The Official Heath/Zenith Computer Users Magazine

REMark®

September 1990

Octoport HC-1032

Page 5

Searching for Binaries

Page 9

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.

REMark®

September 1990

The Official Heath/Zenith Computer Users Magazine

PC Compatible

H/Z-100 and PC Compatibles

H/Z-100 Only

General

Resources

Reader Service No.		Page No.
104	FBE Research Co., Inc.	25
136	Lindley Systems	48
117	Payload	26
193	QuikData, Inc.	32
149	WS Electronics	48

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

Hardware Questions ...... (616) 982-3309

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

Contributing Editor...... Robert C. Brenner

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

H8 - H/Z-89/90

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ACCOUNTING SYSTEM	885-8047-37	CPM	BUSINESS	20.00
ACTION GAMES	885-1220-[37]	CPM	GAME	20.00
ADVENTURE	885-1010	HDOS	GAME	10.00
ASCIRITY	885-1238-[37]	CPM	AMATEUR RADIO	20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PACKAGE	885-1119-[37]	HDOS	UTILITY	20.00
CASTLE	885-8032-[37]	HDOS	ENTERTAINMENT	20.00
CHEAPCALC	885-1131-[37]	HDOS	SPREADSHEET	20.00
CHECKOFF	885-8010	HDOS	CHECKBOOK SOFTWARE	25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY	20.00
DISK UTILITIES	885-1213-[37]	CPM	UTILITY	20.00
DUNGEONS & DRAGONS	885-1093-[37]	HDOS	GAME	20.00
FLOATING POINT PACKAGE	885-1063	HDOS	UTILITY	18.00
GALACTIC WARRIORS	885-8009-[37]	HDOS	GAME	20.00
GALACTIC WARRIORS	885-8009-[37]	CPM	GAME	20.00
GAMES 1	885-1029-[37]	HDOS	GAMES	18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY	30.00
HDOS PROGRAMMERS HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUPLICATION UTILITIES	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PRODUCTS THRU 1982	9.75
HUGMAN & MOVIE ANIMATION	885-1124	HDOS	ENTERTAINMENT	20.00
INFO. SYSTEM AND TEL. & MAIL SYSTEM	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DATABASE	25.00
MAPLE	885-8005	HDOS	COMMUNICATION	35.00
MAPLE	885-8012-[37]	CPM	COMMUNICATION	35.00
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)

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
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
NAVPROGSEVEN	885-1219-[37]	CPM	FLIGHT UTILITY	20.00
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

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.

		OPERATING		
PRODUCT NAME	**PART NUMBER**	**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
DUNGEONS & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSODS	PRINTER PLOTTING UTILITY	25.00
GAMES (ZBASIC)	885-3011-37	MSDOS	GAMES	20.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHIC GAMES (ZBASIC)	885-3004-37	MSDOS	GAMES	20.00
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
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
BACKGROUND PRINT SPOOLER	885-3029	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 II	885-3040	MSDOS	UTILITY	40.00
HELP	885-8040	MSDOS	CAI	25.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
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
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 8 ISSUES 84-95	885-4008	N/A	1987	25.00
REMARK VOL 9 ISSUES 96-107	885-4009	N/A	1988	25.00
REMARK VOL 10 ISSUES 108-119	885-4010	N/A	1989	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	UTILITIES	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
YAUD (YET ANOTHER UTILITIES DISK)	885-6015	MSDOS	UTILITIES	20.00

Now Available!

HUG software is now available on 2" disks. Just put a "-90" at the end of the part number (i.e., 885-6014-90). Also add $3.00 to the purchase price of the software (i.e., $20.00 + $3.00 = $23.00).

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.

The other cats get to sing along!

*That's because HEPCAT runs **with** your other programs, not **over** them. HEPCAT (HUG Engineer's and Programmer's CAlculation Tool) is a powerful pop-up calculator for all Heath/Zenith MS-DOS and Z-DOS based computers. Unlike other pop-up calculators, HEPCAT does not stop the currently running program while it is popped up. That means that you can do calculations while your computer is busy with something else. For example:*

- *While Lotus (tm) is loading a huge spreadsheet, you can check your kid's math homework.*
- *While Dbase (tm) is sorting a large database, you can add up some grocery prices.*
- *While your computer is busy compiling one program, you can work on number base conversions needed for another program.*

HEPCAT is safe to pop-up during just about any running program — even during disk activity. And HEPCAT has other features the other guys can't touch.

HEPCAT gets along with everyone . . .

HEPCAT supports more video configurations than any other pop-up, and always pops up in the current video mode, rather than forcing the screen into a text mode as other pop-ups do. It also works properly with more programs than any other pop-up. You can pop up HEPCAT over Microsoft Windows (tm) and many other programs that other pop-ups can't work with, and even over some other pop-ups.

HEPCAT works harder . . .

*HEPCAT provides a multi-function floating point calculator and a programmer's binary calculator that work together to do more than the basic four (+, –, *, /). The floating point calculator includes the following built-in functions: powers, pi, factorial, square root, sine, arc sine, cosine, arc cosine, tangent, arc tangent, log (natural and base 10), e^X and 10^X. It also includes the following conversions: degrees-radians, radians-degrees, Celsius-Fahrenheit, Fahrenheit-Celsius, centimeters-inches, inches-centimeters, meters-feet, feet-meters, kilometers-miles, miles-kilometers, grams-ounces, ounces-grams, kilograms-pounds, pounds-kilograms, milliliters-fluid ounces, fluid ounces-milliliters, liters-quarts, quarts-liters. The binary calculator works in these number bases: binary, tetral (base 4), octal, split octal, decimal, and hexadecimal; and it supports these operations: MOD, AND, OR, XOR, SHL, SHR.*

The HEPCAT floating point calculator supports 8 significant digits and can display numbers four ways: floating point, fixed point, scientific notation, and engineering notation. Numbers are handled internally in BCD format to eliminate binary round off errors in addition and subtraction.

HEPCAT eats less . . .

HEPCAT uses less than 18k of memory — less than any other pop-up calculator that we know of. It also uses less than 14k of disk space, so you don't have to worry about where to put it on a small system. The HEPCAT window uses less screen space, too. It shows you more real information than other pop-up calculator displays, but it doesn't waste space by showing you a keypad layout. You already know what your keypad looks like! HEPCAT is easier to learn, too, with commands that make sense.

If you are tired of pop-ups that can only sing solo, give HEPCAT a try. HEPCAT is available from HUG as part no. 885-3045-37 for $35.00. It works on any Z-100 PC, Z-200 PC, or Z-100 (not PC) system and any version of MS-DOS or Z-DOS.

The Heath "Octoport" An Intelligent Switchbox, and More

Terry Perdue
Heath Project Engineer

Does your PC have enough ports to support all the peripherals you've accumulated? If not, you're probably in the habit of reaching over the top of your computer to unplug one cable and plug in another.

That was my situation until recently, and whenever I switched cables, the one(s) that weren't connected invariably slipped off the back my computer desk onto the floor.

Heath Company recently introduced an intelligent switchbox that can relieve you of this problem. Using it and the accompanying software, you can simply press a 'hotkey' combination to select the peripheral you want to use.

As the name, "Octoport", implies, it's equipped with eight ports. Six of these are serial ports; two are Centronics-compatible parallel ports. Each of the six serial ports may be configured to match the connected peripheral's baud rate and handshaking polarity. One parallel port is permanently configured for output while the other may be configured for input or output.

Each port has a name associated with it. Initially these names are "P0" through "P7", but they may be assigned names that describe the peripherals, such as "MODEM", "DRAFT", "DAISY", "PLOTTER", etc. Simple commands allow the ports to be configured and linked together in pairs. Up to four simultaneous paths are possible, and data buffering, baud rate conversion, serial to parallel or parallel to serial conversion, and handshaking are automatic.

Although the Octoport is useful in a single-user situation, it may also be used as a simple LAN in a small office environment to allow communications and file transfers between users, and/or printer/plotter sharing. In this case, each serial port could be given a user's name, and one or both of the parallel ports would typically be connected to a shared peripheral. A simple mailbox function is also provided to allow each serial port to leave a short memo for other serial ports. When the Octoport is used to provide printer sharing, it can be made to handle multiple print jobs automatically.

Serial port 0 is the master port. The PC connected to this port has access to several privileged commands, to rename and configure ports, link two remote ports, initialize the unit, etc.

Each of the 19 commands begins with a unique letter so that it may be abbreviated to just a single letter if desired. For example, L ME MOD might be used to Link your PC to your modem. To leave the Command mode and actually communicate with the linked port, you just press the ESC key. (Command mode may be re-entered either through a user-defined sequence of three like characters which must be preceded and followed by a pause of at least one second, or through a sequence of three Breaks. This prevents accidental re-entry to command mode when transferring a binary file.)

The Octoport has three commands that are particularly useful. The ASSIST command gives a brief description of what each command does. The HELP command gives the syntax of each command. Finally, the STATUS command shows the configuration and linkage status of each port, as well as other information. The text returned by these three commands is shown in Figures 1, 2 and 3, respectively.

The software provided with the Octoport includes a special terminal emulation program, TEM, which may be run from the DOS command line, or installed to provide hotkey operation. In this memory-resident mode, a (selectable) hotkey gets the system's attention, as indicated by a sequence of three tones. At this point, you have three choices:

1. If you press the SPACEBAR (and the current video mode is suitable), the terminal emulation screen will appear. Complete control of the Octoport is available here, as well as file handling functions. Pressing the F10 key restores the original screen when you're done.
2. If you press one of the function keys, F1 through F10, TEM will look for a "Setup" file in the \TEM directory. Each Setup file is named SETUPn.OCT, where n is the number of the corresponding function key. These files are similar in function to DOS batch files; they contain a list of commands to be sent to the Octoport.
3. If you press any other key, TEM gives control back to whatever program was running when you hit the hotkey.

In my arrangement at home, I have a PC-AT, a letter-quality printer, a logic analyzer, a modem, and an EPROM programmer connected to five of the six serial ports of the Octoport. A draft printer is connected to one of the two parallel ports. (So I still have room for expansion!)

Since each of the serial ports uses a 9-pin 'D' connector of the same gender and pin-out of the AT's serial port, it's not too difficult to get each peripheral interfaced to the Octoport. A cable is provided that interfaces the Octoport to a PC or AT.

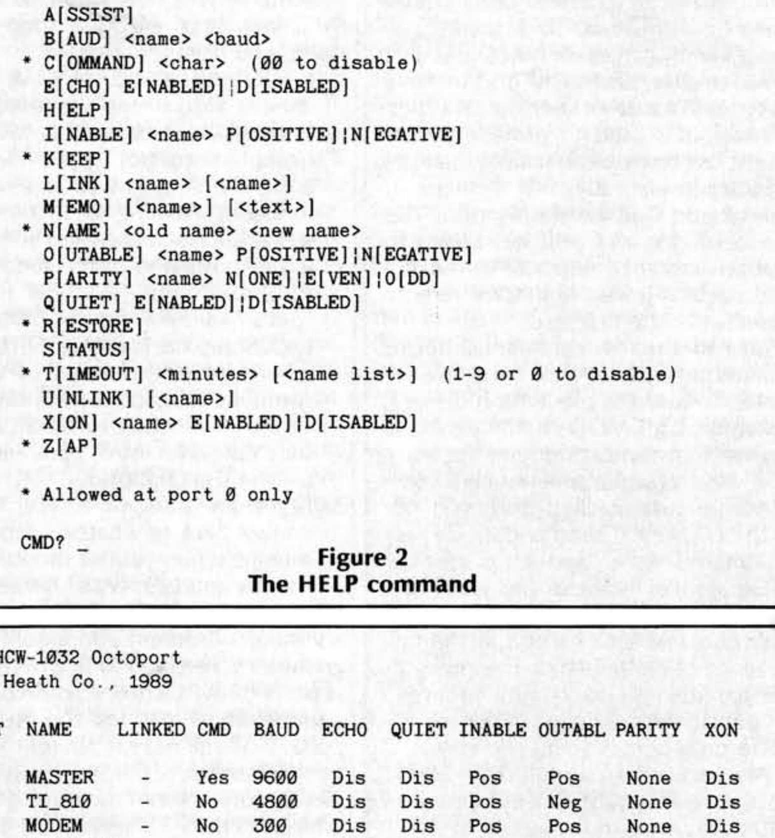
Once the Octoport has been configured properly, you can use the KEEP command to save this configuration so you can RESTORE it should you accidentally change something. Since I use the TEM program in its memory-resident mode, I have also loaded one of the ten possible Setup files, SETUP0.OCT, with all the commands required to configure the

ASSIST — Displays this text.
BAUD — 1[50], 3[00], 6[00], 12[00], 2[400], 4[800], 9[600].
COMMAND — Defines character used (x3) to enter Command mode.
ECHO — Enable for terminal-to-terminal link.
HELP — Displays command syntax.
INABLE — Defines output level to enable input to port.
KEEP — Saves configuration for Restore.
LINK — Links two ports. Only port 0 can link two remote ports.
MEMO — One line of text per port.
NAME — Renames a port.
OUTABLE — Defines input level to enable output from port.
PARITY — Selects output parity.
QUIET — Disables prompts, error messages, and echoing of commands.
RESTORE — Restores configuration saved with Keep.
STATUS — Displays current status.
TIMEOUT — Auto-unlinks after this period of inactivity on chosen ports.
UNLINK — Defaults to self if no name given.
XON — Enables XON/XOFF handshaking.
ZAP — Total reset (Hit ENTER until prompted).

ESC or ^ exits Command mode.

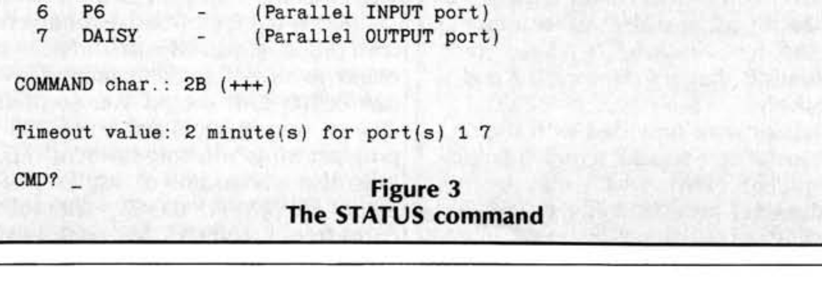
CMD? _

Figure 1
The ASSIST command



```
    A[SSIST]
    B[AUD] <name> <baud>
  * C[OMMAND] <char>  (00 to disable)
    E[CHO] E[NABLED]|D[ISABLED]
    H[ELP]
    I[NABLE] <name> P[OSITIVE]|N[EGATIVE]
  * K[EEP]
    L[INK] <name> [<name>]
    M[EMO] [<name>] [<text>]
  * N[AME] <old name> <new name>
    O[UTABLE] <name> P[OSITIVE]|N[EGATIVE]
    P[ARITY] <name> N[ONE]|E[VEN]|O[DD]
    Q[UIET] E[NABLED]|D[ISABLED]
  * R[ESTORE]
    S[TATUS]
  * T[IMEOUT] <minutes> [<name list>]  (1-9 or 0 to disable)
    U[NLINK] [<name>]
    X[ON] <name> E[NABLED]|D[ISABLED]
  * Z[AP]

  * Allowed at port 0 only

CMD? _
```

Figure 2
The HELP command



```
HC/HCW-1032 Octoport
(C) Heath Co., 1989

PORT NAME     LINKED CMD  BAUD  ECHO  QUIET INABLE OUTABL PARITY  XON

* 0  MASTER     -    Yes  9600  Dis   Dis   Pos    Pos    None    Dis
  1  TI_810     -    No   4800  Dis   Dis   Pos    Neg    None    Dis
  2  MODEM      -    No   300   Dis   Dis   Pos    Pos    None    Dis
  3  LOGIC_AN   -    No   9600  Dis   Dis   Pos    Pos    None    Dis
  4  DATA_IO    -    No   9600  Dis   Dis   Pos    Pos    None    Dis
  5  P5         -    No   9600  Dis   Dis   Pos    Pos    None    Dis
  6  P6         -    (Parallel INPUT port)
  7  DAISY      -    (Parallel OUTPUT port)

COMMAND char.: 2B (+++)

Timeout value: 2 minute(s) for port(s) 1 7

CMD? _
```

Figure 3
The STATUS command

Octoport as required for my situation. No matter what happens, I'm only two keys away from restoring the Octoport to the configuration I normally use.

As an example of how I use my Octoport to select a peripheral, suppose I want to program an EPROM. I have the EPROM programmer connected to port 4. All I have to do is press the hotkey (the TEM program beeps to confirm that it is ready), and the F4 key. That's all! The program automatically places the Octoport's master port, to which I'm connected, into Command mode, then looks for the file \TEM\SETUP4.OCT, which contains commands to unlink me if I'm already linked somewhere, link me to port 4, then exit command mode. It then beeps to indicate that the operation was successful. At this point I'm ready to send the appropriate file to the programmer, either using the DOS COPY command, or the 'transmit file' function of TEM.

One thing to be aware of when using application programs such as spreadsheets, word processors, etc., is that they may need to be re-configured for a different baud rate or port than they were before the Octoport was connected. You will normally have the Octoport's master port configured for 9600 baud, no matter which peripheral you are linked to. If the printer you normally use with your word processor is set for 2400 baud, your word processor must be set for 9600 baud, not 2400. The Octoport makes the conversion to 2400 baud. All this is explained in the Octoport's Operation Manual.

You should also be aware that the Octoport may not work with all serial peripherals. Many pointing devices, such as mice, and possibly some other peripherals, don't use the handshake lines in conventional ways. Some devices even get their operating power from the handshake lines. These devices may not work reliably or at all with the Octoport.

Although it's likely that you'll spend a little time getting the Octoport configured, the Setup files created, and your application software re-configured, once everything is working together, I think you'll agree that the time was well spent. The Octoport is available in either kit (HC-1032) or wired (HCW-1032) form.

In situations where you need to communicate with peripherals or other PCs in remote locations, the Heath HCA-1032-1 is an assembled RS-232 <-> RS-422 converter that allows serial lines to be extended to a length of at least 4000 feet. A converter is required at each end of the cable, which contains two twisted pairs of wire for communication circuits, and two more pairs if handshaking is required. A power cube is included, and the Line Extender may be used with or without the Octoport.

Assembly Language

Pat Swayne
HUG Software Engineer

Part 8

More Ways to Say Hello

This is another part of my series on Assembly Language. In this installment, I will continue the introduction to I/O that I started last time.

Other MS-DOS Output Functions

Last time, I presented the first sample program of this series. That program prints HELLO THERE on the computer screen using one of the I/O functions built into MS-DOS. The particular function used, function 9, prints a string of characters, starting with the address at DS:DX (the segment in DS, and the offset in DX). It prints all characters until it comes to a dollar sign ($), and then stops printing.

There are other functions for printing characters on the screen in MS-DOS besides function 9. You can use the functions normally used for writing to files, but we will not cover that this early in your Assembly Language education. For printing individual characters, including the dollar sign that function 9 cannot print, there is function 2. Since the other methods that we will cover for printing to the screen (BIOS, direct hardware access) require that characters be handled individually, I will show you how to handle them using MS-DOS function 2 first. Here is a program that prints HELLO THERE using function 2.

```
        CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     100H
START:  CLD                             ;CLEAR DIRECTION FLAG
        MOV     SI,OFFSET MSG           ;POINT TO MESSAGE
        MOV     CX,MSGSIZ               ;GET SIZE OF MESSAGE
PMLOOP: LODSB                           ;GET A CHARACTER
        MOV     DL,AL                   ;IN DL
        MOV     AH,2                    ;DOS CODE TO PRINT CHR
        INT     21H                     ;CALL DOS ROUTINE
        LOOP    PMLOOP                  ;LOOP UNTIL DONE
        INT     20H                     ;EXIT TO DOS
MSG     DB      'HELLO THERE',13,10
MSGSIZ  EQU     $-MSG
CODE    ENDS
        END     START
```

This program introduces a technique that can make your work easier in Assembly Language — the loop. I will explain the code line-by-line starting with the first machine instruction, which is CLD. This instruction is used to make sure that the direction flag is clear, because the program uses one of the string manipulation instructions, LODSB, later on.

In the next line, the SI register is loaded with the starting address of the message to be displayed. The DX register, used as the message pointer in the program from last time, is used as a pointer for MS-DOS function 9 only because of its roots in CP/M. The CP/M equivalent function used the 8-bit 8080 processor's DE register. Unless required by a routine such as function 9, the DX register is not a good choice for a pointer register. The SI register, on the other hand, is a natural for the job.

In the next line, the program loads the CX register with the number of characters in the message. Whenever you print a message, you need a way to indicate the end of the message. You can use a marker character, such as the dollar sign required by function 9, or you can use a count of the characters in the message. If you use a marker character, it is better to use one that is not part of the printable character set (such as the dollar sign). If you are writing your own routine that processes characters some way and stops when it comes to a marker, a good choice to use is zero (binary zero, not the character "0"). Another way to mark the end of a text string is to set the high bit on the last character. That way, you avoid taking up an extra byte using a marker that is not printed. But this method only works if you are using the standard ASCII character set, and not the extended (graphic and international) characters.

Notice how the count, contained in the variable MSGSIZ, is specified. The dollar sign used in the expression

```
MSGSIZ  EQU     $-MSG
```

is called the origin pointer. It takes on the value of the offset at the particular place in the program where it is used. Since it immediately follows the message, its value minus the offset at the start of the message (at the label MSG) equals the size of the message, which is stored in MSGSIZ.

The next line of the sample program begins the loop that will display the characters on the screen. The LODSB instruction gets a character from the text string, and increments the pointer register (SI). In the next three lines, the character is loaded into the DL register as required by MS-DOS function 2, the AH register is loaded with the function number, and MS-DOS is called to print the character.

After a character is printed, the LOOP instruction is used to decrement the CX register and jump back to the label PMLOOP if CX is not zero. So the instructions in the loop are repeated until the entire message is printed. Then control falls through to the exit interrupt, which returns control to the MS-DOS command interpreter.

BIOS Output Routines

The routines in the BIOS are organized into groups of functions, and each group is assigned a software interrupt that is used to access the functions in it. Interrupt 10 (hex) is used to access the video output functions, which can be used to output characters to the screen. The BIOS video functions are quite versatile, as I mentioned in the last installment. I will show you how to display "HELLO THERE" on the screen using two of them.

Video function E (hex) is similar in operation to MS-DOS function 2. The last sample program can be easily modified to use it, and it looks like this.

