

Table 1A
Original Zenith Code in MAIN.SPL

Make the changes, compile them with (and within) Sprint, test the results and go back and fix your mistakes. The whole process is quick, easy and fun. Sprint provides the perfect programming environment for itself.

If you've done any programming, you might worry that by modifying something as complex as an editor you could end up with an incomprehensible mess. Not so. Any changes you make to key assignments automatically show up in Sprint's comprehensive menu structure.

Adding new features to existing menus is as simple as typing them in to the existing menu code. Creating your own pop-up menus is child's play, just follow the examples in the source code.

One quick word on menus before we go into installing the program. Although Sprint has layers upon layers of menus, it's never necessary to wade through them. All commands can be issued with a control code or a tap on a function key. Now, the gory details.

Installing Sprint on the Z-100

This is not as straight forward as it ought to be. Sprint comes with an install program, SP-SETUP.EXE. This program is IBM specific and requires ZPC and a ZHS board to run on a Z-100. Even then, it's not quite perfect under ZPC.

Borland has developed a generic MS-DOS program, SPINST.COM that works well on the Z-100. This program is available either directly from Borland International (give them a call and have your serial number handy) or from the Borland SIG on COMPUSERVE where it can be found in the Sprint Library as SPINST.ARC.

I have heard that this program is also available on Genie and BIX, but I haven't checked that out. Although SPINST.COM isn't absolutely necessary, the only other alternatives are to run SP-SETUP under ZPC or to install Sprint for your Z-100 on a true compatible.

Z-100 Screen Installation

Before running the install program, either SP-SETUP.EXE or SPINST.COM, it is necessary to make minor modifications to the file MAIN.SPL. This is an ASCII file and must be edited with an ASCII editor, such as WordStar in the non-document mode.

MAIN.SPL is a "library" file that the installation program refers to. Search this file for Z-100. Starting on line 252 you should find the code listed in Table 1A.

This block of code describes how Sprint will handle the Z-100 screen, it's similar to a UNIX termcap description. Don't worry about the comment on the slow screen. Sprint runs in the character mapped mode and it is as fast as anything I've seen on my Z-100.

First, you must un-comment the first line. Just remove the two semicolons before the word **screen** and make sure the the s in **screen** is in the first column.

The next step, in spite of the comment, is to leave the "key expansion" enabled. Do this by deleting the characters ^ [y? in the **init** entry. More on this in a moment. Again in the **init** entry, the characters ^[x4 turn on the block cursor. If this annoys you, delete them.

The entry set^[%[q%;p%], enables and disables reverse video. If you have color RAM installed, you'll probably want to experiment with enabling color rather than just having two screen attributes. This will be useful even with a monochrome monitor to differentiate type styles on-screen.

Comment out the present **set** command (put ;; as the first characters on the line), this way you can easily recover the original entry. Now type in the new entry found in Table 1B.

This will enable a few of the Z-100's color combinations to be displayed by Sprint. When you enter this line type the ^ (caret) and the [(left bracket), not <CONTROL>[(escape) and don't forget the comma. It's required at the end of each entry except the last.

You might recognize this entry as a string of escape sequences that control the colors displayed by the Z-100. You can refer to Table 2 and change the numbers to suit your taste.

Although this will allow you to display *some* of the available color combinations, it's by no means the most efficient or even the proper way of doing it. You C hackers might recognize the **set** entry as being a printf string. Actually, it's a case statement within a printf string. Once I got this kludge working I gave up on trying to figure out the proper way of doing it.

If you feel brave, refer to page 453-454 in Sprint's "Advanced User's Guide" and use the ANSI.SYS entry in MAIN.SPL as an example. The information is there but it's beyond me. If one of you can figure out how to solve this problem, I'd be glad to hear from you.

Once you have saved these changes to MAIN.SPL, you are almost ready to run the installation program, but first get your modified MAIN.SPL, your installation program (either SP-SETUP.EXE or SPINST .COM), and the proper "library" file for your printer on the same disk. The printer files are on disks 8 and 9 and have names like EPSON.SPL or OKIDATA.SPL.

Now run the installation program and choose **Select the default screen** from the menu. You will then be given the available screen types and asked to choose one. Pick **non-IBM**. The program will then display the various non-IBM screens and you'll naturally pick **Zenith Z-100.**

Finally, you'll be asked where the video output should be sent. Choose **CON**. Now the install program will automatically compile MAIN.SPL and produce a binary file called DEFAULT.SPS. Each time Sprint is invoked it will look for this file and use it to send its output to your Z-100's screen.

The next step is to install your printer. Without leaving the installation program, choose **Select the default printer** and follow the instructions in the User's Guide. The installation will again compile MAIN .SPL and produce a binary file named DEFAULT.SPP. This is your printer driver. Like the screen driver, it must reside in the same directory as Sprint.

The only dirty work left to do is to get the right files on the right disks. If you're doing this on a true compatible, just follow the instructions in the manual.

```
screen non-IBM.Zenith Z100,
;; The bit-mapped screen is slow.
;; When Sprint is running, "key expansion" is disabled, so every key
;; returns a 1-byte code.  This makes it easy to write a macro set for
;; this machine.
rows 25,cols 80,
init ^[x1^[y4^[w,
reset^[y1^[x?^[v^[H^[J,
reinit^[E,cleol^[K,cur^[Y%32+c%32+c,
;;set^[%[q%:p%],
set %[^[m70%;^[m07%;^[m04%;^[m40%;^[m53%;^[m35%;^[m26%;^[m62%;],
```

Table 1B
Zenith code after your modifications. The only changes are the removal of the semicolons before "screen" and in the "init" and "set" entries.

```
Foreground and background colors are set by sending
an  escape sequence to the terminal.

ESC m fore back
Where fore and back are ASCII digits that equal:
0 = Black
1 = Blue
2 = Red
3 = Magenta
4 = Green
5 = Cyan
6 = Yellow
7 = White
Example:   `[m61 will set yellow characters on a blue background.
```

Table 2
Color Codes for the Z-100 (Taken from the Z-100 User's Manual)

With either of the install programs on the Z-100, you must copy the necessary files from your distribution disks to your working disks. SP-SETUP, for some bizarre reason, won't recognize that the Z-100 has any disk drives and the automatic file copying fails. SPINST doesn't even offer any automatic file copying. The documentation is of little help on this so refer to Table 3 for the required files.

Table 3 assumes the worst case, that is installing Sprint on a system with only two 360K floppies. If your system has more disk capacity, you can combine files to suit your needs.

As you can see from Table 3, Sprint is a trifle large. In fact, it comes on eleven disks. If your system has only two 360K drives, Sprint is going to be less than fun; you're probably better off with WordStar 3.x unless you really love swapping disks. If you have one, or better yet, two quad density drives, you can configure an efficient system.

With my single 8" drive, I can run Sprint with both the spelling checker and thesaurus if I forego the help file, SPHELP .HLP. While learning the program I chose to use the help file in lieu of the spelling checker and made up a separate 8" disk that included the spelling checker and help file rather than the thesaurus. These setups have worked well for me. Of course, if you have room on a hard disk, Sprint is a joy.

Gory Function Key Details

After all this work I'm sure you're anxious to type SP and put Sprint through its paces. Not yet. There is still some more grunt work to do. When you edited the Z-100 terminal definition in MAIN.SPL I'm sure you saw the comment about it being easy to write a macro set with key expansion disabled.

This may have been true of an earlier incarnation of the program but doesn't appear to be true today. I've done a great deal of experimenting and have even had considerable help from Borland's technical experts on CompuServe and I'm still unable to get Sprint to recognize the Z-100 function keys.

While Sprint isn't completely reliant on function keys like so many of today's programs, it has so many features that using the function keys, at least for infrequently required features, is almost mandatory. This won't be a problem though, because . . .

KEYMAP Comes to the Rescue

This program is another of Pat Swayne's gems. You'll find it listed in the HUG Price List in this issue. Even if you never intend to run Sprint, you deserve KEYMAP. Go ahead and blow the twenty bucks. The program is more versatile than a Swiss Army knife and it won't wear holes in your pockets.

First, familiarize yourself with Mr. Swayne's well written instructions. Now refer to Table 4. It will show you exactly what to enter to make the Z-100 function keys send the same codes as a true compatible. For a more thorough explanation, refer to pages 437-445 in Sprint's Advanced User's Guide.

Enter these new key codes carefully because they're tricky to edit if you make a mistake. Each of the new codes begins with ^@ which is the ASCII null or zero. Once entered, the nulls and everything after them will show up as null or nothing. Even though all entries in your KEYMAP will appear blank when you go back to make changes, the codes are really there and Sprint will respond to them.

Keyboard Hints

Be sure to make your function keys put out exactly the same codes as a compatible. This will give you a known base to work from. Once you start redefining these keys, and I'm sure you'll want to,

```
This table presents an absolutely bare bones Sprint system as
might be used by those with only two 360K floppy drives.  If your
machine has greater storage capacity use this table to help you
arrange Sprint's files according to your needs.

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

 Volume in drive B is PROGRAM
 Directory of  B:\

SP       EXE   114451   6-16-88   1:00a  Required.
SPFMT    EXE    88065   6-16-88   1:00a  Required.
SPSORT   EXE    14180   6-16-88   1:00a  Only necessary for sorting.
SPMERGE  COM    28801   6-16-88   1:00a  Only necessary for mail merge.
SPRECOVE COM    13085   6-16-88   1:00a  Required.
COLORS   OVL     2728   6-16-88   1:00a  Only necessary to set colors.
CONVERT  OVL     4770   6-16-88   1:00a  Only necessary for file conversions.
SP       OVL    49326   6-16-88   1:00a  Required.
SP       PIF      369   6-16-88   1:00a  Only necessary for Microsoft Windows.
QCARD    SPM    15762   6-16-88   1:00a  Nice to have, not necessary.
STANDARD FMT    14285   6-16-88   1:00a  Required.
STANDARD SPG     2077   6-16-88   1:00a  Very nice to have.
       12 File(s)      8192 bytes free

If you use MS-DOS v. 3.x and anything but the most essential files on
this disk you will not have room for the system files i.e. the disk must
be formatted without the /s switch.  However, COMMAND.COM must be on this
disk.  You must also copy the screen driver DEFAULT.SPS and the printer
driver DEFAULT.SPP that you created with the installation program to this disk.

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

```
 Volume in drive B THESAURUS
 Directory of  B:\


THESAUR  ENG    11299   6-16-88   1:00a
THESAUR  LEX   322816   6-16-88   1:00a
        2 File(s)     26624 bytes free

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

 Volume in drive B SPELL_CHECK
 Directory of  B:\

AMERHYPH LEX    29338   6-16-88   1:00a
AMERICAN LEX   239430   6-16-88   1:00a
HYPHEN   ENG    20477   6-16-88   1:00a
SPELLER  ENG    42947   6-16-88   1:00a
USER     DIC       10   6-16-88   1:00a
        5 File(s)     28672 bytes free

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

 Volume in drive B HELP
 Directory of  B:\

SPHELP   HLP   244543   6-16-88   1:00a
        1 File(s)    117760 bytes free

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

 Volume in drive B DATA
 Directory of  B:\

SP       SWP   131072   1-01-80  12:31a
        3 File(s)    231424 bytes free
```

Table 3

```
This table shows the prompts issued by KEYCON.COM (the
entries up to and including the colons) and the proper
responses to make the Z-100's function keys send IBM
compatible codes to Sprint (the entries after the colons.)
Notice that each of yopur entries begin with the two
characters ^ and @ which represent to KEYCON the ASCII null.

Once KEYCON has produced your KEYMAP file be  sure to run
KEYMAY (or whatever you choose to call it) before you run
Sprint.  Otherwise, the only way to exit Sprint will be via
a reset.

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


Begin by choosing F0 as the "alternate response key."

F0  <Alternate Response Key>              F7  (ESC-Q): ^@^G
F1  (ESC-S): ^@^A                         F8  (ESC-R): ^@^H
F2  (ESC-T): ^@^B                         F9  (ESC-OI): ^@^I
F3  (ESC-U): ^@^C                         F10 (ESC-OJ): ^@^J
F4  (ESC-V): ^@^D                         F11 (ESC-OK): ^@^K
F5  (ESC-W): ^@^E                         F12 (ESC-OL): ^@^L
F6  (ESC-P): ^@^F

The codes above emulate the normal PC function keys.
-------------------------------------------------------------------

SHIFT-F0  (ESC-E):                        SHIFT-F7  (ESC-1G): ^@^W
SHIFT-F1  (ESC-1A): ^@^Q                  SHIFT-F8  (ESC-1H): ^@^X
SHIFT-F2  (ESC-1B): ^@^R                  SHIFT-F9  (ESC-1I): ^@^Y
SHIFT-F3  (ESC-1C): ^@^S                  SHIFT-F10 (ESC-1J): ^@^Z
SHIFT-F4  (ESC-1D): ^@^T                  SHIFT-F11 (ESC-1K): ^@^[
SHIFT-F5  (ESC-1E): ^@^U                  SHIFT-F12 (ESC-1L): reserved
SHIFT-F6  (ESC-1F): ^@^V                                for PrintScreen

The codes above emulate the shifted PC function keys.
-------------------------------------------------------------------

I CHR  (ESC-@): ^@G               I CHR and keypad 7 emulate HOME
D CHR  (ESC-N): ^@I               D CHR and keypad 9 emulate PgUp
INS LINE  (ESC-L): ^@O            INS LINE and keypad 1 emulate END
DEL LINE  (ESC-M): ^@Q            DEL LINE and keypad 3 emulate PgDn
HOME  (ESC-H):
BACK ARROW  (ESC-D): ^@K          Back Arrow and keypad 4 emulate left arrow
FOWARD ARROW  (ESC-C): ^@M        Fwd Arrow and keypad 6 emulate right arrow
UP ARROW  (ESC-A): ^@H            Up Arrow and keypad 8 emulate up arrow
DOWN ARROW  (ESC-B): ^@P          Down Arrow and keypad 2 emulate down arrow
HELP KEY  (ESC-~):

----------------------------------------------------------------------------

Alternate Response Codes
F0  (ESC-J):                              F7  (ESC-Q): ^@'
F1  (ESC-S): ^@!                          F8  (ESC-R): ^@(
F2  (ESC-T): ^@"                          F9  (ESC-OI): ^@)
F3  (ESC-U): ^@#                          F10 (ESC-OJ): ^@*
F4  (ESC-V): ^@$                          F11 (ESC-OK): ^@+
F5  (ESC-W): ^@%                          F12 (ESC-OL): ^@,
F6  (ESC-P): ^@&

The codes above emulate the PC control function keys.
----------------------------------------------------------------------------

SHIFT-F0  (ESC-E):                        SHIFT-F7  (ESC-1G): ^@7
SHIFT-F1  (ESC-1A): ^@1                   SHIFT-F8  (ESC-1H): ^@8
SHIFT-F2  (ESC-1B): ^@2                   SHIFT-F9  (ESC-1I): ^@9
SHIFT-F3  (ESC-1C): ^@3                   SHIFT-F10 (ESC-1J): ^@:
SHIFT-F4  (ESC-1D): ^@4                   SHIFT-F11 (ESC-1K): ^@;
SHIFT-F5  (ESC-1E): ^@5                   SHIFT-F12 (ESC-1L): ^@<
SHIFT-F6  (ESC-1F): ^@6

The codes above emulate the PC shift control function keys.
----------------------------------------------------------------------------

I CHR  (ESC-@): ^@g               I CHR and keypad 7 emulate ^HOME
D CHR  (ESC-N): ^@i               D CHR and keypad 9 emulate ^PgUp
INS LINE  (ESC-L): ^@o            INS LINE and keypad 1 emulate ^END
DEL LINE  (ESC-M): ^@q            DEL LINE and keypad 3 emulate ^PgDn
HOME  (ESC-H):
BACK ARROW  (ESC-D): ^@k          Back Arrow and keypad 4 emulate ^left arrow
FOWARD ARROW  (ESC-C): ^@m        Fwd Arrow and keypad 6 emulate ^right arrow
UP ARROW  (ESC-A): ^@h            Up Arrow and keypad 8 emulate ^up arrow
DOWN ARROW  (ESC-B): ^@p          Down Arrow and keypad 2 emulate ^down arrow
HELP KEY  (ESC-~):
```

Table 4

things could become pretty confusing if you get "creative" with your original definitions.

The Z-100 has a few more function keys available than the boring blue machines. I've assigned these extra keys to a consecutive group of codes that Sprint recognizes but leaves available for you to assign commands to.

Assuming that you'll be using the Borland User Interface, you'll find that many commands have two or more keys assigned to invoke them. ^C and PgDn (keypad 3) are equivalent. All variations of F10, F11 and F12 bring up Sprint's main menu. Keep this in mind and you'll find plenty of ways to customize Sprint. I'll show you how easy it is in a moment.

After you've created and installed your new KEYMAP program there's one more final touch before running Sprint. Find the file BORDERS.SPM. It's hidden away on Disk 3 in the file SPMSOURC .ARC. You'll have to unpack the file with UNPACK.COM found on the same disk. Put both files on a floppy and give the command UNPACK SPMSOURC. All the .SPM files that emerge are the source code for the Borland User Interface. Put BORDERS.SPM on your program disk.

The Acid Test

Now it's finally time to run Sprint. SP<RET> and you're in business. F10 will bring up Sprint's main menu. Ugly, isn't it? Let's fix it. Select **Utilities**, then **Macros**, **Load**, **Borders** and finally **non-IBM**. Now you've got decent looking menus and they'll stay that way.

Next, you might want to try **Customize** and **Colors**. If my **set** command in MAIN.SPL had worked properly, you could spend all day juggling colors around. Even as they stand, you've got enough combinations to keep you busy.

One final trick to show how easy it is to modify Sprint. Enter a sentence or two and delete a word with ^T. Now go to the main menu with F10 and choose **Edit**. The first function displayed is **Undelete**. You'll notice that this command, as useful as it is, hasn't been bound to a key.

You could press <RET> now and undelete the word, but I'm sure you don't want to wade through menus each time you want to recover a deletion. Instead of pressing <RET>, press ^J (or Line Feed, they're the same). The status line will prompt you to enter a "shortcut" key. I chose the Z-100 HELP key. Go ahead and press HELP, it's as good as any. From now until you change it, the undelete command will be bound to the HELP key. What could be easier?

Is Sprint Really for You?

Sprint does have its faults. Mouse control is abysmally slow on anything less than a 16 MHz AT, but what do mice have to do with editing? The spelling checker has some bugs, but they're mere annoyances, not serious. Reading and writing files larger than 64k is too slow. However, once these large files are loaded into Sprint's swap file, accessing them is sudden. Merging three-up labels may be impossible. These are the only drawbacks that I've found.

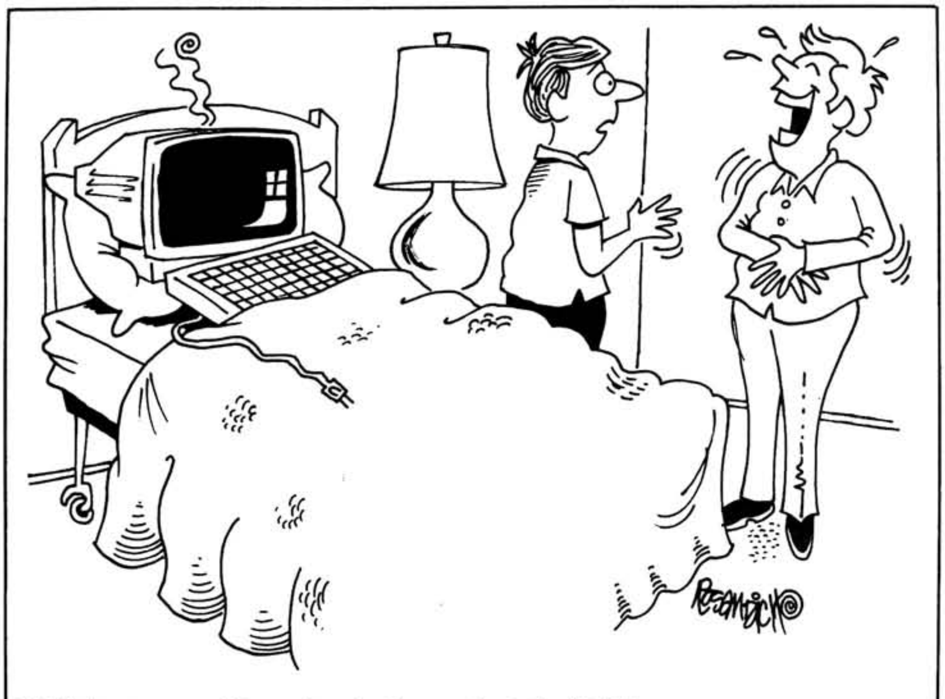
The Borland User Interface, although the most powerful of the nine user interfaces offered, doesn't match the power of a good programmer's editor, such as Brief, Epsilon or my own favorite, vi — not as it comes out of the box. However, all the tools are there for you to add your own features or change any feature that irritates you.

This is not to say that Sprint is a mere collection of tools, an erector set that must be assembled. Right out of the box Sprint offers at least as many easy to access whistles and bells as any word processor aimed at the mass market.

If you're running a system with only two 360K floppy drives, Sprint is not for you, it's just too big. Most of today's best software falls into this category and hard disk prices for the Z-100 are out of sight. However, you needn't be left out in the cold.

Full-height 8" floppy drives are becoming available at reasonable prices. Another relatively inexpensive alternative is to run a 5-1/4" 1.2 megabyte drive off your 8" controller. See George Elwood's "Upgrade Kits Available for the Z-100" in the October 1988 issue of REMark. One megabyte of disk storage will transform your machine and only put a moderate dent in your wallet.

In conclusion, if you work with words, Sprint is the most dramatic improvement you can make to your Z-100 . . . well a hard disk *would* be nicer. Its moderate price, $150 or less if you shop, makes it a bargain compared to the competition. In fact, a careful comparison shows that Sprint really has no competition among the mass marketed word processors. And it runs on our Z-100s! ✱



"WHAT'S SO FUNNY? IT HAS A VIRUS, DOESN'T IT?"

CLASSIFIED ADS

*** ZENITH PC COMPUTER UPGRADES ***

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

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

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

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

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

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

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

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

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

*** IBM COMPATIBLE SOFTWARE ***

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

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

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

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

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

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

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

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

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

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

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

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

*** Z-100 SERIES SOFTWARE ***

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

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

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

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

*** FLOPPY DISK DRIVES ***

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

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

* PAYLOAD CUSTOM ASSEMBLED COMPUTERS *

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

*** VIDEO MONITORS ***

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

*** VIDEO CARDS ***

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

*** MURATA FAX MACHINES ***

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

*** MEMORY CHIPS, ECT. ***

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

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

DISCOVER
MasterCard
VISA

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

PAYLOAD
15718 SYLVAN LAKE
HOUSTON, TEXAS
77062

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.

Reader Service #117

Getting Started With . . .

Pat Swayne
HUG Software Engineer

ASSEMBLY LANGUAGE

Part 1 Fundamentals

Way back in April of 1983, I started a series of articles on "Getting Started with Assembly Language". Those articles were relatively popular, and I have been urged to do the series again. I can't just reprint the old articles, because back in those days we had different computers with different processors in them. One of the drawbacks (as I will discuss later) of assembly language is that it is processor dependent, which means that a program that is written for one kind of processor (for example, the 68000 used in a Macintosh computer) will not run on another processor (such as the 80286 used in a SuperSport 286). However, the processors used in the various computer models currently made by Zenith and Heath are compatible with each other, and programs can be written in assembly language that will run on any of them.

Another difference between this series and the original one is that I intend to introduce more fundamentals. I assumed in the old series that the reader already knew some fundamentals of assembly language. We had run some fundamental articles way back in 1981, and I referred him back to them if he was lost. In this series, I will assume only that you know how to use the MS-DOS operating system, and that you can write simple programs in BASIC. I will not go as deep into the fundamentals as a book on the 8088 processor, for example, does, but hopefully I will present enough for you to get started.

Elements of a Computer Language

Any computer language, whether it is BASIC or Assembler, is a way of giving instructions to a computer. You would have to program your computer using binary numbers if you did not have some kind of computer language, because that is all it actually understands. The language takes your instructions, expressed in a form that you can understand, and changes them into a form that the computer can understand.

The instructions that are available for you to use in any computer language (sometimes called the "instruction set") can be divided into the following categories:

Input/Output — These instructions allow the computer to gather information from the "outside" world, and to send information to it. Some input/output instructions in BASIC are PRINT and INPUT, GET and PUT.

Data Transfer — Much of the work of a computer consists of transferring data from one place to another. Data transfer is done so much in a BASIC program that the main transfer instruction, LET, does not have to be expressed. If I enter LET A = 10, or just A = 10, I have entered an instruction to transfer the number 10 from the program itself to a holding cell called A.

Arithmetic — Just about any kind of program you might write will involve some kind of arithmetic, whether the job the program is doing involves arithmetic or not. For example, a word processing program needs to add up how many characters you have typed on a line before it can perform a "word wrap" operation. In BASIC, mathematical operations are done with algebraic expressions, such as TOTAL = A + B.

Comparison and Logic — These instructions are actually the most important group in any computer language. The ability to make comparisons is what sets the computer apart from all other "calculating machines". A computer can use comparisons to make decisions. Here is an example of a comparison in BASIC.