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

START:   CLD                        ;CLEAR DIRECTION FLAG
         MOV     SI,OFFSET MSG      ;POINT TO MESSAGE
         MOV     CX,MSGSIZ          ;GET SIZE OF MESSAGE
         XOR     BH,BH              ;SELECT PAGE 0
PMLOOP:  LODSB                      ;GET A CHARACTER
         MOV     AH,0EH             ;INT 10 CODE TO PRINT CHR
         INT     10H                ;CALL BIOS ROUTINE
         LOOP    PMLOOP             ;LOOP UNTIL DONE
         INT      20H               ;EXIT TO DOS

MSG      DB      'HELLO THERE',13,10
MSGSIZ   EQU     $-MSG

CODE     ENDS
         END      START
```

This program is so similar to the one using MS-DOS function 2 that I will not explain every line. Besides using a different interrupt and function number, the only changes are that the DL register is not loaded with the character (the BIOS function accepts the character in AL, where LODSB puts it), and the addition of a line to clear the BH register. The BH register is cleared because a color system that is in a text video mode supports more than one "page" of video. The pages are different areas of video memory. The system can only display one page on the screen at a time, which is normally page 0. You can write characters to the undisplayed pages, and later you can switch to them using one of the video functions. By writing to undisplayed pages and then switching, you can make text changes happen instantaneously. One good use for multiple video pages is to have help information available for a program. The help information can be switched onto the screen instantly, and the original screen display can be restored instantly when you are through viewing the help text.

So far, the methods I have presented for putting characters on the screen have not been concerned with the color of the characters on the screen. These methods use the colors that are already on the screen. If you want to specify the colors of the characters in a message, you can use function 9 of the video output functions. This function is a little more complicated to use than function E, however, because it does not automatically move the cursor over to the next character position each time a character is placed on the screen. So a program that uses function E must also use other functions to move the cursor. Here is a program that prints "HELLO THERE" using function 9.

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

START:   CLD                        ;CLEAR DIRECTION FLAG
         MOV     SI,OFFSET MSG      ;POINT TO MESSAGE
         XOR     BH,BH              ;SELECT PAGE 0
         MOV     AH,3
         INT     10H                ;READ CURSOR POSITION
         MOV     CX,MSGSIZ/2        ;GET SIZE OF MESSAGE
PMLOOP:  LODSW                      ;GET A CHARACTER AND COLOR
         MOV      BL,AH              ;COLOR TO BL
         CMP     AL,' '              ;CONTROL CHARACTER?
         JB      CTRL               ;YES, USE FUNCTION E
         MOV      AH,9               ;INT 10 CODE TO PRINT CHR
         PUSH    CX                 ;SAVE COUNTER
         MOV     CX,1               ;PRINT ONE CHARACTER
         INT     10H                ;CALL BIOS ROUTINE
         INC     DL                 ;MOVE CURSOR OVER
         MOV     AH,2
         INT     10H                ;SET CURSOR TO NEW POSITION
         POP     CX                 ;RESTORE COUNT
         LOOP    PMLOOP             ;LOOP UNTIL DONE
CTRL:    MOV     AH,0EH             ;INT 10 CODE TO PRINT CHR
         INT     10H                ;ALL BIOS ROUTINE
         LOOP    PMLOOP             ;LOOP UNTIL DONE
         INT     10H                ;EXIT TO DOS

MSG      DB      'H',9,'E',10,'L',11,'L',12,'O',13
         DB      ' ',13,'T',14,'H',15,'E',14,'R',13
         DB       'E',12,13,0,10,0
MSGSIZ   EQU     $-MSG

CODE     ENDS
         END     START
```

The first thing you will probably notice about this new program is that the message (at the label MSG) is different. There is a number after each character of the message. Each number determines the color of the character that precedes it. Actually, the number, which will be used in video function 9, can do more than just set the color of a character. It can also make it blink, or, in the case of a monochrome (Hercules-type) display system, it can cause the character to be underlined. Therefore, these numbers are often called "attribute codes". Here is a table that shows how the bits of an attribute code affect a character.

Bits	What They Affect
0-3	Foreground color. Actually, bits 0-2 determine the color, and bit 3 is an intensity bit. Bit 0 is the blue bit, bit 1 is the green bit, and bit 2 is the red bit. Therefore, if bits 0 and 1 are high, the associated character is a cyan color. If bits 0, 1, and 3 are high, the character is a brighter cyan.
4-6	Background color. Normally background colors can be only colors 0-7 of the normal 16 colors.
7	This bit causes a character to blink on a color system, or to be underlined on a monochrome system. You can also program a color system to use this bit to intensify the background color (but I will not go into that here).

The first letter in the message (H) has

Continued on Page 12

Searching For Binaries

A. Grattan
138 Dana Street
Brockville, Ontario
Canada K6V 6W8

Insight

In the computer world, at home, or in the office, we are in a time when we consider the performance of equipment to be a critical factor. Monies available to upgrade or replace the existing equipment is tight, and this creates a dilemma for the users. Need speed . . . no money.

At this very moment, there is less and less software available that will run on an 8088 or 8086 processor. Microsoft's Excel and Word for Windows both require a minimum 80286 CPU, while the newest release of Autocad requires an 80386. Users running on 8088/8086 machines, who require products, such as the ones listed above, have a definite problem. They must upgrade their equipment to support the new software or live with what they have.

Others, who may do in-house development, can look at other alternatives. One, is looking at other methods of getting the same result. Or, in other words, "there is more than one way to skin a cat."

Using data bases as an example, many users use packages like R:base, dBase or Enable. It has been my experience that there are probably just as many people who develop custom data base systems using high level languages such as BASIC or C. These in-house data base systems usually work well, especially when they have one or two hundred records. Suddenly, when they hit one or two thousand records, performance becomes nonexistent. Solution . . . we run out and buy a 33 Mhz 80386 from our nearest Heath/Zenith dealer, or we take another look at our program.

The Real Problem

How does the program access records? Sequentially? This is not an uncommon practice for many. Most users programming today are self-taught, or have taken only introductory programming courses at their local college. This usually limits their exposure to the more common techniques and practices used in the programming world. Have you ever heard of a binary search? Do you use binary searches in your programs? If not, the rest of this article may be of interest to you.

A binary search is an excellent method for searching a pre-sorted file or array, and can cut access time to practically zero. For all its benefits, it surprised me to see how few books cover the topic. I would estimate that half the programming manuals and textbooks I have read do not mention binary searches at all. Of those that do, a very few cover the topic completely. Only one, of the books I read, showed how to handle a "not found" condition in a binary search.

Before getting into the thick of things, let's look at how we approach similar situations manually. A good example is a telephone book. If we didn't know any better, we would start at page one and read until we came to the name we were looking for. This method could be tolerable if we were calling Mr. Abernathy, but could you imagine looking up Mr. Zuger's name.

We all know that this is not how we find a name in the phone book. We know that the phone book is sorted alphabetically, "A" to "Z." Generally, when looking up a name in the phone book, we pick up the book and arbitrarily open it to a spot we believe is close to the name. We then find if the name is before or after the page we have opened to. This in essence, is a binary search. So, if we can do this, why can't the computer.

The Theory

A binary search can be written in any language. I have included a sample program following the article, written with Microsoft's QuickBASIC 4.5. I believe BASIC is still the most popular and widely understood programming language.

As the name implies, a binary search is based on the number two. If we were working with an array or table that contained 500 elements, we would begin the search at point number 250 (500 divided by 2). From this point, we would figure out if the item we were looking for was greater than, less than, or equal to the middle point (250). If we find that the item we were looking for was less than that found at the middle point, we would move the search point to 125 (250 divided by 2). If we now find the requested item to be larger than the item found at the new middle point (125), we would move the search point to 187 ((250 + 125) divided by 2 and rounded down). This process would continue until we found what we were looking for, or the "no match found" flag was turned on.

We must not forget that the file [array], that we are working with, must be sorted. The sample program requires ascending order for the binary search to work. This can be changed by swapping the "greater than (>)" and "less than (<)" conditions used in the binary search. The sort order would then have to be in descending order.

A number of events must take place when invoking a binary search:

1. Initialize a low value (0).
2. Initialize a high value (equal to the number of elements in the file or array).
3. Calculate a mid point ((low + high)/2)

As each new middle point is calculated, then the previous middle point will become either the high value, or the low value.

Example:
low = 0
high = 500
middle point = (0 + 500)/2 = 250

If we find that the item we are looking for is less than the item found at point 250, then the following process will occur:
low = 0
high = last middle point = 250
middle point = (0 + 250) / 2 = 125

Let's assume that the item we are looking for is greater than the item found at point 125. This is what will happen then:
low = last middle point = 125
high = 250
middle point = (125 + 250) / 2 187
(187.5 rounded down)

The example shows that after the third seek, we will have narrowed down the window from a maximum of 500 items to 125, falling between 125 (low) and 250 (high). We can see that a sequential search is going to require a minimum of 125 seeks to reach this point.

This explanation and example should help us to understand the point I made earlier about faster processors. Correcting a problem may require a different line of thinking, not more muscle.

The Advantages

Table 1 illustrates the advantages of a binary search over a sequential search by comparing the maximum number of required seeks for each:

Number of Elements	Maximum Seeks (Sequential)	Maximum Seeks (Binary)
20	20	5
100	100	7
2000	2000	11
10000	10000	14

Table 1

We can see that as the number of elements increases, the binary search becomes much more efficient than the sequential search. The formula for determining the maximum number of seeks in a binary search is:

2^n > number of elements
eg., $2^7 > 100$

What is it going to cost in overhead to code a binary search instead of a sequential search? Looking at the program found at the end of the article, and counting the number of statements within the While...Wend loops, we can see that the binary search requires only 3 extra statements. I counted an IF statement as 1 even if it ran across three lines.

Walk Through

The program and data following the article, compares the performance of a binary search to that of a sequential search. Obviously, as illustrated in the above table, the larger the quantity of data, the more beneficial a binary search becomes. If you want to increase the size of the supplied data file, just remember to change the value of MAX in the program, and maintain the sorted sequence for the data.

I wrote the program in a manner that I believe most people can follow. There is a small main program area that does the initial housekeeping tasks and calls five subroutines. I will briefly outline the four supporting subroutines before detailing the binary search subroutine. The other subroutines should be straightforward and not require much explanation.

Initialize Array — Opens a file (eg., ANIMALS.TXT), reads a record, and loads the record into the array WORDS$. The process will continue until an end-of-file condition is detected, or until the counter (I) exceeds the value assigned to MAX (the upper limit of the array).

Prompt For Input — Clears the screen and prompts the user for the item to be searched (FIND$). Before returning, we translate FIND$ to uppercase.

Sequential Search — Performs a search of the array (WORDS$) until the end of the array or, until it is determined that the item has been found or, that the item does not exist in the array.

Display Results — Calculates the elapsed time for each search, and displays the results on the screen.

The Binary Search

Upon invoking the binary search, we must initialize four variables:

FOUND — Used as a switch to decide if the search has completed.

"1" — search successful — terminate
"0" — continue with search
"-1" — search unsuccessful — terminate
Initialize variable to 0

HIGH — Used to represent the upper boundary for the binary search. Initialized to MAX (see start of program for the value of MAX).

LOW — Used to represent the lower boundary for the binary search. Initialized to 0.

PREV — Contains the value of the last middle point as calculated by the routine. We use this to determine if the search has finished without finding a match.

The variables BIN.START! and BIN .END! (the ! declares the variables as single precision numbers), trap the start and end times for the binary search. We use them later to calculate an elapsed time for the search. The result is used as a comparison with the time it took to perform the sequential search. It should be noted that on a 30 element array, the elapsed time generally shows as 0 for both searches.

WHILE...WEND is the loop structure that controls execution of the binary search, and processes while the condition "FOUND = 0" is true.

The first order of business once we are within the loop, is to calculate the middle point (MID = (LOW+HIGH)/2). Immediately following, we figure out if the search should be terminated (IF MID = PREV ...).

"PREV = MID" will assign PREV the current value of MID. This will not be important until the loop comes around again to the previous statement to check for a possible "not found" condition.

The compound IF statement is used to decide which condition (high, low, found) may be true and take the appropriate action. You will notice that this is the point at which the HIGH and LOW variables will be recalculated.

BIN.COUNTER is the variable used to calculate the number of loops made in the search. It is then compared with the results from the sequential search.

The sample program requires QuickBASIC 4.5 to run. It can be compiled and run in memory, or you can create an EXE file and run it. By adding line numbers, commenting out the subroutine labels, and changing the GOSUB statements accordingly, this program should run through a BASIC interpreter as well. If you decide to try using data of your own, you need only change the value for MAX and change the name of the file to be used for input.

Conclusions

Wasn't that easy? It should have been just about as painless as a visit to the dentist. Remember, learning the statements of a programming language isn't the important point. Learning to be innovative with the knowledge you have gained, is what really matters.

```
'   BINSRCH.BAS - compares a sequential search to a binary search.
'   A. Grattan
'   January, 1990
'
'   written using Microsoft s QuickBASIC 4.5
'==============================================================================

   DEFINT A-Z                            ' define all variable to be integers
   max = 30                              ' initialize max to 30
   seq.counter = 0                       '    "      sequential counter to 0
   bin.counter = 0                       '    "      binary counter to 0
   DIM words$(max)                       ' dimension words$ to MAX elements

   GOSUB initialize.array                ' initialize array

   WHILE find$ <> "Q"
      seq.counter = 0                    ' set sequential counter to 0
      bin.counter = 0                    '  "  binary counter to 0
      GOSUB prompt.for.input ' prompt user for input
      IF find$ <> "Q" THEN
         GOSUB sequential.search         ' perform a sequential search
         GOSUB binary.search             ' perform a binary search
         GOSUB display.results           ' display results
      END IF
   WEND

   END                                   ' end program
'==============================================================================
'
'                          Sequential Search
'
'==============================================================================
sequential.search:

   i = 1                                 ' initialize array counter to 1
   found = 0                             ' set found switch to 0
                                         ' 1 = found ... -1 = does not exist
   seq.start! = TIMER                    ' trap start time for seq. search
   WHILE i <= max AND found = 0          ' search array until search criteria
      IF words$(i) = find$ THEN          ' met, or list exhausted
         found = 1                       ' found the name
      ELSE
         IF words$(i) > find$ THEN
            found = -1                   ' name not in list
         END IF
      END IF
      i = i + 1
      seq.counter = seq.counter + 1      ' increment counter showing # of passes
   WEND
   seq.end! = TIMER                      ' trap end time for seq. search

RETURN

'
'                          Binary Search
'
'==============================================================================
binary.search:

   found = 0                             ' set found switch to 0
                                         ' 1 = found ... -1 = does not exist
   high = max                            ' set high value to array maximum
   low = 0                               ' set low value to array minimum
   prev = 0                              ' previous midpoint

   bin.start! = TIMER                    ' trap start time for binary search
   WHILE found = 0                       ' perform binary search
      mid = (low + high) \ 2             ' set midpoint (use integer divide)
      IF mid = prev THEN                 ' has it been determined that
         found = -1                      ' the search word is not in list
      END IF
      prev = mid                         ' store current midpoint
      IF words$(mid) > find$ THEN        ' is the search word less than the
         high = mid - 1                  ' word at midpoint of the array
      ELSE
         IF words$(mid) < find$ THEN     ' is the search word greater than the
            low = mid + 1                ' word at the midpoint of the array
         ELSE
            found = 1                    ' if two previous conditions were false
            END IF                       ' then wc must have found the one we
      END IF                             ' were looking for
      bin.counter = bin.counter + 1      ' increment counter showing # of passes
   WEND
   bin.end! = TIMER                      ' trap end time for binary search

RETURN

'
'                          Prompt For Input
'
'==============================================================================
prompt.for.input:

   CLS                                   ' clear the screen
   INPUT "Enter word to be selected ('Q' to end)"; find$    ' prompt user
   find$ = UCASE$(find$)                 ' convert input to uppercase


RETURN

'
'                          Display Results
'
'==============================================================================
display.results:

   seq.time! = seq.end! - seq.start!
   bin.time! = bin.end! - bin.start!
   LOCATE 5, 1, 1
   PRINT TAB(25); "Elapsed Time"; TAB(65); "Number of Passes"
   PRINT
   PRINT "Sequential Search"; TAB(30); seq.time!; TAB(70); seq.counter
   PRINT "Binary Search"; TAB(30); bin.time; TAB(70); bin.counter
   PRINT
   IF found = 1 THEN
      PRINT find$; " was found in the array."
   ELSE
      PRINT find$; " was NOT found in the array."
   END IF
   PRINT
   PRINT "Strike any key to continue."
   n$ = ""
   WHILE n$ = ""
      n$ = INKEY$
   WEND

RETURN

'
'                              Initialize Array
'
'===============================================================================
initialize.array:

   i = 1                                ' initialize counter to 1
   OPEN "ANIMALS.TXT" FOR INPUT AS #1   ' open data file
   WHILE NOT EOF(1) AND i <= max        ' loop until end of file
      INPUT #1, word$                   ' read words from file
      words$(i) = UCASE$(word$)         ' load array with uppercase words name
      i = i + 1                          ' increment counter
   WEND
   CLOSE #1                             ' close file

RETURN
```

Test Data

Badger Bat Bear
Bird Buffalo Camel
Cat Cougar Coyote
Deer Dog Eagle
Eel Elephant Fish
Fox Giraffe Kangaroo
Leopard Lion Mole
Monkey Moose Mouse
Rhinoceros Snake Tiger
Whale Wolf Zebra

Continued from Page 8

the attribute code 9 following it. That means that the foreground color of the first character will be bright blue (bits 0 and 3 high make 9), the background will be black (no bits high), and the character will not blink.

Now, let's examine the program itself. Near the beginning, video function 3 is used to read the current cursor position, which is returned as a number in the DX register. The DH register contains the row number of the cursor, and the DL register contains the column number. The rows and columns are numbered starting in the upper-left corner of the screen, which is position 0,0. The function also returns a number that represents the shape of the cursor in the CX register, but this program does not use that information.

Notice that the count is loaded into the CX register as MSGSIZ/2. That is because the message now contains two bytes for each character, but the message is loaded (LODSW) two bytes at a time. This places the character in AL and the attribute in AH. The attribute is then moved to the BL register, where function 9 expects it to be. Function 9 also expects a count in the CX register. This feature allows it to be used to draw lines across the screen, etc. Because the program's loop count is in CX, it must be PUSHed onto the stack, and then a 1 is loaded into CX. Then the video interrupt is called to display the character in AL, with the attribute in BL.

Before the program calls the video interrupt, it tests the character to see if it is a non-printable control character, such as a return or line feed. These characters all have an ASCII value that is less than a space character, so each character is compared with the numerical value of a space (CMP AL,' '). If the value of a character is below a space, the program branches to a section that uses function E to process the character. That is because function 9 processes all characters as displayable, and the return and line feed would show up as their graphic equivalents.

As written, this program is only suitable for printing strings that end in a control character. That is because there is no exit following the first LOOP instruction. If a string ended on a non-control character, execution would "fall through" at the first loop into the function E routine. Then the last character would be printed a second time, the loop counter would be "decremented" to 65535 by the second loop, and the program would print whatever "garbage" followed the string — 65535 characters worth! The program can be fixed by adding an INT 20H instruction after the first loop. Be sure to do that if you change the string to one not ending in a control character.

A Little Homework

Here's your first homework assignment for this series. Try to modify the last program so that every other character in the message "HELLO THERE" blinks. You may also want to change the colors to a repeating pattern (for example, red, green, red, green, etc.). I have run out of steam for this installment, so I will save keyboard input for next time. See ya then.

Adding a 3.5" Drive to an H/Z-158

Edouard A. Piche
4127 Fox Hill Drive
Sterling Heights, MI 48310

For a while now, I have been looking enviously at the new 3.5" micro floppy drives, which offer several advantages over the 5-1/4" drives. Specifically, they offer data transfer capability with laptop computers and newer desktop machines; offer increased storage capacity (especially useful for archiving and periodic backups); and allow running software that is increasingly being distributed in the new 3.5" format.

I finally took the plunge into installing a new 3.5" drive in my 3-1/2 year old H-158 when my wife bought me one for Christmas. I encountered some difficulty in the installation, which I eventually overcame. I hope this article will help other HUGgies who want to make the same installation.

Types of Drives

In choosing a 3.5" drive, you must make the decision between one of two formats. The lower capacity drive is rated at 720K bytes (formatted; 1 Megabyte unformatted), and will work with the stock H/Z-158 floppy disk controller. The higher capacity drive is rated at 1.44 Megabyte formatted, but requires a high-density floppy disk controller.

In addition to the cost of the controller card and the loss of an expansion slot, there was at one time a cost penalty on the high capacity disks. This cost penalty is slowly decreasing as these floppies become more popular. A local computer store advertises boxes of ten Sony 720K floppies at $13.50; Sony 1.44 Meg floppies run $29.90 per box of ten. This is about a 10% price penalty for equivalent storage capacity. However, unlike the 1.2 Meg 5-1/4" floppies, the 1.44 Meg drives will reliably read and write at the lower capacity. For purposes of this article, I will assume that you are installing a 720K drive.

3.5" Drive and Support Hardware Requirements

My system has one stock Z-207-7 Panasonic JU-455-5 5-1/4" floppy disk drive, and an aftermarket Seagate ST-225 20 megabyte hard disk. The 3.5" drive is a Mitsumi 720K model from CompuAdd, P/N 45202, at a cost of $89. See Figure 1. This drive comes mounted in an adapter frame to fit a 5-1/4" half height bay, and includes a black face plate and mounting screws. There are power and data connector adapters included in this kit, since the native 3.5" drive has different connectors than a 5-1/4" drive. Conversationally, this is referred to as an "XT" installation kit. Be aware that an "AT" installation kit will get you the same hardware but with a gray face plate.

Before you dig into the installation, make sure you have all the necessary hardware and software. Since the H/Z-158 power supply has only two power connectors for floppy drives, you will need a Y adapter to power the third drive. I purchased one at the local Computerland store for $6.50.

I also found that I needed a new floppy disk drive 34 conductor ribbon cable, since the edge connectors on the original cable were separated by only 2-1/4". This works fine if the disk drive edge connectors are directly above each other. However, the Mitsumi and the Panasonic drive signal edge connectors were on opposite sides of the drives. The existing cable would not reach both drives; you will need about 6" between the connectors. If the 3.5" drive you select has the edge connector directly below the A drive connector, you will not need to purchase this cable. If you find you need one, be sure to call around for prices. Computerland quoted a price of $25.00. I bought a cable from an independent computer store for $12.95 plus tax. See Figure 2 for these cables.

Be sure the cable you buy does not have a twist in the ribbon between the two edge connectors. This twist is used with the high capacity drive controller card to identify the A and B floppy drives. It switches the motor enable and drive select signals on lines 10 through 16 for the two floppy drives. With the Heath/Zenith controller, this twist will cause both drives to operate simultaneously, causing damage to the disk in drive A when you attempt to format drive B. I did not discover this fact until after purchasing the connector cable. Fortunately, the cable I bought had a snap-on connector. I overcame this problem by prying off the top half of the connector; removing, straightening and reinstalling the ribbon cable; and snapping the top half of the connector back on. If you are only able to find a twisted ribbon cable, you may need to resort to the same tactics. Be sure to inspect the connector to make sure it can be reworked.

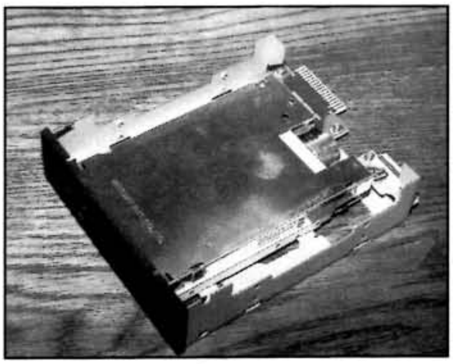


Figure 1
Mitsumi 720K 3-1/2" Floppy Disk Drive

Software Requirements

If you have DOS 3.0 or 3.1, you will find that the 3.5" drive will format properly — at 360K! This will be fine if you simply want to have a second drive that acts like a 360K 5-1/4" floppy; however, you will not be able to read or write at the higher capacity. Support for the 3.5" drive was not added until MS-DOS release 3.2. Support for the 1.44 Meg drive was added in MS-DOS 3.3, and Zenith Data Systems MS-DOS 3.21.

There are two solutions. The obvious one is to upgrade to the version of DOS that supports the drive you bought. A second alternative is to buy a software driver along with the 3.5" drive that will work in conjunction with the older version of

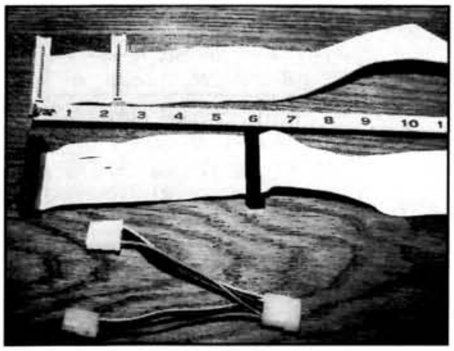


Figure 2
Top to bottom: Original H/Z-158 34 conductor ribbon cable; New ribbon cable; Power Y adapter.

DOS. If you are going to go this route, be sure to buy the driver when you buy the drive. Cost should be between $15 and $30. Also be aware that CompuAdd does not sell this driver.

Installing and Checking Out the Drive

I am assuming that you will be installing the three drives as follows: The 5-1/4" drive in the top of the three drive rack as drive A; the 3.5" drive in the middle as drive B; and the hard disk buried in the bottom as drive C. You should park the heads on the hard disk before starting (a wise practice at all times before shutting the computer down). You may also want to back up the hard disk before starting this installation.

First, disconnect all power from the computer. Once you remove the cover, you can remove the disk drive rack. The power supply will eventually have to be removed to reinstall the drives, but it can stay in for now.

According to the 5-1/4" drive installation manual, a terminating resistor must be removed from all but the last drive. This terminating resistor is used by the disk controller card to identify the last drive in the system. However, I found that it was not necessary to remove this resistor since the 3.5" drive has a different resistor value than the 5-1/4" drive.

Remove the hard disk from the drive rack. Remove the face plate from the drive, since it will interfere with installation of the front face plate in its new location. I suggest that the hard disk be set aside and not reinstalled at this time. You may need to play around with the floppy disk drive settings a couple of times before you get it running properly. You don't want to risk damage to the hard disk.

Before installing the 3.5" drive, make sure it has the proper jumper settings. On the Mitsumi drive, there are only two jumpers to be set. The first selects between XT and AT operation; set this to XT. The second is the drive number for this drive. The drive jumper is on the outboard rear corner of the Mitsumi drive. The CompuAdd instruction manual stated that the jumper should be set to the first position (drive 0) and that placement of the drive on the cable determines whether the drive will be addressed as A or B. This presumes a twisted ribbon cable as mentioned above. However, the Heath/Zenith controller card is different, and using the twisted cable will cause the two drives to respond simultaneously to commands like DIR and (worse) FORMAT. I found that the 3.5" drive had to be set up as drive 1 (B) with the appropriate jumper. Make sure to refer to the manual that comes with your drive for any additional instructions.

Loosen the drive A mounting screws in the rack to allow the sides to be spread apart slightly. Install the 3.5" drive in the middle position. The drive rack can be temporarily placed in the computer next to the power supply.

Install the ribbon connector onto the two edge connectors. Since the connectors are electrically identical, it theoretically should not matter how they are connected. However, I suggest that the middle connector be installed onto drive A and the end connector be installed onto drive B to allow cooling air flow past the drives. Just be sure to install the connectors with the red stripe on the slotted end of the connector. Plug in the two power connectors, and the drives should now be functional.

To check out the drives at this time, install a bootable floppy in drive A. Reconnect power and turn on the machine. The system should power up and ask for date and time, then give you the familiar A:> prompt.

On a 3.5" disk, the write protect tab (the sliding plastic block on the bottom of the disk), must cover the hole to write to the disk. Sliding the block to uncover this hole write protects the disk. This procedure is exactly the opposite of that of the 5-1/4" floppy.

Try to format drive B at this time. The 3.5" floppy disk is inserted with the drive hub down, and the sliding cover into the drive opening. If the disk does not drop into place, it was installed incorrectly. Try again with the disk in the proper orientation. To remove the disk when you are done with it, simply push the button and the disk will pop out.

If you do not have a CONFIG.SYS file, or if it does not have a DRIVPARM statement in it, drive B should format at 360K at this time. The FORMAT program will tell you that there are 362496 bytes total disk space. To reach the full 720K (730112 bytes) of this drive's capacity, add the following statement to a CONFIG.SYS file in the root directory of the disk in the A: drive, and later to your hard disk.

```
DRIVPARM = /D:1 /F:2
```

This statement tells the system that drive 1 (the new 3.5" B drive; remember drive A is referenced as drive 0) is to be set up with a form factor of 2, or double-density. The /F:2 is actually the default, and can be omitted if you desire. If you buy the "add-on" software driver, make sure to follow the instructions with this software.

Reassembling the Computer

Once the 3.5" disk drive checks out at the higher density, it is now time to reinstall the hard disk and reassemble the computer.

If your hard disk is like mine, you will find an interference between the case of the hard disk and the plastic disk drive mounting brackets in the middle position. I solved this problem by sanding the horizontal legs of these brackets with a disk sander. Use whatever tool is handy (hacksaw, bandsaw, etc.), but these brackets must be reinstalled for proper mounting of the B floppy disk drive. Make sure to leave at least 1/8" on the horizontal legs so drive B can be located correctly. See Figure 3.

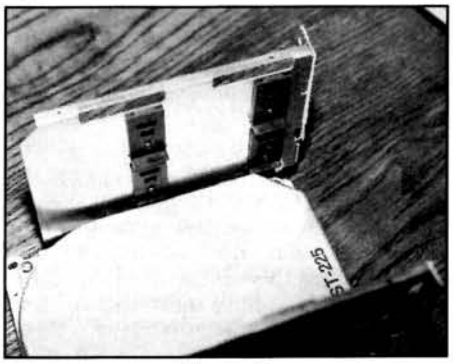


Figure 3
Note that the middle floppy adapter bracket legs have been cut down to clear the hard disk.

Now that the hard disk is buried in the bottom of the rack, the drive activity light is no longer visible. I solved this problem by drilling 4 holes, 15/64" in diameter, in the front panel. Three of these holes are for cooling to replace the Seagate front panel cooling slots. The fourth is lined up with the hard disk LED. It takes some careful measurement to get it lined up properly. To allow the drive light to be seen, I used (don't laugh) a yellow Lite Brite peg in line with the LED on the hard disk. A dab of plastic model cement holds it in place.

By serendipity, the peg is exactly the right length to pass through the front panel and just clear the hard disk LED. The drive activity light is clearly visible. I originally used a red peg, but the red light cancelled the green light from the LED, and caused the light to be very dim. As an alternative to a Lite Brite peg, a piece of clear acrylic rod or anything similar to act like a light pipe could be used.

Reinstall the two floppy disk drives in the rack. You will notice that the end of the hard disk sticks out of the back of the

Continued on Page 17

The World of WP50 and Its Wonders

Salli Brackett
2201 Sycamore, #123
Antioch, CA 94575

Traveling through the marvelous world of WordPerfect 5.0 (WP50) is indeed an adventure. Although confusing at first, it is worth the trip.

In the next few articles, we will set up WP50 in subdirectories; create a file, save it and print it; learn basic editing techniques and cursor movements; begin the world of macros; study how to create a style; do several types of merging; learn how to set up a table of contents and index; and set up a laser printer.

The Following Paragraphs are for Hard Disk Users

The WordPerfect manual suggests you put the WP50 program files in a subdirectory, i.e. \wp50. I suggest putting your files in another, such as \data, if you have no other programs, or \data\wp50. If you plan to have files in a variety of subjects, more directories are advisable, i.e. letters = \data\wp50\letters or reports = \data\wp50\reports.

Setting up directories saves a great deal of searching later for a file. Think of a hard drive as a file cabinet. You wouldn't think of putting a bunch of papers in a file cabinet without file folders and probably pendaflexes.

FILE CABINET	**HARD DRIVE**
(Drawer)	**(Root Directory)**
Pendaflex	First Subdirectory (\data)
File Folders	Second subdirectory (\data\wp50)
Papers	Files

Note: DOS only allows a limited number of files in the root directory and an unlimited number in a subdirectory. If you don't use subdirectories, you will be limited depending on the size of the media you're using.

If you wish to create a menu batch file to access subdirectories now, turn to the last page of this article. The example assumes your program files are in a separate directory.

1. If you are using subdirectories, you need to make some changes in the autoexec.bat.
2. Put the PATH command in your autoexec.bat. (DOS looks for autoexec.bat when booting up. Your manual will have further explanations on how to use this file.) The PATH command makes using directories almost invisible. It is used to tell DOS where to look for a file that is not in the ACTIVE directory. ONLY use it for PROGRAM files, for it will save any corrections to a file to the ACTIVE directory. In other words, you may have two versions of a file; one in the ACTIVE directory and the original in another directory.
3. Use a text editor (i.e., Edlin) or go to WP50 and save the autoexec.bat as a DOS text file (^F5). Below is a sample autoexec.bat file without the menu option. (Menu is a series of batch files on the last page which will help you to set up a screen menu for choosing different software.)
4. `Path = c:\;\wp50;\dbase;\lotus`
   <sets path to root and 3 subdirectories>
   `rtclock`
   <sets real-time clock, see DOS manual>
   `zspool 40`
   <sets spooler with 40 k memory, see DOS manual>
   `cd \data\wp50`
   <makes directory \data\wp50 the active directory>
   `wp`
   <starts wp50>

All this may seem like a lot of work just to make a few files. BUT there is a rule when dealing with PCs:

- If you plan to create 10 files, they will multiply by 5.
- If you plan to create 50 files, they will multiply by 10.

A Note on Naming Files

Remember DOS only permits 8 characters plus 3 for the extension. (The extension being the 3 characters to the right of the dot, i.e., smith689.ltr,) Try to name files by subject matter. You can use extensions to indicate whether a file is a letter, memo, or report. (Extensions are optional.)

```
aa589.rpt or aa589
Report to AAA, May 1989

wp501.art
first article on wp50
wp502.art
second article on wp50
```

The main thing is to be consistent. Art = article; ltr = letter; rpt = report, etc.

You say, "Why do I need all this organization?" Beware my friends, no matter what your age, tomorrow you will have forgotten what you called that file. We could lose you for a week while you melt into frustrated mush looking for that file you created last month. Placing the name of the file in the bottom left-hand corner of your document is also a way to save your sanity.

We Begin Our File

If you have not used the menu batch file:

Type `cd \data\wp50`
Press ENTER
Type `wp`
Press ENTER

Tip: The reason for changing the directory before your request WP50 is that WordPerfect will save all files in the ACTIVE directory. The DEFAULT (automatic) ACTIVE directory is the root (C: or C>>). By using CD (change directory), you actually change the ACTIVE directory.

Your screen will look like the figure below:

```
Doc 1 Pg 1 Ln 1 Pos 10
```

The screen shows 25 lines on screen. The right-hand corner (Status line) tells the position of your cursor at all times. It gives the following information:

Doc 1 — *WONDER #1*: WP50 allows you to be in two documents at once. Press Shift F3. The status line shows Doc 2. Press Shift F3 again and you are back in Doc 1.

Pg 1 — This shows the page the cursor is on.

Ln 1 — This shows the line the cursor is on. The default top margin is 6. Therefore, the first typed line will print on line 7.

Pos 10 — This is the column the cursor is on. The far left-hand column will be (by default) 10 or 12, depending on if your default font is a 10 pitch or 12 pitch font.

(4.2 and 5.1 will be considered)

Some Useful Information and Terminology

An 8-1/2" × 11" paper has:
- 85 columns
- 66 lines
- 6 lines = 1 inch
- 10 pitch = 10 characters (columns) per inch (as on a typewriter)
- 12 pitch = 12 characters (columns) per inch

WP50 Default (automatic) settings:
- 10 pitch
- 1 inch margins top and bottom
  - 1 inch = 6 lines
  - 1page = 66 lines
- 1 inch margins left and right

This leaves 54 lines of available typing space per page.

Tip: I have found it less confusing to have the left and right margins and tab menus show the position by columns (4.2 does this by default). This will became clearer in further discussions. For now just follow the instructions below. You will change the default of inches to 4.2 units or columns.

Press Ctrl-F1 (Setup)
Press U (in WP51 press E then U)
Press D
Press U
Press F7

You now have your original screen back.

You ask "Can I begin my letter yet?" Be patient, dear friends. Learning the above will prevent much confusion and retyped files later. The last thing is to learn a few basic cursor movements. (The complete list is in your manual). I suggest you write the following list on a post-it or index card and set it on your computer for easy reference.

<- ->	To move by character
| |	To move by line
Ctrl -> Ctrl <-	To move by word
Ctrl Home	To move to specific page (Go to)
Home Home |	Top of Document
Home Home |	Bottom of Document

Now We Type

This sample letter will be used throughout the articles. The letter is typed based on 10 pitch font (the WP50 Default). If your left- and right-hand margins are not 10, your letter's right-hand margin will look different. Therefore, you will have to adjust the cursor movement instructions. The following has TYPOS. Please type exactly as is!! The instructions in the parens are NOT TO BE TYPED.

Before starting the second paragraph, Press F4 (the indent key). *WONDER #2:* The indent key will indent the left margin by the tab sets until you press the ENTER key. (This is my personal favorite feature in word processing.)

The CTRL END is a required page break. *WONDER #3:* You can have a page break any place on the page at the 'flick of the wrist.'

Tip: Type with Insert OFF. Press INS. The word 'typeover' will be in the lower left-hand corner.

(Press ENTER 4 times)

January 1, 2000 (Press ENTER 4 times)

John Tweedledee
1234 Wonder Lane
Looking Glass, LS 00000 (Press ENTER twice)

Dearmr. Tweedledee: (Press ENTER 3 times)

I am writing you this letter to remind you of our date at the rabbit hole. We must not be late, because the white rabbit will be highly upset. (Press ENTER once, then Press F4)

> I am certain the tea party will be much fun for many are invited. I was told that the Queen of Hearts will be present, the Mad Hatter is always good for laughs, and the Cheshire Cat will always brighten us will a smile.

(Press ENTER once)
So in closing, my dear Mr. Tweedledee, remember to be punctual and don't forget to bring the white gloves. (You know that the white rabbit always forgets his.) He is so absent-minded.

Sincerely yours, (Press ENTER 4 times)

Alice (Press CTRL ENTER)

You want to save your file and leave it on the screen.

Press F10
Type `alice`
Press ENTER

If, after following the instructions below, your letter *doesn't* look like the one on page 17, clear the screen and retrieve **alice**. Then you can start over again. (To clear screen: Press F7, Press **n** twice; To retrieve file: Press Shift F10, type **alice**, press ENTER.) Be sure to follow these instructions below, exactly. Use the cursor movements as outlined (even if you know an easier way). They are designed to demonstrate the different possibilities. Be sure you are in the TYPEOVER mode before starting this exercise. Go to the top of the document. (Home Home Up Arrow.)

1. **To INSERT a space and change 'm' to 'M'.**

   Using Arrow Down 12 times and Arrow Left 4 times, cursor to the 'm' in 'Dearmr.'.

   Press INS
   Press SPACEBAR
   Press INS
   Type `'M'`

2. **To DELETE a blank line.**

   Using Arrow Down, cursor one line down.

   Press DEL

   (Notice this deletes one of the blank lines. You actually deleted a code. You can delete/insert codes just as you do text. If the one blank line didn't delete, check your display codes (ALT F3). There may be another code there besides the [HR+] (which stands for hard return).)

3. **To DELETE a word.**

   Using Arrow Down once and Ctrl Arrow Right 4 times, cursor to 'to' in the first line.

   Press Ctrl BACKSPACE

4. **To INSERT a character.**

   Using Arrow Down once, END, and Ctrl Arrow Left twice, cursor to the 'r' in 'rabbit'. Using the Arrow Left twice, cursor to the 'e' in 'whie'.

   Press INS
   Type `t`
   Press INS

   (Notice that the character INSERTS to the left of the cursor.)

5. **To TYPEOVER two characters.**

   Using the Arrow Down 6 times and the Arrow Right 5 times, cursor to the first 'l' in 'will' in the last line of the second paragraph.

   Type `th`

6. **To DELETE a blank space.**

   Use the Arrow Down 3 times and Home Arrow Left. Using Ctrl Arrow Right 6 times, cursor to the 'T' in 'Tweedledee' in the first line of the third paragraph.

   Press BACKSPACE

   (Notice that the cursor moves back one but nothing deletes while you are in the typeover mode.)

   Press INS
   Press BACKSPACE
   Press INS

7. **To DELETE TO END OF LINE.**

Using Arrow Down twice and Ctrl Arrow Right twice, cursor to the 'H' in 'He' in the last line.

Press Ctrl END

(This is the EOL or Delete End of Line Key. It will delete any text or codes to the right of the cursor.)

8. **To SAVE** the corrected file.

   Press F7
   Press ENTER
   Type `alice2`
   Press ENTER twice

   You now have two files. Keep both of them. **alice** is the uncorrected file. **alice2** is the corrected file. This is the one you will use throughout the rest of the articles.

Stay tuned for more trips through WordPerfect. The next article will deal with merges and some tips on block copy/move/delete.

MENU.BAT

```
echo off
type menu.dsp
```

MENU.DSP

```
1. WP
2. DBASE
3. LOTUS
Enter Choice by Number and Press ENTER
```

The following is the AUTOEXEC.BAT that calls the menu program.

```
Path = c:\;\wp50;\dbase;\lotus
<sets'path'to'root'and'3'subdirectories>
rtclock
<sets'real-time'clock,'see'DOS'manual>
zspool'40
<sets'spooler'with'40'k'memory,'see'DOS'manual>
menu
<starts'menu'program>
```

January 1, 2000

John Tweedledee
1234 Wonder Lane
Looking Glass, LS 00000

Dear Mr. Tweedledee:

I am writing you this letter to remind you of our date at the rabbit hole. We must not be late, because the white rabbit will be highly upset.

> I am certain the tea party will be much fun for many are invited. I was told that the Queen of Hearts will be present, the Mad Hatter is always good for laughs, and the Cheshire Cat will always brighten us with a smile.

So in closing, my dear Mr. Tweedledee, remember to be punctual and don't forget to bring the white gloves. (You know that the white rabbit always forgets his.)

Sincerely yours,

Alice

1.BAT

```
echo off          <will not print on screen>
cd\data\wp50      <change directory for WP50 files>
wp                <start WordPerfect>
cd\               <return to root directory>
menu              <call menu program again>
```

2.BAT

```
echo off          <will not print on screen>
cd\data\dbase     <change directory for DBASE files>
dbase             <start Dbase>
cd\               <return to root directory>
menu              <call menu program again>
```

3.BAT

```
echo off          <will not print on screen>
cd\data\lotus     <change directory for LOTUS files>
123               <start LOTUS>
cd\               <return to root directory>
menu              <call menu program again>
```

Continued from Page 14

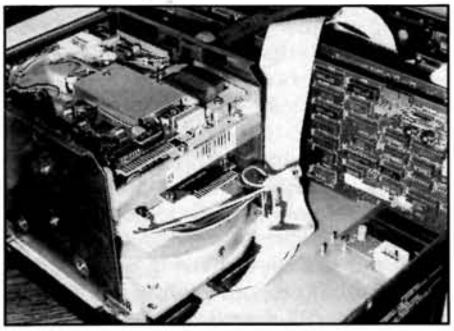


Figure 4
Drives reinstalled with hard disk controller cables and Y power adapter. Power supply removed at this time.

drive rack about 1-1/2" further than the floppy drives. You will also find that the drive rack cannot be installed with the power supply in the computer because of interference with the hard disk and the connectors.

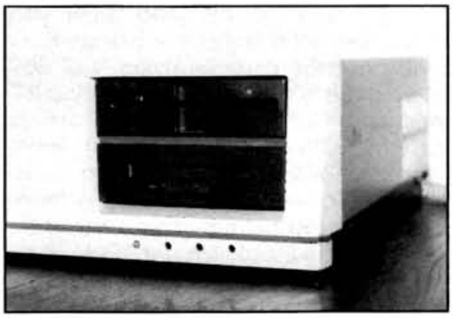


Figure 5
The finished product. Note plastic "light pipe" in the far left hole.

Remove the power supply at this time, and reinstall the disk drive rack in the computer. See Figure 4. Install all the connectors. Remember the data ribbon connectors go on with the red mark on the slotted side of the data connector. I suggest the Y power adapter be installed in the hard disk and drive B, since drive A is readily accessible. This will make for easier removal later. Power connectors P1 and P2 are equivalent, and can be connected to either drive. Fold the hard disk cables down to allow air flow past the floppy and hard disk drives. Reinstall the power supply, and test the system one more time before installing the cover. Everything should be working at this time. The finished product is shown in Figure 5.

Conclusion

I hope this information has been of benefit to you, and that the 3.5" drive increases the usefulness of your H/Z-158.

References

CompuAdd
12303 Technology
Austin, TX 78727
(800) 626-1872
Also, local stores in major metropolitan areas.

dBASE III
Part 6

D. R. Cool
7421 Troy Manor Road
Dayton, OH 45424

In the previous article, the revised project update program (PROJUPD2.PRG) was presented along with five subprograms. In this article, I will analyze the main program. The program (Listing 1 in Part 5) included line numbers which I will refer to.

Lines 6 through 8 set the working environment. The reason for SET TALK OFF has been discussed previously. Setting BELL OFF is mainly a matter of how annoyed you get hearing a computer beep at you every time you make a menu selection. My opinion is that the system should beep only to call the user's attention to a prompt or error message. Setting DELETED OFF insures that deleted records will show up in the data base in the event that you would want to recover a deleted record.

Lines 10 through 13 open the data base and index files that will be used by the program. Previously, only one data base has been opened at a time. DBASE III, however, allows up to ten data base files to be opened at once using the SELECT command. Files are opened in work areas numbered 1 to 10. Normally, when you open a file with the USE command, it is automatically assigned to work area 1. If you then use the USE command to open another data base, the first data base will be closed and the second data base assigned to work area 1.

Using the SELECT command allows you to open more than one data base at once. This facilitates transfer of data from one file to another. Since we have two data base files that are linked by the common field PROJNR, using the SELECT command allows both files to be open simultaneously. To give you an idea of what is accomplished by this, suppose for a minute that dBASE III did not provide this capability. In order to display both the project data and cross reference data, the programming would have to open the PROJECTS data base, get and display the data, save the record number, close the data base, open the PROJXREF data base, find the corresponding record, get and display the data, save the record number for later reference, and, in general, make the programming much more complicated.