```
10 IF BALANCE < 0 THEN PRINT "You are overdrawn."
```

The computer "decides" whether to print "You are overdrawn" depending on the result of the comparison. Comparison and logic instructions are grouped together, because they are both sometimes used in decision making. Consider this BASIC example:

```
10 IF (AGE < 8) OR (HEIGHT < 48) THEN PRINT "Go away kid."
```

Branch — Branch instructions are used to transfer control from one part of a program to another. They are often used with comparison instructions, as in IF A > B THEN 20.

Machine Control — These instructions are rarely used in languages other than assembly. They will be explained later.

These categories actually apply specifically to assembly language, and I have taken the liberty to apply them to other languages, to (hopefully) help you understand assembly better.

Levels of Languages

As I stated previously, a computer works internally using binary numbers, and you would have to use binary numbers to program it if you did not have some kind of computer language. Most computer languages, such as BASIC are a type called "high level languages". Their instructions bear no resemblance to the instructions used by the computer itself, but are patterned after instructions that one person would give to another person. Consider the line

```
10 PRINT "HELLO THERE"
```

There is no instruction called "PRINT" in your computer's instruction set. In order to put the message "HELLO THERE" on your screen, your computer has to use several of its instructions to get each character of the message from where it is stored, and output those characters to the screen. Your BASIC interpreter translates the BASIC instruction PRINT into machine instructions that can do the job of displaying the message on the screen. (Actually, a BASIC interpreter does not translate an instruction, but it executes a subroutine within the interpreter itself that performs the function of the instruction. A BASIC compiler, on the other hand, can actually translate a program by combining subroutines into a program that can run by itself.)

Assembly language is a "low level language". Its instruction set consists of "words" that represent the computers instruction set. These words are usually an abbreviation of English words that describe the instruction. For example, a data transfer instruction that moves data from one location to another is called MOV.

A low level language is one whose instruction set represents or is closely related to a computer's instruction set. A high level language is one whose instruction set resembles human instructions.

Why Assembly Language?

Obviously, it should be easier to write a program in a high level language than to write it in a low level language. So why should anyone bother with assembly language? The answer is efficiency. Programs written in assembly language are generally faster and much smaller (use less memory) than programs that do the same, but were written in a high level language. To help you see how efficient assembly language programming can be, let's look at some of the inefficiencies built into high level languages.

When you compile something like

```
10 PRINT "HELLO THERE"
```

the compiled result contains a subroutine that can execute the BASIC instruction PRINT. This subroutine comes from a library of subroutines containing computer instructions that can execute all of the BASIC instructions. It uses the same subroutine for all the PRINT instructions that a program might contain. As it turns out, however, there are several ways to put a message on a computer screen. Some ways are faster, and some are "tighter" (they use fewer CPU instructions and thus less memory). When you write in assembly language, you can use the way that is best for each situation.

One of the worst inefficiencies built into many high level languages is that they attach a fixed library of basic subroutines to every compiled program. This library almost always contains code that is not used by the program, and therefore, just takes up memory space. To see just how inefficient this is, consider that you can write a program in assembly language to print a message on the screen that assembles to 9 bytes, plus the length of the message. (We'll show you how in a future episode.) A program to print HELLO THERE would, therefore, take 23 bytes (the message includes CR and LF characters, and a terminator). When I assembled

```
10 PRINT "HELLO THERE"
```

using QuickBASIC version 2.01, and I used the /O switch to create a stand-alone program that would run by itself, the result was a program containing 24,877 bytes! There is a lot more code in that file than is required to print a message on the screen! When I compiled the program without the /O switch, the result was a much smaller 3,453 byte program, but programs compiled without the /O switch will not run by themselves. They require the BRUN utility, which is 69,454 bytes!

Well, you might say, BASIC is not the most efficient high level language around. What about Pascal? A program to say HELLO THERE in Pascal looks like this:

```
PROGRAM HELLO;
BEGIN
WRITELN ('HELLO THERE')
END.
```

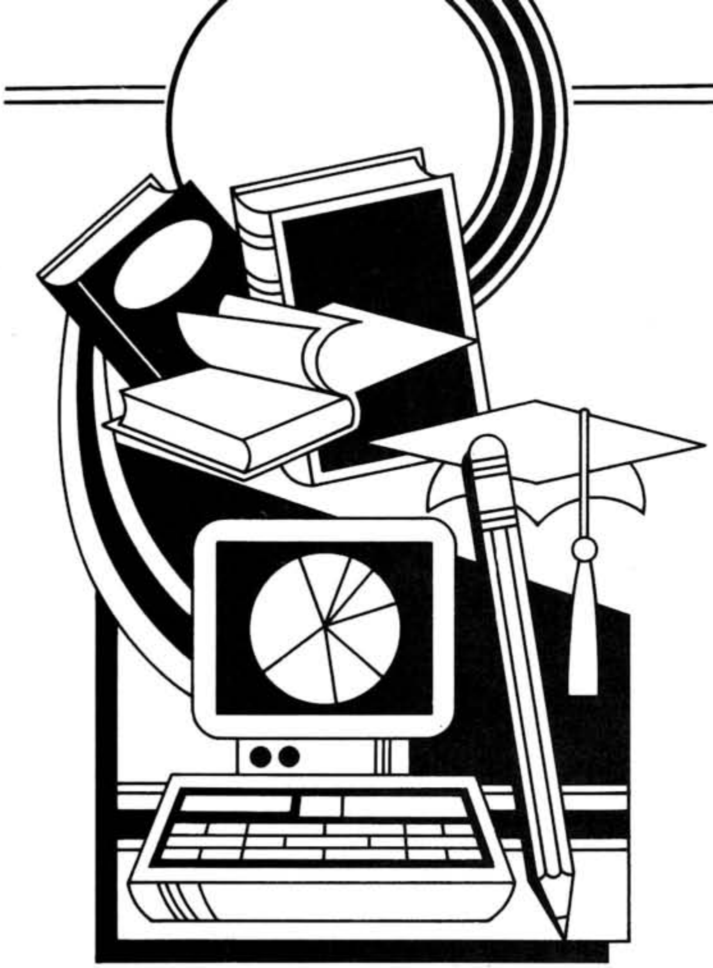
When you compile this with Turbo Pascal version 3.01, you get a program containing 11,457 bytes! Once again, there is a lot more there than is required to put a message on the screen.

Another inefficiency built into most high level languages has to do with the way data is passed around in the program. For example, if the main program must call a subroutine, it usually must pass information to the subroutine or get information from it. The most efficient way to pass a few bytes of information is to use the CPU's "registers" (special memory cells inside the CPU itself). A program compiled in a high level language usually uses the "stack" (part of the system's memory outside the CPU), because the compiler is not "smart" enough to know which registers might have to be used by a subroutine. It is always faster, and sometimes it saves memory, to use registers to pass information. The difference in speed is not noticable in human terms for a few subroutine calls, but a program may do thousands of calls while executing a loop, and the difference can become noticable. An assembly language programmer can use the best method for passing information in each case.

Because of its complexity, assembly language is not the best language for every programming job. However, there are some jobs that simply cannot be done using any other language. Consider my program HEPCAT (HUG disk no. 885-3045). HEPCAT is a pop-up calculator that provides a computer replacement for the calculator you may have on your desk. In addition to the basic mathematical functions, it can do trig functions, square roots, and other "transcendental" functions. Since HEPCAT is a memory resident pop-up utility, it was important to keep it as small as possible. It uses less than 16k of memory, and the part of the program that does the actual calculations (while it is in the floating point mode) uses only slightly more than 4k of memory. It would be impossible to write a program that does what HEPCAT does using any high level language. I am making that statement even though I am not familiar with every high level language there is. However, I challenge anyone to prove me wrong.

Next Time . . .

In this article, I have introduced the concept of different levels of computer languages, and pointed out some of the inefficiencies of high level languages. In the next installment, I will take you on a tour through a Central Processing Unit, and introduce the assembly language instruction set.



BETTER WRITING

GEORGE ELWOOD
1670 N. LADDIE COURT
BEAVERCREEK, OH 45432

We all write, for better or worst. At times worsers than better. Yo'all do the best that you can but still nothing comes out right.

Hopefully, you do better than the first few sentences in this article by planning your writing in great detail and with a lot of thought. You write at a level that is consistent with your audience. You make sure that the nouns and verbs are in agreement. Most of your writing is done by you in the active voice. Your sentences are short and to the point without a lot of extra words to confuse the reader and lose their train of thought.

While I know you write better than the first paragraph, the second paragraph may come closer to home. Writing is not easy and it does take some effort on your part to make it correct. To help you, two grammar checking programs, RightWriter 3 and Grammatik III, are available. Both of these programs use a set of rules to check your writing and make suggestions on how to improve it. They check using a basic set of artificial rules. If you do a lot of writing, these programs may provide you with the edge in getting an excellent product. These products will not change your document, they will only suggest changes.

Both of these products list for under $100. You can find them in discount software houses for less. RightWriter has a generic mode which will permit it to run on the Z-100. It will not display the screen, but it will generate the same output. RightWriter will run on the Z-100 with ZPC without patching and it will display all the PC screens.

These two products work in different ways. RightWriter 3 will produce a marked up copy of the document with the errors and suggestion inserted. The output from the program is either a file that is compatible with your word processor or an ASCII file. The file has the same name but with an extension of OUT. It also outputs a summary of the document it checked showing the reading level and other points. Figure 1 shows the output for paragraph two.

```
Hopefully you do better than the first few sentences in this
          ^<<* S9. WEAK SENTENCE START: Hopefully *>>
article by planning your writing in great detail and with a lot
        ^<<* U11. WORDY: in this article *>>
      <<* U9. IS THIS JUSTIFIED? great *>>^
of thought.  You write at a level that is consistent with your
            ^<<* G3. SPLIT INTO 2 SENTENCES? *>>
audience.  You make sure that the nouns and verbs are in
agreement.  Most of your writing is done by you in the active
   <<* S5. USE VERB FORM. REPLACE are in agreement BY agree *>>
voice.  Your sentences are short and to the point without a lot
of extra words to confuse the reader and lose their train of
thought.
```

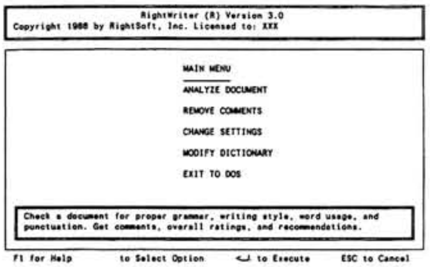
Figure 1
RightWriter Marked Up Paragraph

RightWriter 3 installs on your system by copying the files from the two disks. The first contains the basic program and the rules. The second disk contains a dictionary. The first time you run the program it asks you for your name, which is displayed on the top of all screens. It also asks for the type monitor you are using, color, monochrome, or not PC compatible. This is where the Z-100 users would select Not PC compatible.

For Z-100 users, you must start the program and type the file you wish to check. The Non PC Compatible mode does not permit you to enter files from the menu. You would simply type RIGHT FILE NAME including extension and location of file and the path. An example would be, RIGHT E:\EN200Z\WRITE.WPF. The output marked up file is written on the same drive as the file unless you say otherwise.

While the program is running, the screen displays a counter showing you the number of words checked. This is not displayed if you are using the Z-100 without ZPC. You will have to wait until it completes the check. I have run one very long document (200K) that took ten minutes to run on the Z-100. After the check is complete you can load the OUT file and fix the suggested problem areas. RightWriter places the sentence comments on the next line enclosed in double greater than and less than signs. If the problem is a word or phrase, there is a caret pointing to the part that needs attention. Working with ENABLE, I have the basic document in one window and the OUT file in another. I will then move between the windows, correcting the basic file.

RightWriter 3 supports most popular word processors. It automatically selects most word processors. There are two types of word processors, fully compatible and others. The fully compatible systems are: Bank Street Writer, Edix/Wordix, IBM Writing Assistant, Mircosoft Word, MultiMate, NewWord, OfficeWriter, PC-Write, PeachText 5000, pfs:First Choice, pfs:Professional Write, pfs:Write, Volkswriter, WordPerfect, WordStar, XyWrite. If you use these pacakges, RightWriter will generate a compatible output file. RightWriter recognizes these word processors and converts them to an ASCII format: ENABLE, Leading Edge, Microsoft Windows Write, Samna Word, and WordStar 2000. If you have another word processor, you must convert the file to an ASCII format.

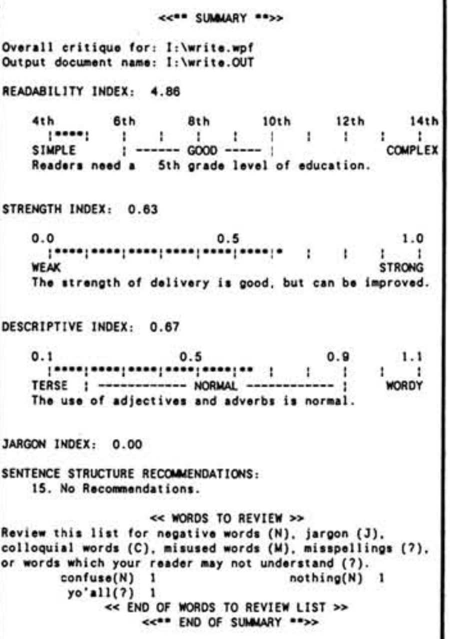


```
                    RightWriter (R) Version 3.0
Copyright 1988 by RightSoft, Inc. Licensed to: XXX

                         MAIN MENU

                         ANALYZE DOCUMENT

                         REMOVE COMMENTS

                         CHANGE SETTINGS

                         MODIFY DICTIONARY

                         EXIT TO DOS

  Check a document for proper grammar, writing style, word usage, and
  punctuation. Get comments, overall ratings, and recommendations.

F1 for Help        to Select Option      ↵ to Execute      ESC to Cancel
```

Figure 2
RightWriter Opening Screen

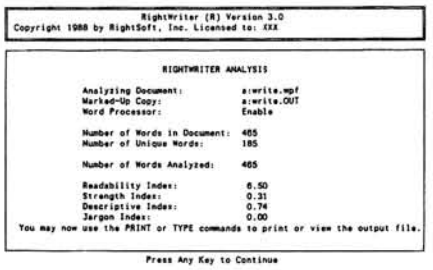
At the end of the marked up document is a summary. This section of the file provides you with information on the reading level, readability, strength, a descriptive index, and a list of words to check.

The Readability Index will tell you what level the document is written for. Most documents should be written for between 6th and 10th grade level. You should avoid higher levels. RightWriter uses three formulas to compute the readability index, the Flesch-Kincaid, Flesch, and the Fog. The Flesch-Kincaid is the formula used by the Department of Defense for their publications.



```
                         <<** SUMMARY **>>

Overall critique for: I:\write.wpf
Output document name: I:\write.OUT

READABILITY INDEX:  4.86

    4th        6th       8th       10th      12th      14th
      |****|    |    |    |    |    |    |    |    |    |
    SIMPLE      | ------ GOOD ----- |                 COMPLEX
    Readers need a  5th grade level of education.


STRENGTH INDEX:  0.63

    0.0                       0.5                      1.0
      |****|****|****|****|****|****|*   |    |    |    |
    WEAK                                             STRONG
    The strength of delivery is good, but can be improved.


DESCRIPTIVE INDEX:  0.67

    0.1                  0.5                  0.9      1.1
      |****|****|****|****|****|**  |    |    |    |    |
    TERSE  | ----------- NORMAL ----------- |        WORDY
    The use of adjectives and adverbs is normal.


JARGON INDEX:  0.00

SENTENCE STRUCTURE RECOMMENDATIONS:
    15. No Recommendations.

                        << WORDS TO REVIEW >>
Review this list for negative words (N), jargon (J),
colloquial words (C), misused words (M), misspellings (?),
or words which your reader may not understand (?).
        confuse(N)  1                    nothing(N)  1
         yo'all(?)  1
                << END OF WORDS TO REVIEW LIST >>
                     <<** END OF SUMMARY **>>
```

Figure 3
RightWriter Summary



```
                    RightWriter (R) Version 3.0
Copyright 1988 by RightSoft, Inc. Licensed to: XXX

                      RIGHTWRITER ANALYSIS

         Analyzing Document:            a:write.wpf
         Marked-Up Copy:                a:write.OUT
         Word Processor:                Enable

         Number of Words in Document:   465
         Number of Unique Words:        185

         Number of Words Analyzed:      465

         Readability Index:              6.50
         Strength Index:                 0.31
         Descriptive Index:              0.74
         Jargon Index:                   0.00
 You may now use the PRINT or TYPE commands to print or view the output file.

                      Press Any Key to Continue
```

Figure 4
RightWriter Summary Screen

The Strength index calculates the delivery of the document. This index reflects a wordy writing style. It also reflects the number of passive voice sentences, slang, cliches, and ambiguous wording. The descriptive index measures the use of adverbs and adjectives. Too many make the document hard to read. The jargon index reflects the use of nouns and verbs used in the opposite manner. It checks for abbreviations, and buzz words.

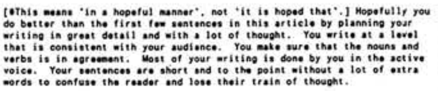
The second of these two programs is Grammatik III. This program works slightly different from RightWriter. You will normally run the file through the program, correcting as you go. Running the same file through both programs does result in some differences. The rules used are slightly different in this program.

Grammatik III comes on three disks and has an install program. The install program simply copies the files into a directory called GMK3. Loading on the Z-100 was interesting in that I did get the WILD INTERRUPT. Even though this appeared, the program continued to load. At the end it did bomb, but not until it was ready to run. It will not run in the native Z-100 mode. Using ZPC, it ran without a problem.

Grammatik III requires that you configure the system. You must select your word processor or you can use ASCII files. The word processors supported are: WordPerfect, WordStar, Microsoft Word, pfs:Write, pfs:Professional Write, pfs:First Choice, Q&A Write, MuliMate, Bank Street Writer, Framework II, Freestyle, IBM Writing Assistant, XYWrite, Easy, Edix/Wordix, Electric Desk, MS Windows Write, NewWord, Palantir, PC Write, Spellbinder, Superwriter, Ventura Publisher, Volkwriter, WordStar 2000, Microsoft Works, DisplayWrite, Office Writer, WordMARC, Sprint, and Byline. If you use ENABLE, you must save the file as an ASCII file or one of the other supported file types, like WordStar.

You can also set many other parameters with the system. You can turn on or off different checking points. The dictionary is one of the things you can turn on. Once you select the file top check, Grammatik III will check the document, stopping at every error. It will give you a short description of the error and what needs to be done. You can correct the error at this time or mark it for later correction. At the end of the document is a summary with the Readability, paragraph, sentence, and word statistics. Unlike RightWriter, Grammatik shows individual readability for the Flesch, Fog, and Flesch-Kincaid levels.

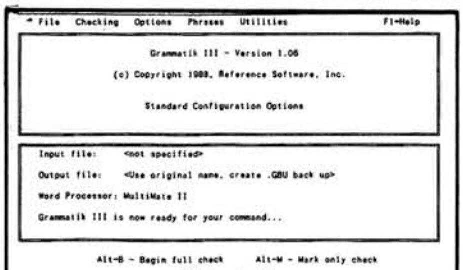
The paragraph statistics displays the number of paragraphs and the average length. The sentence statistics show the number of sentences, average number of words and endings.



```
[#This means 'in a hopeful manner', not 'it is hoped that'.] Hopefully you
do better than the first few sentences in this article by planning your
writing in great detail and with a lot of thought.  You write at a level
that is consistent with your audience.  You make sure that the nouns and
verbs is in agreement.  Most of your writing is done by you in the active
voice.  Your sentences are short and to the point without a lot of extra
words to confuse the reader and lose their train of thought.
```

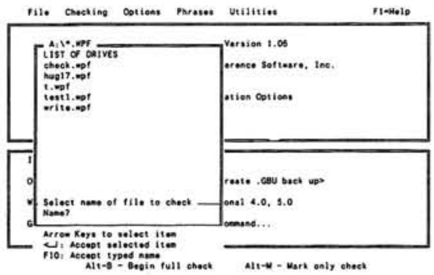
Figure 5
Grammatik III Marked Up Paragraph

One of the options that can be turned on is the Marked Up copy. Grammatik will generate a marked up copy of the file. The marking consists of a bracketed statement about the problem. A pound sign (#) is used to indicate the problem. As an option, you can also output a summary file. This file consists of all



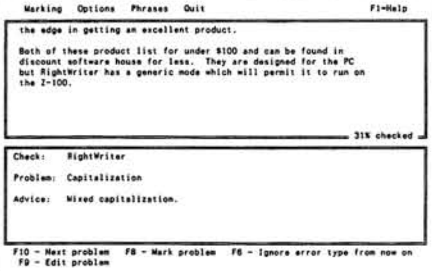
```
 * File   Checking   Options   Phrases   Utilities                  F1-Help

                     Grammatik III - Version 1.06

               (c) Copyright 1988, Reference Software, Inc.

                    Standard Configuration Options

   Input file:      <not specified>

   Output file:     <Use original name, create .GBU back up>

   Word Processor: MultiMate II

   Grammatik III is now ready for your command...

             Alt-B - Begin full check      Alt-M - Mark only check
```

Figure 6
Grammatik Opening Screen



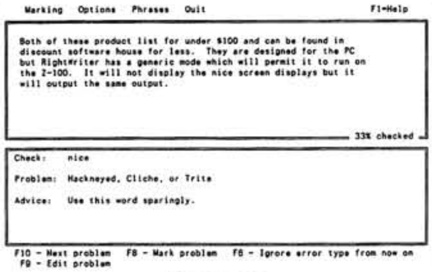
```
File   Checking   Options   Phrases   Utilities                    F1=Help

  A:\*.WPF                                Version 1.05
  LIST OF DRIVES
  check.wpf                               erence Software, Inc.
  hug17.wpf
  t.wpf
  test1.wpf                               ation Options
  write.wpf

I

O                                         reate .GBU back up>

W Select name of file to check           onal 4.0, 5.0
  Name?
G                                         ommand...
  Arrow Keys to select item
  <┘: Accept selected item
  F10: Accept typed name
          Alt-B - Begin full check      Alt-M - Mark only check
```

Figure 7
Grammatik File Selection Window



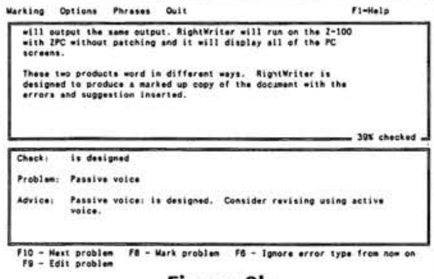
```
Marking   Options   Phrases   Quit                              F1=Help

 the edge in getting an excellent product.

 Both of these product list for under $100 and can be found in
 discount software house for less.  They are designed for the PC
 but RightWriter has a generic mode which will permit it to run on
 the Z-100.

                                                         31% checked

Check:    RightWriter

Problem:  Capitalization

Advice:   Mixed capitalization.

F10 - Next problem   F8 - Mark problem   F6 - Ignore error type from now on
 F9 - Edit problem
```

Figure 8
Correction Screen



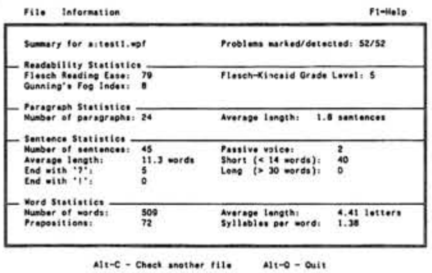
```
Marking   Options   Phrases   Quit                              F1=Help

 Both of these product list for under $100 and can be found in
 discount software house for less.  They are designed for the PC
 but RightWriter has a generic mode which will permit it to run on
 the Z-100.  It will not display the nice screen displays but it
 will output the same output.

                                                         33% checked

Check:    nice

Problem:  Hackneyed, Cliche, or Trite

Advice:   Use this word sparingly.

F10 - Next problem   F8 - Mark problem   F6 - Igrore error type from now on
 F9 - Edit problem
```

Figure 8a
Grammatik Correction Screen



```
Marking   Options   Phrases   Quit                              F1=Help

 will output the same output. RightWriter will run on the Z-100
 with ZPC without patching and it will display all of the PC
 screens.

 These two products word in different ways.  RightWriter is
 designed to produce a marked up copy of the document with the
 errors and suggestion inserted.

                                                         39% checked

Check:    is designed

Problem:  Passive voice

Advice:   Passive voice: is designed.  Consider revising using active
          voice.