Line 14 stores a message used throughout the program to the memory variable PRESS_MSG. If you have a number of prompts or error messages that are used throughout a program, it makes sense to store them in memory variables. This not only saves on typing and program size, but makes it easier to make a change in a message since the change need be made in only one location.

Line 17 establishes the numeric variable ROW. The next command — do SCREEN — runs a subprogram which "paints" the data entry screen. This SCREEN program uses the TEXT and ENDTEXT commands (explained later) to draw the data entry screen. Once the data entry screen is displayed, the main program uses @...GET commands to position the different data fields on the screen. As frequently happens when writing the code for this type of program, you may miscalculate the starting row position of the data entry screen with the result that all of the @...GET commands position the data fields one or more rows offset from where they belong. If all of the @...GET commands used absolute numeric values (e.g., @ 10,15 get PROJNR), then you would have to change each row number throughout the entire program. The way to avoid this is to simply make all @...GET commands relative to a variable such as ROW (@ ROW+5,12 get PROJNR). Then, if for some reason you wanted to move the entire screen up or down one or more rows, all you would have to do would be to change the initial value of ROW.

The SCREEN subprogram, called by Line 18, uses the TEXT command to paint the PROJECTS edit/inquiry screen. Using this command, all text characters between the words TEXT and ENDTEXT are displayed on the screen exactly as typed. This provides a convenient way to see from within a program exactly how a screen display will appear. Also, a screen is painted slightly faster than with @...SAY commands.

Line 20 begins the main program loop, which ends with Line 331. Lines 22 through 29 display the main menu and read the user input into the variable MAIN_OPT. Intensity is set OFF during the read process in order to avoid the appearance of a blank space in reverse video at the cursor position. Note that Line 23 initializes MAIN_OPT to a single space, insuring that only a single keystroke will be required to select any menu option.

Line 30 starts the DO CASE structure which evaluates the user's input. The first case, SELECT = "X", resets the default environment, closes all files, clears the screen and returns to the calling program, which in this case is the dot prompt.

The next selection, SELECT = "S", allows the user to search the data base for a specific drawing. Lines 39 and 40 initialize two memory variables to accept the user input for drawing number and revision letter. To avoid confusion with the field names, the letter "M" (for "memory") is prefixed to the field name. Note that both memory variables are initialized to the number of spaces equal to the widths of the corresponding fields, DWGNR and REV. A new function is introduced with Line 39, the SPACE function. The SPACE function eliminates the need to count spaces. Line 39 could also be written 'store " " to MDWGNR'.

After getting the user's input for drawing number and revision letter, Lines 45-47 give the user a convenient way to exit from this option in case of error. By simply pressing ENTER, the program will loop back to Line 20.

Line 48 stores the current record number — represented by the dBASE function recno() — to the memory variable RECNO. This is done so that, in the event the specified drawing is not in the data base, the record pointer will not remain at the last record. The programming can then return the user to the record prior to the search. This is accomplished with Lines 50-57. This programming also sounds the bell (Line 52), clears row 23 (Line 53, displays an error message (Line 54), waits for the user to press a key (Line 55), then loops back to the start of the main loop (Line 56).

Line 49 uses the SEEK command to search for the combination of MDWGNR plus MREV. The SEEK command conducts a very fast search of a data base, as opposed to LOCATE, which is very slow. This becomes apparent with very large data bases (10,000 records or more). Even with such a large data base, SEEK can find an existing record in less than a second, whereas LOCATE could take up to 40 seconds, depending on the location of the record. The key to understanding how SEEK can be so much faster than LOCATE is the index file. An index is an actual file separate from the data base file. It contains only the values of the key and the record number of the record in the data base which corresponds to the key. The index file could be compared to an index of a book. If you wanted to find a particular subject in the book, you would look in the index and find the page number of the subject. Then you can go directly to that page. Similarly, the SEEK command causes dBASE III to do a very fast search of the index for the specified key. If it finds a match, it immediately gets the record number of the record in the data base that corresponds to the key. The system can then go immediately to that record in the data base.

In contrast to SEEK, the LOCATE command would be similar to trying to find a particular paragraph in a book without an index, you would have to start on page one and scan each page until you found what you were looking for.

Line 58 calls the subprogram GET__PD, which gets all of the project data and displays it on the screen. The next line clears all of the GETs since we only wish to display the data and not change it. Lines 60 and 61 call two additional subprograms which display the project status and display whether or not the project record has been deleted.

Line 62 stores the PROJNR to a memory variable. This information will be used to find the project's associated device type record in the PROJXREF data base. Line 63 selects the PROJXREF data base and Line 64 SEEKs for the previously saved PROJNR. The reason the SEEK command is used instead of FIND, is that SEEK can look for an expression, while FIND cannot. If the expression MPROJNR equals "5962-E1234", then the command SEEK MPROJNR will cause dBASE to look for a record for which the PROJNR field equals "5962-E1234". In contrast, the command FIND MPROJNR would cause dBASE to look for a record for which the PROJNR field equals "MPROJNR".

Line 65 starts an IF...ELSE...ENDIF structure. Whenever you use the SEEK command, you should always consider the possibility that the expression being searched for will not be in the data base. This program has built-in controls that prevent a project record from being initiated without entering at least one device type record in PROJXREF. However, my experience has been, whenever you think something is impossible, it happens. This IF...ELSE...ENDIF simply covers the possibilty that a project will not have a corresponding device type record. In that event, an error message is displayed (Line 71). Otherwise, Line 66 calls another subprogram which gets and displays the device type data and Line 67 clears the GETS.

The next option (Line 74) advances to the next record if not at end of file or displays an error message if already displaying the last record. The SKIP command (Line 76) is the command to advance to the next record. This is followed by a series of commands (Lines 77-85) which get the project data, display the project status and deleted status, and then gets the device type data.

The next option (Line 93) gets the previous record. This is essentially the same as option "N", except the command SKIP -1 is used and the BOF (beginning of file) function is used instead of EOF.

The update option begins with Line 112. In the event that no record has been selected, the record pointer will either be at the beginning or end of the file and the user will be told to "Search first" (Line 116). Otherwise, the user will be asked to select the data base to update (Lines 119-125). This is immediately followed by a DO CASE to evaluate the user's choice.

Line 127 starts the case for the projects update routine. The project update menu is displayed (Line 131) and again the user has a choice to make (Lines 133-135). Another DO CASE structure (Line 137) evaluates the value of the PRUP__OPT (PRoject UPdate OPTion) variable.

The first option, PRUP__OPT = "X", causes the program to EXIT the current DO WHILE loop (the one which began at Line 128). This returns the user to the main menu. The next option, PRUP__OPT = "C", allows the user to edit the current record. Again, the subprogram GET__PD gets the project data. This time, however, it is not followed by a CLEAR GETS statement, but a READ command instead. This allows a full edit of all of the fields in the record. Lines 143-145 implement one of the system requirements: If the user enters a date into the APPDATE field, the project is considered completed and the STATUS field is automatically changed to "C".

After editing the data, the system should run a validation check, which is the purpose of Line 146. The program, VALIDATE, checks the same fields as the PROJVAL program developed in Part 4. Any field found to be invalid will cause the user to make an on-the-spot correction before being permitted to proceed with the next menu choice. After validation, the program again displays projects status (Line 147).

The "D", or "Delete", option requires the user to confirm that he really intends to delete the current record. Note the structure of Line 150. In effect, this line says "do the following for as long as the value of CONFIRM does not equal "Y" or "N". In other words, this forces the user to reply with either the "Y" key or the "N" key. Any other key will keep the user in the loop. If the user replies "Y", then Line 157 deletes the current record, Line 158 calls the subprogram DISP__DEL to display the deleted status, and Lines 159-160 tell the user that the record has been deleted.

Actually, since the delete process does not actually physically delete the record and since it can be easily recalled, the "confirm" routine is not really necessary. In many applications that require continuous daily updating of a large data base, the confirmation routine simply adds more keystrokes to the process.

The next option (Line 163) recalls, or undeletes, a deleted record. Note that this routine does not require the confirmation of the deletion routine. By the way, dBASE III will not complain if you RECALL a record that has not been deleted. (Note: Lines 164 and 170 should be deleted, as they are not necessary. Unfortunately, I caught this error too late.)

The next option, change the STATUS of the project, must deal with three possibilities: 1) STATUS = "C"; 2) STATUS = "A"; 3) STATUS = "D". For the first possibility, remember the system requirement, the status of a completed project cannot be changed. If the user tries to change the status of a completed project, Lines 174-177 will warn the user "Sorry, not allowed".

The other two possibilites, "A" and "D", are covered in one CASE statement (Line 178). This routine shows how you can use memory variables to advantage in not having to write similar code more than once. The command to REPLACE the STATUS field (Line 186) uses the memory variable NEW__STAT. If the current status is "A" (Line 179), then NEW__STAT = "D" (Line 180). Likewise, if the current status is "D" (Line 182), then NEW__STAT = "A" (Line 183). After the status has been changed and the new status displayed (Line 187), the program displays a message (Line 190). This message includes the memory variable NS__WORD, which is initialized to "DISCONTINUED" in Line 181 or "ACTIVE" in Line 184. Thus, the message can say two different things, depending on the current status.

I have, of course, overlooked a fourth possibility — namely, the STATUS field is either blank or does not contain an "S", "D" or "A". Again, this is not supposed to happen. But, for the rare situation where it does, you could simply add an "OTHERWISE" option to the DO CASE structure. In this instance, you might ask the user to enter the correct status.

The final option for the project update menu, the "A" or "Add a new record" option, begins at Line 193. Lines 194-199 prompt the user to enter the drawing number and revision letter of the new project. If the user presses ENTER without entering data, the program LOOPs back to the update menu (Line 201). Otherwise, the program attempts to find an existing record with the same drawing number and rev letter. If a duplicate record exists, the program generates an error message (Line 207). If a duplicate record is not found, the program APPENDs a blank record (Line 210), replaces the DWGNR and REV fields with the data already supplied and stores an "A" in the STATUS field.

Lines 212-213 blank out any device type data of the previous record. Lines 215-216 allow the user to enter the remainder of the project data. The program then validates the data (Line 217), displays the project status (Line 218) and the deleted status (Line 219). The project number is saved to a variable MPROJNR, thereby saving that information for use when adding a device type record. It is at this point that the program forces the user to add at least one device type record for every project record.

Lines 222-241 set up a DO WHILE loop that allows the user to add as many device type records as is necessary. Exit from this loop occurs only if the user answers anything other than "Y" to the prompt (Line 231).

The "otherwise" of Line 243 belongs to the DO CASE started at Line 127. This DO CASE structure ends with Line 245. The ENDDO of Line 246 completes the DO WHILE loop started with Line 128.

Line 247 starts the CROSSREF UPDATE LOOP. This is the second selection possibility for the DO CASE of Line 126. As in the PROJECT UPDATE loop, this loop also starts with a DO WHILE (Line 248). The user is again presented with a menu (Line 252) very similar to the PROJECT UPDATE menu. The NEXT and PREVIOUS options are also included, since more than one device type record could exist for any given project record.

The first selection routine, CRUP_OPT = "N", checks first to see if an end-of-file condition exists. Line 263 checks to see if the project number of the device type record still matches the project number of the current project. If it doesn't match, the SKIP if Line 261 is canceled by the SKIP -1 of Line 264 and an error message is displayed (Line 267). Otherwise, Line 270 calls the program to display the device type data of the next record.

Lines 273-286 process the PREVIOUS record selection, the main difference being that the SKIP command is replaced by SKIP -1. The next option, CHANGE, is pretty straightforward — a command to get the device type data (Line 288) followed by a READ command.

The next option, DELETE a record, and the option following "RECOVER a record" follow pretty much the same format as the DELETE and RECOVER options for the PROJECT menu. Lines 303 and 308 are included to give the user immediate visible indication that the record has been deleted or recovered.

Line 313 starts the routine for the "ADD record" option. This routine appends a blank to the data base, replaces the PROJNR field with the project number previously saved to MPROJNR, then gets the device type data from the user.

The final valid option for the CROSSREF UPDATE LOOP exits from the DO WHILE loop of Line 248. Again, the DO CASE structure includes an OTHERWISE (Line 320) for invalid input. Line 322 ends the DO CASE of Line 258 and Line 323 ends the DO WHILE of Line 248. The ENDCASE of Line 326 completes the DO CASE of Line 126.

The ENDIF of Line 327 completes the IF structure of Line 113. The OTHERWISE of Line 328 is part of the DO CASE beginning with Line 30, which is completed with the ENDCASE of Line 330. Finally, the program loop which began with Line 20 is completed with the ENDDO of Line 331.

As you can see from this program, the use of indentation is almost a necessity in keeping track of so many nested DO WHILE and DO CASE structures. Many apparently inexplicable program errors occur simply because an ENDDO or ENDCASE was forgotten somewhere in the coding of a program.

If you have typed this program (leaving out the line numbers, of course!), I would suggest trying it with the two data bases using various selection combinations. Although I have tested it extensively, I can't guarantee that it will never under any circumstances cause a problem. If you think the program can be improved in any respect, by all means do so. In my next article, I will discuss procedure files and show how the PROJUPD2 program and the five subprograms can be combined into a single file. We will also develop a more generalized menu program which will include an option for monthly reports.

SupersPort SX; Reducing Bad Sectors on Hard Drives; SIMMs for the Z-286, Z-386/20, Z-386/25 and Z-386/33

In a previous article, I mentioned that I was able to significantly reduce the number of bytes in bad sectors on the hard drives on my computers. Many of you sent letters asking me about the details of how I did that, so I will discuss the procedure in some detail in this article. I recently performed that same procedure again on my latest computer, a SupersPort SX. If you are careful and follow the procedure exactly, you can use the same technique to "recover" a significant number of bytes in bad sectors on your hard drive too. But first, let's take a look at the SupersPort SX.

The SupersPort SX

If you have been reading this column for a while, you know that I bought a SupersPort 286 last year to use for writing reports and articles while I am traveling. And although that has been an excellent computer, its CGA screen resolution left something to be desired when I used some graphics programs that I need from time to time. The SupersPort 286 has a double-scan CGA screen that effectively provides a 640 × 400 resolution, but I still needed something better for graphics. The latest ZDS laptops — the SupersPort 286e and the SupersPort SX — both have VGA resolution of 640 × 480 with the "Page White" display that make working with graphics programs (and others too) much easier. The SupersPort SX runs at 16 MHz with an 80386SX CPU as compared with the SupersPort 286's 12 MHz with an 80286 CPU, but I have found a number of interesting improvements in these new laptops that are worth mentioning.

Physically, the SupersPort SX (and SupersPort 286e) look the same, but they are noticeably different from earlier models. Because the LCD (Liquid Crystal Display) screen has VGA resolution, it is thicker than the CGA version, and that shows up as a much thicker panel which contains the LCD. Overall, the VGA unit is about 1/4" thicker, and there is an obvious ridge near the hinged part of the LCD to accommodate that. Most of the other external features are virtually identical to earlier SupersPort models. Other than that, there really does not seem to be a lot of physical difference until you look a little closer. I found two things that are not particularly obvious features, but I think they are worth mentioning.

If you have ever looked at one of the earlier SupersPort models, you may have noticed a "door" on the bottom of the unit that provides easy access to add a numeric coprocessor, such as an 80287. But if you wanted to change the SupersPort's backup battery or ROM, you had to disassemble the computer just to get at those items. Although it is not particularly difficult to change the backup battery in those computers, it did require some disassembly. And changing the ROM was a major task because the ROM is on the underside of the motherboard, which required a TOTAL disassembly of the SupersPort 286. The details of changing the ROM on the SupersPort 286 can be found in my November 1989 article on that subject. On the latest SupersPort models, however, no disassembly is required.

ZDS has thoughtfully added two more "doors" to the bottom of the latest SupersPort models. On my SupersPort SX, you can easily change the backup battery through a door that is located under the keyboard's left front corner. Another door provides easy access to the ROM, which avoids all of the gyrations and disassembly shown in my November 1989 article. Also, I noticed that the SupersPort SX has a "standard size" 5-pin keyboard connector that can accept a regular 101-key keyboard plug. The older SupersPort models had the "tiny size" keyboard connector that could be used with the optional keypad. The addition of these small features and improvements is one of the reasons I continue to buy Heath/Zenith computers.

I also noticed that the SupersPort SX's SETUP program now provides three "options" (using the PgDn key) instead of my SupersPort 286's two. On the SupersPort 286, I have the general SETUP screen and one for the Expansion Chassis. On the SupersPort SX, there is a SETUP screen for AC Power, one for Battery Power, and one for the Expansion Chassis. At this point, I have only used the AC Power SETUP, but there are some very useful new features included in it, too.

Although there are a number of new items in the SupersPort SX's SETUP program, I will only mention two that I think are especially noteworthy. I have stated numerous times that I really dislike the "IBM standard" design of the 101-key keyboard with the CAPS LOCK key to the left of the A key and the CTRL key on the bottom row. Unfortunately, many manufacturers, including ZDS, followed that same design for their 101-key keyboards, and I dislike that design so much that I found the Northgate OmniKey/102 keyboard is much more to my liking. I even bought a second OmniKey/102 keyboard so that I can have one for my Z-248 and one for my Z-386/16. And from my brief test, the OmniKey/102 seems to work just fine with the SupersPort SX. But I have never liked that design on the SupersPort keyboards, and ZDS has fixed that for us in a neat way.

The SupersPort SX has a SETUP parameter that provides an option to adjust the location of the CAPS LOCK and CTRL keys. All you have to do is select the *Caps Lock is:* "Below Shift Key" or "Above Shift Key", using the Space Bar to toggle between the two options. As you might expect, the default setting is "Above Shift Key" to be consistent with the 101-key keyboards and the key labels. I selected the "Below Shift Key" option so that the CTRL key is now to the left of the A key where it really belongs in the first place. I managed to raise some eyebrows when some friends saw me reboot the computer with the CAPS LOCK-ALT-DEL key sequence, but like most touch typists, I rarely look at the keyboard anyway.

A *Keyclick* feature has also been added to the SupersPort SX. I personally like the keyclick, but you can select either "Enable" or "Disable" in the SETUP program, depending on your own preference.

The only thing I do not like in the SETUP program is the fact that I cannot

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

use the "Wini Powerdown" feature to turn off the hard drive in the AC Power mode. I guess the assumption is that you will always want the hard drive running because power is no problem when you are running from the AC line, but I prefer to have the hard drive power down after 10 minutes of no use. My SupersPort SX has ROM version 3.2B, so I hope that will be changed in a future version. Of course, the Battery Power "Wini Powerdown" feature DOES work to conserve battery power, but I would like to be able to do that when I am using AC.

Despite the fact that the 40 MB hard drive on the SupersPort SX had already been PREPed, I reran PREP because I always like to check out my systems from the beginning. In ZDS MS-DOS 3.3 Plus, PREP makes four passes on the hard drive, and it took only about 25 minutes to complete the PREP on this system. Several people have written to me about an "error" in the MS-DOS 3.3 Plus Command Reference about PREP (page 2.12) which states that "This PREP operation requires approximately 30 minutes per 5 MB of hard disk capacity." That statement is based on a slow 8088 system with a slow hard drive. Faster systems with an 80286 or 80386 CPU and faster hard drives (and controllers) will of course be able to complete the PREP operation in far less time.

After running PREP, which creates a single 40 MB partition, I ran PART to create two partitions: a primary partition of about 32 MB and a second partition of about 10 MB. As usual, I had to delete the existing partition created by PREP before I could add those two partitions.

Then I ran FORMAT on both the C: and D: partitions to see how many bytes had been found in bad sectors by PREP. The first partition (C:) had 215,040 bytes "unavailable", and Microsoft changed the wording on that. What "unavailable" really means is that the disk space is marked as bad sectors. And the second partition (D:) had 98,304 bytes in bad sectors. Partition C used a cluster factor of 4 (2,048), so it is easy to calculate that that partition had 105 clusters that were marked as "bad." Partition D has a cluster factor of 2 (1,024), so I calculated that there were 96 clusters marked as "bad." Bad sectors were detected by PREP, recorded in the Bad Sector Table, and then the clusters were marked as bad when FORMAT was run on each partition. By using the technique that I will describe in a minute, I was able to reduce the bytes in bad sectors in partition C to 45,056 bytes (22 bad clusters) and partition D to 12,288 bytes (12 bad clusters). In short, I was able to "recover" 83 clusters on partition C (169,984 bytes) and 84 clusters (86,016 bytes) on partition D. But before we get into the details of how I was able to achieve that kind of reduction, let's take a look at how ZDS MS-DOS marks bad sectors.

Finding Bad Sectors

The PREP program in ZDS MS-DOS 3.3 Plus actually performs six passes on a hard drive. The first pass is an "initialization" step which essentially creates the track and sector headers on the disk. Then PREP performs a two-step pass: the first writes data to every sector and the second is to verify the data was written correctly. After this is performed four times, PREP then makes another initialization pass, which also records any bad sector addresses in the Bad Sector Table. PREP is the most sensitive and thorough hard drive test that you can find in any version of MS-DOS, but that kind of sensitivity to bad sectors is more than I really need. PREP will mark a sector as bad if it fails the verify test on any pass, and the verify test is NOT the same thing that DOS does when it displays an "Abort, Retry, Ignore" type of error.

In any case, the Bad Sector Table is not the end of the process because it simply is a running record of all bad sectors found on the hard disk: initially by PREP, but additional bad sector addresses may also be added by the ZDS DETECT command. When you FORMAT a hard disk partition, the ZDS FORMAT program checks the Bad Sector Table for addresses and marks the CLUSTER containing a bad sector in both File Allocation Tables (FATs). As you will see in a minute, there is a special mark or indicator that the FORMAT program places in the FATs which tell DOS not to use that cluster for any file.

But that is still not quite the end of the story. In ZDS MS-DOS, you can check a hard drive for bad sectors at any time using the DETECT command. The DETECT command checks the entire hard drive for bad sectors, and if any are found, the program updates the Bad Sector Table. DETECT is a non-destructive test program that only reads each sector and updates the Bad Sector Table if a sector cannot be read the first time. Unfortunately, the DETECT command *ONLY* updates the Bad Sector Table; it does NOT mark bad clusters in the FAT to prevent DOS from using that area on the disk. Again, you must run FORMAT to mark the bad clusters in the FATs, which of course means that you must back up the entire hard drive (ALL partitions) first. Although this is a good approach because it keeps a permanent and running record of all bad sectors on a hard drive, it has the disadvantage that you must FORMAT all partitions again when you see a DETECT message like "Bad sectors located. Tables modified." Perhaps DETECT could be modified to tell you which partition contained the new suspect cluster(s), but I think there is a better way with one caution.

A Caution!

If you decide to try this procedure on an existing system, be SURE that you back up ALL partitions on the hard drive first. And before you proceed, be SURE that you have a good backup of all files because you will completely wipe out all file references when you run the FORMAT command, which is a part of this procedure.

If you are starting with a brand new system, then you simply run PREP, PART (if required), and FORMAT. Remember that you must FORMAT each partition, and I recommend NOT using the /S switch to transfer the system. That makes it easier when you make changes to the File Allocation Tables. For a bootable partition, you can use the SYS command to transfer the system files (IBMBIO.COM and IBMDOS.COM) and COPY COMMAND.COM to the root directory of the bootable partition (usually C:).