F10 - Next problem   F8 - Mark problem   F6 - Ignore error type from now on
 F9 - Edit problem
```

Figure 8b
Grammatik Correction Screen



```
File   Information                                              F1=Help

  Summary for a:test1.wpf              Problems marked/detected: 52/52

— Readability Statistics
  Flesch Reading Ease:   79            Flesch-Kincaid Grade Level: 5
  Gunning's Fog Index:   8

— Paragraph Statistics
  Number of paragraphs: 24             Average length:   1.8 sentences

— Sentence Statistics
  Number of sentences:   45            Passive voice:          2
  Average length:        11.3 words    Short (< 14 words):     40
  End with '?':          5             Long  (> 30 words):     0
  End with '!':          0

— Word Statistics
  Number of words:       509           Average length:         4.41 letters
  Prepositions:          72            Syllables per word:     1.38

           Alt-C - Check another file      Alt-Q - Quit
```

Figure 9
Grammatik Summary Screen

```
Version 1.06 ------------------

Summary for i:\write.asc

Problems marked/detected: 106/106

Readability Statistics

  Flesch Reading Ease:           71
  Gunning's Fog Index:           10
  Flesch-Kincaid Grade Level:     7

Paragraph Statistics

  Number of paragraphs: 21
  Average length:       4.1 sentences

Sentence Statistics

  Number of sentences:  88
  Average length:       13.1 words
  End with '?':         5
  End with '!':         0
  Passive voice:        7
  Short (< 14 words):   56
  Long (> 30 words):    2

Word Statistics

  Number of words:      1153
  Prepositions:         113
  Average length:       4.59 letters
  Syllables per word:   1.45
```

Figure 10
Grammatik Printed Summary

of the problems listed one at a time. At the end of the summary are the statistics.

Both of these programs will help you in checking your writing. They will not change the text but will make suggestions. It is still up to you to revise the text. It is easier to correct if they have been flagged by these programs.

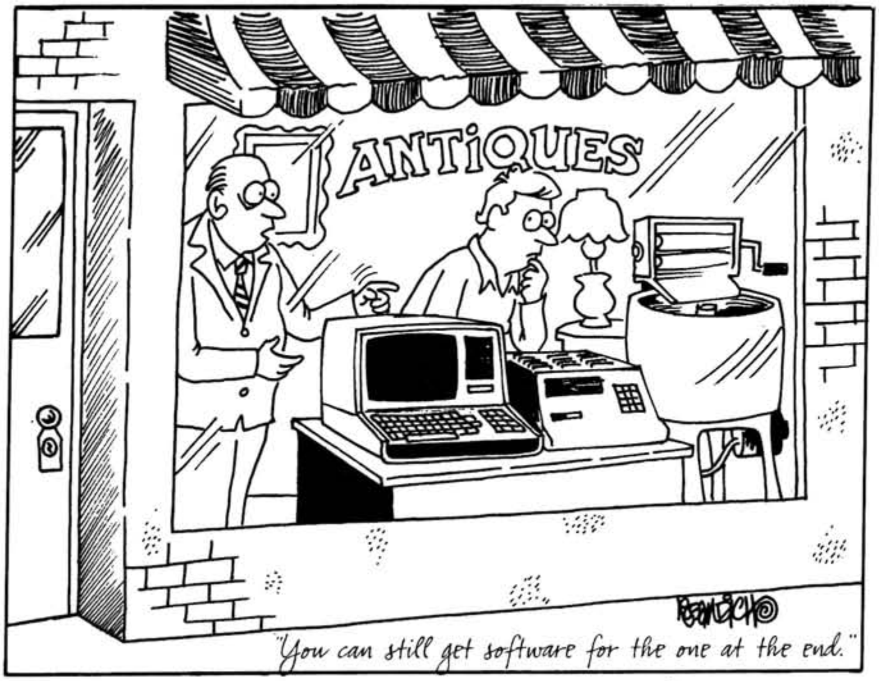
Your requirement will dictate the program you choose. If you want to work from an output file, use RightWriter 3. If you wish to correct the document on screen, choose Grammatik III. Happy writing.

Grammatik III $95.00
Reference Software
320 Townsend Street
Suite 123
San Francisco, CA 94107
(415) 541-0222

RightWriter 3 $95
RightSoft, Inc
4545 Samuel Street
Sarasota, FL 34233
(813) 923-0233

✱

CARTOON CONTEST!!

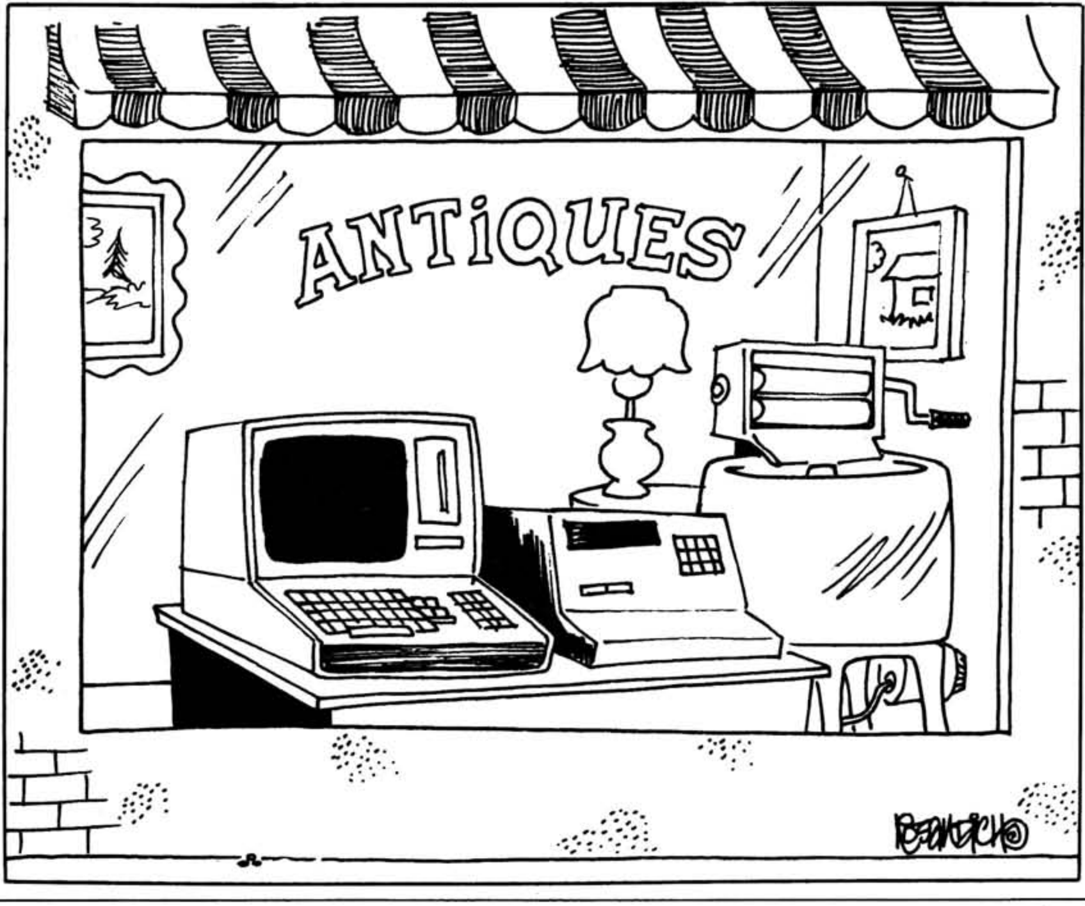


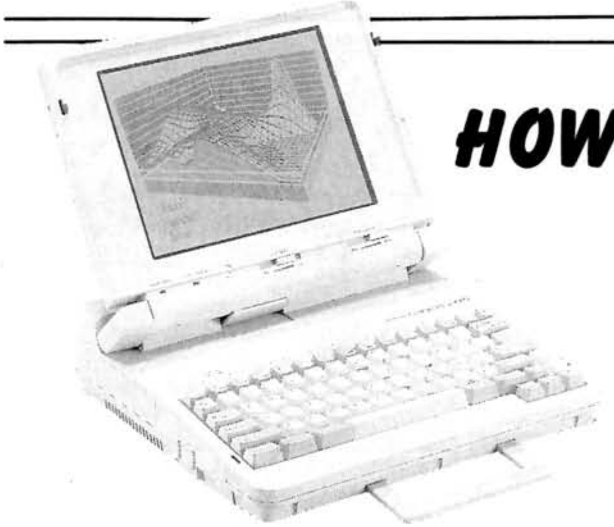
"You can still get software for the one at the end."

It's contest time again! This one is easy. Just write a funny caption for the cartoon at the bottom of the page. As you can see from the picture, Mr. Seymour Junque's store window has items for sale, but no advertising. It's been rumored by Mr. Junque that one night he thought he heard: "Your H8 is up and running . . . weep, weep".

We must receive your entry before February 1st, 1990. Are you ready for the prizes? The first place prize is a ZSW-184-2 SupersPort Laptop, which includes a 20 meg. hard disk. The second place winner will receive a ZSS-184-1, dual floppy drive SupersPort Laptop.

Entries will be judged upon the loudest laughs from the HUG staff. The decision of the judges are final. All entries become the property of the Heath Users' Group, and those in good taste, will be printed in future issues of REMark. Send all entries to: "Cartoon Contest", Heath Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217.





HOW TO CHANGE THE ROM IN THE SUPERSPORT 286

Changing the ROM in most Zenith desktop computers is usually fairly easy and straightforward. It is usually a matter of just removing the cabinet, finding the current ROM on a printed circuit board, replacing it (or them), and reassembling the computer. Changing the ROM in the Zenith laptop computers, like the SupersPort 286, is not quite so easy because of the compact design. In addition, it is not exactly obvious how to disassemble the SupersPort to even get at the ROMs, even if you know where they are. The SupersPort ROMs are physically located on the underside of the motherboard, and getting to them requires complete disassembly of the computer and removing virtually all of the parts from the case. Even if you don't have one of these laptops, you might find it useful to take a look at some of the photographs to see some of the new parts.

Why Replace the ROM?

In my "On the Leading Edge" column published in the September 1989 *REMark*, I mentioned that it was difficult to see the cursor when using some software. When I wrote about that problem, I was using ROM version 2.3D (the -1 part number). About the time the article was published, I learned that Zenith had a new ROM with specific enhancements to fix the problem of the "disappearing cursor." As of the time of this writing, the latest ROM version is 2.8F which has a "-3" part number (e.g., 444-671-3). If you order the parts from Heath Company at the toll-free number listed at the end of this article, they always ship the latest version which may be different than the one mentioned here. When I got the new ROMs, I learned very quickly that installing them was trickier than I had imagined. Even finding them was a little challenging because I did not know for sure where they were located. Fixing the problem of the disappearing cursor was replaced by the more immediate problems of how to disassemble the computer and locating the "disappearing ROMs."

Before You Begin

There are four things you should know before you attempt to replace the ROMs in the SupersPort. First, make a complete backup of all partitions on the hard drive if your SupersPort has one. Backing up all hard drive partitions is good practice whenever you change or add hardware or software to any computer system. A complete backup is your safety net in case something goes wrong with the installation, and this backup should be in addition to normally scheduled ones.

Second, use the Print Screen (PrtSc) key to make a copy of the SETUP screen. Use the CTRL-ALT-INS key sequence to activate the ROM Monitor, type *SETUP*, press RETURN, and print the screen. When the screen is printed, be sure to mark what choices are selected by the "highlights" because they do not show up on the printed version. For example, I circled the "EMS" value in the "Exp. memory" field so I would know exactly how my system was configured. This is important because you will need to run SETUP after the ROM change is completed because you have to disconnect the backup battery. When you disconnect the battery, all of the configuration information is "lost," just like any of the 80286 or 80386 systems. By the way, I strongly recommend that you always keep a "Print Screen" version of the SETUP screen for any of these computers so you will easily be able to configure the system when the backup battery eventually fails.

Third, if you have any doubts about replacing the ROMs in the SupersPort, don't even try it. Like all computers, the SupersPort 286 is a complicated piece of electronic equipment, and everything in the system must be absolutely perfect or the computer may not work reliably or at all.

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

CAUTION: If you have any doubts about your ability to disassemble your SupersPort 286 computer, *DON'T*. Although both HUG and the Author have used considerable care to verify the accuracy of this article, the steps listed describe the complete disassembly of the Author's Zenith SupersPort 286 and may not apply to later models. It is the user's responsibility to ensure that these steps apply to a specific Zenith SupersPort 286 computer. The Author, HUG, and Heath Company offer no warranty, either expressed or implied, and are not responsible for any losses due to the failure to follow the procedure or other use of information presented in this article.

Last, but not least, do *NOT* even attempt to replace the ROMs if your SupersPort 286 is less than a year old. Opening the shell will void the one-year warranty, and there are *NO* exceptions. If you want to replace the ROMs in a newer SupersPort, take the system to an authorized Zenith dealer, such as a Heath/Zenith Computer Center, and have them do it.

WARNING: The following procedure requires major disassembly of the SupersPort 286 and will void the Zenith One-Year Warranty. Do not attempt to replace the ROMs if the computer is less than one year old.

The Tools You'll Need

The number of tools required to disassemble the SupersPort is minimal: a number 2 philips screwdriver and a small, flat blade screwdriver is all you'll need. A small piece of tape is also needed to mark one of the connectors that must be unplugged. Two hands are required, and you may find three or four helpful when you begin to reassemble the computer. I also found it helpful to have a few plastic dishes to place small parts in as they are removed. That is extremely important because you will need to keep track of which screws fit into which holes. But before you begin, one caution is in order.

Although the disassembly and assembly are easy once you know how, it is important to take your time. Plan on spending at least an hour or so, but do not try to beat the clock. It took me nearly four hours to perform this process, but that includes setting up for the photographs and making a lot of notes during the disassembly.

Do NOT pull on or try to forcefully remove anything that seems to be "stuck." Read the discussion of that particular step again to be sure you have removed all of the screws holding that particular part. Be gentle, but firm, when you go through these steps, and when in doubt, be gentle. When some particular step requires more force than you might expect (e.g., removing a connector), it will be specifically mentioned in that step. Keep a count of the screws and parts you have removed, and check them. Be sure you have plenty of work space, and get a piece of cardboard from an old box or use an old rug to place the computer on so the cabinet will not be scratched. To be sure that you understand the general steps that follow, be sure to read this entire article before you start. And if you decide to disassemble your SupersPort 286, be sure to follow the exact steps discussed in this article because some things must be done in a specific order.

Disassembling the SupersPort 286

Be sure that you have powered off the computer, disconnected the external power supply, and removed the external battery pack.

1. **Disassemble the cabinet:** Turn the computer upside down (on the cardboard or a small rug) with the handle facing you, as shown in Photo 1.

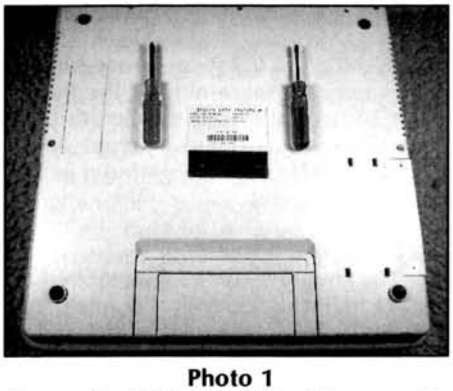


Photo 1
SupersPort 286 Ready for Disassembly

Photo 1 also shows the two small screwdrivers that I used for the disassembly. Using the Philips screwdriver, remove 2 screws in the "holes" on the back, and remove 3 screws in the holes on the front. In the middle on each side, there is a flat-head philips screw — remove both of them. Note that the flat-head screw on the right side (near the modem cover plate) is shorter than the left one. You should have removed a total of SEVEN SCREWS during this step. There are two other flat-head philips screws that hold a cover for the internal modem, and these screws should NOT be removed.

Grasp the *entire* computer with both hands and turn it over with the handle facing you. The top half of the cabinet should be loose, and it is very important to keep the cabinet together as you turn the unit over.

2. **Remove the LCD:** Carefully lift up the top half of the cabinet straight up, and you will find that it is "connected" to the bottom half with a cable and pin connector for the LCD screen, as shown in Photo 2.

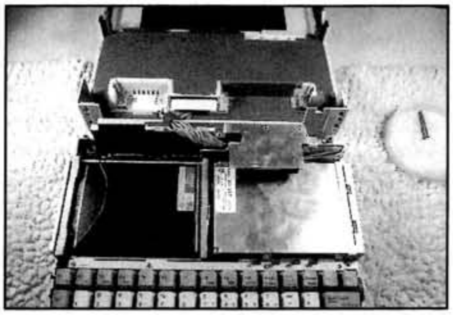


Photo 2
Disconnecting the LCD Cable

Before you remove the LCD connector, place a small piece of tape on the LEFT side of it so you can tell how to reinstall it later. This is an important step because the connector is not polarized, and it can be installed incorrectly. Grasp the top half of the plastic connector firmly, and pull it straight up. The connector may be quite firmly attached, so some force may be necessary to remove it. Be sure that you pull straight up on the connector, *not* the cables. I found that inserting a fingernail between the connector and the plug is an easy way to loosen the connector. Place the top half of the cabinet in a safe place so you will not inadvertently bump or push it off the work table. Note that Photo 2 shows a black ground wire for the keyboard that will be removed in the next step.

3. **Remove the Keyboard:** Removing the keyboard is next. Use the philips screwdriver to remove the two screws at the front of the computer. Photo 3 shows a screwdriver in the area at the bottom of the keyboard assembly where the two screws need to be removed. The plastic dish used to keep the small screws from getting lost or misplaced.

Photo 3
Removing the Keyboard Screws

After removing the two keyboard screws, note the black ground wire that extends from the upper left-hand corner of the keyboard to the back left corner of the hard drive (see Photo 2). Use a philips screwdriver to remove this one screw and place it aside. I found it is easier to leave the screw in the terminal connector when you remove it.

Carefully lift up the keyboard and you will find it is attached to the motherboard with a flat cable, as shown in the center of Photo 4.

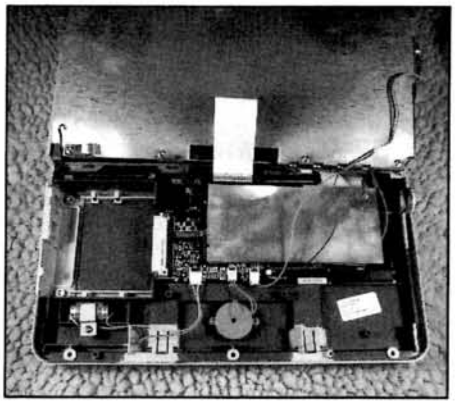


Photo 4
Removing the Keyboard Connectors

The flat cable is secured by a "locking" socket. Note that each side of the socket has a tab on it which must be lifted to release the lock. I used a fingernail to lift each side of the lock. It will move less than 1/8" when fully unlocked, and the flat cable will be released. When I removed the cable from the socket, I found that it had a blue "stripe" on it, as shown in Photo 4. The keyboard also has a two-wire connector running from the right-hand side of the keyboard to the motherboard, as shown in Photo 4. Grasp the connector firmly and pull it straight out (toward you) from the motherboard. In this step, you should have removed the two short screws (about 1/4") from the keyboard (Photo 3), the ground wire and longer screw (about 3/8"), a flat cable connector, and a two-wire connector. A total of THREE SCREWS were removed during this step.

4. **Remove the Hard Drive:** The next step is to remove the disk drive assemblies, starting with the one on the left. This SupersPort 286 has the usual 3.50" floppy drive on the right side of the cabinet and a 40 MB hard drive unit on the left side. I found it's easier to remove the hard drive unit on the left side first, even though the Service Manual (part number 860-197) suggests removing the floppy drive first.

Use the Philips screwdriver to remove the 2 remaining screws (located in about the middle of the motherboard) on the hard drive unit. I found it's easier to leave the 2 screws in the hard drive mounting holes when the drive is removed. Carefully lift up the "back" (or inside where the screws were removed) of the hard drive to get access to the drive cable. Carefully pull the drive cable connector out of the hard drive. The connector is usually tight, and this may take some time. Do *NOT* pull on the cable, just the connector. Set the hard drive assembly and the TWO SCREWS aside. Photo 5 shows the plastic cover plate and the disconnected hard drive.

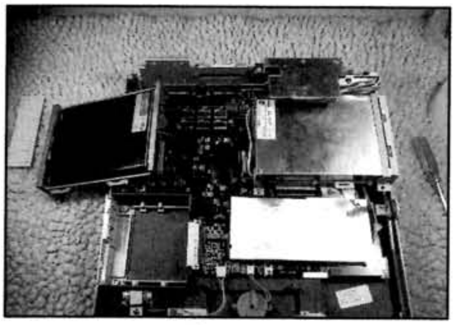


Photo 5
Removing the Hard Drive

5. **Remove the Floppy Drive:** For the floppy drive, disconnect the two-wire power cable from the unit, as shown in Photo 6.

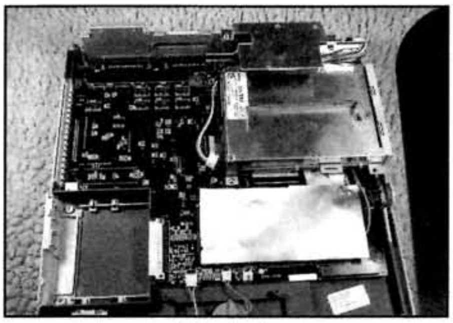


Photo 6
Disconnecting the Floppy Drive Power Cable

Use the Philips screwdriver to remove the one screw holding the bracket on the floppy drive in the back right corner of the computer, lift out the bracket and screw together, and set them aside. Remove the 2 remaining screws from the floppy drive mounting near the center of the motherboard (back of the floppy drive). Again, it's easier to leave the 2 screws in the floppy drive mounting holes when the drive is removed. Carefully lift up the "back" (or inside) of the floppy drive to get access to the flat drive cable. Pull the drive cable connector carefully out of the floppy drive. Like the hard drive, the cable connector is usually tight, and this may take some time. Do *NOT* pull on the cable, just the connector. Place the floppy drive assembly and the 2 screws near its bracket and 1 screw (TOTAL 3 SCREWS).

6. **Remove the Hard Drive Controller Screw/Spacer:** The next step is to remove the screw on the hard drive controller that is covered by a copper sheet near the front of the computer. It is attached to the cabinet by 1 Philips screw — remove it. A small plastic spacer is located underneath the board where the screw was removed; remove it and place it with the 1 screw.

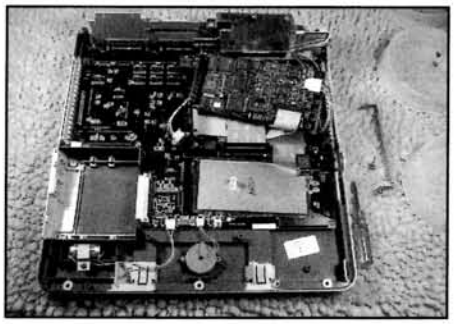


Photo 7
Removing the Hard Drive Controller

There is really no need to unplug the hard drive controller from the motherboard, but it is shown that way in Photo 7 for clarity only. Do not unplug this board, and other photos will show this board still in place as it should be. In this step, you should have removed only the ONE LONG SCREW and the plastic spacer. Have patience, you're almost there.

7. **Remove the Power Supply Module:** The next step is to remove the power supply module in the back right corner of the computer. Take a look at Photo 8 showing the metal bracket that has four screws on the power supply module.

There are four screws on the power supply module. Photo 8 only shows three of them on the top, and there is also a fourth screw located on the side of the power supply module as indicated by the screwdriver. Remove *ONLY* 1 screw on the extreme left of the bracket (on the top) and 1 screw on the extreme right of the bracket (on the side where the screwdriver is shown in Photo 8). You must also remove 1 screw located in the back right corner near the external power connector. A total of 3 screws were removed — two machine screws from the metal bracket and a self-tapping screw located near the external power connector. Be *SURE* you remove the correct screws in this step before you attempt to remove the power supply module. Disconnect the 4-wire cable that goes from the power supply module to the hard drive controller board shown in Photos 7 and 8.

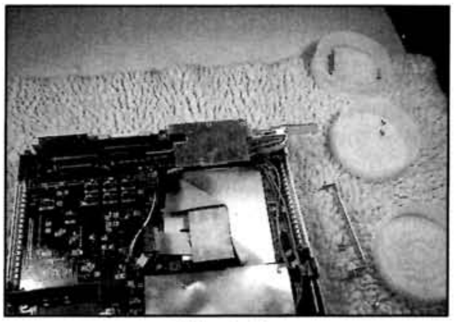


Photo 8
Removing the Power Supply Module

The power supply module is connected to the motherboard by two pin connectors. To remove the power supply module, grasp each side of the unit, and, while pulling straight up, gently rock the unit from side to side (not front to back). This step will take some time and a little force, and you may need a third hand to hold the remaining part of the computer in place while you are doing this. Note that the external power supply connect assembly also plugs into the power supply module. These units should not be separated. Use extreme care when removing this assembly, and note that the entire unit includes the power supply module, the external power supply connector, and the slide switch for the power supply. Set the power supply module aside.

8. **Remove the Motherboard:** Note that there are two remaining connectors on the front of the motherboard: one for the backup battery and one for the sound transducer. Unplug each connector by carefully grasping the plastic plug, and pull it straight out of its socket (towards you). Photo 9 shows how the computer should look at this point with these connectors unplugged.

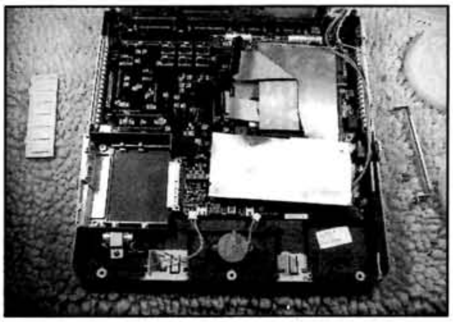


Photo 9
The Motherboard

There should be two remaining screws on the motherboard. One screw is located on the metal power supply bracket in the back right corner of the motherboard. The other screw is located on the plastic socket connector near the internal modem enclosure. Remove both screws. You should have removed TWO SCREWS in this step, and there should be no other screws holding the motherboard in place.

Now you can remove the motherboard. Grasp the motherboard on both sides, and lift the front up at about a 30-degree angle to disengage the connectors from the holes on the back of the cabinet. Gently lift the motherboard out of the cabinet. Move the now-empty cabinet away from your work area, and turn the motherboard upside down on the cardboard or rug. Be sure to keep the front of the motherboard nearest you. Now you are finally ready to replace the ROMs.

Removing the Old ROMs

The ROMs are located on the front of the motherboard as shown in Photo 10. They should be easy to identify because they have a label showing "Zenith Data Systems" with a copyright date as shown in the photo. Each ROM should also have a part number shown on the label. Note that part number 444-672 is located on the top and 444-671 is located on the bottom as shown in Photo 10.

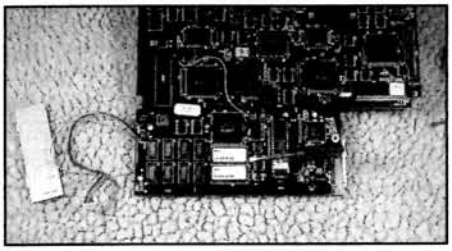


Photo 10
ROM Location and Part Numbers

9. **Remove the old ROMs:** Remove each ROM by carefully inserting the small *flat* blade screwdriver between the ROM and the motherboard and lifting up that end VERY slightly, as shown in Photo 10. Also, take a quick look at Photo 11, and note that these ROMs are installed in "holes" in the motherboard, rather than the usual socket. Then, carefully insert the screwdriver at the other end of the ROM (between the ROM and the motherboard), and lift that end up VERY slightly. Continue alternating between each end of the ROM until it is released from the "socket." Be very careful that you do not attempt to pry too many of the pins out of the "socket" at once because a pin may break off or you may damage the socket. Set the first ROM aside.

This procedure is not difficult, but it is important to be very careful while doing it. By the way, you can use this same trick to remove other *socketed* chips on a printed circuit board, but of course, you should not attempt it if the chips are soldered to the board.

Remove the second ROM in the same way as the first, and set it aside.

Inserting the New ROMs

Photo 11 shows the motherboard with the old ROMs removed and the new ROMs ready for installation.

10. **Install the new ROMs:** Again, note the absence of the standard IC socket — the "holes" for the ROM mounting are part of the motherboard. Also note that the ROMs are shown in the black, anti-static protective foam blocks.

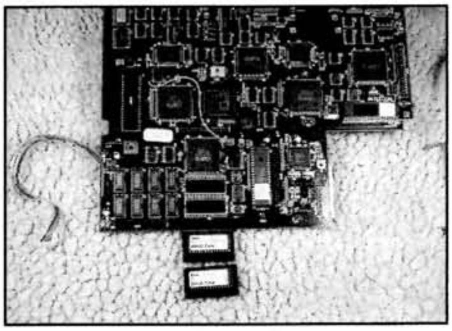


Photo 11
New ROMs Ready for Installation

Be careful with the new ROMs. All ICs — the usual RAM chips, ROMs, numeric coprocessors, Micro Processor Units (MPUs), etc. — are complex units that can be damaged or their contents destroyed by static electricity. When you remove an IC from the anti-static protective foam, do not let go of the IC until it is installed in the socket or inserted in the protective foam.

If you have ever built a Heathkit that uses an Integrated Circuit (IC), then you have undoubtedly seen the neat trick to use when installing one of these chips in a socket. This trick helps ease the difficulty of inserting a chip in a socket, and it minimizes the possibility that a pin will be bent underneath a chip during insertion. It is so important that it is also described in Chapter 6 (page 6-3) in the SupersPort 286 Owner's Manual in the context of how to install a numeric coprocessor. And it is especially worth mentioning if you have never done it before.

As you probably noticed when you removed the old ROMs, they are U-shaped with the pins extending perpendicular to the case. If you remove the new ROM from the black "foam" that helps protect the ROM against static electricity, you may find that the pins are not quite perpendicular to the case, as shown in Figure 6-2 in the Zenith Owner's Manual. As the SupersPort Owner's Manual suggests, you will probably need to "roll" both sides of each ROM on a flat surface to ensure that the pins are perpendicular to the ROM's case. Virtually all of the current Zenith manuals describe this trick for adding a coprocessor to a system.

Also be sure that the ROM pins are evenly spaced and that they are not bent at some strange angle that would prevent proper insertion into a socket. Check each ROM for correct pin alignment.

If you look closely at a ROM, you will see that one end contains an index mark that usually looks like a "notch." A numeric coprocessor also has a similar notch, as shown in Figure 6-3 on page 6-3 of the SupersPort 286 Owner's Manual. Note that the "socket" location on the motherboard also has a corresponding notch indicated on the left side of each ROM location. Be sure to look at Photo 10 to see which ROM part number is inserted into which socket.

Select the ROM part number 444-672 for insertion into the top "socket" on the motherboard, and gently lay it on top of its socket. Check and double-check all pins to be sure they are *INSIDE* the socket "holes." Then, place a thumb on each end of the ROM, and press it firmly into the socket while applying even pressure to both ends. When the ROM is completely seated in the socket, pick up the motherboard, and examine the rows of pins to be sure that a pin was not bent during insertion. Sometimes a flashlight is helpful in providing enough light to check the pins.

Using the same procedure, insert the remaining ROM part number 444-671 in the bottom socket. Now you are ready to assemble the computer.

Assembling the Computer

Putting everything back together is much easier since you basically know the steps, which will be reversed during assembly. Again, take your time, and don't try to force anything.

11. **Install the motherboard:** Locate the bottom half of the cabinet. Place it on the work surface (with the handle toward you), and gently place the motherboard inside it. Remember that you will have to angle the back of the motherboard into the cabinet. It is important to note that the L-shaped metal bracket tabs fit into (and between) the two plastic tabs on the back of the cabinet shell, as shown in Photo 12.

Install the long, self-tapping screw on the modem pin connector, as pointed out by the tip of the screwdriver on the left side of the motherboard. Install the 3/8" self-tapping screw on the right side of the metal bracket, as shown by the tip of the screwdriver on the right side of the motherboard. As with all self-tapping screws, be sure to tighten them firmly, but be careful that you don't strip any of the threads. Plug in the transducer and back-up battery connectors to the appropriate

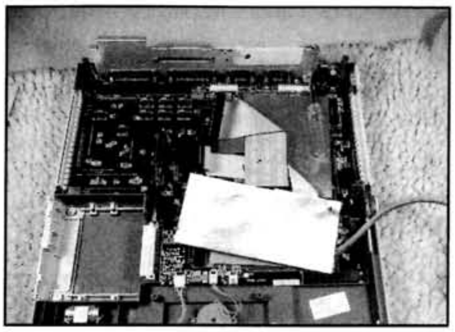


Photo 12
Installing the Motherboard

sockets on the motherboard.

12. **Install the power supply module:** Take a look at Photo 13 to see how the power supply module is installed.

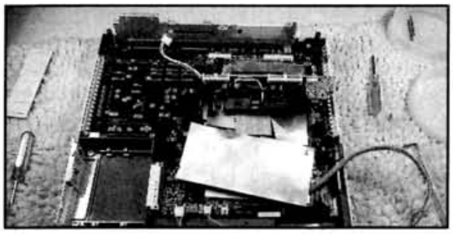


Photo 13
Installing the Power Supply Module

The power supply module is shown upside down so that it is easy to see how the connectors on the unit fit into the matching pin-plugs on the motherboard. Install the power supply module by carefully matching the two connectors. When the connectors match exactly, slowly press straight down on the metal bracket on top of the power supply, while being *sure* that the external power supply connector and the slide switch fit into the appropriate slots. Double check to make sure the power supply connector and the slide switch are correctly positioned. Then secure the printed circuit board by installing the 1 short, self-tapping screw in the back right corner of the computer. Then, install the 2 *machine screws*: 1 on the top, left side of the metal bracket and 1 on the right side of the metal bracket. Note that the 2 screws for the metal bracket are the only machine screws inside the computer — the others are all self-tapping.

13. **Install the Hard Drive Controller Screw/Spacer:** Install the hard drive controller board by sliding the small plastic spacer directly under the hole in the upper, right corner of the board, as shown in Figure 14.

Note that I have removed the hard drive controller board in Photo 14 for clarity, but it should *not* have been removed in a previous step. Also, plug in the 4-wire connector from the hard drive controller to the power supply module.

14. **Install the Floppy Drive:** Install the floppy drive by inserting 1 mounting screw in the back, left corner mounting bracket and carefully placing the drive in position, as shown in Photo 15.

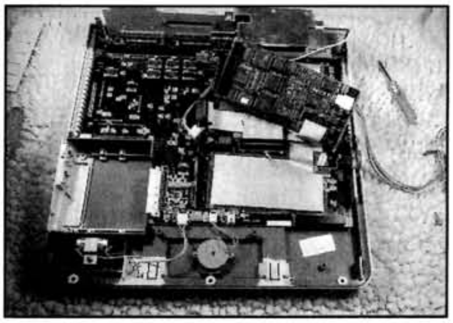


Photo 14
Installing the Hard Drive Controller

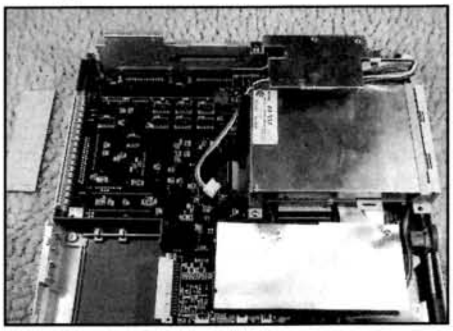


Photo 15
Installing the Floppy Drive

When installing the floppy drive, be careful not to crush any of the cables underneath the unit or mounting brackets. See Photo 8 for the correct orientation of the cables, especially the power cables, and be sure that you do not crush any of the cables. Install 1 screw in the other drive mounting bracket in the middle of the motherboard. Install 1 screw on the back, right corner of the floppy drive with the separate mounting bracket (TOTAL THREE SCREWS). Plug in the flat cable and 2-wire power cable connectors to the floppy drive, and note that both are unplugged in Photo 15. Be sure that all pins for both connectors are correctly positioned.

15. **Install the Hard Drive:** Install the hard drive by placing a mounting screw in the back, right corner of the drive mounting, and move the drive so that you can plug in the flat drive cable. Be sure that you correctly match all drive pins to the plug so that all connections are made, and double check it before proceeding. Also, verify that none of the floppy drive cables are crushed under the hard drive mountings. Then secure the 1 screw that was previously positioned. Similarly, install another mounting screw in the front, right corner of the drive (TOTAL TWO SCREWS). Note that the remaining screw for the hard drive will be installed with the ground wire for the keyboard in a later step. Also note that each drive is only secured by three screws — the "extra hole" will be secured by the flat head screws installed in the last step.

16. **Install the Keyboard:** Carefully lay the keyboard assembly on top of the disk drive units, as shown in Photo 16.

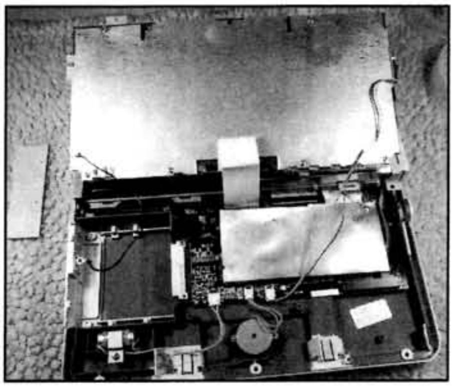


Photo 16
Plugging in the Keyboard

Insert the cable for the LCD screen into its socket. While holding the cable, press down on the plastic lock to secure the cable in the socket. Also, plug in the 2-wire connector from the keyboard to the motherboard. Note the metal tabs on the keyboard assembly fit into corresponding slots on each disk drive mounting bracket.

Turn the keyboard assembly over (toward you), and carefully match the metal tabs to the corresponding slots on the drive brackets while laying the assembly in place. Be careful not to crush the black ground wire with the left tab. Place the remaining drive mounting screw in the terminal lug on the black ground wire, and install the screw and terminal lug in the back, left corner of the hard drive, as shown in Photo 17.

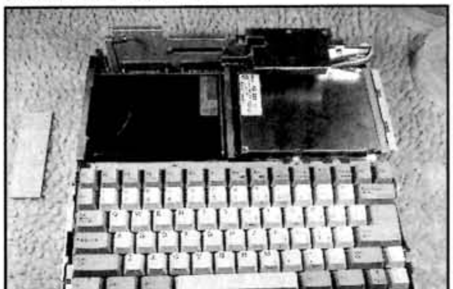


Photo 17
Installing the Keyboard

Install the 2 short keyboard mounting screws in the appropriate holes at the bottom (front) of the keyboard assembly (not shown in Photo 17). A total of THREE SCREWS were installed in this step.

17. **Install the Hard Drive Cover Plate:** Photo 18 shows the plastic cover plate for the hard drive.

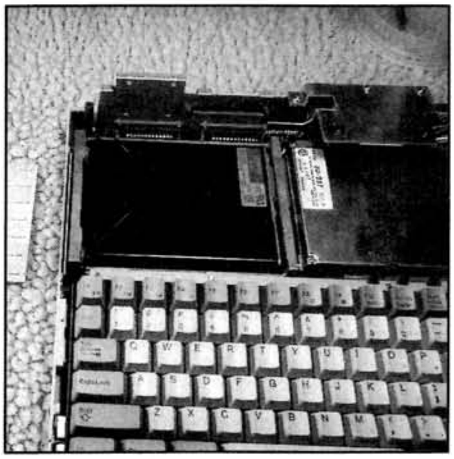


Photo 18
Installing the Cover Plate

Note the tabs on the cover plate shown in Photo 18, the plate should be inserted into the appropriate slot on the side so that these tabs are up and visible.

18. **Install the LCD Cable:** Find the top part of the cabinet, and position it as shown in Photo 19.

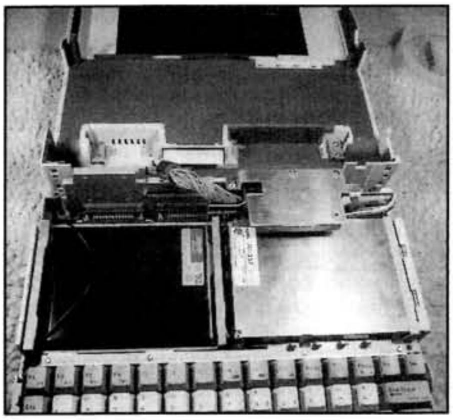


Photo 19
Installing the LCD Cable

The side with the tape should be on the left. Plug in the cable for the LCD screen as shown. On my system, there are two open pin positions in the plug, oriented toward the front (toward you), right side of the unit.

19. **Assemble the Cabinet:** Grasp both sides of the top half of the cabinet, and very carefully place it on the bottom half. Check the alignment of the keyboard and the external power connector on the back. If the top half of the cabinet does not seem to lay correctly in place, also check to make sure that the LCD cable is not being crushed.

When you are satisfied that everything fits and aligns correctly, grasp each side of the entire unit with both hands so that it is upside down with the handle toward you, as shown in Photo 1. Install the 2 long, roundhead screws in the appropriate holes in the back of the unit. Install the 3 short, roundhead screws in the appropriate holes in the front of the computer. Install the 1 long, flathead screw in the middle, *left* side of the computer, and install the remaining shorter, flathead screw on the right side.

Check It Out

Finally, you are ready for what I like to call the "smoke test." That is, if it does not go up in smoke when you power it on, you probably did everything correctly.

20. **Check Out the System:** Connect either the battery pack or the external AC power supply adapter to the computer, and turn on the power switch.

When you power up the system, be prepared to see a lot of error messages like "ERROR: Please replace the backup battery!", "ERROR: Bad Configuration found in CMOS!", and "ERROR: Incorrect video configuration — Please run SETUP!" These error messages are caused by unplugging the battery, and all of your stored configuration was lost. You will also see these error messages when the backup battery eventually loses power.

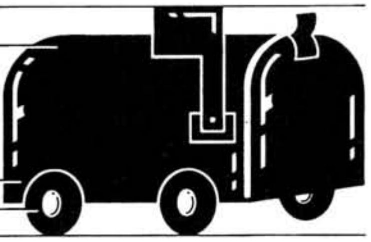
When you see these error messages, just press the ESC key each time as instructed by the prompt, and you will finally be returned to the ROM Monitor prompt that looks like: "->". Type *SETUP*, and press RETURN to access the program. Re-enter all of the data as shown on the printed copy of the SETUP screen, including the current time. You did make a copy of the SETUP screen, didn't you? When you have finished resetting all the values, press ESC, type a Y, and press RETURN to save the configuration. You should again see the "->" prompt for the monitor ROM. If you have a hard drive, just type *BW0* and press RETURN to boot the system.

If you have followed these instructions exactly, you should have no problem in booting the computer. The procedure is quite straightforward, and I have suggested that the alignment of connectors and such be checked in each step as they are installed. But if you do have any problems, you will have to take everything apart again and check for a bad or incorrect cable connection. Just start from the beginning, and take your time.

Products Discussed

SupersPort 286 w/40 MB HD . . . $5499.00
SupersPort 286 w/20 MB HD . . . $4999.00
ROM Upgrade for SuperSport 286
444-671-x. $13.00
444-672-x. 13.00
Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-0570
(Heath Catalog orders only)

*



Modifying the Cluster Size of a 4 MByte Hard Disk to Obtain More Efficient Use of Its Disk Space and to be Able to Boot From It

Daniel Ng
P.O. Box 18090
Pittsburgh, PA 15236

PC users who have hard disks in their systems can boot up from the hard disk by just switching on the computer without having to insert a floppy disk to bring up the operating system, a great convenience. They also find out that the way disk space is assigned to a file is very wasteful. Very often they will discover that to store just one byte of information in a file, 8k of disk space is assigned to that file. It is like always using a $100 bill to buy an ice cream cone and insisting on no change. The reason is that in most PCs, (at least those running DOS version 2.x) when disk spaces are assigned to a file, they are allotted in chunks of 16 sectors (8k), called the cluster size. By contrast, the big computers (such as Digital Equipment's VAX) allots disk space to files in blocks of 1/2k each.

I have been using a hard disk for some time, without the convenience of booting from it, but suffering from the shortcoming of wasteful usage of disk space. If you have a 30 or 40 megabyte hard disk, or perhaps a 10 or 20 megabyte hard disk, 8k being the smallest file may not be too wasteful. But for me, my hard disk is a meager 5 megabyte ST506; 5 megabytes will only allow me to have, at most, 600 files of 8k sizes. My hard disk, when initially formatted by and working with Zenith MS-DOS version 2.x (now 3.x), has cluster size of 8 sectors, i.e., the smallest file is 4k. That still wastes disk spaces on the average of 2 k per file. When there are 500 files in the hard disk, a megabyte of disk space is not available for storing information. Incidentally, the hard disk interface that PC disk controllers use are known as ST506 interface, but installing the 5 Meg ST506 in my H-160, which I completed some time ago is a project by itself, and before I changed the cluster size, I can use it only as a "nonstandard" drive by booting up from floppy.

AT hard disk files have allocations of 4 sectors per cluster, that means files can be as small as 2k. If I can use AT hard disk cluster size in my 5 megabyte hard disk, I will immediately have 500k more useful disk space.

I, therefore, started a project aiming at introducing the smaller cluster in my hard disk's format, it could be as small as 1 sector (512 byte). Actually, 1 sector is also the cluster size in AT 1.2 Meg 5-1/4 inch floppy disk, in the new 3-1/2 inch 1.44 Meg microfloppy disk, and also in my modified (non-standard) floppy disk format that allowed me to use a 96 tpi 720k 5-1/4 inch floppy disk in my H-160. The initial indication was that it can be done if I use DOS version 3.x, the 16 sector inefficient cluster size is a limitation due to DOS 2.x's 12 bit FAT.

This article describes my experience in finishing this project. In retrospect, it turned out to be quite easy. I have studied the disk (floppy) formats when I patched Zenith MSDOS version 2.11's BIOS to allow me to use a 96 tpi 720k 5-1/4 inch floppy disk ("Modifying the MS-DOS BIOS of the H-160 to use a 96 tpi Drive", REMark, Vol 7, Issue 4, April 1986, page 35-36). It is no longer necessary to do so when MSDOS 3.2 begins to support 3-1/2 inch 720k microfloppy drives. The 720k microfloppy drive has the same electrical interface as the 96 tpi 5-1/4 inch floppy drive. Their only difference is the physical medium.

A floppy disk can identify its disk format to the operating system for reading and writing by two methods: (1) by the parameter block at the boot record sector and (2) by the media descriptor byte at the FAT.

In the discussion which follows is information learned from experimenting with floppies and hard disks. Beginning at byte 11 of the boot (first) record sector of a disk, and ending at byte 30 are the values of the parameter block specifying the disk's format. Typical parameter block values for a 360k floppy can be inspected using DEBUG. Remember that DEBUG uses hexadecimals. At DEBUG's (-) prompt, issue the command

```
L0,0,0,1
```

it means to L(oad) into DEBUG's memory location 0 (the meaning of the first 0), from drive 0, i.e., drive A (this is the meaning of the second 0, if one wants to look at drive B, use 1 instead, similarly to look at drive C, use 2, etc) begin loading from the floppy disk's sector 0 (this is the meaning of the third 0, if you want to begin at other sectors, e.g., sector 5, then use 5 instead) and load a total of 1 sector (this is the meaning of 1, if you want to load 10 sectors, then the value will be A, the hexadecimal of 10).

Next, issue the command

```
DB
```

it means D(isplay) the values beginning at memory loation B (B is hexadecimal for 11). You will see the values shown in Table 1.

I have inserted the vertical lines to show the demarcation of byte and word items. The meaning of the items numbered 1 to 11 are described below:

Item 1 is a word (2 bytes) identifying the number of bytes in the disk's sector, 00 02 is the hexadecimal 0200 in reverse byte order meaning decimal 512 bytes per sector.

Item 2 is a byte identifying the number of sectors per cluster.

Item 3 is a word identifying the number of reserved sectors at the beginning of the disk, 1 for floppy disk, also for the hard disk.

Item 4 is a byte identifying the number of copies of FAT; 2 for floppies and hard disk.

Item 5 is a word identifying the total number of root directory entries in a disk; for floppy disks, the hexadecimal 0070 means decimal 7 × 16 = 112.

Item 6 is a word identifying the total number of sectors on the disk, here 02D0 is the hexadecimal for the decimal 720 coming from 2 sides × 9 sectors × 40 tracks.

Item 7 is a byte identifying the disk format; FF for 2 sided 8 sector, FE for 1

Table 1
Values of Disk Parameter Block for a 360k Floppy Disk

```
00 02 02 01 00 02 70 00 D0 02 FD 02 00 09 00 02 00 00 00
     ¦  ¦     ¦  ¦     ¦     ¦  ¦     ¦     ¦     ¦
  1   2    3   4    5     6   7   8      9     10     11
```

sided 8 sector, FD for 2 sided 9 sector, FC for 1 sided 9 sector, F8 for hard disk, F9 for quad density 9 sector and 15 sector, and F0 for 1.44 Meg high density 3-1/2 inch microfloppy.

Item 8 is a word identifying the number of sectors per FAT, 2 in all cases.

Item 9 is a word identifying the number of sectors per track, it is 8 or 9 for floppy, 15 (hexadecimal 0F) for AT 1.2 Meg floppy and 17 (hexadecimal 11) for hard disks, and 18 (hexadecimal 12) for the 1.44 high density microfloppy.

Item 10 is a word identifying the number of sides (head), e.g., 1 or 2 for floppies, 4 or even 6 for hard disks, depending on types.

Item 11 is a word identifying the number of special reserved sectors, I do not know much about this item.

As mentioned earlier, the disk format information is also identified by the media descriptor byte at the beginning of the FAT. There are two copies of the FAT, the first copy immediately follows the boot record sector, and the second copy follows the sector after the end of the first. The media descriptor byte is Item 7 in the parameter block, FC, FD, FE, FF, F8, F9 and F0, etc. described above. However, it seems that MSDOS versions 3.1 and earlier do not use this information contained in the parameter block and only rely on the media descriptor byte. Those versions of DOS use predefined internal disk parameters identified by the media descriptor byte to read and write floppy disks.

In the earlier project, I reported introducing a different disk format to the operating system by patching MSDOS version 2.11's BIOS at locations where the internal parameters like those shown in Table 1 are stored. In doing so, I had to change (effectively giving up) one of the supported disk formats. I have been able to use such a drive long before it was finally supported in MSDOS 3.2.

F9 and other disk formats are unknown to DOS version 3.1 and earlier. Thus, some floppy disks formatted by DOS version 3.2 and later were not readable by those earlier DOS versions still in use. It appears that versions of DOS later than 3.2 ignores the media descriptor byte and uses only the parameters in the boot record to read and write floppy disks. In this sense, they are smarter and more versatile because even the nonstandard disk format that I modified earlier were all readable by these newer versions of MSDOS.

Having understood the (floppy and hard) disk formats up to this point, it is quite easy to get the 1k cluster size into my 5 Meg hard disk. My observation and experimentation seem to indicate that the format of a hard disk, such as cluster size, number of FAT, the FAT's size, etc., is contained in the boot (first) record sector of the disk also. It is not clear to me how this information was recorded there. Is it by Zenith's PART (equivalently IBM's FDISK) or by the low level formatting program in the hard disk controller's ROM? DEBUG is not able to examine the hard disk after low level format. Nevertheless, when the FORMAT program is later run, it uses the information contained there to reset the FAT and the directory.

I backed up all the files on the hard disk into floppies. It is quite easy for me, it only took six 720k floppy disks. Next I deleted all the files from the hard disk. This step is not necessary, but I did so effectively to reset the FAT. With that done, I ran the DEBUG program. At the (-) prompt, I issued the command L0,4,0,4B to tell DEBUG to go to drive E, load 75 sectors of data into its memory beginning at memory location 0, the sector count begins at sector 0. The reason for loading in 75 sectors will become apparent later. My drive A is a 360k 5-1/4 inch floppy drive, drive B is a 720k 5-1/4 inch floppy drive masquerading as a 3-1/2 inch microfloppy drive, drive C is a real microfloppy drive, drive D is declared to be also a microfloppy drive, thus drive E is my hard drive. Next, I issued the command D8 to display the values of the disk parameters.

What I see is shown in Table 2.

Table 2
Values of Parameter Block for a Hard Disk with 4k Cluster Size