Again, make sure you have a good backup of ALL partitions on the hard drive before you proceed.

What You Will Need

In order to perform this procedure, you will need a copy of the Mace Utilities which includes the REMEDY program. The major feature of REMEDY is that it will perform a non-destructive read of each sector in each partition and if that sector cannot be successfully read on the first try, then REMEDY will mark that cluster in the FATs and move the file (if any) to another location on the disk. I have used REMEDY for a number of years on all my Heath/Zenith computers and have found it to be incredibly reliable.

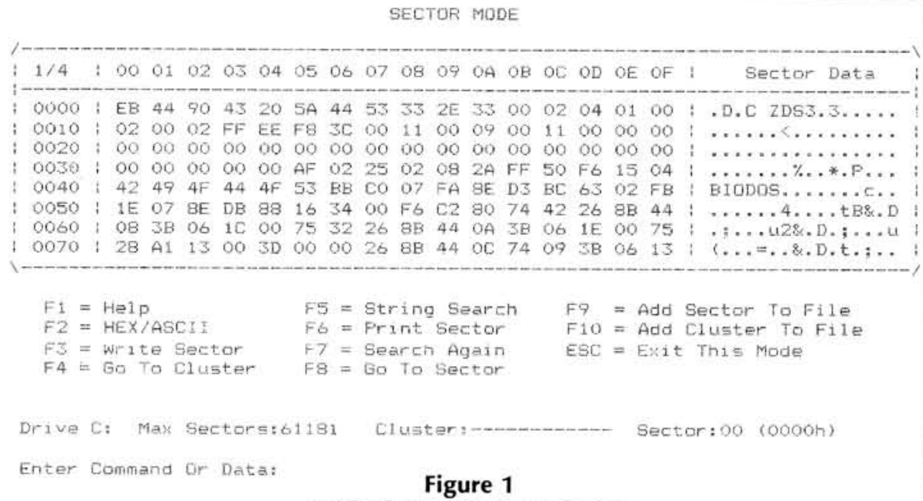
You will also need a special editor that allows you to perform sector-level editing. For example, Mace Utilities includes MUSE (Mace Utilities Sector Editor), but I prefer HUG's HADES program because it has special features that make it especially easy to use. In particular, HADES' capability to perform a search on each sector of a disk make it quite easy to locate bad cluster marks in the FATs.

Last, but not least, you will need a bootable floppy disk which is created with a command like *FORMAT A:/S*. As a minimum, you will also need to copy HADES (or another editor) and FORMAT (from DOS) to that floppy, and I recommend that you also add CHKDSK (from DOS) to the disk. CHKDSK will be used to make sure that the FATs are updated completely.

The Procedure

First, be sure you have a good backup for all hard drive partitions, and boot the computer from the floppy you created. After the system is booted, run FORMAT on each partition. For example, you can use the *FORMAT C:* command to format the first partition. If you have a second partition, then use the *FORMAT D:* command to format it. Do NOT use the /S switch because that complicates things.

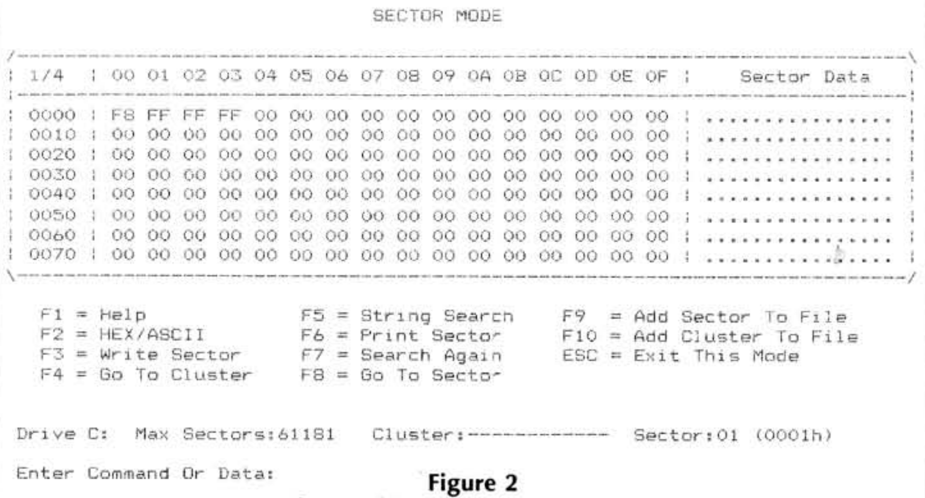
After you have completed running FORMAT on all partitions, it is time to go into the FATs. At this point, I will assume that you are using HADES. Run the HADES program (from the floppy), use the F6 key to change to drive C, and press the F3 key to use the Sector Mode editing feature. At this point, you should see a display similar to Figure 1.



```
                                  SECTOR MODE

/-----------------------------------------------------------------------------\
: 1/4   : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F :    Sector Data    :
:-----------------------------------------------------------------------------:
: 0000  : EB 44 90 43 20 5A 44 53 33 2E 33 00 02 04 01 00 : .D.C ZDS3.3.....  :
: 0010  : 02 00 02 FF EE F8 3C 00 11 00 09 00 11 00 00 00 : ......<.........  :
: 0020  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0030  : 00 00 00 00 00 AF 02 25 02 08 2A FF 50 F6 15 04 : .......%..*.P...  :
: 0040  : 42 49 4F 44 4F 53 BB C0 07 FA 8E D3 BC 63 02 FB : BIODOS.......c..  :
: 0050  : 1E 07 BE DB 88 16 34 00 F6 C2 80 74 42 26 8B 44 : ......4....tB&.D  :
: 0060  : 08 3B 06 1C 00 75 32 26 8B 44 0A 3B 06 1E 00 75 : .;...u2&.D.;...u  :
: 0070  : 28 A1 13 00 3D 00 00 26 8B 44 0C 74 09 3B 06 13 : (...=..&.D.t.;..  :
\-----------------------------------------------------------------------------/

   F1 = Help               F5 = String Search     F9  = Add Sector To File
   F2 = HEX/ASCII          F6 = Print Sector      F10 = Add Cluster To File
   F3 = Write Sector       F7 = Search Again      ESC = Exit This Mode
   F4 = Go To Cluster      F8 = Go To Sector

 Drive C:  Max Sectors:61181   Cluster:------------   Sector:00 (0000h)

 Enter Command Or Data:
```

Figure 1
HADES Boot Sector Display

Note that you are currently editing sector zero (0), which is the first sector on the disk. Also note that HADES displays 128 bytes of data at one time, so you will need to press the PgDn key four times to get to the first FAT. If you have followed this procedure so far, the first FAT will look something like Figure 2.



```
                                  SECTOR MODE

/-----------------------------------------------------------------------------\
: 1/4   : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F :    Sector Data    :
:-----------------------------------------------------------------------------:
: 0000  : F8 FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0010  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0020  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0030  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0040  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0050  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0060  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
: 0070  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................  :
\-----------------------------------------------------------------------------/

   F1 = Help               F5 = String Search     F9  = Add Sector To File
   F2 = HEX/ASCII          F6 = Print Sector      F10 = Add Cluster To File
   F3 = Write Sector       F7 = Search Again      ESC = Exit This Mode
   F4 = Go To Cluster      F8 = Go To Sector

 Drive C:  Max Sectors:61181   Cluster:------------   Sector:01 (0001h)

 Enter Command Or Data:
```

Figure 2
First 16-bit FAT (HADES Display)

Figure 2 shows a 16-bit FAT because the first two clusters (0 and 1) are shown as *F8 FF FF FF*, where F8 is called the FAT ID. A FAT ID of F8 means this is a hard drive. This example was taken directly from my SupersPort SX with a partition size of just under 32 MB. Smaller partitions, say on the order of 10 MB or so, will have a 12-bit FAT. It can easily be recognized by a slightly different set of hex characters of *F8 FF FF*, which also represents the first two clusters (0 and 1). You can look at any kind of newly formatted disk (if you have not used the /S switch with FORMAT), floppy or hard, and determine whether it has a 12-bit or 16-bit FAT by simply examining the first few characters of the FAT. If there are three non-zero characters (e.g., F8 FF FF), then it is a 12-bit FAT. If there are four non-zero characters (e.g., F8 FF FF FF), then it is a 16-bit FAT. And the FAT ID is different for different kinds of disks. For example, the FAT ID for a high density (1.2 MB) 5.25-inch floppy disk is F9.

Now that you are looking at the FAT, you can do one of two things to find the bad cluster marks. First, you can simply scroll through the FAT using the PgDn key to look at each FAT. Or, you can use the HADES search feature by pressing the F5 key to look for an F7 hex. Why look for F7?

The answer is that FF7 (12-bit FAT) or FFF7 (16-bit FAT) is used to identify a bad cluster on the disk. The objective is to find ALL of the bad cluster marks in the FATs (BOTH of them), change them to all zeros, and write the data back to the disk. For example, let's assume that you have found the bad clusters with HADES as shown in Figure 3.

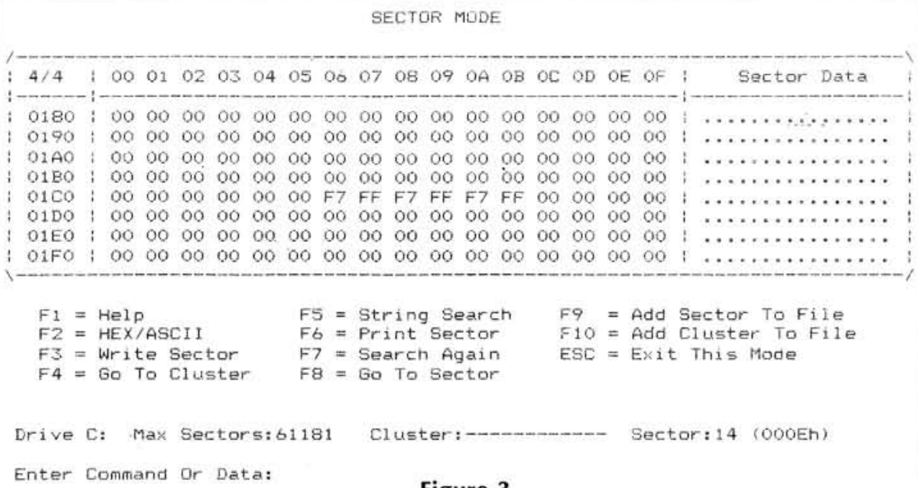
Notice that the bad clusters in this example are marked as: F7 FF F7 FF F7 FF (three bad clusters in this 16-bit FAT). To zero out the bad clusters, just use the arrow keys to move the cursor to the first character set of the bad cluster mark, which is F7 in this example. Then, press and hold the "zero" key (on the main keyboard, not the keypad) to write zeros into those locations. To save the changes, press F3 to write the changes to the disk, and press Y (Yes) to confirm the write. If you have used the HADES' search feature, you can press F7 key to repeat the search for F7 hex. Continue doing this until you have zeroed out all of the bad cluster marks in both FATs.

The process is slightly different if you are searching for an F7 in a 12-bit FAT. Remember, a 12-bit FAT will have a FAT ID of *F8 FF FF* (instead of F8 FF FF FF for the 16-bit FAT), and bad clusters will be marked in the 12-bit format. For example, you may see a series of bad cluster marks that look something like: *70 FF F7 7F FF* or *F7 7F FF F7 0F*. It is extremely important to zero out ALL of these values because they all indicate bad clusters. In other words, you want to be sure to zero out everything in each FAT except for the FAT IDs of F8 FF FF of course.

The easiest way to tell when you have fixed both FATs is to keep an eye on the Cluster (number) in the bottom center of the HADES' screen. Note that no Cluster number is shown in Figure 3 because you are looking at the FAT, and the cluster number will not appear until you begin looking at the Data (or File) portion of the disk. If that number begins to change during the search for F7 hex, then you have completely searched both FATs, and you are done with the procedure. I like to review each FAT manually using the PgDn key after the update, just to make sure that I have zeroed out all of the bad clusters. At this point, you can press ESC twice to exit from HADES and return to the DOS prompt.

Remember that you have been running HADES from a floppy, so be sure that the floppy drive is the current drive. Then, check your work by running CHKDSK (e.g., CHKDSK C:), and you should find that it does not display any "bytes in bad sectors" or "bytes unavailable", depending on the DOS version you are using.

Now it's time to run the Mace REMEDY program from the floppy. The purpose of running REMEDY is to mark the bad clusters that it finds, and I have always found that REMEDY marks only the bad clusters. For configuration purposes, be sure that you have entered the SET command pa-



```
                              SECTOR MODE

/---------------------------------------------------------------------------\
: 4/4   : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F :    Sector Data    :
:-------:-------------------------------------------------:-------------------:
: 0180  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
: 0190  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
: 01A0  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
: 01B0  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
: 01C0  : 00 00 00 00 00 00 F7 FF F7 FF F7 FF 00 00 00 00 : ................ :
: 01D0  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
: 01E0  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
: 01F0  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................ :
\---------------------------------------------------------------------------/

   F1 = Help                F5 = String Search     F9  = Add Sector To File
   F2 = HEX/ASCII           F6 = Print Sector      F10 = Add Cluster To File
   F3 = Write Sector        F7 = Search Again      ESC = Exit This Mode
   F4 = Go To Cluster       F8 = Go To Sector


 Drive C:  Max Sectors:61181   Cluster:------------   Sector:14 (000Eh)

 Enter Command Or Data:
```

Figure 3
Bad Clusters in 16-bit FAT
(HADES Display)

rameter defined in the Mace manual, and I suggest running the REMEDY program for at least 10 passes on each partition to make sure that all bad clusters are marked. Of course, you will want the report to go to the floppy drive.

That's all there is to it. Once REMEDY has completed, you can check your hard drive with CHKDSK again, and you should find that there are far fewer bytes lost due to bad sectors. Now it's time to reload the partition, and assuming that you will start with drive C, here is how to do that.

First, run the *SYS C:* command to transfer the BIOS and the DOS kernel to the hard drive. Then, use the *COPY COMMAND.COM C:\* to copy the Command Interpreter to the root directory of the hard drive partition. Remember that the SYS command does NOT copy COMMAND.COM to the hard drive, so you will have to do that manually. Whatever you do, do *NOT* run the FORMAT command again or you will waste all of the work you have just completed. After loading the BIOS, DOS Kernel, and the Command Interpreter to the hard drive, you can reload all of the files to that partition using the DOS RESTORE command or whatever backup software you have been using.

For maintenance purposes, I suggest running the REMEDY program at least once a week (one or two passes) for the next month or so. After that, I just use the "test" feature of the UNFRAG program when I run it to correct any file fragmentation that might have occurred. The process is simple and effective, and I have used this process for over three years on a variety of Heath/Zenith systems, including laptops, with no problems. And if you have more than one partition on a hard drive, remember that you will have to repeat the process on EACH partition.

When I write a semi-technical article like this, I invariably get a lot of letters on various problems that are virtually impossible to troubleshoot by long distance. If you have any doubts about your ability to follow this procedure using the software I mentioned, then don't do it. Even though the process is quite simple, you may create more problems than you will solve if you do not follow this procedure exactly. And do not even try to attempt it if you have ANY files on a hard drive because you will not know exactly where the bad clusters are.

In summary, this procedure involves six steps. First, backup ALL files on ALL partitions on the hard drive. Second, run FORMAT on that partition (e.g., *FORMAT C:*) to zero out all of the FAT entries except for the bad cluster marks. Third, run HADES on that partition to search for all of the F7 bad cluster marks in BOTH FATs, and change all of the bad cluster marks to zeros. Fourth, exit from HADES, and run the CHKDSK (e.g., *CHKDSK C:*) command to verify that all of the bad cluster marks have "disappeared." Fifth, use the *SYS C:* command to copy the BIOS and DOS Kernel to partition C:, and use the *COPY COMMAND.COM C:\* command to copy the Command Interpreter to the root directory of partition C:, which makes that hard drive partition bootable. And sixth, use the DOS RESTORE command or whatever backup software you are using to restore all files to that partition. You can use this same procedure on any partition, but you will not normally use the SYS and COPY commands to copy the system files to any partition other than C:. For maintenance, I suggest running the UNFRAG program at least once a week for the first month or so, and once a month thereafter.

Z-286 and Z-386 Memory

I have recently received a number of letters about memory for the newer Z-386 computers, especially the Z-386/25 and the Z-386/33. These computers use a SIMM (Single In-line Memory Module) for adding memory and can contain up to 6 MB of 1 MB SIMMs on the motherboard. Yes, these newer computers are back to using a motherboard, probably because the backplane concept would not work too well with these faster computers. The problem is that these newer computers do not use the standard, run-of-the-mill SIMMs, and many of you have asked what the difference is and where you can get them.

As far as I can determine, there are basically two types of SIMMs used by Heath/Zenith desktop computers. The first is the widely available SIMM (1 MB × 9, 80 ns) that fits the Z-286 (12 MHz), and there are many vendors who sell this SIMM, including Payload, who regularly advertises in *REMark*. Like most Heath/Zenith computers which use SIMMs, you can add up to 6 MB on the motherboard because there are "sockets" for them. Finding SIMMs for the Z-386/20, Z-386/25 and Z-386/33 is not so easy because they are physically different in that there is a "key" slot on the edge of the SIMM connector, which prevents the insertion of the "standard" SIMM used on the Z-286. For these newer Z-386 systems, I have found a couple of sources that seem to have reasonable SIMM prices.

Both of these sources are listed in the May 28, 1990, *PC WEEK*. I have not tried either of these sources, and I am mentioning them only as a service to HUG members. The first source, shown on page 98 of that issue, is Unitex, Inc., which advertises 1 MB modules for $120 and 2 MB modules for $269. In that same advertisement, they also list SupersPort 286 memory upgrades: 1 MB for $359 and 4 MB for $1179. According to the advertisement, there is no surcharge for MasterCard and VISA. Their address and phone number is listed at the end of this article.

The second source is South Coast Electronics (page 99), which has 1 MB modules for $110, 2 MB modules for $220, and 4 MB modules for $795. That advertisement also lists the SupersPort 286 1 MB memory upgrade for $450.

Since I have not ordered anything from either of these vendors, I cannot actually recommend them, but I have found that other vendors who advertise in *PC WEEK* are quite reliable. Again, this information is provided only as a service to HUG members because of the outrageous ZDS prices for memory upgrades. If you need additional memory for one of these systems, I will leave it to you to decide which vendor is best.

Powering Down

As I was finishing this article, I received the upgrade notice for ZDS MS-DOS 4.0, so I will be discussing the changes and enhancements later this year. According to the upgrade card, there are two new device drivers (EMM386.SYS and HIMEM.SYS), two new commands (DSKSCAN and MEM), and two new CONFIG.SYS commands (INSTALL and REM). If you have correctly registered your version of ZDS MS-DOS (by returning the registration card included in the package), you should have also received your upgrade notice by the time you read this.

Many new Heath/Zenith computer users ask me about getting upgrades to ZDS MS-DOS. The upgrade notices are sent out automatically to all *REGISTERED* users. In order to be a registered user, you MUST correctly complete and return the registration card included with the ZDS MS-DOS package. In particular, you must complete the name and address section of the registration card (and sign it). The folks at ZDS Software Registration tell me that you would not believe how many registration cards they receive that do not have a name or valid mailing address. Some other software vendors tell me that they receive registrations for less than 25% of the software that is actually sold, so you should be sure to return the registration card for all software as soon as you open the package. Like ZDS, most software vendors make upgrades available at a reduced cost for registered users, but it is obviously impossible for them to notify users without a registration card that contains legible and complete name and address. Be sure to take a couple of minutes to fill out these registration forms completely and legibly.

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

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

Products Discussed
HUG Software

Powering Up (885-4604)	$12.00
HADES II (885-3040)	40.00

Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (HUG Software only)

Software

Mace 5 Utilities	$99.00
Mace Gold Utilities	149.00

Fifth Generation Systems, Inc.
11200 Industriplex Blvd.
Baton Rouge, LA 70809
(800) 873-4384 (Orders only)

Hardware
Z-286 SIMMs

1 MB, 80 ns	$89.00

Payload
15718 Sylvan Lake
Houston, TX 77062
(713) 486-0687
(Orders/Customer Service)
(713) 486-8994 (FAX)

Z-386/20/25/33 SIMMs

1M	$110.00
2 MB	220.00
4 MB	795.00
SupersPort 286 1 MB Upgrade	450.00

South Coast Electronics
10920 Wilshire Blvd., Suite 1100
Los Angeles, CA 90024
(800) 289-8801 (Orders)
(213) 208-3260 (Customer Service)
(213) 208-3282 (FAX)

Z-386/20/25/33 SIMMs

1 MB	$120.00
2 MB	369.00
SupersPort 286 1 MB Upgrade	450.00
1 MB	359.00
4 MB	1179.00

Unitex, Inc.
17222 Armstrong St.
Irvine, CA 92714
(800) 533-0055 (Orders)
(714) 251-8689 (Customer Service)
(714) 251-8943 (FAX)

*

Don't Miss A Single Issue!

Let us know 3-4* weeks

before you move!

FREE MICROSOFT WINDOWS * DETAILS BELOW

Seagate HARD DRIVES

MODEL	CAPACITY/FORMAT/SPEED/SIZE	DRIVE ONLY	XT KIT
* ST-125	21 MEG / MFM / 40 MS / 3.5"	$229.00	$279.00
* ST-138	32 MEG / MFM / 40 MS / 3.5"	$277.00	$327.00
* ST-138-1	32 MEG / MFM / 28 MS / 3.5"	$307.00	$357.00
* ST-151	42 MEG / MFM / 24 MS / 3.5"	$353.00	$403.00
* ST-138R	32 MEG / RLL / 40 MS / 3.5"	$258.00	$313.00
* ST-157R	49 MEG / RLL / 40 MS / 3.5"	$286.00	$341.00
* ST-225	21 MEG / MFM / 65 MS / 5.25"	$199.00	$254.00
* ST-251-1	42 MEG / MFM / 28 MS / 5.25"	$312.00	$362.00
* ST-4096	80 MEG / MFM / 28 MS / 5.25" FH	$582.00	$633.00
* ST-238R	32 MEG / RLL / 65 MS / 5.25"	$218.00	$273.00
* ST-277R-1	65 MEG / RLL / 28 MS / 5.25"	$348.00	$403.00
* ST4144R	122 MEG / RLL / 28 MS / 5.25" FH	$623.00	$678.00

* IDE, SCSI, ESDI AND OTHER SEAGATE MODELS AVAILABLE. PLEASE CALL.

*** ZENITH PC COMPUTER UPGRADES ***

SmartWatch from FBE RESEARCH

⇒Installs in ROM Socket on the CPU Board in Zenith computer series Z-100/ 138/148/150/160 and most all other XT computers. This clock/calendar contains a ten year battery and keeps your computer informed of both date and time at each boot-up. Instructions and software included. $35.00

Z-150/160 MEMORY UPGRADE

⇒This kit includes a replacement memory decoder PAL chip for the standard Z-150/160 memory card (not for the Z-157/58). With this PAL and the 18 pieces of 256K RAM chips (included), you will expand the memory on the card to 640K or 704K. ZP640+/18 Kit.....$59.00. PAL chip only ZP640+....$18.00

Z-150 SERIES HARD DISK DRIVE KIT

⇒These kits include high speed Seagate drives with autopark heads. Each kit includes all cables, hardware and instructions to mount the hard drive under your two floppy drives in your Z-150 series computer.