```
00 02 08 01 00 02 00 02 4F 28 F8 04 00 11 00 04 00 11 00
     ¦  ¦     ¦  ¦     ¦     ¦  ¦     ¦     ¦     ¦
  1   2    3   4    5     6   7   8      9     10     11
```

This shows that each sector is 512 bytes long; there are 8 sectors in a cluster (4k); there is one reserve sector; there are two copies of the FAT and there are 512 root directory entries; and the total number of sectors in my hard disk is 284F in hexadecimal or 10,320 in decimal, this value comes from the number of cylinders that were assigned to this particular partition; the disk type F8 means it is a hard disk; each FAT is 4 sectors long; each track has 17 sectors; there are 4 heads and 17 special reserved sectors.

My goal is to change the parameters in the hard disk's boot sector so that it will have a cluster size of 1k. To do that I will need to change item 2 in Table 2 from 08 (4k) to 02 (1k) sectors per cluster. There are other coupled changes that also need to be made, e.g., the size of the FAT. I have to use the 16 bit (2 bytes) FAT that DOS 3.x allows. Since there are 10,320 sectors in my hard disk, and each cluster contains 2 sectors, I will need 5,160 cluster entries in the FAT to describe all of them. Each entry in the FAT describes one cluster, to completely describe 5,160 clusters, I need 2 × 5,160 = 10,320 bytes. That is a little bit over 20 sectors (10,320/512 = 20.16), therefore, I need a FAT that is 21 (15 hexadecimal) sectors long. For item 8 in Table 2, the new value is 15 00 (reverse byte). The parameters should now have the values shown in Table 3. The rest will remain unchanged. The changes that need to be made are easily done by using the E(xamine) command of DEBUG.

The first FAT begins at the next sector. The second old FAT originally is located at memory location 2k (4 sector FAT means 2k in memory) further down in DEBUG's memory. The first FAT appears at DEBUG's address xxxx:0200, where xxxx is the segment of memory that DEBUG was loaded. The first 3 bytes were F8 FF FF, corresponding to a 12 bit FAT. I changed the first four bytes beginning at xxxx:0200 to F8 FF FF FF corresponding to 16 bit FAT. The other memory locations showed 00 until the second old FAT is reached at memory location xxxx:0A00, with values F8 FF FF appearing again. If I have not deleted all the files already, the FAT will have all kinds of numbers in them making it more difficult to see things.

Since the new FAT would be larger than the old one, I changed these 3 bytes to 00 and filled in the memory locations up to sector 43 with 00 (the first copy of the FAT occupies 21 sectors, the second copy another 21, plus the boot sector makes 43). Beginning at location xxxx:2C00, I changed the values of the 4 bytes to F8 FF FF FF to signal the beginning of the second new FAT. The locations for root directory entries comes after the second copy of the FAT. Since there are 512 directory entries, each entry requires 32 bytes, a total of 32 sectors are, therefore, required. I prepared that part of the disk for directory entries by storing 00 in all the memory locations of DEBUG using the F(ill) command. The correct way to reset the directory should have been to fill the

Table 3
Values of Parameter Block for a 5 Meg Hard Disk with 1k Cluster Size

```
00 02 02 01 00 02 00 02 4F 28 F8 15 00 11 00 04 00 11 00
     ¦  ¦     ¦  ¦      ¦      ¦  ¦     ¦     ¦      ¦
  1   2    3    4    5      6   7   8      9     10     11
```

first byte with E5 and the remaining 31 bytes with 00. Recall that we have been dealing with 1 boot sector, 21 × 2 = 42 FAT sectors and 32 root directory sectors, for a total of 75 sectors, that is the reason for having loaded so many sectors from the disk at the beginning. I next wrote all these changes back to the hard disk using DEBUG's W(rite) command W0,4,0,4B to write DEBUG's memory beginning at location 0, into drive E, starting at sector 0 of that drive, for a total of 4B (75 decimal) sectors. Then, exited DEBUG with the Q(uit) command.

If the number of root directory entries has been more or less than 512, 512 is the value in my hard disk, the number of disk sectors for directory entries would have been different, but the arithmetic of arriving at that different number will be the same. A cluster size of 2 sectors results in an average waste of 1 sector (1/2k) per file, going to a cluster size of 1 sector, the average waste would be 1/4k, a savings of exactly 1/4k. On the other hand, using a cluster size of 1 sector results in each FAT requiring 21 more sectors, that comes to 21k total disk space. The break even point in savings is reached at 84 files. The decision to select a cluster size of 2 sectors (1k) instead of 1 sector (1/2k) was made as a compromise between efficient use of disk space and performance.

These changes to the hard disk are not known to MSDOS until the next reboot. Upon reboot, I used Norton Utilities to verify that the cluster size is indeed 1k. After that, I restored all the files from the backup floppies. I found that I have gained about 800k of additional useful disk space. Running Norton Utilities' FILESIZE showed that the percentage of slack of all the files is only about 2% compared to as much as 10 to 15% before.

I have since discovered that the elaborate calculation and efforts that I made to reset and relocate the FAT, the directory entries, etc., were not necessary. Once the changes have been made to the parameters in the boot record sector, one needs only to reboot and then reformat the hard disk. The FORMAT program will use the changed information to create the correct FATs, directory sectors, and the desired cluster size, etc. For some Heath/Zenith users, before you can format the hard disk, depending on the version of your DOS, you may need to run DSKSETUP to remove the format protection first.

This procedure of improving the disk usage in a hard disk can be applied to 10, 20 and even 30 Meg drives. Though I only have the opportunity to try it on a Miniscribe 10 Meg full height hard disk, and it works. The corresponding values can be calculated as follows.

For a 10 Meg drive, there will be about twice as many sectors (20,640) as the 5 Meg, the size of the FAT thus needs to be 41 (29 in hexadecimal) sectors. For a 20 Meg drive, the number of sectors will be 41,280, requiring 81 (51 hexadecimal) sectors. For a 30 Meg drive, there will be 61,920 sectors requiring 121 (79 hexadecimal) sectors. The results are shown in Tables 4, 5 and 6. In the entries marked with xx xx, the exact values corresponding to the drive should be used. That value is obtained by examining the boot sector with DEBUG.

The procedures for patching the boot sector can be done using batch files. To do so, one needs to prepare a batch file and a data file to go along with it. These files can be create using WordStar, DOS's EDLIN or plainly from the keyboard as shown below.

These files will be called CLUSTER.BAT and CLUSTER.DAT. To create CLUSTER.BAT, at DOS's prompt, type

```
COPY CON: CLUSTER.BAT
DEBUG<CLUSTER.DAT
^Z
```

^Z means control-z.

To create CLUSTER.DAT at DOS' prompt, type:

```
COPY CON: CLUSTER.DAT
L0,4,0,40
EB
00 02 02 01 00 02 00 02 xx xx F8 yy 00 11 00 04 00 11 00

W0,4,0,40
Q
^Z
```

The extra blank line before W0,x,0,40 is necessary. Execute this program by typing CLUSTER. After this, reboot, run the FORMAT, and restore the files from floppies. However, you must make sure that DEBUG is on the default disk or directory or accessible through a path. The underlined quantities differ from system to system, if your hard drive is drive C, it should be L,0,3,0 40, and W0,3,0,40. Also, different from system to system is the value yy, the number of sectors for the FAT.

As a result of my ability to modify the cluster size of the hard disk, I can now also boot from my 5 Meg hard disk. You might say big deal, who can't? Ironically, as mentioned earlier, because of the "non-standardness" of the ST506 hard disk drive, most hard disk controllers using the ST506 interface standard do not support it. When I came into possession of a phased out ST506 for $25, I looked for a hard disk controller that would support it. I purchased a DTC (Data Technology Corporation) 5150CX controller, because it would support a total of 16 different hard disk formats. In spite of that, it still would not support the ST506 directly.

The DTC controller nevertheless has a provision for supporting a non-standard drive. A hard disk controller needs to know the parameters of the hard disks (number of cylinders, number of heads, etc.) in order to control them. Such information is contained in a table located in the controller's ROM. The beginning location of this table is stored as interrupt vector 65 (41 hexadecimal) at the two bytes in memory location 0000:0104 to 0000:0107 (in reverse byte order) during system initialization. Because the parameters corresponding to ST506 are not in ROM, the controller is not able to communicate with it. In order for the DTC controller to access my hard drive, it is necessary to set

Table 4
Values of Parameter Block for a 10 Meg Hard Disk with 1k Cluster Size

```
00 02 02 01 00 02 00 02 xx xx F8 29 00 11 00 04 00 11 00
     ¦  ¦     ¦  ¦      ¦      ¦  ¦     ¦     ¦      ¦
  1   2    3    4    5      6   7   8      9     10     11
```

Table 5
Values of Parameter Block for a 20 Meg Hard Disk with 1k Cluster Size

```
00 02 02 01 00 02 00 02 xx xx F8 51 00 11 00 04 00 11 00
     ¦  ¦     ¦  ¦      ¦      ¦  ¦     ¦     ¦      ¦
  1   2    3    4    5      6   7   8      9     10     11
```

Table 6
Values of Parameter Block for a 30 Meg Hard Disk with 1k Cluster Size

```
00 02 02 01 00 02 00 02 xx xx F8 79 00 11 00 04 00 11 00
     ¦  ¦     ¦  ¦      ¦      ¦  ¦     ¦     ¦      ¦
  1   2    3    4    5      6   7   8      9     10     11
```

the controller's selection switches to correspond to the non-standard drive, boot up using a floppy. Once booted up, run a terminate-and-stay-resident program to create a new disk parameter table in RAM to replace the one in ROM. In this new table, the entry corresponding to the non-standard drive contains the correct parameters describing the ST506. The vector at interrupt 65 is changed to point to the new table's memory location in RAM by this program. From this point on, the hard disk is usable. The program which performs the table being modified was obtained from Mr. Michael Snyder, presently with the Computer Science Department of Ohio State University. The listing of this program is given in Listing 1, it is specifically written for the DTC controller.

I nevertheless attempted to boot from the ST506 by setting the switches in the hard disk controller to a type of hard disk that is closest to the ST506, e.g., that of a 10 Meg drive with 304 cylinders, 4 heads (the ST506 has 152 cylinders, 4 heads). This way, the first two or three tracks of the ST506 can be accessed reliably. In fact, all tracks can be read if they are accessed sequentially. Norton Utilities' DISKTEST can read the disk with no error. This is so, because the average access time of the ST506 is something like 85 ms compared to the 60 ms or less for the others, the reading errors occur when the head is expected to have finished moving rapidly from tracks separated far apart by the controller. Three tracks corresponds to 204 sectors (102k), that could be enough storage space to hold the disk operating system which consists of the files IBMBIO.COM, IBMDOS.COM and COMMAND.COM. I should be able to boot up from the hard drive, then immediately run the terminate-and-stay-resident program to create the new drive parameters table with information corresponding to the selected disk type (304 cylinders, 4 heads, etc.) replaced by information corresponding to the ST506, etc.

Indeed I was able to do so with MSDOS 3.1. When MSDOS 3.2 appeared, because of its support for the 3-1/3 microfloppy drive, I was attracted to use it. But version 3.2 has grown much bigger, there is not quite enough room. When booting up from the hard drive in this manner, the COMMAND.COM file, which is last to be read, was found to be bad by the operating system. It must have occupied sectors on track 4 and the hard disk controller cannot read it. This is due to the original inefficient 4k cluster size combined with the large numbers of overhead sectors (1 boot sector, 8 FATs sectors, 32 directory sectors, and 17 special reserved sectors, etc.) for the entire disk.

The way I get the ST506 to boot now is to create a bootable partition containing 4 cylinders with a cluster size of 1 sector (256 bytes) and the remaining cylinders contained in another partition with a cluster size of 2 sectors (1k). For the bootable partition, the values of the parameters are shown in Table 7.

Table 7
Values of Parameter Block for a 4 Cylinder Partition with 1/2k Cluster Size

```
00 02 01 01 00 02 00 02 xx xx F8 02 00 11 00 04 00 11 00
    |  |     |  |     |     |  |     |     |     |
 1  2     3  4     5     6  7     8     9    10    11
```

Four cylinders produce a hard disk of about 136k. The use of disk space becomes more efficient, the overhead is reduced to 1 boot record sector, 2 FATs of 1 sector each, 2 directory sectors, and the 17 special reserved sectors.This way I am able to put the operating system (IBMBIO .COM, IBMDOS.COM, COMMAND .COM) in this partition by formatting with the /s option. In addition, the other files, such as CONFIG.SYS, (for configuring the system, e.g., installing ANSI.SYS, number of files and buffer sizes, etc.), AUTOEXEC .BAT, (for customizing my working environment, e.g., reading the date and time from the realtime clock, declaring the path and customizing the prompt, etc.), the special terminate-and-stay-resident hard disk parameter table creating program, and ASGNPART.COM, (for access to the second partition) are also in this bootable partition.

My PC can now boot and configure itself automatically by executing the necessary programs to access the second partition, a big improvement.

```
DATA SEGMENT
ORG  0
FDTBL       DW    512
            DB    8
            DW    0200H
            DW    0100H
            DB    0BH,5,19H,0FFH
            DB    'P'
            DB    0,0,0,0

            DW    615
            DB    2
            DW    0267H
            DW    0267H
            DB    0BH,5,19H,0FFH
            DB    'P'
            DB    0,0,0,0

            DW    612
            DB    4
            DW    0264H
            DW    80H
            DB    0BH,5,19H,0FFH
            DB    'P'
            DB    0,0,0,0

            DW    306        ;THIS ENTRY IS CHANGED TO
            DB    4          ;CORRESPOND TO ST506
            DW    0132H      ;IN THE BOOTABLE
            DW    0          ;VERSION
            DB    0BH,5,0CH,0B4H
            DB    '('
            DB    0,0,0,0

            DW    306
            DB    2
            DW    80H
            DW    0132H
DB    0BH,5,0CH,0B4H
DB    '('
DB    0,0,0,0

DW    375
DB    8
DW    0177H
DW    0
DB    0BH,5,0CH,0B4H
DB    '('
DB    0,0,0,0

DW    306
DB    8
DW    84H
DW    0
DB    0BH,5,0CH,0B4H
DB    '('
DB    0,0,0,0

NSD   DW 153  ; *** NON-STANDARD DRIVE ***
DB    4
DW    128       ; FILLED WITH PARAMETERS
DW    128       ; CORRESPONDING TO ST506
DB    0Bh,00,0Ch,0B4h
DB '('
DB    0,0,0,0

DW    306
DB    6
DW    80H
DW    0100H
DB    0BH,5,0CH,0B4H
DB    '('
DB    0,0,0,0

DW    512
DB    6
DW    0100H
```

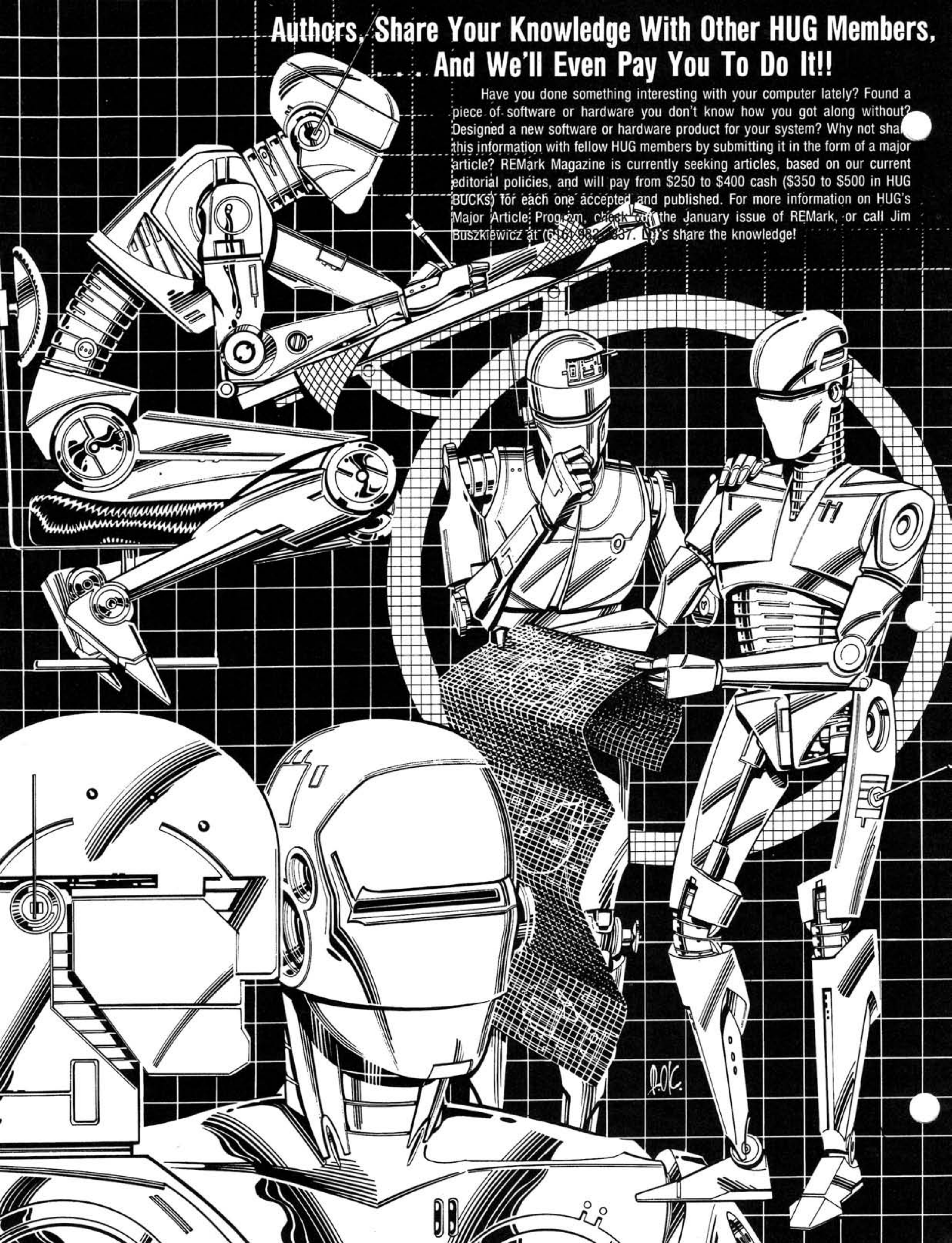
```
          DW    Ø1ØØH
          DB    ØBH,5,19H,ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø

          DW    612
          DB    2
          DW    Ø264H
          DW    8ØH
          DB    ØBH,5,19H,ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø

          DW    64Ø
          DB    4
          DW    Ø28ØH
          DW    Ø1ØØH
          DB    ØBH,5,19H,ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø

          DW    697
          DB    3
          DW    Ø2B9H
          DW    Ø2B9H
          DB    ØBH,5,19H,ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø

          DW    697
          DB    5
          DW    Ø2B9H
          DW    Ø2B9H
          DB    ØBH,5,19H,ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø

          DW    64Ø
          DB    6
          DW    Ø28ØH
          DW    Ø1ØØH
          DB    ØBH,5,19H,ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø
          DW    3Ø6
          DB    4
          DW    Ø132H
          DW    Ø
HFPORTØ   DB    ØBH
HFPORT1   DB    5
```

```
HFPORT2   DB    19H
HFPORT3   DB    ØFFH
          DB    'P'
          DB    Ø,Ø,Ø,Ø

TAIL DB   Ø

ORG  1Ø4H
LØ1Ø4     DW    ?
LØ1Ø6     DW    ?

DATA ENDS
PROGRAM   SEGMENT

ASSUME    CS:PROGRAM, DS:DATA

LØØ71:    PUSH DS
          XOR  AX,AX
          MOV  DS,AX
          MOV  SI,WORD PTR LØ1Ø4    ; EXAMINE INT VECTOR 41
          MOV  DX,WORD PTR LØ1Ø6
          MOV  DS,DX
          CMP  WORD PTR [SI+OFFSET NSD],99ØØH      ; DOES IT POINT INTO MY DATA?
          JNZ  LØØ8B
          POP  DS
          INT  2ØH              ; IF SO, TERMINATE

LØØ8B:    MOV  AX, DATA
          MOV  DS, AX                    ; GET CORRECT DATA SEGMENT
          MOV  DX,OFFSET FDTBL           ; SET INTERRUPT VECTOR 41h TO DS:Ø
          MOV  AX,2541H
          INT  21H
          MOV  DX, 2ØØh           ; TERMINATE AND STAY RESIDENT,
          INT  27H                ;   - SAVING UP TO DS:2ØØ

PROGRAM ENDS

          END  LØØ71
```

*

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!



Jazzing Up Your Desktop Publishing Articles with Scanning

Matt Elwood
1670 North Laddie Court
Beavercreek, Ohio 45432

If you do any desktop publishing, you probably know that scanned pictures make a big impression. Of course, to make a big impression, you need big bucks, right? Not now. Until now, a good scanner costed around $1000, but now a new breed of scanners called hand scanners might change the face of desktop publishing, making it more affordable and introduce it to private citizens.

The hand scanner I will be reviewing is the DFI HS-3000 Handy Scanner. It is a selectable 100-400 dpi scanner that can scan a segment of a page four inches wide easily. It is bundled with its own scanning program, and HALO DPE, a desktop publishing editor.

Installation is simple for the hand scanner. All I needed was an extra slot and a screwdriver. The scanner has a short PC-style board card that I installed in the outermost slot on our H-386. The scanner interface is a small, circular DIN cable. Installing the board took me a total of five minutes.

Installation of the software is simple. The software that comes with the scanner is a small EXE file. Copying this over straight to another disk or a hard disk will suffice as installation.

The scanner software is simple to use, also. When you first call up the program, there is a video board selection screen. With the Z-449 card, you can use EGA mode. The VGA will not work without this patch.

To use the Z-449 patch below, you must have DEBUG in the path or current disk drive:

```
copy scan.exe scan449
debug scan449
-e1679
    11.12
-w
-q
ren scan449 scan449.exe
```

The patch actually changes the video mode to an identical mode that the Z-449 can use. If you do not see the "11" when you enter "e1679", then do not patch the program. You have a different version than the version I have of the program.

Now, you can load up the "SCAN" or "SCAN449" program. The program is

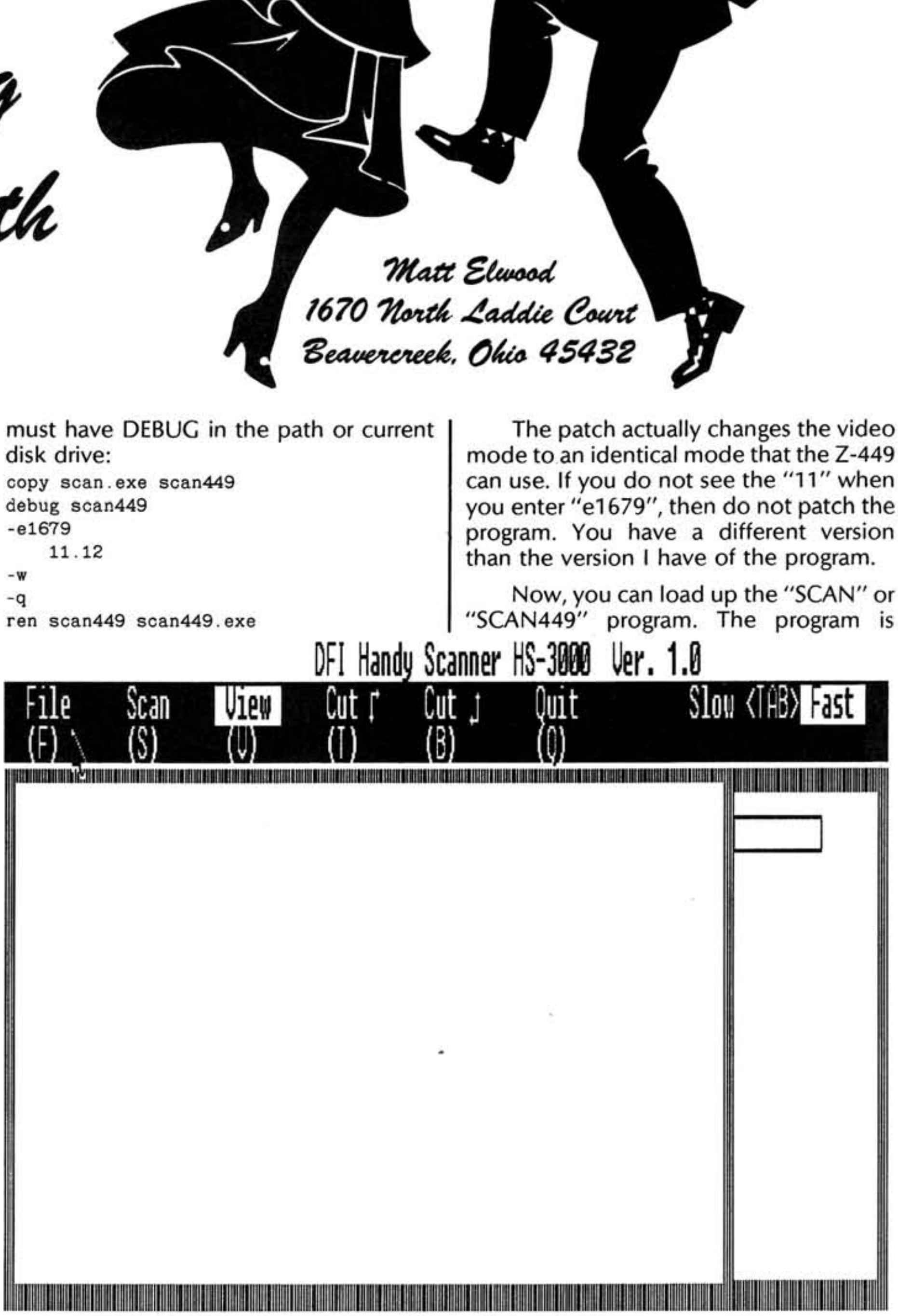


Figure 1
The Screen for the Scanner Program

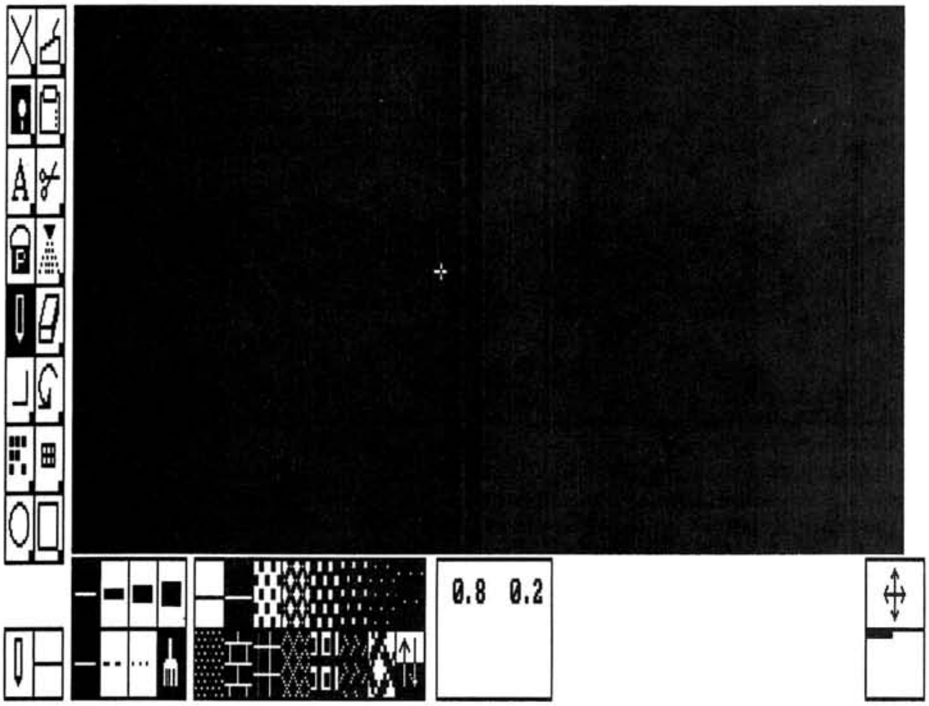


Figure 2
A Blank Halo DPE Screen

menu driven and simple to use. You can also use the mouse if you have the Microsoft Mouse driver loaded. After the video board selection screen, a screen looking like Figure 1 appears. It consists of a top line menu, which you can select with the keyboard commands or with the mouse. The large window at the bottom shows part of the page you've scanned and the small window at the right shows the entire scanning buffer. You can scroll around with the arrow keys.

To scan a picture, either type "S" (no return key) on the keyboard or select "Scan" with the mouse. A screen showing the current resolution and the height of the scanning buffer (the amount you can scan) is shown. To change the resolution, select either 1, 2, 3, or 4 at the base of the scanner to select 100, 200, 300, or 400 dpi. You may change resolutions in the middle of scanning, too. After pressing return, you are ready to scan. Align the scanner, push the start button located at the left side, and VERY SLOWLY move the scanner down the page. Be especially careful to go slow when using 400 dpi. If the scanned picture is too light or too dark, adjust the brightness control next to the resolution control. If you have a halftone or text document, you might want to change the button on the side of the scanner next to the start button to letter. If you are scanning a photo, try all three photo modes for the best.

Once you have scanned a picture, you can save a block of it. You save it by first specifying the top-left corner and the bottom-right corner of the picture using the T and B keys. This shows a block around the area to be saved. To save the picture, click on File or type F. You can select four image formats: GEM, PC Paintbrush, Halo DPE, and Windows Paint. You can also load pictures in these formats, and save them to another format, using the program as a conversion utility. To actually save the file, type F1 or click on save. If you want to save the picture rotated ninety degrees, use F2 or click on the second save command.

As you have seen, the scanning program is simple to use. For greater image manipulation, you can use the Halo DPE. In Halo DPE, you can read in files saved in that format from the SCAN program, or scan in a picture yourself. Halo doesn't provide as much capability, though, because you can't see as you are scanning and you must stay with the same DPI as you set in the SETUP program.

Halo DPE is not fully optimized when it comes to devices. The EGA driver only allows 2 colors. There are only 24 printers, most of them laser. The unit uses EMS memory, but will not use the 256k in the first 1 Meg of RAM. I recommend a mouse, and this is what I use in the examples. To use the mouse, just select a tool with the left button. When you press the right button on one of the tools, it will bring up a display of options.

When you first call up Halo DPE, a screen that looks like Figure 2 will appear. The strip on the side is the selectable graphics tools. The bottom strip shows the coordinates and the fill patterns. The large area in the middle is part of the logical drawing page.

The size of the drawing page varies by printer, resolution, and memory. I had to go into SETUP many times and reduce the size. With the HP LaserJet + at 300 dpi, I used a page size of 1300 by 1300. Since the screen only shows part of the page, you can move around with the four small arrows shown in the lower right-hand view. To toggle between full page and part page mode, you select the box below the four arrows in the right-hand corner. Full page mode looks like Figure 3.

Our first tool we will talk about is the pencil tool. This tool is the "doodling" tool. Pen widths can be selected at the

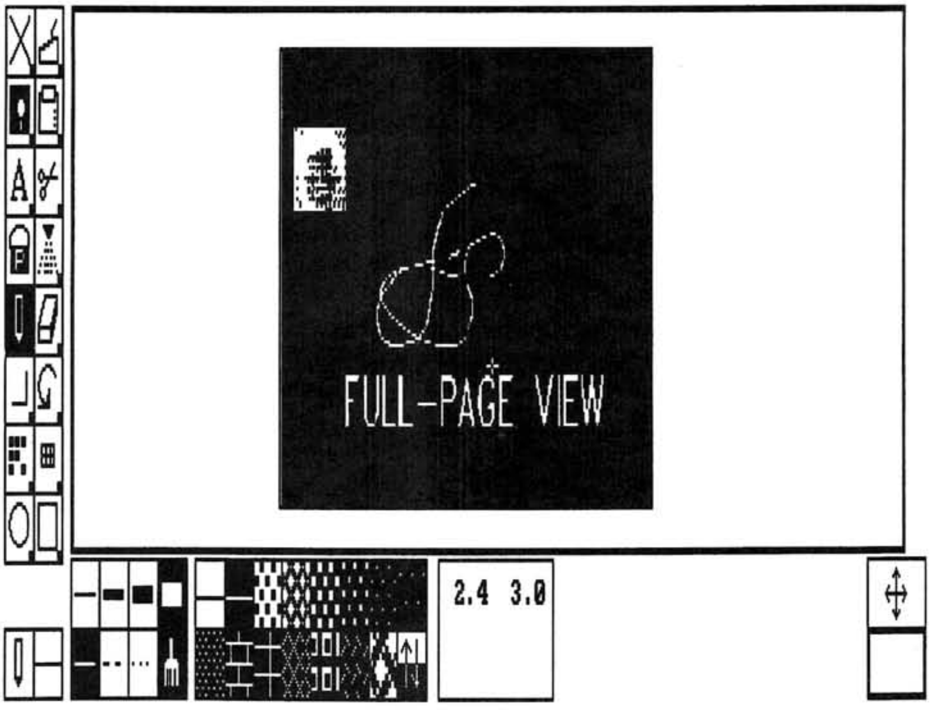


Figure 3
The Full-Page View in Halo DPE

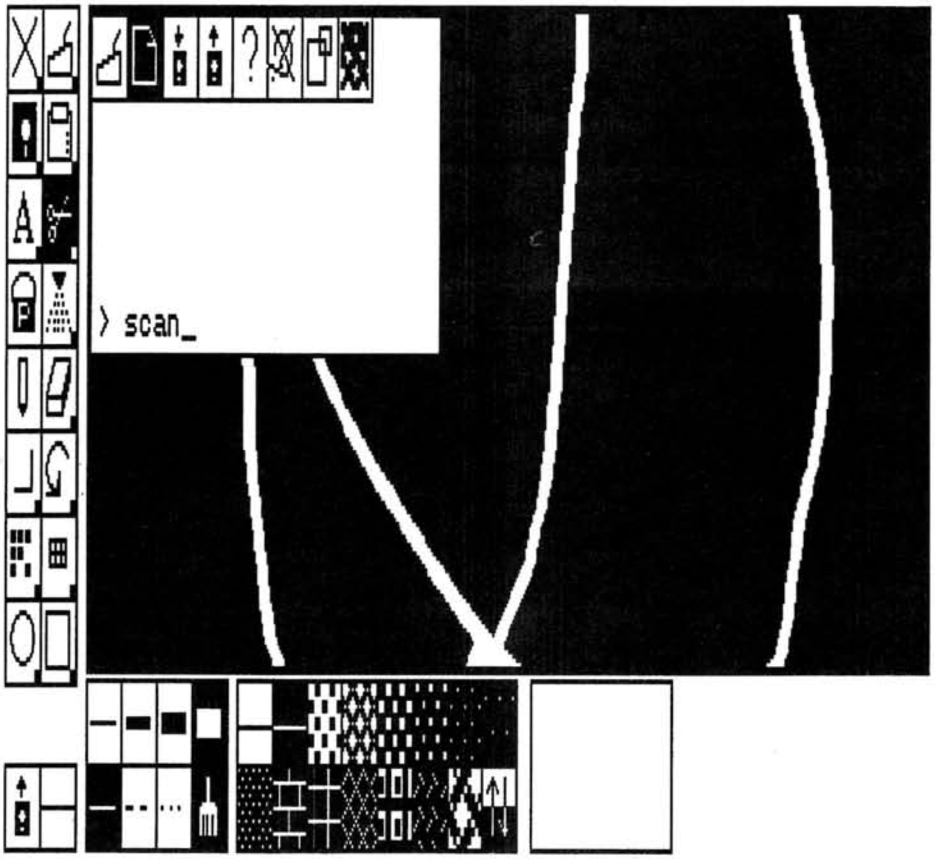


Figure 4
The Scanning Menu in Halo DPE

bottom left corner. To erase an entire area, use the eraser icon to the right of the pencil icon. The eraser works like the pencil, except you can make the area larger or smaller by holding down the right button and dragging the box shown in the drawing area.

The simple scanning tool looks like a side view of a typewriter. This is the upper right-hand icon in the tools area. When you select it, the screen will blank and then you scan. When finished, press the escape key. The scanning options are few, as shown by Figure 4.

To draw boxes or circles, use the bottom two icons in the tools corner. When you click them with the right button, they bring up menus with options, also. You can select special modes as filling. Remember to select the fill pattern you want from the patterns at the bottom.

The two icons above the bottom two are for different reasons. The icon on the left is for straight bit edit mode. In this mode, the left button edits pixels in one color and the right button edits pixels in the background color. When you first select the icon, a box appears on the editing screen. You can select the size of this box by holding the right button and "dragging" to the right and up to make the box larger. The largest this can go is 64 by 64 pixels, as shown in Figure 5.

The icon to the right of the bit edit icon is the grid icon. This draws a visible or invisible grid. To select how many grid "ticks" there will be in an area, invisible or visible, click the icon with the right mouse button. Once again, when in the drawing area, you can make the area larger or smaller with the right button.

To fill in a closed area, there is a "paint can" tool furnished. It will fill in a closed area with the specified fill pattern selected from the bottom list.

A rather powerful feature of Halo DPE is the ability to cut images from a picture and paste them somewhere else, or make a "rubber stamp" to repeatedly copy the cut image. The resize icon will change the size of an area. You can also save or restore images. When you restore the images, you can change the size of the image. You can also change the bitwise way you insert the image. The four icons in the second row, when you select the right button, control this. Figure 6 shows the range of cut and paste commands.

The first cut and paste command simply is to cut an area of an image and replace it somewhere else. This command leaves nothing behind but a blank area of the place that is being cut, so be careful.

The next command is the cut-and-copy or rubberstamp command. This leaves the original of the area being cut down, and makes as many copies as you want.

The third command is the resize command. It is just like the cut and paste command, except you can resize the graphics by holding the right button down and dragging the graphics to the right size.

After that are the two commands to retrieve and save cut files. You specify the area and use the right button to size the box containing the area to be saved for the save command.

The question mark icon to the right of the disk access commands gives you a list of the available cut files. You can se-

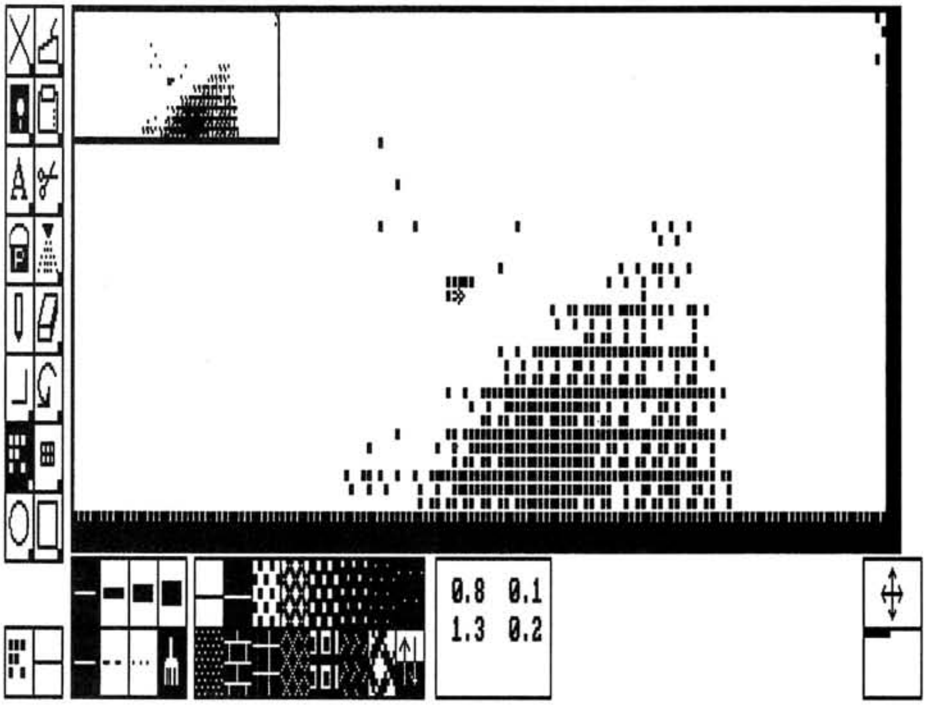
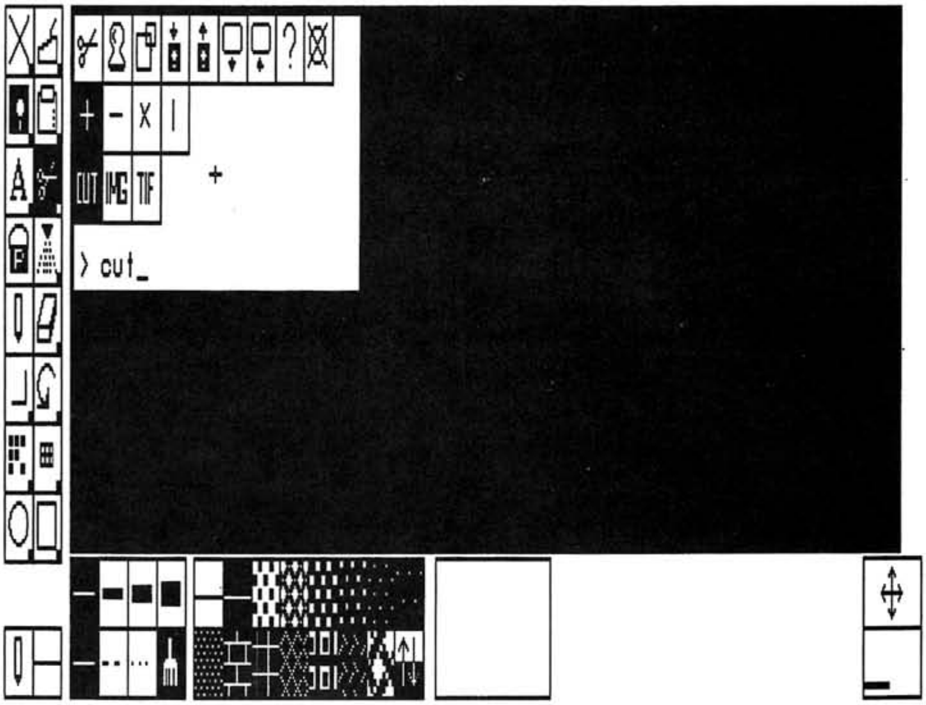


Figure 5
The Magnified Bit-Edit Mode



```
> cut_
```

Figure 6
The Cut-and-Paste Screen Menu

lect from these files. If you have one selected and you want to delete it, the right-most icon will delete the file.

The second and third row of cut and paste commands are not really important, but I will mention what they are. The second row is the type of bitwise operation (AND, OR, XOR) that the image will be pasted with. The third row specifies the type of file to write.

Halo DPE has very powerful text capabilities. It includes many fonts and many font sizes. It also includes the capability to rotate, fill in, shadow, and scale fonts.

Selecting these fonts can be done in two ways. You can select the first eight boxes on the top line, or you can select the right-most star icon, which brings up a directory of all the fonts available. You can select straight from any of these names.

Any font can be resized to an infinite number of possibilities using the size commands on the next line. The first symbol lets you select any size for your fonts. The other size selections offer you four preset sizes in points.

You can also select the direction the text will flow in. The icons to the right of the font size select the font direction. Selecting the shadowing is just as simple. The four icons below the size icons (labeled 0, 1, 3, 5) select the degree of shadow used. The last icons just underline the text and allow you to rotate it.

Those are about the features that Halo DPE has. The scanner works great, and with the modification, the scanning program is a small, but powerful, utility. As for Halo DPE, the program was nice, but because of the graphics limitations, I think I'll just stay with my Deluxe Paint II.

Diamond Flower Electric Instrument Co.
2544 Port St., W. Sacramento, CA 95691
(916) 373-1234

Photo 1
The Package for the Scanner

Photo 2
The Actual Scanner with DPI and Brightness Switches Down

SOFTWARE POTPOURRI

Robert S. Hudson
5012 1/2 Field St.
San Diego, CA 92110

INTRODUCTION

This article discusses several software programs, and their application on the Zenith 248 (some of it is also available for the H/Z 100). The primary focus will be on Inset, Software Carousel, and some engineering programs from BV Engineering. Hence, the title, Potpourri. We will also briefly discuss Norton Commander, but not in great detail, because it was the subject of a well written article in the February REMARK. Now, obviously, all these programs can't be covered in one article, so we will not go into great detail on any specific program, perhaps leaving that for a future article if the interest warrants.

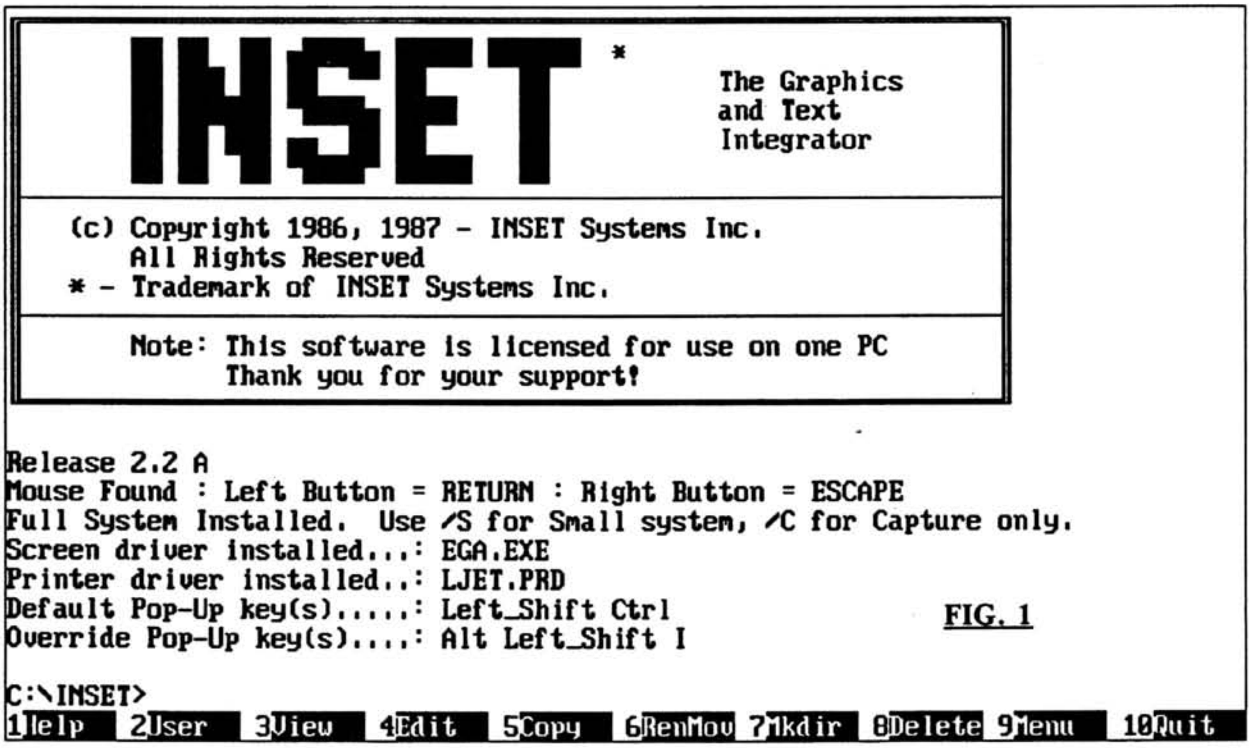
So we are all on the same wavelength, I'll fill you in on the subject system and its software. I have a Zenith 248, my fourth computer, and the third Heath/Zenith. It has 2 hard drives (a 20M and a 40M), and 2 floppies (both 1.2M); with 640K base memory and 1.5M expanded memory (NOT extended; see REMARK, Mar 1989, pp 31-32). This extra memory is very useful for some of the applications software.

The "C" drive (20M) is the boot drive. It holds the "basic" computer software, such as spreadsheet, word processor, communications, and recreation. The "D" drive (40M) contains the engineering related software, such as OrCad, Software Carousel, Microsoft Project, and BV Engineering. It presently has a lot of spare room, but that probably won't last, because I plan to get some more CAD software similar to OrCad.

I also have a HP LaserJet series II printer, and a HP 7550A plotter. The plotter will take "A" and "B" size drawings and transparencies, and is the primary output for schematics and circuit analysis plots.

LE MENU

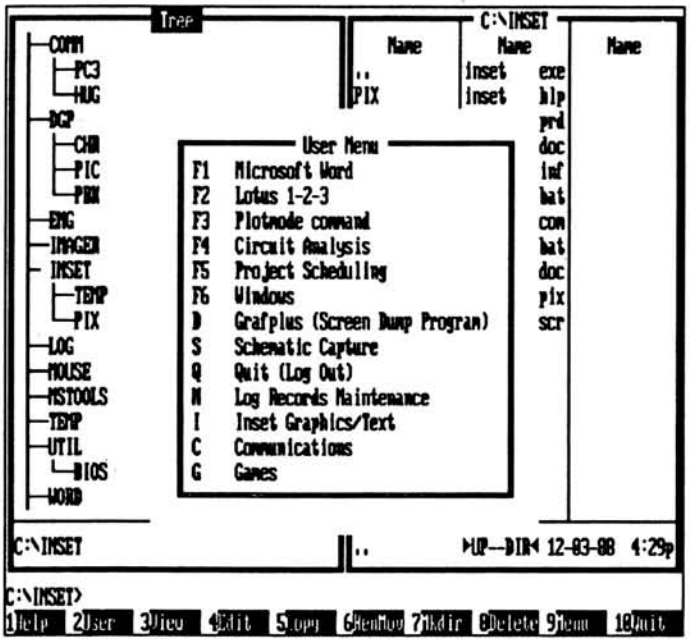
At work, we have XTREE on most of the computers, and I have never liked it. I have always felt that it was much easier and faster to tool around in DOS than in XTREE. With the Zenith version of DOS, its very easy to locate files with the



```
INSET*        The Graphics
              and Text
              Integrator

(c) Copyright 1986, 1987 - INSET Systems Inc.
    All Rights Reserved
 * - Trademark of INSET Systems Inc.

    Note: This software is licensed for use on one PC
          Thank you for your support!