* ST-125/Z150 Kit	21 Meg, 40 MS,	$283.00
* ST-138/Z150 Kit	32 Meg, 40 MS,	$331.00
* ST-151/Z150 Kit	42 Meg, 24 MS,	$409.00

Z-148 HARD DISK DRIVE KIT

⇒Includes the hard disk drive and a Z-148 compatable controller together with the Z-148 Expansion Card described below. All required cables, hardware and instructions are included for you to replace one floppy with a Seagate Hard Drive in your Z-148. Add only $30.00 the the following price if your would like us to include a SmartWatch.

* ST-125/Z148 Kit	21 Meg, 40 MS,	$354.00
* ST-138/Z148 Kit	32 Meg, 40 MS,	$402.00
* ST-151/Z148 Kit	42 Meg, 24 MS,	$478.00

Z-148 EXPANSION CARD

⇒Adds one full length and one half length IBM expansion slot to your Z-148 for hard drive controller, video card, modem, etc. ZEX-148......$79.00

Z-150 VIDEO ELIMINATOR

⇒For the Z-150 or Z-160 only. Not required for the Z-157/158/159 computers. A small piggyback board which replaces the scratch pad memory on your current video card. This allows the removal of the original Zenith video card and replacment with an EGA, VGA or any other 8 bit video card. Order VCE-150 .... $54.00

2400 BAUD MODEMS

⇒Fully Hayes compatible 2400/1200/300 Baud with Software. Internal $99.00 External Model $128.00 Cable for External Modem $8.50

15718 SYLVAN LAKE * HOUSTON TX 77062

FLOPPY DISK DRIVES

⇒MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	$ 68.00
⇒MIITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	$ 81.00
⇒ MITSUBISHI M-353	3.5" in 5.25" frame 720K	$ 84.00
⇒MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	$ 94.00
⇒TOSHIBA ND352	3.5" with 5.25" frame 720K	$ 74.00
⇒TOSHIBA ND356	3.5" with 5.25" frame 1.44 MEG	$ 79.00

⇒M-355 and ND356 run on AT compatible or special controller only.

PAYLOAD CUSTOM ASSEMBLED COMPUTERS

⇒We assemble 8088 XT, 80286 AT, 80386SX 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-1492	ZENITH Color Flat Screen VGA	$ 679.00
⇒MA2565	SAMSUNG Amber TTL 720x350	$ 89.00
⇒CW4656	SAMSUNG Color RGB 640x200	$ 223.00
⇒CM4592	SAMSUNG Color EGA 640x350	$ 339.00
⇒CJ4681	SAMSUNG VGA 720x400	$ 360.00
⇒CVB4581	SAMSUNG Multi-sync VGA 1024x768	$ 469.00
⇒CM1440	SEIKO VGA 1024x768 .25 dot	$ 609.00

VIDEO CARDS

⇒EGA480	PARADISE AUTOSWITCH 640x480	$ 99.00
⇒VGAPLUS	PARADISE AUTOSWITCH 800x600	$ 168.00
⇒VGAPLUS16	PARADISE AUTO 16 BIT 800x600	$ 207.00
⇒VGA 1024-256k	PARADISE AUTO 1024x768	$ 249.00
⇒VGA 1024-512k	PARADISE AUTO 1024x768	$ 309.00

MEMORY CHIPS, ETC.

⇒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 80 ns.....$2.95	SIP 1Mx9 80 ns $80.00
⇒41256 256x1 100 ns $2.65	SIM 1Mx9 80 ns $79.00
⇒41256 256x1 120 ns $2.30	SIM 256x9 80 ns $24.00

Z-100 SERIES COMPUTER UPGRADES

High Density 1.2 Meg Drives

⇒External floppy drive set-up complete with drive, power supply, case and cables. 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

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. ZMF100A$60.00

UCI Memory Upgrade Pal Chip Set

⇒For the Z-100's with the newer motherboard part number 181-4918 or greater. Allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K 150 ns RAM chips (sold separately) a total memory of 768K is obtained. Chip Set....... $64.00

UCI EasyWin HARD DRIVE SYSTEM

⇒Complete Hard Disk System for mounting inside your Z-100. 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

Z-100 SERIES SOFTWARE

⇒PART NUMBER /DESCRIPTION	LIST PRICE	SALE PRICE
⇒CD-463-2 Condor File Manager	$175.00	$10.00
⇒MS-253-1Basic-80 (8-bit)	$175.00	$10.00

MICROSOFT WINDOWS version 1.04 for PC (not Z-100)

⇒FREE with any order from this ad over $30.00. Just ask for it and add $5.00 for
⇒shipping and handling. Includes 5 disks and 300+ page manual. Offer good until
⇒existing inventory given away (about 300 packages).

⇒Please Mail, Phone or FAX your order. All hardware carries the manufacturers warranty plus the PAYLOAD 90 day guarantee. No surcharge on credit card orders. COD shipments on request. Add $5.00 to all prepaid orders for handling and shipping in the Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders. We accept purchase orders from schools, government and approved accounts. Mail or phone your order for prompt friendly service. Texas residents please add 8.0% state sales tax.

PHONES: ** ORDERS AND INFO **(713) 486-0687** ** FAX: **(713) 486-8994** **

PAYLOAD * PAYLOAD * PAYLOAD * PAYLOAD

Reader Service #117

WordPerfect Macros

Programming Language

Part 2

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

Introduction

In an earlier REMark article, August 1990, I covered keystroke macros and gave a brief introduction to the WordPerfect macro programming language (MPL). This article covers MPL in considerably greater detail. First, I present two useful macros and explain how they work. One stores text in a variable for future retrieval. The second does super- and subscripting based on user input. These examples should help you understand some of the important MPL commands. Next I mention some important commands that are not used in the examples. Finally, I explain how to use the standalone WordPerfect macro editor.

A Macro to Store Text in a Variable

If you store text in a variable, you can insert that text anywhere in your document by pressing <ALT-variable number>. Text can be stored in a variable in several ways. One of the most convenient ways is shown in the macro below. When the macro is activated, a prompt asks for the text to be stored. Pressing the <RETURN> key terminates text storage and displays a prompt that instructs the user how to retrieve stored text.

Please note that the line numbers are not present in the actual macro. They have been added here so that I can easily direct your attention to various commands.

Also note the tilde characters (~) at various points. These tildes are essential, and they're easy to omit. If you leave one out, the behavior of the macro may be quite odd. When debugging macros, missing tildes is the first thing to look for.

```
1 {TEXT}1~The text you type will be
  stored in variable 1 until
2           you press <RETURN>{Enter}~
3 {PROMPT}{Up}{Del to EOL}Retrieve your
  text by typing <ALT-1>~
4 {BELL}
5 {WAIT}10~
6 {Screen}r
```

Line 1 prompts the user to enter text and stores what the user types until <RETURN> is pressed. The 1 immediately following the {TEXT} command designates variable 1 for text storage. A tilde separates the variable number from the screen prompt, which continues on Line 2. Raised dots are placed in the prompt by the macro editor when the space bar was pressed. They do not appear on the screen during macro execution. The {Enter} command just before the tilde moves the prompt up one line so the user can type text on a blank line. <RETURN>, which appears just before the {ENTER{ command, is part of the prompt. It is text, not a command.

During execution of macros, carriage returns and tabs are ignored, so you can use these keys to make the macro more readable. Lines 1 and 2 are all part of one command. They were split into two lines during macro creation to make the command easier to read.

The {PROMPT} command in Line 3 erases the previous prompt and creates a new one. The {Up} and {Del to EOL} commands erase the first prompt. The rest of the line creates a new prompt which instructs the user how to retrieve the stored text. {Up} is placed in the macro by typing <CTRL-V> followed by the up arrow key. {Del to EOL} is created by typing <CTRL-V> and then <CTRL-K>. These codes are covered in the manual under Control Characters.

{BELL} sounds a beep and {WAIT}10~ creates a delay. Delays are specified in tenths of a second, so 10 is one second.

The command in Line 6 redraws the screen. Pressing <CTRL-F3> during macro creation inserts the {Screen} command. When the macro executes, this command calls up the screen menu. The macro Selects "r" from the menu, which causes the screen to be redrawn. This is necessary in order to avoid confusion in the display.

Each time the macro is executed, new text is stored in variable 1. This text can be inserted into the document as many times as needed by pressing <ALT-1> with the cursor at the desired point. As I wrote this article, I executed the macro and stored "<CTRL-" in variable 1. This was a big help in typing the paragraphs above.

A Subscripting & Superscripting Macro

Depending on user input, the macro listed below subscripts or superscripts the character to the left of the cursor.

The program listing is written in readable form with one command per line and indenting to show how parts of the program relate to each other. Since MPL ignores hard returns and tabs, the macro can actually be written in the format shown. This is not necessary. You can write the whole macro as one continuous word wrapped line, but if you do, you won't be able to read it later.

```
1 {LABEL}TOP~
2 {CHAR}1~
3     {^]}1{^\} su{^V}B{^Q}script
    {^]}2{^\} su{^V}P{^Q}erscript~
4 {CASE}{VAR 1}~
5     1~subs~
6     B~subs~
7     b~subs~
8     2~supe~
9     P~supe~
10    p~supe~~
11    {PROMPT}INVALID CHOICE,  TRY AGAIN~
12    {BELL}
13    {WAIT}5~
14    {GO}TOP~
15 {LABEL}subs~
16    {DISPLAY OFF}
17    {Left}{Block}{Right}
18    {Font}12
19    {DISPLAY ON}
20 {QUIT}
21 {LABEL}supe~
22    {DISPLAY OFF}
23    {Left}{Block}{Right}
24    {Font}11
25    {DISPLAY ON}
26 {QUIT}
```

How the Macro Works

Since MPL ignores carriage returns, the {LABEL} command must be used to establish reference points. Thus line 1 establishes a reference point at the beginning of the macro. Note that line 14 directs macro execution back to this point if the user makes an invalid menu selection.

Lines 2 and 3 form the {CHAR} command. This command displays a prompt on the screen and waits for the user to press a key. It stores the character typed by the user in a variable. In this macro, the character is stored in variable 1, since a 1 follows the {CHAR} command. The tilde following the 1 separates the variable number from the screen prompt. {CHAR} is similar to {TEXT}, but {CHAR} takes only the first character typed and does not wait for a carriage return.

Line 3 is the screen prompt to be displayed. It looks rather cryptic, doesn't it. If you ignore the bracketed characters, you can see that it says "1 suBscript 2 suPerscript." The bracketed characters control the appearance of 1, 2, B and P on the screen. Characters typed between {^]} and {^\} will appear bold faced. Characters typed between {^V} and {^Q} will appear like menu selection letters in the standard WordPerfect menus.

When the macro is activated, the screen prompt appears beginning at the bottom left corner of the display. The user may press 1, B or b to subscript the character to the left of the cursor and 2, P or p to superscript it. If any other key is pressed, an error message is displayed.

Line 4 is the beginning of a {CASE} structure. This line causes the contents of variable 1 to be compared with the choices in lines 5 through 10. If a match is found, execution transfers to the labeled location. If the value of {VAR 1} is 1, B or b, macro execution is transferred to {LABEL}subs~ at Line 15. If {VAR 1} has the value 2, P or p, execution is transferred to {LABEL}supe~ at Line 21. Note the double tilde (~~) at the end of line 10. One tilde marks the end of that particular line; the other marks the end of the {CASE} structure. Both tildes must be present.

If the user presses anything except 1, B, b, 2, P, or p, the {CASE} statement fails, and execution continues with Line 11. This is the error message mentioned above. It appears at the bottom left of the screen. {BELL} in Line 12 causes a beep to sound to alert the user to the error message, and {WAIT}5~ in Line 13 causes a pause of 5 tenths of a second.

Lines 11 through 14 are executed only if the user has not made a valid selection from the menu, and the {GO}TOP~ statement in Line 14 transfers execution back to the beginning of the macro so the user can try again.

The {LABEL} statements in Lines 15 and 21 are the starting points of two subroutines. The one that begins in Line 15 handles subscripting, and the one beginning in Line 21 handles superscripting. These points in the macro can only be reached from the {CASE} structure.

At the beginning of each subroutine (Lines 16 and 22) the {DISPLAY OFF} command disables the display of the font menu. Since the macro, not the user, makes a selection from the menu, display is unnecessary, and execution is faster if menu display is suppressed. Lines 17 and 23 highlight the single character to the left of the cursor. Line 18 calls up the font menu and selects 1 (size) and 2 (subscript). Line 24 calls up the font menu and selects 1 (size) and 1 (superscript). Lines 19 and 25 enable future menu display. Lines 20 and 26 terminate macro execution.

Other MPL Programming Structures

The two macros described above illustrate several important MPL commands and programming structures. Others not illustrated are included in Table 1.

Command	Comment
{IF}...{ELSE}...{ENDIF}	Commands between {IF} and {ELSE} are executed when the {IF} test is true. Otherwise, commands between {ELSE} and {ENDIF} are executed.
{ASSIGN}variable~value~	Stores a number or a string in a variable.
{CALL}label~	Transfers execution to a location named "label". This differs from {GO} because execution will later return to the command below {CALL}.
{CHAIN}	Initiates another macro when the current one ends.
{NEST}	Initiates another macro too, but execution will later return to the command below {NEST}.
{PAUSE}	Halts macro execution until <ENTER> is pressed. The user is free to type text or issue commands during a pause.

Table 1

The Stand-Alone Macro Editor

As described in my earlier article, command language macros can be created with the macro editor built into the WordPerfect word processing program, but this editor is rather cumbersome for all but the simplest macros. The stand-alone macro editor program ME.EXE makes macro creation and editing much easier.

If you try to load a macro file into the WordPerfect word processing program, it won't load, but you can load a macro file into the macro editor. Think of the macro editor as a word processing program for creating and editing macros. Its command structure and operation are very similar to the word processor. Bold facing, underlining, centering and indenting are missing; they aren't needed in macros. Block move, delete and copy work as you would expect. Searching, printing, saving and retrieving work as they do in the word processor.

The function key template for the macro editor is shown below. In each block, the top line is <CTRL-function key>, the next line down is <ALT-function key>, second from the bottom is <SHIFT-function key> and the bottom is <function key>.

```
    Shell             Wrap
F1   ***              REPLACE        F2
    Comment           <-Search
    Cancel            ->Search
    Screen            Move
F3  CODES             BLOCK          F4
    Switch            Append
    Help              Copy
    Interpret         Macro Def
F5   ***              COPY LINE      F6
    Date              Dup Line
    List Files        Dup Word
     ***              Print Format
F7   ***              OPTIONS        F8
    Print             Macro Summary
    Exit              Switch
    Top               Functions
F9  BOTTOM            MACRO          F10
    Middle            Retrieve
    Block             Save
```

Look closely at the F10 block. Pressing <F10> alone initiates the process of saving the macro displayed on screen. <SHIFT-F10> retrieves a macro from disk into the macro editor. <ALT-F10> enables you to run macro editor macros! Creation of these macros is started by pressing <CTRL-F6>. They automatically get the file extension MEM to distinguish them from word processor macros, which have the extension WPM.

If you press <CTRL-F10>, every key inserts its name into the macro when you press it. Thus pressing <ENTER> causes ⅛Enter⅜ to appear in the macro, and pressing the back space key causes {Backspace} to appear. These commands will do what their names say when you run the macro. If you had not pressed <CTRL-F10>, pressing <ENTER> would move the cursor to the beginning of the next line of the macro, and pressing the back space key would delete the macro character or command to the left of the cursor. Pressing <CTRL-F10> a second time restores normal key function.

<F3> activates the help facility, which works like the word processor's help facility. When help is active, pressing any key produces a help screen that applies to that key. This system is so good you can almost learn to use the program from it.

<F1> has about the same function as in the word processor. The undelete feature of <F1> is especially handy. It not only lets you recover from an unintended deletion; it can also be used as a substi-

Continued on Page 46

Programming the LaserJet

Part 2

Alan Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

The PCL command language that controls Hewlett-Packard printers is divided into five downward compatible levels. For example, a level 4 device can print files containing level 1, 2, 3, and 4 commands.

Non-laser HP printers are usually level 1 and 2 devices. By using a subset of the PCL command language, these printers can handle basic text files and documents using a standard HP-compatible printer driver.

The original LaserJet printer was a level 3 device because it could not handle downloadable fonts, or softfonts. The LaserJet Plus, LaserJet II, IID and IIP are level 4 printers, the highest level until the recently announced LaserJet III, the first level 5 printer.

This new level introduces to the LaserJet line automatic font scaling (as in PostScript) and the ability to modulate the laser beam for highest resolution printing.

In this article, I'll review a basic set of level 4 PCL commands that are used most frequently. My book *"Your HP LaserJet Handbook,"* published by Sybex Computer Books, includes full details and examples on programming with PCL.

Escape Code Conventions

In a previous article, I explained how to transmit PCL commands to your printer in the C, Pascal, BASIC, and dBASE programming languages. That article explained how to send the escape code, ASCII code 27.

In this article, I'll refer to the escape code simply as <ESC>. When you see this code in a PCL command, it means to send the escape character to the printer. For example, the code <ESC>E, used to reset the printer, would be transmitted as CHR$(27)+"E" in BASIC.

The commands also include specific parameterized and parameter characters, as well as a user-added field value. The field value will be shown as #. For example, <ESC>&a#H means to transmit the escape code, the characters &a, your selected field value, then the terminator character H. Valid field values will be illustrated.

Now let's look at some useful PCL commands.

Cursor Movement

Cursor movement is performed in either decipoints, dots, lines or columns. Movement can be absolute, measured from the "home" coordinate position, or relative to the current coordinate. If the value field (#) has no sign, the cursor is positioned absolutely. A sign indicates relative movement to the right or left of the current position.

In addition to the ASCII control codes for carriage return, back space, and horizontal tab, you can set the horizontal cursor position by either character positions, decipoints, or dots. The command <ESC>&a#C, for example, sets the position by column number. For instance, to move the cursor five positions from the 0 position on the X axis, enter the command <ESC>&a5C. Transmit <ESC>&a +5C for a relative move five positions from the current position.

<ESC>&a#H sets the position by decipoints, while <ESC>*p#X sets the position by dots.

Cursor movement vertically is performed by the line feed and form feed ASCII control codes, as well as escape codes for lines, decipoints or dots.

<ESC>&a#R sets the position by row (line) number. The size of each row is set by the vertical motion index, normally the point size of the font. The first row is row 0.

<ESC>&a#V sets the vertical position by decipoints and <ESC>*p#Y sets it by dots.

<ESC>= performs a half-line feed, based on current vertical motion index.

Job Controls

Job Controls commands set the overall nature of the print job, including the type, source, and size of the paper. <ESC>&l#X sets the number of uncollated copies (up to 99).

Select the paper feed with <ESC> &l#H. Valid field values depend on your printer model, can range from 0 to 6 as shown below:

Field Value	Action
0	No Change — Form Feed
1	Upper Tray
2	Manual Paper
3	Manual Envelope
4	Lower Tray
5	Paper Deck
6	Envelope Feeder

Use the command <ESC>&l#A to select the paper size with these possible field values:

Field	Size
1	Executive
2	Letter
3	Legal
6	Ledger
26	A4
27	A3
80	Letter Envelope
81	Business #10 Envelope
90	DL Envelope
91	C5 Envelope

(This command cannot be used on models earlier than the LaserJet II, so use the Page Length command instead.)

<ESC>&l#P sets the page length in lines of the logical page — the field value is the number of lines per page. With the LaserJet, LaserJet Plus, and LaserJet 500 Plus, use this setting to determine the page length since the Page Size command is not available.

<ESC>&l#O determines the orientation. The field value can be either 0 for portrait or 1 for landscape.

Duplex Printing

These commands are available only in the LaserJet IID and LaserJet 2000.

<ESC>&l#S selects either single- or double-sided printing and the binding edge. Use a field value of 0 for single-sided, 1 for duplex with vertical binding, or 2 for duplex with horizontal binding.

Text Area

These commands determine the number of lines and characters that will print on each page. While there is a top margin setting, no PCL command directly establishes the bottom margin. Instead,

LaserJet PCL Commands

calculate the bottom margin by subtracting the top margin and text length from the physical page size.

<ESC>&a#L sets the left margin in columns, <ESC>&a#M sets the right margin in columns.

<ESC>9 resets the margins to the left and right boundaries of the logical page.

<ESC>&l#E determines the top margin in lines based on the currect vertical motion index or line spacing. The top margin becomes the Y-axis 0 coordinate.

<ESC>&l#F sets the number of lines that will print starting at the top margin.

Spacing Commands

These important commands control the distance the cursor moves with each line or character printed.

<ESC>&l#D sets the number of lines per inch, with a default value of 6. Valid field values are 1, 2, 3, 4, 6, 8, 12, 16, 24, and 48.

<ESC>&l#C controls the vertical motion index, or VMI, which is the number of 1/48 inch increments between rows. Use this command as an alternative method of changing line spacing, particularly for settings not available with the Line Spacing command. Field values range from 0 to 336.

The VMI effects all commands using lines (or rows) as a measure. For example, the command to change line spacing to four lines per inch is <ESC>&l12C. If you then used the command <ESC>&a6R to move the cursor down six lines, the cursor moves 1-1/2 inches. The same command at the default VMI would move the cursor just one inch.

<ESC>&k#H is the Horizontal Motion Index (HMI), the number of 1/120 inch increments between columns. The default value of 12 accommodates 10 pitch characters. Field values range from 0 to 840. The HMI effects all commands using columns as a measure. Use this command to add extra space between characters or compress text.

Font Selection

By default, the LaserJet uses the internal 12-point Courier font for all text. Using an application with a LaserJet driver, you can select any available font, then format characters with attributes, such as size, boldface and italic.

In your own program, or using PCL commands from within applications, you have the same capabilities, even more. You select the font to use (called the primary font) by specifying its attributes or, for a softfont, its ID. But you can also choose a secondary font, then quickly switch between the two with the shift-in and shift-out control codes like this:

```
10 LPRINT CHR$(14);"This is the secondary font"
20 LPRINT CHR$(15);"This is the primary font"
```

Use shift-out (ASCII code 14) to select the secondary font; shift-in (ASCII 15) to return to the primary font.

Designating primary and secondary fonts is called font selection.

Let's first take a look at using attributes to select fonts.

Selecting Fonts by Attribute

<ESC>(SET-ID designates the font with this symbol set as the primary font. The SET-ID is a unique two or three character code that has been assigned to all HP compatible symbol sets. If not found, the default set is used. To select the PC-8 set, for example, use the command <ESC>(10U.

<ESC>(s#P sets the spacing of the primary font as either fixed (0), or proportional (1).

The field value of the command <ESC>(s#H is the pitch in characters per inch of the primary font. When one with this value isn't found, the LaserJet first looks for the next greater value, then the closest lower value.

<ESC>(s#V selects the point size of the primary font. If the exact point size isn't found, the closest font will be selected.

Use <ESC>(s#S to select either an upright (0) or italic (1) primary font.

The command <ESC>(s#B determines the stroke weight of the primary font using these field values:

Field	
Value	**Weight**
-7	Ultra Thin
-5	Thin
-3	Light
0	Medium
3	Bold
5	Black
7	Ultra Black

If the designated weight isn't located, the next thickness is used, and if not found, the next thinnest. As an example, suppose you have a font available in medium (normal) and bold. The commands <ESC>(s3B, <ESC>(s5B, and <ESC>(s7B would all select the bold font.

<ESC>(s#T determines the font family of the primary font. Your printer manual will list the field values for most commonly used type families.

The command to set all new characteristics can be quite complex. In this example, we'll select an 18-point proportionally spaced, Helvetica bold italic font in the ASCII symbol set, printed in landscape. Because the font is not fixed width, we don't have to specify the pitch.

First, look at the individual commands:

Landscape Orientation	<ESC>&110
Symbol Set — ASCII	<ESC>(OU
Spacing — Proportional	<ESC>(s1P
Height — 18 Point	<ESC>(s18V
Style — Italic	<ESC>(s1S
Weight — Bold	<ESC>(s3B
Typeface — Helvetica	<ESC>(s4T

Putting the commands on one line yields <ESC>&l1O<ESC>(0U<ESC>(s1P <ESC>(s18V<ESC>(s1S<ESC>(s3B<ESC> (s4T.

Then combining the codes using the rules outlined in my previous article, the final code is <ESC>&l1O<ESC> (0U<ESC>(s1p18v1s3b4T.

In BASIC, the command would look like this:

```
10 LPRINT CHR$(27);"&l10";CHR$(27);
   "(ØU";CHR$(27);"(s1p18v1s3b4T"
```

If you're using the default font, or have already selected a new primary font, you can change specific attributes by just specifying their characteristics.

For example, to print in bold using the current font, use the command <ESC>(s3B. To select italic print, use the command <ESC>(s1S.

Selecting Automatic Underlining

Underlined characters are not created with a font but with the printer's automatic underline feature. Use the command <ESC>&d#D to turn on underlining, <ESC>&d@ to turn it off.

The field value, either 0 or 3, determines the position of the underline. If the field value is 0 (or not included) the line is drawn starting on the fifth dot under the baseline, three dots thick. This is called *fixed position underlining*.

With a field value of 3, the line position is determined by the fonts on the line. All fonts contain an internal descriptor called *underline distance*, the recommended starting position under the baseline. Using floating position underlining, as this mode is called, the line is placed at the lowest underline distance of all of the fonts on the current line.

Selecting Softfonts by ID

Because so many codes are needed to select a unique font by attribute, it is more efficient to select downloaded softfonts by their ID number using the command <ESC>(#X.

The field value is the ID number assigned to the font when it was downloaded. To use the font with ID five as the primary font, for example, use this command <ESC>(5X.

Selecting by Pitch

The original LaserJet, LaserJet Plus, and LaserJet 500 Plus have a quick way of changing from standard 10 pitch to the 16.66 pitch line printer font.

The command <ESC>&k2S selects line printer as the current primary and secondary font. Reselect 10-pitch using <ESC>&k0S.

Returning to the Default Font

To return to the default font after selecting fonts, use the command <ESC>(3@. This sets all of the attributes to those of the default font.

Managing Softfonts

If you have a font downloading program, the business of assigning ID and permanent or temporary status is handled for you. The same occurs when downloading certain fonts using the DOS COPY command. The ID and status information is contained in binary data in addition to the font bitmap itself.

Most softfont files, however, do not contain this information and can only be downloaded with the COPY command if you manually specify the ID number and status using PCL commands.

These commands, and the PCL code to delete fonts, are classified as *font management*.

The command <ESC>*c#D transmits the ID number of the next font downloaded to the printer. If you are about to download a font as ID 3, for example, use the code <ESC>*c3D.

Use the command <ESC>*c#F immediately after downloadng the font. The field value designates the font as temporary (4) or permanent (5) status. Other field values will be discussed shortly. Until you designate a font as permanent, it is considered temporary. So the field value of 4 is useful for changing the status of a permanent font. As an example, if you just downloaded font 1, make it permanent with the command <ESC>*c5F.

This command changes the status of the last softfont referenced. To make sure you're setting the correct font, combine the Font ID code with the Status command. For instance, to set font number 0 as permanent, at any time, use the command <ESC>*c0d5F.

Use <ESC>*c#F to delete softfonts from memory. Use a field value of 0 to remove all fonts, or 1 to delete just temporary softfonts. If you want to delete a specific font, use the field value 2, but first designate the font's ID using the <ESC>*c#D command. For instance, delete font 3 with the command <ESC>*c3d2F.

As an example, here's how you would download a font called HEL240-RPN.USP, with the ID 3 and permanent status.

/t/

1. If this is the only font you want loaded, delete all current softfonts with the command <ESC>*c0F.
2. Send the PCL code to designate the font ID: <ESC>*c3D.
3. Use the DOS copy command, with the /B option, to download the bitmap: c:> COPY /B HELV240RPN.USP LPT1.
4. Make the font permanent with the command <ESC>*c5F.
5. If you want to use the font immediately, select it as the primary font with the command <ESC>*(5X.

Graphics

There are two categories of PCL commands for printing graphics: raster graphics and rectangular area fill.

Raster graphics commands are used to print images that are transferred to the LaserJet bit-by-bit. You get complete control over the design, but it is really only practical for those with very small images, or with a great deal of time since at 300 dpi resolution, a one square inch graphic image is 90,000 dots.

Creating raster graphics is a time-consuming process that requires an understanding of the concept of resolution. When the LaserJet is set at 300 DPI, each dot you transmit is printed as one dot, 1/300th of an inch large. So if you transmit a graphic that's 300 dots high and 300 dots wide, the printed image will be one inch square. At 150 dots per inch, however, the LaserJet prints each transmitted dot as a two-by-two matrix, using four printed dots for each dot in the graphic. This makes the image twice the size.

A graphic designed to be 300 dots square will print 300 dots high and 300 dots wide, one inche square. The same dots print as a three-by-three matrix at 100 dpi — three times the size, and a four-by-four grid at 75 dpi — four times the size.

As the matrix used to represent each dot grows larger, two visual effects degrade the quality of the image. First, each white space where a dot doesn't print — call them "white dots" — is the same enlarged size. These wider white areas give the image a more grainy, less defined look.

Second, the larger grid gives diagonal lines a stepped looked. The rounded shape of single dots are replaced by what appears to be the straight edge of a black square.

Rectangular Fill Graphics

Fortunately, graphics using rectangles can be created easier using other PCL commands. These define the shape of the rectangle and density and pattern used to fill it. Rectangles can be empty, filled with various degrees of gray or one of six different patterns.

The basic rectangular fill commands are:

<ESC>*c#A Specifies the length of the rectangle in dots.

<ESC>*c#H Specifies the length of the rectangle in decipoints.

<ESC>*c#B Specifies the height of the rectangle in dots.

<ESC>*c#V Specifies the height of the rectangle in decipoints.

<ESC>*c#G The field value represents either the gray scale value (a number from 1 to 100) or pattern (a number from 1 to 6).

<ESC>*c#P Causes the rectangle to be printed. The field value determines the type of fill: 0 for solid black, 1 for a gray scale, or 2 for a pattern.

For example, this program prints a solid 1/4-inch line, five inches long:

```
10 LPRINT CHR$(27);"&a720h3960V";
20 LPRINT CHR$(27);"*c3600h180V";
30 LPRINT CHR$(27);"*c0P";
```

The line starts one inch from the left and 5-1/2 inches down. All measurements are given in decipoints.

Conclusion

This summary of PCL commands and their uses is intended to give you a basis for writing your own printer control programs. Because of the very small increments in cursor movement, and the complex nature of PCL itself, produce test printouts frequently to make sure you're on the right path.

If you have an earlier LaserJet, the toner indicator may seem to move quickly from green to yellow to red. But keep in mind that the indicator only measures drum rotation, not the actual level of toner remaining in the cartridge. If your printouts are of small areas, you'll have plenty of toner remaining even after the indicator reaches the red mark.

QUIKDATA - 13 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.

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 - **$295**
EXP386 - **$395** Much faster 16Mhz 80386 SX version

MEMORY UPGRADES

Note: All memory upgrades come without memory chips. Call for current chip pricing. Memory chip prices are down. Example: 256K 150ns DRAM is $1.95 as of this printing.
Z150MP - $17 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.

Z159 EMS LOGIC UPGRADE includes the PAL chip, logic chips and one bank of 256K (9) chips. Up to two banks (18) can be installed for a total system RAM of 1.2 meg on the Z159 main board.
Z315-1 - $79!!

WINCHESTER UPGRADE KITS

PCW20 - $259 Complete winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes 21 meg formatted half-height Segate ST-225 65ms drive, WX1 controller, cable set, doc.

PCW30	- $329	32 meg with 38ms Segate ST-138.
PCW40	- $369	42 meg with 28ms Segate ST-251-1.
PCW80	- $629	80 meg with Segate ST-4096 full size drive.

We also have the DTC controllers ($59) and daughter board expansions ($69) to place a hard drive in the **H/Z148** computers

BARE WINCHESTER DRIVES

ST-125	- $249	21 meg 37ms autopark 3.5" drive in 5" frame
ST-225	- $219	21 meg 65ms HH 5" drive
ST-138	- $275	32 meg 38ms autopark 3.5" drive in 5" frame
ST-251-1	- $319	42 meg 28ms autopark HH drive
ST-4096	- $575	80 meg 28ms autopark FH drive
ESDI	- $949	**160MB ESDI** drives - several types - call or write

FLOPPY DRIVES

MF501	- $75	5" 360K DS/DD drive
MF504	- $83	96 TPI 1.2 meg AT/Z100 drive
MF353	- $79	720K 3.5" drive in 5" frame
MF355	- $89	1.4 meg 3.5" AT drive in 5" frame
SIEM-R	- $39	40 track SS refurb (H8/H89 type)
TM100-4R	- $75	90 trk DS refurb (H8/H89 type)

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 - $33 (add $2 for Z100 installation for spacers)

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 (CCARD) will handle up to four drives, and the CompatiCard II (CCARD2) will handle up to 2 drives. CCARD4 has boot ROM to allow it to be used as primary boot controller in systems that allow you to remove floppy controller in some systems. Also handles 2.8MB 3.5" floppy drives. Additional cables and external enclosures may be required.
CCARD2 - $79 **CCARD** - $99 **CCARD4** - $125

OUR QD LINE OF COMPUTERS

Excellent quality bare bones PC/XT/AT 0-wait state clone systems (our name is on them - we sell them with pride!) with 101 keyboard and one year warranty. Buy the basic system and choose from our product line to add what you need for a complete system. FCC certified. Pick any system, order other items you want installed from us and we'll install the whole works, test it, and ship it out pronto! No extra charge.

QD88/12 is our high-powered entry level complete starter system with 640K RAM, keyboard, 360K floppy drive, 12Mhz V20 CPU, co-processor socket, clock/calendar, 4 bay cabinet with turbo/reset button and keylock, 150 watt switching power supply, 8 card expansion slot, parallel port, serial port, game port. Add TTL video monitor and go, or add hard drive, additional floppy drives, color video, etc. $475

QD286/12 is a solid 12Mhz 80286 AT compatible work-horse with a 200 watt switching supply, 1 meg RAM expandable to 4 megs on motherboard, 80287 socket, clock/calendar, 5 bay cabinet with turbo/reset button and keylock, 8 card expansion slot. Roll your own! Add controller, ports, video, drives and go. $625

QD386/25 is a fast top of the line 25Mhz 80386 system with the same specs as 286/12 except expandable to 16 megs on main board, 80387 or Weitek socket. $1245

QD386SX 16Mhz 28ms 40MB 1:1 interleave hard drive/controller, 1.2 meg 5", 1.4 meg 3.5", 1MB RAM, 2 serial, 1 parallel, 1 game port, 101 keyboard, baby case (8 slots total). Add video and go! $1495

8-BIT/Z100

We carry a full line of replacement boards, parts and power supplies for the H/Z89/90 and Z100. We also have some H8 boards available. We continue to fully support and carry a full line of hardware and software products for the H8/H89/90 and Z100 computers including almost the full line of Software Toolworks items. Other items we carry include diskettes of all types, printers, modems, hard drives, etc.

OTHER STUFF

Quikdata also carries spike protection filters, backup power supplies, modems, printers, disk drives, drive enclosures, cables and connectors, laptop batteries, video monitors and video cards, memory cards, memory chips and ICs, joysticks, accessory cards, serial and bus mouse, all parts for H/Z89/90 and H/Z100 computers, some H8 parts, accessories, a variety of useful and most popular software and much more!

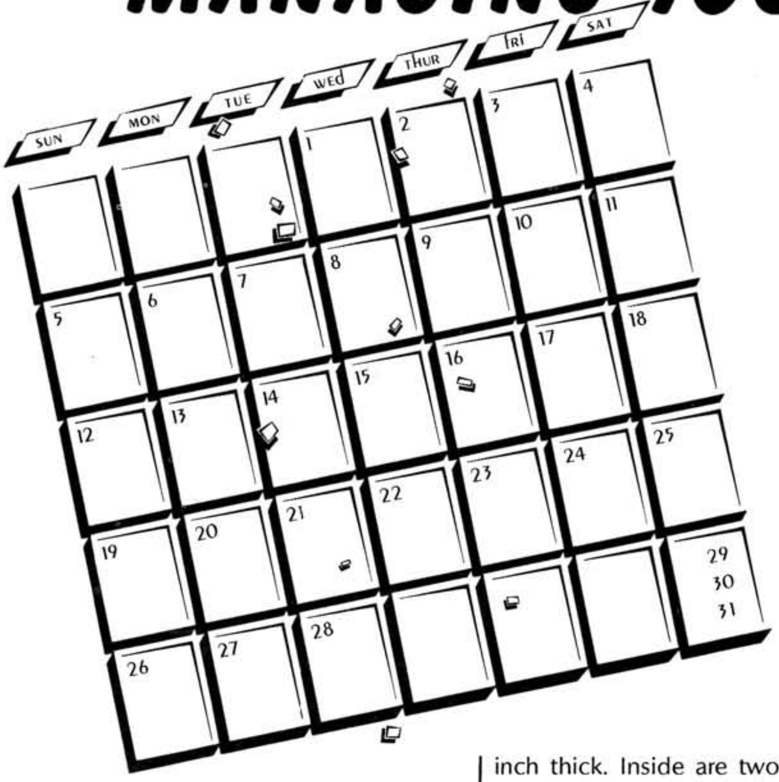
Call or write in to place your order, inquire about any products, or request our **new** free no obligation catalog. VISA and Master Card accepted, pick up 2% S&H. We also ship UPS COD and accept purchase orders to rated firms (add 5% to all items for POs). All orders under $100 add $4 S&H. Phone hours: 9AM-4:30PM Mon-Thu, 9AM-3PM Friday. Visit our bulletin board: (414) 452-4345. FAX: (414) 452-4344.

QUIKDATA, INC.

2618 PENN CIRCLE
SHEBOYGAN, WI 53081-4250
(414) 452-4172

Reader Service #193

MANAGING YOUR SCHEDULE WITH CALENDAR CREATOR PLUS



TOM BING
2755 CAROLYN DRIVE
SMYRNA, GA 30080

Calendar management? It's about as exciting as watching paint dry — tracking your appointments and planning meetings. It's dull, that is, until you neglect it. Then you miss meetings, forget appointments, enrage customers and generally push your blood pressure to unhealthy levels. If you would like to dispense with this kind of excitement in your life, read on. This article describes an inexpensive ($49.95 and less) DOS software package called Calendar Creator Plus (CCPLUS) from Power Up! Software. Basically, all you do is tell CCPLUS what kind of printer you have, what events you need to schedule, and what style of calendar you prefer. CCPLUS will print calendars in several formats on many different brands of printers, using the best print quality available from each. It even includes a "soft font" with several point sizes to get the most impressive type style possible from your printer. Eleven basic calendar styles are available, covering intervals from one year to one day. CCPLUS will consolidate events from many different lists, called "overlays", into one master calendar. Events can be easily added or changed to each overlay, making the old wall calendar, with its mass of hand-scribbled changes in twelve different colors, a thing of the past.

What You Get

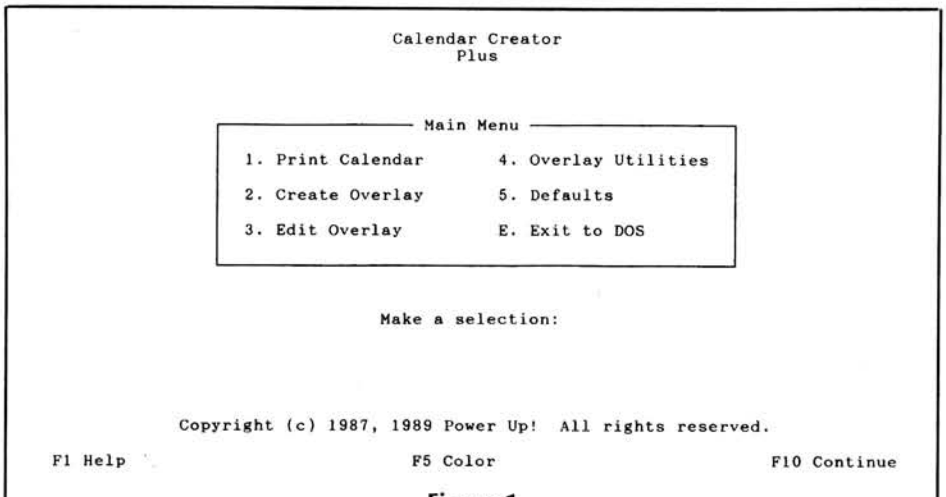
CCPLUS comes shrink-wrapped in a box that's 8 inches by 9-1/4 inches, 7/8-inch thick. Inside are two copies of the program; one on a 5-1/4 inch disk, the other on a 3-1/2 inch; I was really impressed that such an inexpensive program was supplied in two disk formats. There is also a User's Guide, a 51-page document that's neatly typeset with many illustrations. The chapters are organized as follows:

1. Introduction — Covers installation, starting up the program and option selection, and use of on-line help.
2. Events — Covers event lists (overlays) and how to add or edit events in the overlay file. Also explains the difference between "fixed" and "floating" events, which we will discuss later.
3. Hints — Shows efficient ways to do some non-obvious tasks, such as adding a meeting that occurs every other Monday.
4. Printing — Shows pictures of the various calendar output formats and lists the options available with each one. Also shows the options available under each format, as well as describing various supported printers. The calendar examples in this article were produced by a LaserJet Series II; the 300 dot-per-inch resolution and snappy type font really show what CCPLUS can do. I was particularly impressed by the small-size sheet options, which will go right in a Day-Timer™ or Lefax™ binder.
5. Defaults — Covers setting of default date formats, such as MM/DD/YY, and time formats, such as 2:30 PM. Alternative formats such as YY/MM/DD and 14:30 are available.

The Appendix following Chapter 5 describes fixes for common user problems and tells how to convert overlay files from earlier Calendar Creator versions to the current Version 3.0. The appendix also lists new features in Version 3.0 and tells how to use CCPLUS on a LAN. Finally, a very detailed index follows the Appendix.

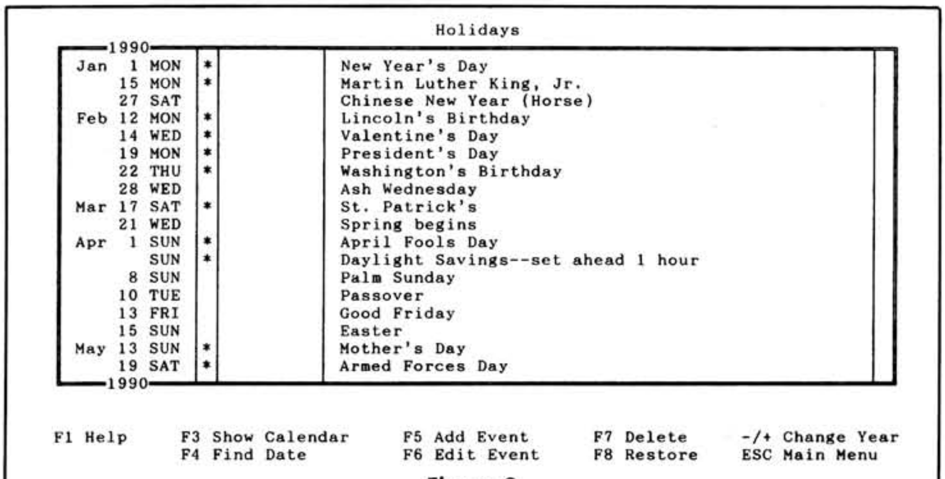
How It Works

Figure 1 shows the CCPLUS main menu. It's a good idea to examine the overlays provided with CCPLUS first, using the Edit Overlay option. Figure 2 shows part of the Holidays overlay which comes with CCPLUS. CCPLUS also includes a Sample Company Events overlay, showing how the program is used in a business setting. As the User's Guide says, the overlays may be thought of as transparent sheets lying on top of your calendar. Each one has a list of events which form some kind of natural group, such as all company holidays, all meetings scheduled for an entire office, or all out-of-office time (vacations, trips, etc.) for a single employee. When it's time to print a calendar, the user simply selects the desired overlays from a list, and they are merged as part of the printing process. If I'm in a work group of 5 or 6 people and I want to do a monthly schedule for my group, I would select a Company Events overlay, plus maybe an "out of office" overlay for each employee. Figure 4 shows a monthly calendar printed using multiple overlays.