Release 2.2 A
Mouse Found : Left Button = RETURN : Right Button = ESCAPE
Full System Installed. Use /S for Small system, /C for Capture only.
Screen driver installed...: EGA.EXE
Printer driver installed..: LJET.PRD
Default Pop-Up key(s).....: Left_Shift Ctrl
Override Pop-Up key(s)....: Alt Left_Shift I

C:\INSET>
```

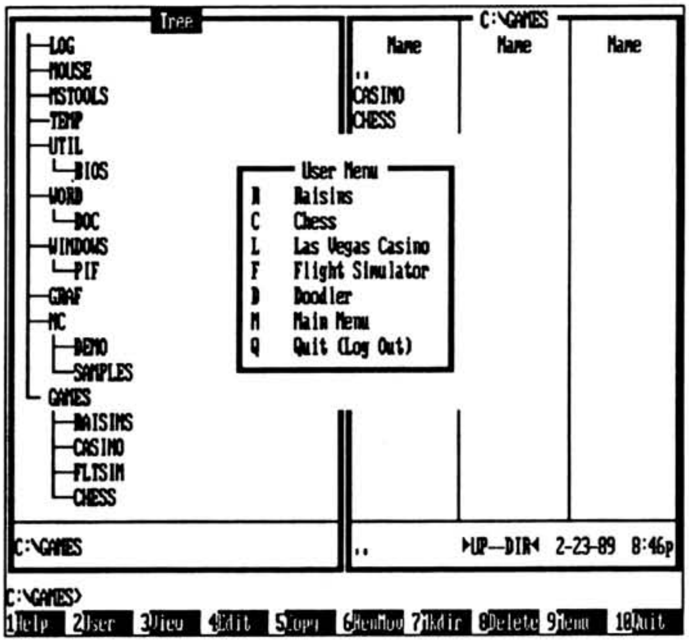
1Help 2User 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9Menu 10Quit

FIG. 1



```
Tree                                         C:\INSET
├─COMM                              Name        Name          Name
│ ├─PC3                           ..          inset   exe
│ └─HUG                           PIX         inset   hlp
├─DGP                                                 prd
│ ├─CHR          ──── User Menu ────                  doc
│ ├─PIC          F1   Microsoft Word                  inf
│ └─PBK          F2   Lotus 1-2-3                     bat
├─ENG            F3   Plotmode command                com
├─IMAGER         F4   Circuit Analysis                bat
─ INSET          F5   Project Scheduling              doc
│ ├─TEMP         F6   Windows                         pix
│ └─PIX          D    Grafplus (Screen Dump Program)  scr
├─LOG            S    Schematic Capture
├─MOUSE          Q    Quit (Log Out)
├─MSTOOLS        M    Log Records Maintenance
├─TEMP           I    Inset Graphics/Text
├─UTIL           C    Communications
│ └─BIOS         G    Games
├─WORD
C:\INSET                          ..          ►UP--DIR◄ 12-03-88  4:29p
C:\INSET>
1Help 2User 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9Menu 10Quit
```

FIG. 2



```
Tree                                         C:\GAMES
├─LOG                               Name        Name          Name
├─MOUSE                           ..
├─MSTOOLS                         CASINO
├─TEMP                            CHESS
├─UTIL
│ └─BIOS           ──── User Menu ────
├─WORD             R    Raisins
│ └─DOC            C    Chess
├─WINDOWS          L    Las Vegas Casino
│ └─PIF            F    Flight Simulator
├─GRAF             D    Doodler
├─NC               M    Main Menu
│ ├─DEMO           Q    Quit (Log Out)
│ └─SAMPLES
└ GAMES
  ├─RAISINS
  ├─CASINO
  ├─FLTSIM
  └─CHESS
C:\GAMES                          ..          ►UP--DIR◄  2-23-89  8:46p
C:\GAMES>
1Help 2User 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9Menu 10Quit
```

FIG. 3

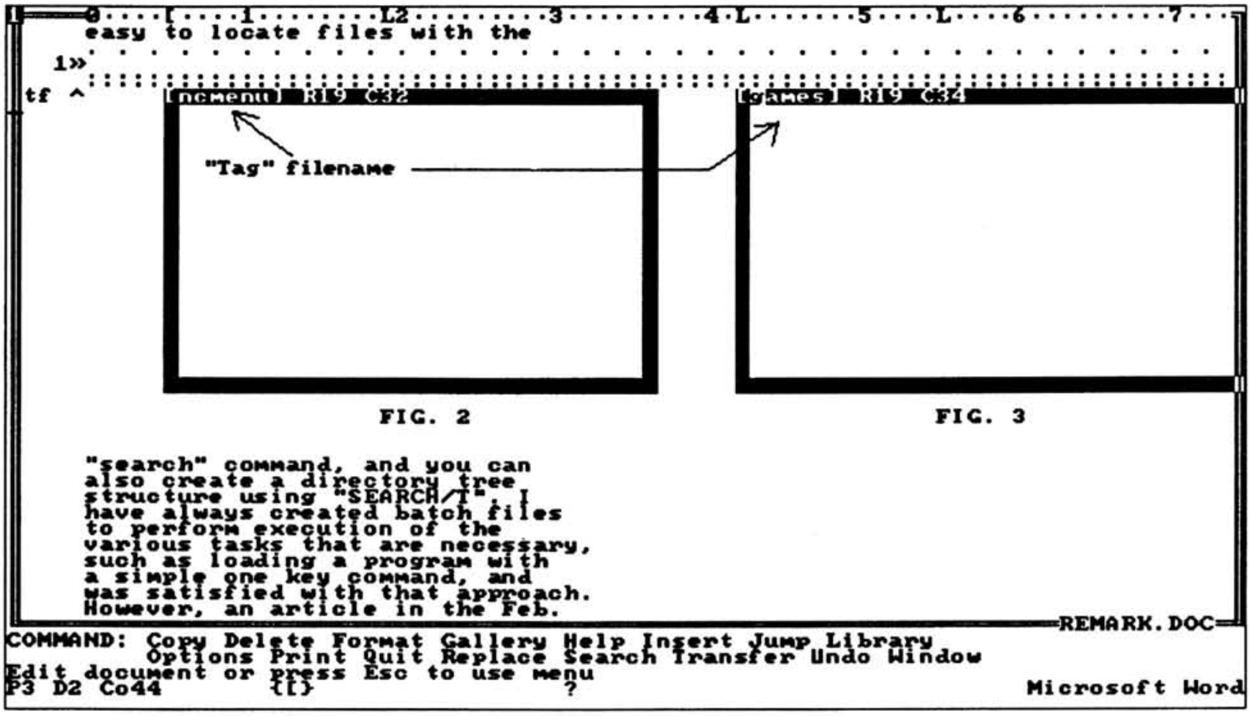
"search" command, and you can also create a directory tree structure using "SEARCH/T". I have always created batch files to perform execution of the various tasks that are necessary, such as loading a program with a simple one key command, and was satisfied with that approach. However, an article in the Feb. 1989 issue of REMARK ("A Power User's Shell", by Kevin R. Grantham) made me look at Norton Commander. I purchased a copy at Egghead Software shortly after reading the article.

Now, I won't go into a lot of detail on Norton Commander, because the referenced article does a very good job. Rather, I would like to explain how the menus have been arranged for convenience in the subject system, and maybe it will give some of you some ideas for your own systems. I like the ability of Norton Commander to create menus and sub-menus. The sub-directories can each have a menu, if desired, which is automatically displayed when that sub-directory is the default. Otherwise, the main menu is displayed. Certain selections on the main menu change the default directory to one which contains a sub-menu. There is a selection on each sub-menu which will return you to the root directory and the main menu. Fig. 2 shows the main menu on my system, with the Norton Commander panels in the background. Now, I am continually modifying my system, as I suppose most people do, so by the time this article is printed (IF it is), the menus will probably be different. But that's the beauty of Norton Commander; it makes reorganization much easier.

On the main menu, function keys [F1-F6] have been designated for most of the hot keys, the remainder have been assigned alphabetical characters. Most of these key code functions are obvious, I'll explain those that are not quite so apparent. As each of the programs or commands assigned to these hot keys terminate, the main menu is automatically brought up. All of these menu commands replace batch files which I used previously. So this system actually saves disk storage area by consolidating many batch files into a few menus.

Function key [F3] invokes the DOS mode command to set up the HP 7550A plotter configuration on COM1:. [F4] changes the default drive to D: and invokes the Software Carousel, which in turn loads various BV Engineering software programs. More about that later. [D] for Grafplus is a TSR (Terminate and Stay Resident) screen dump program for outputting screen graphics to a printer. [S] for Schematic Capture, invokes OrCad SDT, a schematic capture program which is on drive D.

There is an automatic logging system which tracks the hours spent on the computer; or rather, tracks the hours the computer is on. The [Q]uit command executes the Logout program, which is written in BASIC. When the computer is powered up, or re-booted, the Login program automatically executes. These programs maintain a file which reads the



```
[=====0····[····1·········L2·········3·········4·L·······5····L····6·········7···]
     easy to locate files with the
  1»
tf ^      [ncmenu] R19 C32                        [games] R19 C34

             "Tag" filename

                    FIG. 2                                    FIG. 3

     "search" command, and you can
     also create a directory tree
     structure using "SEARCH/T". I
     have always created batch files
     to perform execution of the
     various tasks that are necessary,
     such as loading a program with
     a simple one key command, and
     was satisfied with that approach.
     However, an article in the Feb.
=====================================================================REMARK.DOC=
COMMAND: Copy Delete Format Gallery Help Insert Jump Library
         Options Print Quit Replace Search Transfer Undo Window
Edit document or press Esc to use menu
P3 D2 Co44          {[}               ?                           Microsoft Word
```

FIG. 4

time on and time off. When the file is full, a message prints out on the screen after booting up. The [M] command is for maintaining these records, when the file is full, which happens about once a month.

The Inset program [I] allows integrating text and graphics. It, like Grafplus, is a TSR program. I use this program for writing technical reports and papers where graphs are needed along with the text. For instance, a graph created with one of the circuit analysis programs can be saved, and integrated into a document being composed in WORD. The Inset program has to be loaded before creating the graph so it can be saved, and it has to be in memory before WORD is loaded, in order to integrate the text and graphics.

The communications submenu accesses the directory with the HUGMCP modem communications program. There is also a Games [G] submenu which has Flight Simulator, Chess, and a couple of novelty games which my granddaughter enjoys when she is visiting. Both of these menus have options which will return you to the main menu, or you can log out (Quit) directly from that menu.

Fig. 3 shows the Games menu. Behind the menu are the left and right panels of Norton Commander. I normally don't have the panels on, but turned them on for this illustration.

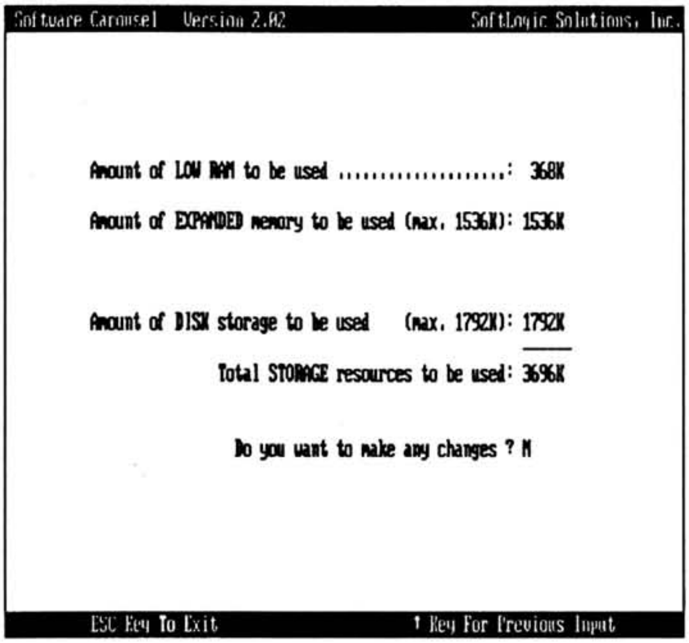
INSET

As mentioned earlier, InSet allows integrating text and graphics in a word processor. I have only used it with Microsoft Word, so my comments are restricted to applications with Word.

I am doing this article with Word and InSet. Except for the title at the beginning of the article, the graphics have been created directly from the screens displaying Norton Commander, or as in Figure 4, from the Word screen. The title was created with Windows Paint. I have also created graphics with Lotus 1-2-3 and BV Engineering's PCPLOT, and integrated them into Word documents. In each case, the graphics is saved with InSet in a PIX file format, and inserted in the Word document at the desired location.

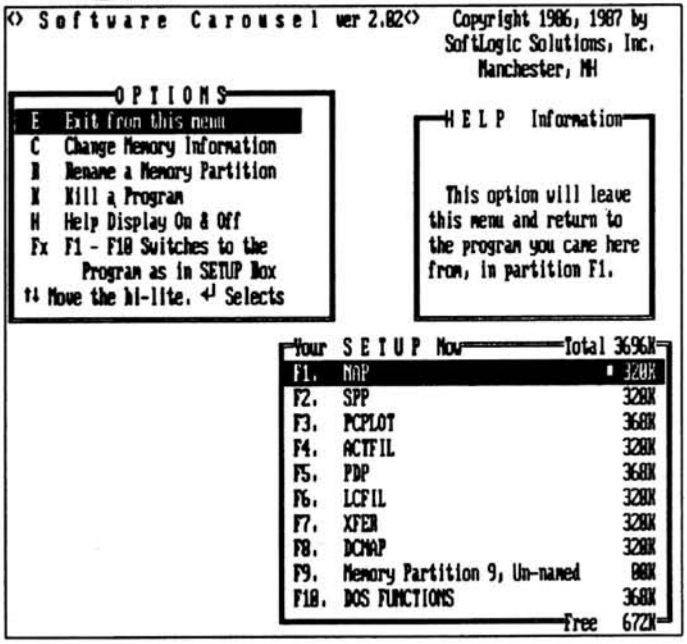
The program is a little awkward to use, but with some practice, you can do a credible job. The difficult part is locating the graphics in the text. That is, Word 4.0 does not have a WYSIWYG (what you see is what you get) display. If you are using a proportional spaced typeface, such as Times Roman, the spacing on the screen for the text and the graphics is not the same, because the graphics is based on fixed spacing. So the only thing you can do, is make a few test runs on the printer until the graphics is located as you want it.

The graphics to be integrated is first created and saved before loading Word,



```
Software Carousel   Version 2.02                    SoftLogic Solutions, Inc.

        Amount of LOW RAM to be used .....................:  368K

        Amount of EXPANDED memory to be used (max. 1536K): 1536K


        Amount of DISK storage to be used    (max. 1792K): 1792K
                                                           -----
                        Total STORAGE resources to be used: 3696K

                         Do you want to make any changes ? N

        ESC Key To Exit                        ↑ Key For Previous Input
```

FIG. 5



```
<> S o f t w a r e   C a r o u s e l  ver 2.02<>   Copyright 1986, 1987 by
                                                   SoftLogic Solutions, Inc.
                                                       Manchester, NH
         O P T I O N S
 E   Exit from this menu                      H E L P   Information
 C   Change Memory Information
 R   Rename a Memory Partition                 This option will leave
 K   Kill a Program                           this menu and return to
 H   Help Display On & Off                    the program you came here
 Fx  F1 - F10 Switches to the                 from, in partition F1.
        Program as in SETUP Box
 ↑↓ Move the hi-lite. ↵ Selects

                             Your  S E T U P  Now            Total 3696K
                             F1.   NAP                          ▪ 320K
                             F2.   SPP                            320K
                             F3.   PCPLOT                         368K
                             F4.   ACTFIL                         320K
                             F5.   PDP                            368K
                             F6.   LCFIL                          320K
                             F7.   XFER                           320K
                             F8.   DCNAP                          320K
                             F9.   Memory Partition 9, Un-named    00K
                             F10.  DOS FUNCTIONS                  368K
                                                             Free  672K
```

FIG. 6

unless it is a Word screen as in figure 4. For instance, Figure 1 at the beginning of this article, was saved right after I loaded InSet. The "hot key" combination for InSet is <left-shift control>. This combination brings up a menu in the lower two lines of the screen, which allows you to Save or Modify the displayed image, whatever it is. Now, I wanted to save an image with the Inset menu showing for this article, but couldn't figure out how to do it. When the "Save" command is invoked, naturally the menu disappears until the save operation is complete. I guess that makes sense, because you wouldn't ordinarily want that menu showing at the bottom of your graphics.

In the Word document, you leave space for the graphics, and place a "tag" in the upper left hand corner of the graphics area. The default characters for the tag is "[filename]" (without the quotes); where "filename" is the name given the graphics when it was saved by InSet. It is automatically assigned a ".PIX" extension, which is not used in the tag. InSet recognizes the square brackets [], and looks for a PIX tag within them. Figure 4 is the Word screen for this article directly beneath figure 2 and figure 3. Figure 2 was saved as "NCMENU.PIX" and figure 3 as "GAMES.PIX". The numbers following the tags in the illustration were put there by Inset, and indicate the size of the frame for the graphics. As you can see, figure 2 was 32 columns wide and figure 3 was 34 columns. I corrected this later so they both have 34 columns. This figure was created by invoking Inset with the <left-shift ctrl> key combination. When the InSet hot keys are invoked with a pix tag on the screen, an outline forms where the graphics image will be merged. This outline can then be sized, both vertically and horizontally. You can also add text, and draw lines and circles in this mode. I added the caption and arrows pointing to the "Tag" filenames.

As mentioned before, it takes some practice to get the printout to look as you want it, but it is much easier than the old cut and paste method with which I am very familiar; and the end result is well worth the effort.

SOFTWARE CAROUSEL

Software Carousel is a task switching program. It is memory resident, and in fact utilizes any type of memory that your computer may have; extended or expanded. In addition, it will use available disk space for temporary storage of programs. It assigns memory "partitions" for each program assigned in a given situation.

In my system I have eight programs, all related to electronic circuit design and/or analysis, which can be loaded into memory and switched between with Carousel. This is very convenient, because, for instance, I may be analyzing a circuit with ACNAP and want to do a Bode plot. I can switch to PCPLOT, the screen plotting program, and do a plot without removing ACNAP or the circuit from memory.

When Software Carousel first comes on, the screen in

Figure 5 is shown. At this point, you can modify the memory resources which you want to use with Carousel. It automatically recognizes your system memory and disk storage capacity. It will use both extended and expanded RAM if you have it available. The first time it is loaded, a file called "CAROUSEL.OPT", or Carousel Options file is created. This can also be created and/or modified with a text editor. This allows you to change memory and program partition data.

Figure 6 is a view of the Software Carousel menu. This menu can be invoked at any time by [Alt-Space] after Carousel is loaded into memory. The programs listed under "SETUP" are: [F1] ACNAP (AC circuit analysis), [F2] SPP (signal processing and FFT), [F3] PCPLOT (graph plotting software), [F4] ACTFIL (active filter design software), [F5] PDP (plotter driver program), [F6] LCFIL (LC filter design software), [F7] XFER (transfer function analysis/synthesis software), and [F8] DCNAP (DC circuit analysis program). [F10] has been reserved for DOS functions. Under ordinary conditions, this menu doesn't appear. It is used for setting up the memory partitions, and changing them. After Carousel has been set up, it can be made to automatically load the first program [F1]; and then each succeeding program can be loaded automatically with [Alt-left shift]. After the programs have been initially loaded, you can switch between them with [Alt-F#] using the appropriate function key. Carousel uses any memory that may be available, and disk space for swapping programs. Of course, when it is swapping to the disk, it is somewhat slower than in memory. Some speed is gained by loading only part of the programs.

Normally, only those programs which will be used in a particular session are loaded, because it speeds up the switching action. For instance, there may be a new circuit design to analyze, so ACNAP, SPP, PCPLOT, and PDP are loaded. You can switch between these programs very rapidly, because the switching will take place solely in memory, not on the disk.

There are a number of ways in which Software Carousel can be utilized. I have described how it is applied to my particular needs. If, for instance, you do a lot of writing, you can have different chapters of a book in each of the memory partitions. This is accomplished by starting your word processor in each of the partitions and loading a different chapter for each partition. This allows you to switch between the chapters without saving them. In effect, you are expanding the memory capabilities of your word processor. Microsoft WORD is limited to the base memory available in the system. When it is installed in Carousel partitions, each partition can have 512K memory. The same game can be played with spreadsheets, with a different spreadsheet in each partition.

CONCLUSIONS

I like Software Carousel; it offers a lot of flexibility in the work I do. Unfortunately, when I am through with the programs it is running, due to the nature of Software Carousel, I have to re-boot, unless I am ready to quit for the current session anyway. Carousel literally takes command of your system. You can still execute DOS commands, but Norton Commander, for instance, has to be re-loaded if you want to use it. One problem is, Carousel uses memory like its going out of style. Of course, that's really what it is designed to do. It is a task switching program, with which you can switch from one program to another, while still maintaining both of them in memory. You don't lose a thing when you switch between the various programs. And this takes memory, a lot of it if you have many programs to switch.

One of the problems commonly encountered with TSR programs, is that they all want to be the last one loaded. The Norton Commander manual claims that it is not a memory-resident program and, as a result that it won't interfere with other memory resident programs. I have a problem comprehending how it is not memory resident. It didn't appear to cause any problems, at any rate. However, care must be exercised when using programs such as Grafplus, Inset, and Software Carousel. I was able to hang the system when Inset was in memory and Carousel loaded. I did this to capture the Carousel screens. I was able to do that, but immediately after the screen is "saved" by Inset, the system would lock up. So to capture the two Carousel screens, I had to re-boot twice.

While doing a print-out from WORD for this article, the LaserJet ran out of paper, which happens from time to time, and the system hung again. Fortunately, I had just done a "save" before I started the printout (I always do; I learned that a long time ago!). This was somehow related to Inset, because it was in the middle of doing one of the graphics print-outs when it hung up. If I use WORD for doing text, without Inset loaded, the print out simply aborts when the paper-out error occurs, and the system does not hang up.

I have been practicing Engineering as a profession for over 30 years. The transistor was invented while I was in college. I have witnessed a number of metamorphoses in the electronic industry; the transition to transistors, IC's, digital electronics, and computers. I would have had difficulty believing 30 years ago, the power of the personal computer I have in my home office today. In minutes, I can solve problems with it that would have taken days or weeks to solve 30 years ago. So with that little bit of perspective, the so-called "problems" mentioned here are mere pittances!

PRODUCTS MENTIONED

Inset
Inset Systems, Inc.
71 Commerce Drive
Brookfield, CT 06804
(800)828-8088

Software Carousel
SoftLogic Solutions
530 Chestnut Street
Manchester NH 03101
(603)644-5555

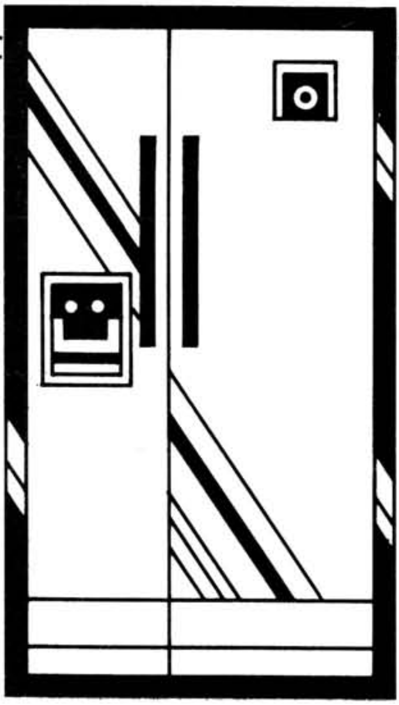
BV Engineering
2023 Chicago Ave.
Suite B13
Riverside, CA 92507
(714)781-0252

Norton Commander
Peter Norton Computing
100 Wilshire Blvd.
9th Floor
Santa Monica, CA 90401
(213)319-2000

*

How to Get your Computer to Talk to your Refrigerator

A Discussion of Computer Control of Non-computer-related Devices Via the A-Bus System



William G. Nabor
2772 Huerta Street
Mission Viejo, CA 92692

Zenith computers are provided with a number of I/O ports that allow them to communicate with you, the user, as well as other computers and a host of computer peripherals. These peripherals could be printers, modems, digitizers, frame grabbers, scanners, plotters, or any collection of specialized pieces of equipment whose numbers are increasing daily. As awesome as this list is, all of the above devices share one thing in common: They are designed, either primarily or incidentally, as computer-related devices. They have a specialized mechanism for connecting to a PC, by a plug-in card, serial port, or whatever. But what if you want to have your computer talk to your refrigerator? There is no RS-232 port next to the compressor (not on the current crop of refrigerators, anyway). The following discussion will explain one low-cost way to do this.

No, I don't really think that anyone would want to connect their refrigerator to a computer. I have chosen this example because it demonstrates the principles involved, as well as any other illustration I might have chosen (lights, alarms, door openers, literally all the bells and whistles you can imagine). If you can get your computer to talk to your refrigerator, you can connect it to almost any electrical device. But how is this done?

If you are only concerned with turning the appliance on and off, then an X-10 system, such as the Heath BC-290P, might work. For those who are unfamiliar with the X-10 system, it consists of a controller and a set of one or more modules. The modules plug into a wall socket, the appliance to be controlled plugs into the module. The controller can be operated by either the human user by pushing a button or, in this case, a computer via the serial port. When given the command to, say, turn on the appliance attached to module #1, the controller sends a coded pulse through the building wiring to all the modules currently plugged in. If the module is not #1, the pulse is ignored. Module #1 senses that this pulse is meant for it and turns on (or off, as the case may be). Wonderful, but there are several limitations to the X-10 system:

1. The modules must be plugged in to an AC outlet on the controller's side of the nearest transformer and reasonably near the controller. This is not a problem in your home, but what about your place of business?
2. The controller can only switch on and off 120 volt AC current (or in the case of incandescent lights, dim them). If you want to switch a low voltage and/or DC circuit, then you must install a 120 volt relay.
3. The modules are about the size and shape of a cigarette package and while this is not large, several of these modules can take up an unreasonable amount of space. Let's say you want to create an annunciator panel with 8 LED indicator lights. You would have to install 8 X-10 modules, together with 8 relays. This could easily consume more space than you have. You cannot plug in more than one module in a standard duplex outlet (they won't fit). Therefore, if you want to use more than one module with the same outlet you have to use adapters and/or extension cords. Ugh!
4. Switching is rather slow. The controller must receive the command, then code and send the pulse. This takes at least 2 seconds. While this is not a long time, it does preclude using this system for precision control. What's more, only one command can be given every couple of seconds. You could control several devices by plugging them in to a single module, or into several modules all of which are designated with the same code, but then you give up the ability of controlling them separately.
5. While the X-10 system is reasonably reliable, there are times when the modules fail to recognize their code and do not respond. For critical applications (turning off a pump, for example, when the tank is full) this is too dangerous to work. The human user might be able to see this and hit the button again, but the computer couldn't, which leads to the biggest problem with the X-10 system:
6. While you can easily get information from the computer to the devices in the field, there is no provision for getting information back to the computer. The computer can turn on the refrigerator, but cannot determine the inside temperature. Therefore, there is no way of constructing a feedback loop for real process control. The computer must act blind — a major failing.

Of course, computer control of equipment is too valuable a service to ignore. There is a whole standard system, IEEE 488, which provides first class methods of interfacing a computer with the real world. The trouble here is that IEEE 488 systems are expensive and not always easy to program. Most advertisements for IEEE 488 systems do not list prices (well, if you have to ask . . .). Also, the IEEE 488 cards, which come in a variety of different configurations for different purposes, each take up a slot in your computer. If your needs expand beyond what you purchased, you lose another slot. This can easily necessitate more expense for an expansion box.

Enter the A-Bus system from Alpha

Products of Darien, Connecticut. This collection of modules and controllers is designed to be affordable, easy to use, versatile, and expandable without hogging your computer's resources. It is a very good way to provide real-world connections for your Zenith computers, including the Z-100 or '89. An A-bus system would consist of a controller, power supply, and one or more modules (somewhat like the X-10 system mentioned above, but with vastly improved versatility). There are 12 different kinds of modules and 3 types of controllers. A single controller can manage up to 25 different modules, using special cables from the controller to the modules and/or a "motherboard" that accepts up to 5 modules. These can be "daisy-chained" as needed, up to the 25 module limit. It would be hard to come up with an application that would require more than 25 modules (especially since most modules provide several individual I/O ports each). If you did have such a need, then you can add more controllers, each of which can manage another 25 applications modules. You will need either an expansion slot or a serial port (but not both) for each controller card, or else a serial port and one or more A-bus "nodes". No software is available, the system is designed to be accessed via standard BASIC commands, such as OUT and IN or PEEK and POKE, for which there are examples given in the A-bus instructions. You can also use other high-level languages (e.g., FORTRAN, C), but here you are on your own for the exact syntax to correspond to BASIC's IN and OUT commands as there are no instructions for these languages in the A-bus literature.

One of the nice features is that the system is modular. You buy only what you want, but you have to buy each piece separately. As a minimum you will need a controller, power supply, one applications module, a 50-conductor connector cable, and BASIC (or other suitable high-level languages). There is nothing proprietary about the cables or power supplies, you can make them yourself form readily available parts, although I recommend purchasing them directly from Alpha Products. Any 12V DC supply will work, if you have one already.

The Controller Cards

At the heart of the A-Bus system is the controller. It provides the interface between your computer and the applications cards. By separating the applications and the computer interface in this way, the A-bus system allows for 25 different applications to be ganged together without requiring any additional computer resources. There are three different types of controller cards: For expandable PC's (e.g., Z-150, Z-248, Z-386) there is a plug-in card that is inserted in an available slot in the computer. Borrowing from mouse terminology, I will call this type the "bus controller". There are versions for Zenith PC's and clones, Apple, Tandy, and Commodore. For other types of computers (e.g., Z-100, Z-89), there are two types of serial controllers that connect via an RS-232 serial port. The first serial controller is just that, a controller card that connects via a serial port with or without a modem. The second one is like the first, but also contains a nifty Intel on-board computer complete with a special built-in BASIC interpreter. This ultimate controller can be programmed via your computer, then disconnected and allowed to do its thing on its own, freeing your computer for other work. At this point, the computer can even be switched off. In short, this card is a little computer in itself.

You may connect as many as 16 serial controllers (of either type) to a single RS-232 port by the use of an A-bus Serial Node (one node per controller). You may also put the serial controller(s) anywhere in the world that has phone access and connect them to your computer via a pair of modems and your own communications software. You want to turn on the lights in your home from your hotel room in Tokyo and then confirm that they, indeed, went on? Plug in your SupersPort in Tokyo and activate the A-bus serial controller at your house, command the relay card to activate relay #1, then query the digital input card's input #1 (connected to a simple Radio Shack light sensing diode) to see if there is light. While you are at it, check on the temperature (analog input card w/thermistor).

Connections

Applications modules plug into the controller(s). Each module contains a 50-pin edge card connector, which must be connected to the corresponding 50-pin header on the controller. These can be daisy-chained together up to a 25-unit limit. To do that, you need a 50-conductor cable with a 50-pin header connector on one end and one edge connector for each module on the other. You can make one yourself, but it is wise to get one already made, as it's a little more involved than making a simple D-25 serial cable. I advise getting the CA-162 cable which has two edge connectors, allowing for 2 modules to be attached. If you have more than two applications modules, you might want to go with the A-Bus "motherboard", a rigid aluminium frame with firm slots for 5 modules. Again, these mothers may be daisy-chained.

Power

Of course, the applications modules must be powered. They need 12 Volts DC. Any suitable power supply may be used for this, or you can purchase one from Alpha Systems. Usually, the connection is made via a mini plug (each module has an appropriate jack), but you can solder the connection if you are willing to give up the warranty. A single power supply can power many modules. The actual number depends on the modules and applications. For example, it takes more current to activate 8 relays than one, and it takes more power to activate a mechanical relay than a transistor. If you daisy-chain several modules, you need connect the power supply to only one of them. The 50-conductor cable devotes one pair to supplying power to all modules connected, including the serial controllers. The bus controller, of course, is powered by the computer.

Applications Modules

The business end of the system is the applications module. Currently there are 12 different modules available, plus a prototyping card if you want to cook up your own. Together these do almost everything you could reasonably want to do. Here is a brief description of each:

1. Relay card: This card provides 8 individual "industrial" mechanical SPST relays. The computer can activate these relays in any combination with a single command. This card provides on-off control, and is the counterpart of the X-10 system, except that it will switch any low-current (3 amps @ 30V DC or 120V AC resistive; 1.2 Amps @ 120V inductive) device you attach. Think of it as a card with 8 toggle switches that the computer can flip at will.
2. Reed Relay card: This is the same as the above, except that it uses reed relays for switching extremely low (20 mA or less) current.
3. Analog Input card: This card allows you to send eight analog (0 to +5 volts or 0 to +100V with a resistor) signals to the computer. The card will, upon a command from the computer sample the voltage at any of the 8 inputs and convert this voltage to a number which the computer can then use for further decision making.
4. 12-bit Analog card: Same as above, but a single input in the range of -4 to +4 V DC or -100 to +100 V DC.
5. Digital to Analog converter: The opposite of the previously mentioned analog cards. This one takes a number generated by the computer and converts it into a 0 to 8 V DC signal for output.
6. 24-line TTL I/O card: This one controls 24 in and out analog signals.
7. Digital Input card: This accepts up to 8 incoming on-off signals of up to 100 V DC and tells the computer which channels are on or off. The connections are optically isolated so that the computer will not be damaged by the incoming voltage.
8. Digital Output Driver: Similar to the relay cards, but provides its own voltage

(250 mA @ 12 V) for directly activating solenoids, relays and such.

9. Clock card: Provides a clock, calendar, alarm, buzzer, and alarm relay for timing.
10. Touch-Tone Decoder card: This card takes as input the tones generated by a telephone-style touch-tone generator (not included) and converts them into a number for the computer's use.
11. Counter Timer: Use this to count events (3 channels of input).
12. Stepper Motor Controller: This card is used in conjunction with stepper motors for robotic control applications.

More applications cards are promised in the future.

Use/Programming

The A-Bus system, as I said before, is designed to be programmed using BASIC, but other high-level languages can be used. The BASIC commands needed for most of the cards are INP and OUT. You can refer to your Zenith documentation for a complete description of these commands, but what they do here is instruct the computer to send a number to or receive one from a designated port. The port in this case is the A-bus controller card. The syntax of the command is OUT X,Y or Y=INP(X). X represents the port address where the data are to be sent (OUT) or received (IN) and Y is the actual number to be sent or received. Since you must tell the computer where you want the number sent (the address), you must assign the A-bus controller card and each applications card a number representing an instruction to the computer telling it where to send or receive data. This is done at the time of installation by inserting a jumper into a well marked DIP socket on the A-bus card. Because the addresses are hardware determined, they are meant to remain the same for the duration of the use of the A-bus system, but they can be changed by removing the card(s) and moving the jumper(s) — you would then have to change all your A-bus-using programs to reflect this change, so it is a step not to be done for light or trivial reasons. You must take care not to assign the cards an address that conflicts with another port you are using for something else. If you don't know what addresses you are currently using, try leaving the A-bus card at the default position (0) and see by trial and error if it works. It probably will. If not, reinstall the jumper at another position.

The address you select for the controller card becomes the base address for the entire A-Bus system. For the purposes of discussion, let's say you decided to set the base address to 64. You would then insert the jumper wire in the slot marked 64. Each application module must now have its address set to a number that is different from all other modules (but one of them can be the same as the base address on the controller). This is also done by configuring jumpers. Again for discussion, let's assume you are installing the relay card and set its card address to 128. The port address of the relay card then becomes the base address (64) plus the card address (128) or 192. A second applications card would have to have its card address set to a number different from the relay card's 128. Let's say you install a digital input card and set its card address to 0. The port address of the input card would then be the base address (64) plus 0 or 64.

To activate the relay card you would then use the BASIC command OUT. The command would be OUT X,Y; where X is the port address of the card and Y is the number code of one of the 8 relays on the card. The 8 relays are assigned numbers 0, 1, 2, 3, 4, 5, 6, and 7. Their code is simply the number 2 raised to the power of the relay's number. For example, the first relay (#0) has the code of 20 or one. The eighth relay (#7) is assigned the code 27 or 128. To turn on the first relay, using our above example, then, you would simply issue the BASIC command OUT 128,1. Remember that the number 128 is the relay card's address, so the 128 in the OUT command tells the computer to send the number 1 to the relay card, which then interprets that as a command to turn on relay #1. To turn all relays off, send OUT 128,0. To turn on certain relays, add up their codes and send that number. For example, if you want to activate relays #0,1 and 6, send their total (1+2+64=69) — OUT 128,69. To turn relays #0 and 1 off while leaving #6 on, send only the code for #6 (64) — OUT 128,64. Whatever the configuration the output card was in before this command was given, OUT 128,64 would turn off all relays but number 6, which would then be turned on (or left on). You can also assign variables and send the commands that way:

```
10 X=128
20 Y=2^6
30 OUT X,Y
```

will also turn on relay #6 and turn all others off.

```
10 X=128
20 R=2+3+4
30 Y=2^R
40 OUT X,Y
```

would turn on relays #2,3, and 4 and turn all others off.

The input card works the same general way, but in reverse. The command is Y=INP(X), or in the case of our example would be Y=INP(64). Remember that 64 is the address we chose for the input card. Issuing this command would cause the input card to return a number which the computer would assign to the variable Y:

```
10 Y=INP(64)
20 PRINT Y
```

The input card assigns Y a value from 0 to 255, with 255 representing no connections and 0 representing all eight connections. Since this is somewhat the reverse of what humans would expect, then changing line 10 above to

```
10 Y=255-INP(64)
```

would return the binary code of the closed inputs. That is, if Y was 1, then the input switch #1 was closed. If Y was 16, then switch 4 (24) was closed. If Y was 24, then switches 3 and 4 were closed (23 + 24 = 24), and so forth. Examples are given in the instructions for the use of the boolean logic functions (e.g., AND) to do "bit masking" to return the actual number of the switches closed without having to resort to complicated programming. For example, to test whether or not a particular input is connected, use the commands

```
10 INPUT
   "Which port do you wish to test";N
20 IF (INP(64) AND 2^N)=0 THEN PRINT
   "Input "N" not connected"
30 IF (INP(64) AND 2^N)<>0 THEN PRINT
   "Input "N" not connected"
```

Line 10 asks the user to input the number of the port he wishes to test. Lines 20 and 30 do the checking and report the results. These commands are not any more difficult than most BASIC instructions and the examples given in the instructions are adequate to let you take full advantage of the A-bus system (even if these instructions are not quite up to Zenith standards).

The serial controllers are used by opening up a COM channel and sending the commands that way:

```
10 OPEN "COM1:9600,N,8" AS #1
20 WRITE #1, X,1 'REM 'X'
   in this case represents the serial
   card's base address plus the output
   card's address
```

Back to Your Refrigerator

Here, then would be a system to control your refrigerator. You would need a controller card (set to port 0), relay (output) card (set to port 2), digital input card (set to port 4), analog input card (set to port 8), mother board, power supply, cable, hook-up wire, 15A 12V power relay, thermistor, and a few appropriate resistors and capacitors. Use the output card's #1 relay to drive the power relay. This power relay would switch on the compressor (which draws more power than the relay card can safely handle). Use the output card #2 relay directly to switch the light. Wire the thermistor in series with the power supply and the analog input's #1 input. Using resistors, connect the power relay's output to the digital input card's #1 input. Connect the light in parallel with the digital input's #2 input. Attach alarms to the output card's other relays. The program could go something like this:

```
10 CLS
20 IF (INP(4) AND 2)=0 THEN LOCATE
   10,1: PRINT "Refrigerator is not on"
```

```
30 IF (INP(4) AND 2)<>0 THEN LOCATE
   10,1: PRINT "Refrigerator is on"
40 Y=INP(8)
50 Y=Y/CF 'REM CF is a calibration fact
   or to convert the Y value to degrees
60 IF Y>40 THEN OUT 2,1 ELSE OUT 2,0
70 IF Y>50 THEN OUT 2,8
80 Q$=INKEY$
90 IF Q$="" THEN GOTO 20
100 INPUT "Do you want the light on",L$
110 IF L$="Y" OR L$="y" THEN OUT 2,4
120 GOTO 10
```

Line 10 would clear the screen. Lines 20 and 30 test the digital input card (at port 4) #1 input ($2^1=2$) to see if power is coming from the power relay. If it is, then we assume that power is also going to the compressor and state that the refrigerator is on (line 30). If not, then we assume that the compressor is off and so state in line 20. Line 40 requests input from the analog input card. This card is connected to the thermistor which is placed inside the refrigerator. This device changes its resistance with temperature, and since it is in series with the 12V power supply, sends a voltage from 0 to 12 to the input card. The card then converts this voltage to a number which is then calibrated by the user to a temperature. If this temperature is greater than 40 degrees F, then we activate the output card (port 2) #1 relay (line 60), which turns on the compressor. If the temperature is 40 degrees or less, then we turn the compressor off (also line 60). If the temperature increases to more than 50 degrees we activate the output card's #3 relay ($8=2^3$) which turns on an alarm connected to relay #3 (line 70). Line 80 checks to see if the user wants to do something. If the user has not pressed a key, the program loops back to check the status again (line 90). If the user has pressed a key, then the program asks if he wants to turn on the light (line 100). If he answers Y(es), then relay #2 is activated (line 100 - $4 = 2^2$). If not, then we loop back to the start. Magic!

Conclusions

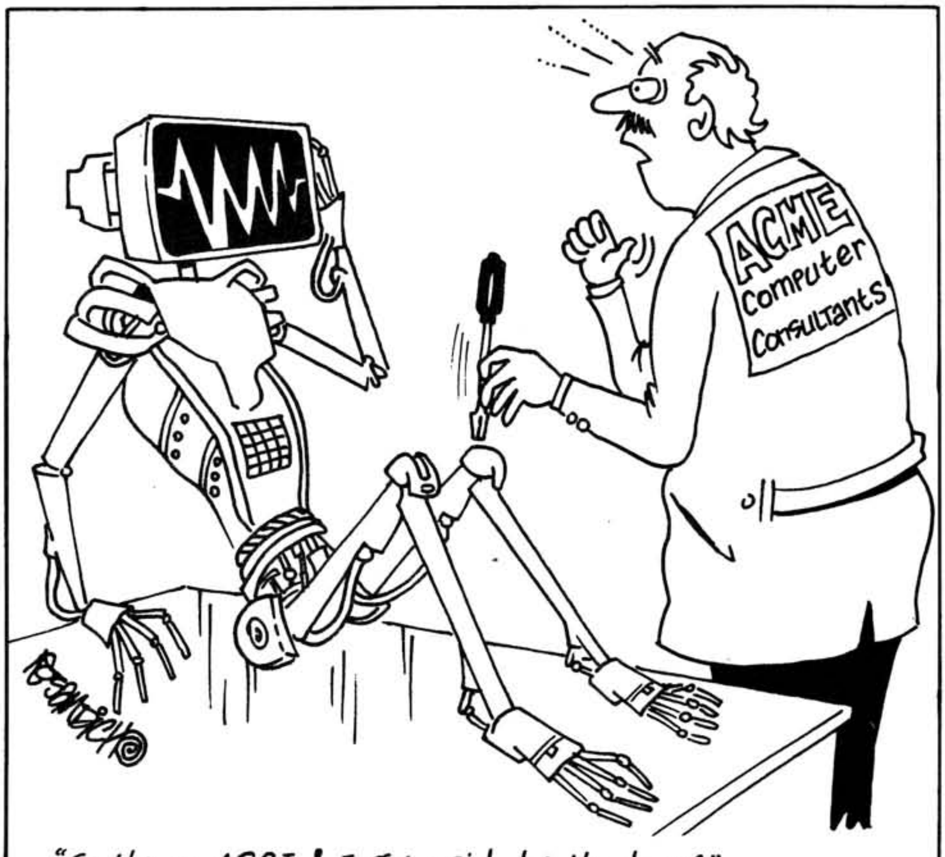
Even if you have no immediate need to interface your computer to the outside world, the A-bus system is fun. We computer hobbyists are well served by Alpha Products and its A-bus system. The modules are simple, relatively inexpensive, and easy to use. They would make excellent Heathkits (hint, hint, Heath). What you don't get, however, is a snazzy box to hide the works from view and accidental damage. All you get are bare circuit boards which (with the exception of the bus controller) cannot be tucked away inside your computer's case out of harm's way. You will be responsible to house your A-bus system in some appropriate manner to prevent, among other things, shorting out any leads. The warranty is a meager 90 days. However, there is no reason to suspect that these cards would not last as long as any card in your computer. These drawbacks are a small price to pay for the convenience of low-cost access to the world outside the computer.

If you are interested in any way in using computers as a hobby these boards are the best way of getting into environmental control. For those who need to use computers in this fashion and are on a budget, the A-bus system is the cheapest way to do it, and the cards do work as advertized.

Products & Prices

Alpha Products
242-B West Avenue
Darien, CT 06820
(203) 656-1806

Ar-133 Bus Controller	$69
SA-129 Serial Controller	$149
Sp-127 Serial Controller w/BASIC Processor	$189
CA-163 2-Conductor Cable	$34
PS-126 Power Supply	$12
MB-120 Motherboard	$108
SN-128 Serial Node	$49
AD-142 Analog Input	$142
AN-146 12-bit Analog Input	$153
RE-140 Relay Card	$142
RE-165 Reed Relay Card	$109
DA-147 Analog Output	$149
DG-148 24-Line TTL I/O	$72
IN-141 Digital Input	$65
ST-143 Digital Output	$78
CL-144 Clock w/Alarm	$98
PH-145 Touch-tone Decoder	$87
PR-152 Prototyping Card	$16
CT-150 Counter Timer	$132
SC-149 Stepper Motor Controller	$299



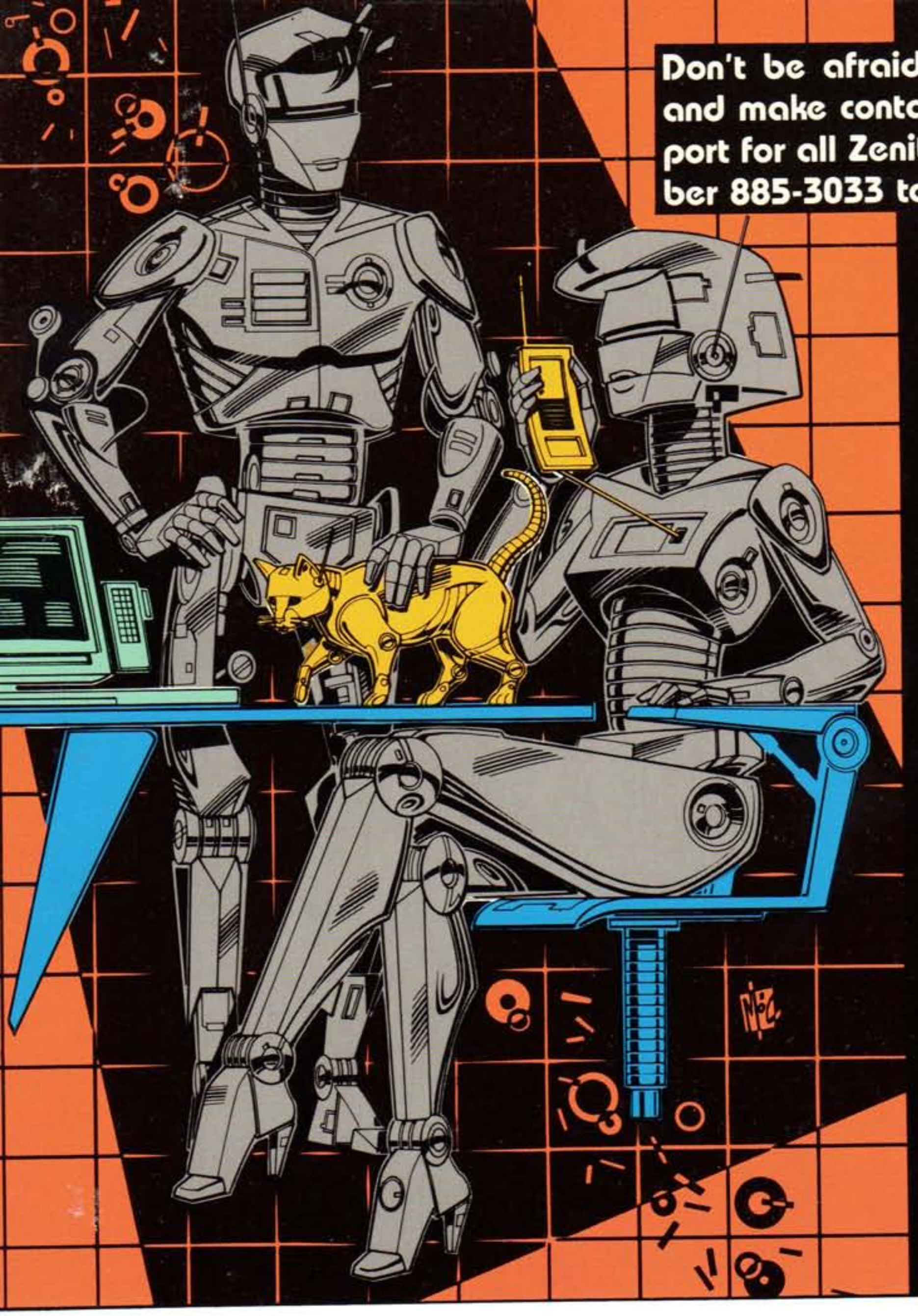
"Gentle my ASCI! I get paid by the hour!"

HADES II

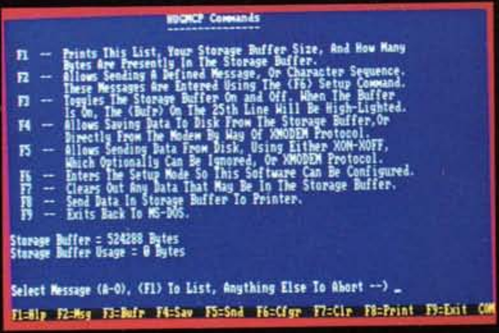
It's HOTTER than ever! Jam-packed with new features, HADES II still remains the easiest-to-use disk editor ever! Just look at some of the features:

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

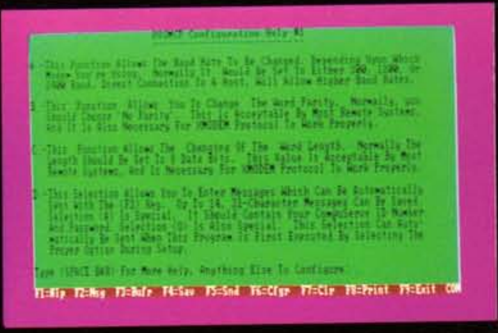
HADES II is still only $40, and original HADES owners can upgrade their distribution disk for only $15. Call HUG today at: (616) 982-3463.



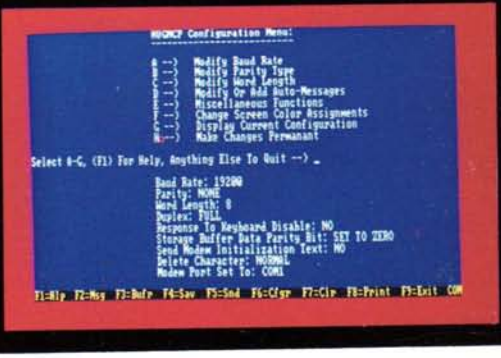
Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all Zenith Laptops, order HUG Part number 885-3033 today.



```
                        HUGMCP Commands
                        ---------------

F1  --  Prints This List, Your Storage Buffer Size, And How Many
        Bytes Are Presently In The Storage Buffer.
F2  --  Allows Sending A Defined Message, Or Character Sequence.
        These Messages Are Entered Using The (F6) Setup Command.
F3  --  Toggles The Storage Buffer On and Off. When The Buffer
        Is On, The (Bufr) On The 25th Line Will Be High-Lighted.
F4  --  Allows Saving Data To Disk From The Storage Buffer,Or
        Directly From The Modem By Way Of XMODEM Protocol.
F5  --  Allows Sending Data From Disk, Using Either XON-XOFF,
        Which Optionally Can Be Ignored, Or XMODEM Protocol.
F6  --  Enters The Setup Mode So This Software Can Be Configured.
F7  --  Clears Out Any Data That May Be In The Storage Buffer.
F8  --  Send Data In Storage Buffer To Printer.
F9  --  Exits Back To MS-DOS.

Storage Buffer = 524288 Bytes
Storage Buffer Usage = 0 Bytes

Select Message (A-O), (F1) To List, Anything Else To Abort --) _
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```



```
                  HUGMCP Configuration Help #1
                  ----------------------------

A -This Function Allows The Baud Rate To Be Changed. Depending Upon Which
   Modem You're Using, Normally It Would Be Set To Either 300, 1200, Or
   2400 Baud. Direct Connection To A Host, Will Allow Higher Baud Rates.

B -This Function Allows You To Change The Word Parity.  Normally, you
   Should Choose 'No Parity'.  This Is Acceptable By Most Remote Systems,
   And It Is Also Necessary For XMODEM Protocol To Work Properly.

C -This Function Allows The Changing Of The Word Length.  Normally The
   Length Should Be Set To 8 Data Bits.  This Value Is Acceptable By Most
   Remote Systems, And Is Necessary For XMODEM Protocol To Work Properly.

D -This Selection Allows You To Enter Messages Which Can Be Automatically
   Sent With The (F2) Key.  Up To 14, 31-Character Messages Can Be Saved.
   Selection (A) Is Special.  It Should Contain Your CompuServe ID Number
   And Password. Selection (O) Is Also Special.  This Selection Can Auto-
   matically Be Sent When This Program Is First Executed By Selecting The
   Proper Option During Setup.

Type (SPACE BAR) For More Help, Anything Else To Configure
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```



```
                   HUGMCP Configuration Menu:
                   --------------------------

                   A --)   Modify Baud Rate
                   B --)   Modify Parity Type
                   C --)   Modify Word Length
                   D --)   Modify Or Add Auto-Messages
                   E --)   Miscellaneous Functions
                   F --)   Change Screen Color Assignments
                   G --)   Display Current Configuration
                   H --)   Make Changes Permanant

Select A-G, (F1) For Help, Anything Else To Quit --) _

                    Baud Rate: 19200
                    Parity: NONE
                    Word Length: 8
                    Duplex: FULL
                    Response To Keyboard Disable: NO
                    Storage Buffer Data Parity Bit: SET TO ZERO
                    Send Modem Initialization Text: NO
                    Delete Character: NORMAL
                    Modem Port Set To: COM1
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```

Heath/Zenith Users' Group

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

$2.50

P/N 885-2117