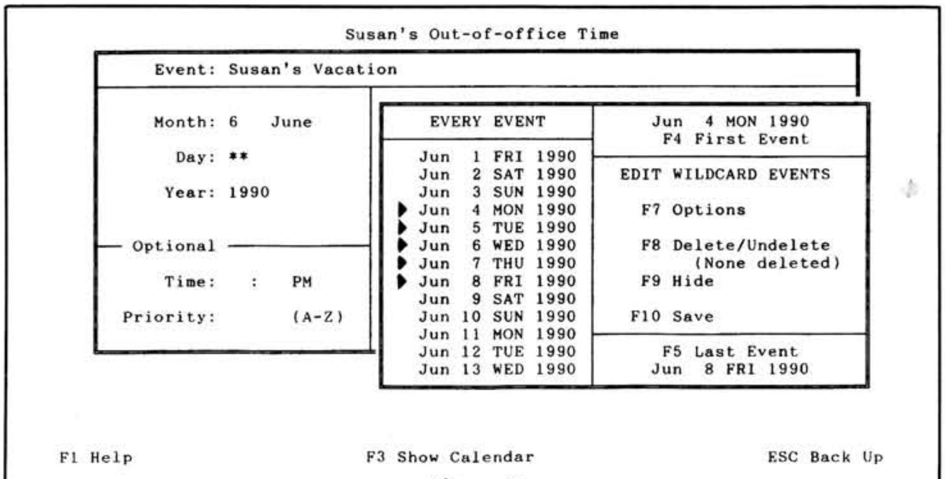
```
                         Calendar Creator
                               Plus

             ─────────── Main Menu ───────────
              1. Print Calendar       4. Overlay Utilities

              2. Create Overlay       5. Defaults

              3. Edit Overlay         E. Exit to DOS


                        Make a selection:


     Copyright (c) 1987, 1989 Power Up!  All rights reserved.

F1 Help                      F5 Color                      F10 Continue
```

Figure 1
Calendar Creator Plus Main Menu



```
                              Holidays
 ──1990──
 Jan  1 MON | * |    New Year's Day
     15 MON | * |    Martin Luther King, Jr.
     27 SAT |   |    Chinese New Year (Horse)
 Feb 12 MON | * |    Lincoln's Birthday
     14 WED | * |    Valentine's Day
     19 MON | * |    President's Day
     22 THU | * |    Washington's Birthday
     28 WED |   |    Ash Wednesday
 Mar 17 SAT | * |    St. Patrick's
     21 WED |   |    Spring begins
 Apr  1 SUN | * |    April Fools Day
        SUN | * |    Daylight Savings--set ahead 1 hour
      8 SUN |   |    Palm Sunday
     10 TUE |   |    Passover
     13 FRI |   |    Good Friday
     15 SUN |   |    Easter
 May 13 SUN | * |    Mother's Day
     19 SAT | * |    Armed Forces Day
 ──1990──

F1 Help   F3 Show Calendar   F5 Add Event    F7 Delete    -/+ Change Year
          F4 Find Date       F6 Edit Event   F8 Restore   ESC Main Menu
```

Figure 2
"Edit Overlay" Option; "Holidays" Overlay Selected



```
                     Susan's Out-of-office Time
     Event: Susan's Vacation

     Month: 6    June       EVERY EVENT        Jun  4 MON 1990
                                                F4 First Event
       Day: **              Jun  1 FRI 1990
                            Jun  2 SAT 1990    EDIT WILDCARD EVENTS
      Year: 1990            Jun  3 SUN 1990
                          ▶ Jun  4 MON 1990     F7 Options
                          ▶ Jun  5 TUE 1990
  ── Optional ──          ▶ Jun  6 WED 1990     F8 Delete/Undelete
                          ▶ Jun  7 THU 1990         (None deleted)
      Time:   :   PM      ▶ Jun  8 FRI 1990     F9 Hide
                            Jun  9 SAT 1990
  Priority:      (A-Z)      Jun 10 SUN 1990    F10 Save
                            Jun 11 MON 1990
                            Jun 12 TUE 1990      F5 Last Event
                            Jun 13 WED 1990    Jun  8 FRI 1990

F1 Help                 F3 Show Calendar                ESC Back Up
```

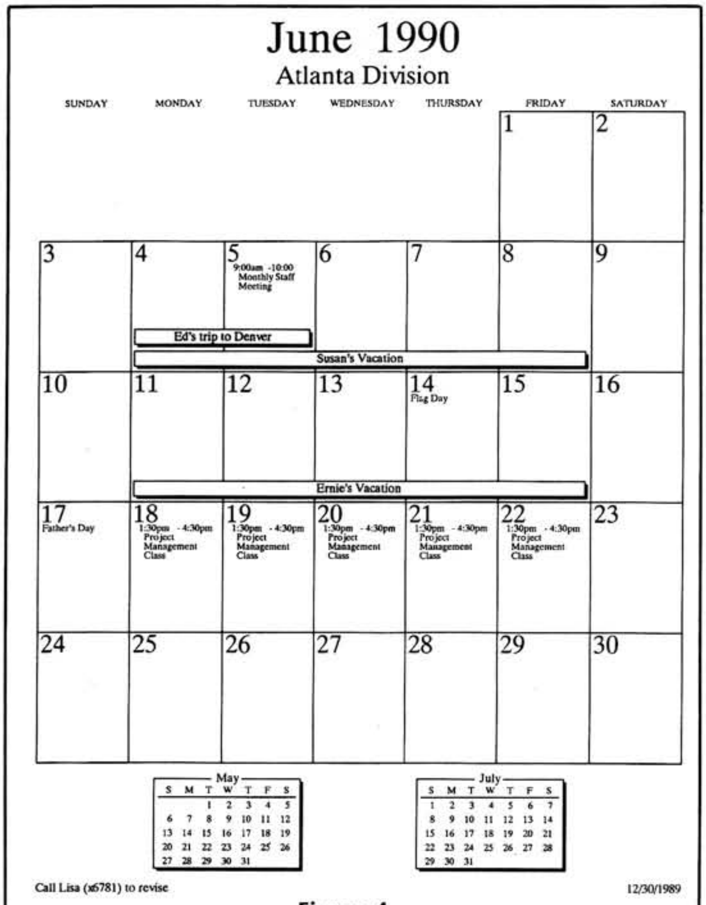
Figure 3
Using the "Edit Wildcard" Feature on the "Add Event" Screen

If a secretary prints a calendar like this and sees a conflict, then one or more of the overlays can be changed and the people involved can be notified; the calendar is then reprinted using the corrected overlays. The power of overlays is that they can be maintained separately and brought together only when you need to print a calendar. Also, if a particular calendar is concerned only with personal events, for instance, you don't need to drag in a company events overlay, or worry that the company events list is not up-to-date. Each person can maintain his or her own overlay, and they only need to be brought together at print time. Overlays can easily be combined permanently using the "Copy/Merge" selection on the Overlay Utilities menu.

When a user creates an overlay, he specifies a name of up to 28 characters. CCPLUS translates this to a DOS file name. For instance, an overlay called "Joe's Out-of-office Time" became JOE_S_OU.CAL on my hard disk. Even if two overlays are created with *identical* titles, CCPLUS will give them unique DOS file names. If the overlays are copied from several different PCs to a secretary's machine using the "Copy/Merge" option of Overlay Utilities, CCPLUS will rename files as necessary during the copy to make the names unique. Users should agree on standards for naming overlays; otherwise, a secretary could wind up with six overlays numbered 1-6 on the menu screens, all having the name "My Vacation". The overlay files would all be different, but a user would not be able to tell which was which from a menu screen. If distinct overlay names are used, and the CCPLUS Copy/Merge function is always used to copy overlays (instead of DOS COPY), managing overlays will not be a problem.

Another valuable "real-world" feature of CCPLUS is that it uses two types of events: fixed and floating. A fixed event is one that falls on a specific *date*, such as the 12th of December. Floating events always fall on the same *day of the week*, such as the second Monday of each month. A repeating event of either type may be indicated by placing a '*' in either the day, month, or year field when the event is added to the overlay. For instance, if a wedding anniversary occurs on the 7th of June, the user would enter the month as 6, the date as 7, and the year as '*' on the "Add Fixed Event" screen (it's the same date every year).

When adding fixed or floating events that repeat, it's best to use wildcards (asterisks) so freely that *too many* calendar entries result, and then prune them down using the Edit Wildcard feature (Function Key F7) of the Add Event screen. For instance, if an employee has a week's vacation in June (see Figure 4), I would add it as a fixed event, entering '6' for the month, '*' for the date, and '1990' for the year. Then, by pressing F7, I would specify June 4 as the start date (using F4) and June 8 as the end date (using F5). Figure 3 shows the "Add Event" screen for this procedure with the Edit Wildcard feature in use. When this change is saved by pressing F10, the number of instances is automatically reduced from 30 to the desired 5. This same basic technique works for repeated floating events, such as every other Monday. In this case, the "Add Floating Event" screen would be used, a



June 1990

Atlanta Division

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
					1	2
3	4 Ed's trip to Denver Susan's Vacation	5 9:00am -10:00 Monthly Staff Meeting	6	7	8	9
10	11 Ernie's Vacation	12	13	14 Flag Day	15	16
17 Father's Day	18 1:30pm -4:30pm Project Management Class	19 1:30pm -4:30pm Project Management Class	20 1:30pm -4:30pm Project Management Class	21 1:30pm -4:30pm Project Management Class	22 1:30pm -4:30pm Project Management Class	23
24	25	26	27	28	29	30

May

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

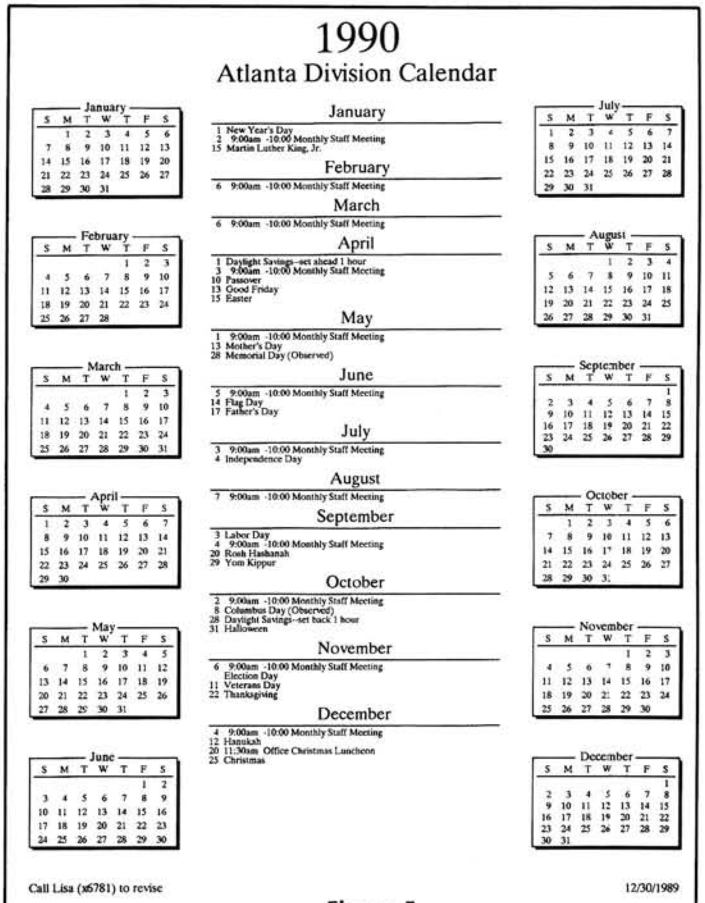
July

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Call Lisa (x6781) to revise

12/30/1989

Figure 4
Monthly Calendar Output



1990

Atlanta Division Calendar

January

1 New Year's Day
2 9:00am -10:00 Monthly Staff Meeting
15 Martin Luther King, Jr.

February

6 9:00am -10:00 Monthly Staff Meeting

March

6 9:00am -10:00 Monthly Staff Meeting

April

1 Daylight Savings--set ahead 1 hour
3 9:00am -10:00 Monthly Staff Meeting
10 Passover
13 Good Friday
15 Easter

May

1 9:00am -10:00 Monthly Staff Meeting
13 Mother's Day
28 Memorial Day (Observed)

June

5 9:00am -10:00 Monthly Staff Meeting
14 Flag Day
17 Father's Day

July

3 9:00am -10:00 Monthly Staff Meeting
4 Independence Day

August

7 9:00am -10:00 Monthly Staff Meeting

September

3 Labor Day
4 9:00am -10:00 Monthly Staff Meeting
20 Rosh Hashanah
29 Yom Kippur

October

2 9:00am -10:00 Monthly Staff Meeting
8 Columbus Day (Observed)
28 Daylight Savings--set back 1 hour
31 Halloween

November

6 9:00am -10:00 Monthly Staff Meeting
Election Day
11 Veterans Day
22 Thanksgiving

December

4 9:00am -10:00 Monthly Staff Meeting
12 Hanukah
20 11:30am Office Christmas Luncheon
25 Christmas

January

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

February

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

March

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

August

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

September

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

October

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

November

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

December

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Call Lisa (x6781) to revise

12/30/1989

Figure 5
Annual Calendar Printout

'*' would be placed in the Number field, 'M' would be entered as the day of the week, and a '*' would be entered for month and (possibly) year. Then, by pressing F7 (Edit Wildcard), the user would select an appropriate start date using F4 and then select every 2nd (every other) event with the arrow and Enter keys.

Repeating events like this, either fixed or floating, are handled as one event. When they are deleted from the "Edit Overlay" screen, it's an all-or-nothing proposition. However, individual dates within a repeating event can be deleted using the "Edit Wildcard" option on the "Edit Event" screens. Events with wildcards also take up a lot less space in the .CAL files than, say, five single fixed events to represent a vacation.

When the number of an event is important, such as an anniversary or birthday, CCPLUS also automatically takes care of that. If I create an event called "Mark's ##### birthday", and then specify the day he was born as the first date on the "Edit Wildcard" screen, then all calendar references to his birthday will be correctly numbered, with "23rd", "24th", etc., in place of the pound signs. Numbered events like this can go up to 999.

Ease Of Use

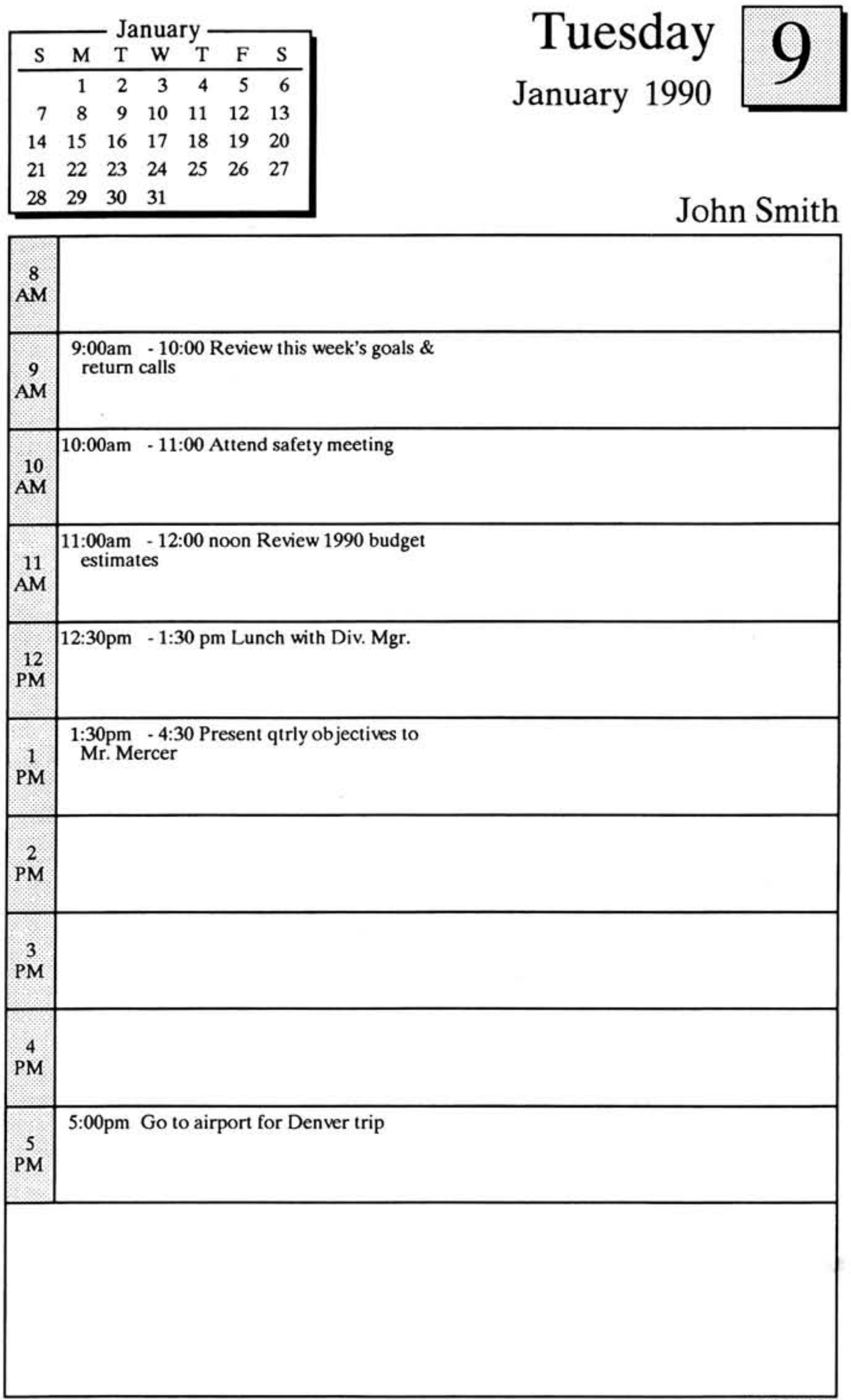
The term "user friendly" has been beaten to death, but I think CCPLUS deserves it. On the main menu, the selections are digits 1 through 5 or E (Exit). Each is activated only after the Enter key is pressed, giving you time to change your mind and switch options. Help is available from any screen by pressing F1, and it's context-sensitive. You can either get page-by-page help by using the PgUp and PgDn keys, or press the Home key for a help index with numbered topics. When the index appears, you can select the numbered topic of interest. The User's Guide gets high marks for clarity and good organization; also, it includes product support telephone numbers.

The use of the ten function keys is reasonably consistent, given the number of functions in this package. For instance, pressing F3 always shows the current calendar month on-screen, which can be quickly "flipped" to past and future months using PgUp-PgDn and "+/-" keys. F10 is always "save" or "continue". The arrow, Enter, and Tab keys all work in a fairly intuitive way. The ESC key always backs you up to the previous screen. When ESC is pressed, current data will either be saved or the user will be given a choice to either confirm or discard the last change. The active or current user entry field on a screen is always highlighted, and some common-sense validation is provided. For instance, if a user types "A" in a month number field, CCPLUS will simply not accept it or display it.

Requests to delete events or overlays will produce an "Are you sure?" prompt. Users can even restore the most recently deleted event by pressing F7 at the "Edit Overlay" screen. The user always sees a numbered list of overlays, with full titles (not disk file names) listed in alphabetical order; CCPLUS takes care of naming and saving DOS files all by itself.

Print Options

From the "SELECT OUTPUT STYLE" menu under "Print Overlay", the user can select one of eleven calendar styles covering from one day to one year, and three page sizes (plus multi-page). The calendars can even be printed or written to disk as straight ASCII text files or comma-delimited files for import to spreadsheets. The "Change Printer Defaults" screen, accessed from option 5 of the main menu, lists 15 models of supported printers. The User's Guide lists scores of others on page 42 as emulating either Epson X series or LQ series, so these can all be used also. CCPLUS supports both serial and parallel printers. As Figures 4, 5, and 6 show, the combination of 300 dpi resolution and the CCPLUS soft font produce stunning results on an HP Laserjet; the HP Deskjet



January

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Tuesday 9

January 1990

John Smith

8 AM	
9 AM	9:00am - 10:00 Review this week's goals & return calls
10 AM	10:00am - 11:00 Attend safety meeting
11 AM	11:00am - 12:00 noon Review 1990 budget estimates
12 PM	12:30pm - 1:30 pm Lunch with Div. Mgr.
1 PM	1:30pm - 4:30 Present qtrly objectives to Mr. Mercer
2 PM	
3 PM	
4 PM	
5 PM	5:00pm Go to airport for Denver trip

12/30/1989

Figure 6
Personal Organizer Printout

and several 9-pin and 24-pin printers are also supported. You can still get the shaded boxes on printouts as long as you have at least 24-pin resolution. Notice that some repeated events appear in a continuous bar across the bottom of the date box; this style is the default if the events have no clock time. Otherwise, they appear in time order at the top of the date boxes, repeating for each day (Figure 4). Incidentally, each calendar page in Figures 4 through 6 was printed on a Laserjet Series II in 30 to 40 seconds. The CCPLUS soft font produced the lettering, so you don't need any Laserjet font cartridges. The soft font appears to be downloaded each time, presumably to guard against some other user changing the contents of the Laserjet memory. This can happen in a printer-sharing situation. The screen dumps in Figures 1 through 3 were printed on an Epson model LQ-510 printer, courtesy of the Jade Computer store in Smyrna.

I am especially impressed by the personal organizer size outputs (Figure 6). These allow people who rely on their Day-Timer or other organizer to benefit from computer-coordinated schedules.

Besides the many variations discussed above, each output style has its own set of options, such as sideways printing for most of the styles, and user-entered footnotes. Another handy option allows you to include or omit the small monthly calendars for previous and later months at the bottom of 1-month calendars. The starting date for a calendar can be varied, so that a school calendar can start in September, for instance.

Event Priority

The priority of an event may be either left blank, or entered as either A-Z or "$". On calendars, non-timed events (i.e., those without a clock time) will print in order of priority, no matter which overlay they belong to. Events with blank priority will appear in random order after all events with A-Z priority. Timed events will always print in time order. If the user changes the print defaults, non-timed events can appear ahead of timed events. Events with "$" priority are hidden events; they appear in the overlays, but not on any calendars. This allows you to keep an event on file when the date is uncertain and you are not yet ready to commit to it on a calendar.

CCPLUS Pros And Cons

I can't really think of anything but "pros". Once while preparing the illustrations for this article, the printer cable worked loose from the Laserjet; an on-screen warning of printer problems would have saved a lot of time in this situation. However, it's not a trivial task to write a diagnostic program that can sense problems in a wide variety of parallel and serial printers. Some problems, such as an intermittent open in a wire in the printer cable, are hard to pinpoint using software.

There may be large numbers of users managing their calendars with Lotus Metro, Borland Sidekick, or some other widely used program. If so, it might be nice to have a utility program to convert their existing calendar entries to CCPLUS. However, I doubt if any such calendar program now has a large enough following to justify writing a conversion program. To. put it another way, converting from manual calendars or from some other program to CCPLUS is probably well worth the effort, even if you have to re-enter all the schedule information by hand.

Conclusion

Is CCPLUS better than keeping up your calendar by hand? Yes, definitely. Is it worth the money? Again, an emphatic yes. It is shown for $49.95 in the Power-Up! catalog and is described as their best

Continued on Page 46

Holy Heathkit Batman

George Elwood
1670 N. Laddie Court
Beavercreek, OH 45432

5.5 Meg Floppies for the Z-100

The Z-100 is not dead. It is alive and doing OK. There are still some of us in HUGLAND that continue to use the Z-100. I have an H-386 at home along with my Z-100. At work I use a Z-248, Sun 3/60 and other types of computers. The one that has the best keyboard is the Z-100 and I still use it. Over 40,000 Z-100s were sold to the Air Force and they continue to compute on. In the December 11, 1989 Government Computer News in an article called "CLASSICS", the Z-120 is detailed. The headline in 2.5 picas or 30 point states "Air Force Finds Zenith Z-120 Elegant, Incompatible."

Having started with an H-89 (beep, beep) with a whole 48K of memory and a tape drive, I soon discovered the joy of a floppy over the tape. Those first floppies where hard-sectored and held a whole 90K per side. By using a paper punch it was possible to use both sides of the disk. My H-89 ended up with four drives in the end, two hard-sector and two soft-sector.

I traded the H-89 for my Z-100. With two 360K floppy drives and 192K of memory, I was set for at least a year or two. As requirements increased I added the 2 Meg. UCI RAM card, the UCI 8 MHz speedup kit, and the NEC V-20 processor. This helped out, but additional storage was needed. Enough of the history. I have already described in an earlier article how I added a 1.2 Meg. 5.25 inch drive to my Z-100 system. With my 2 Meg. of added RAM and four floppy drives it meets my needs, although you never have enough disk space. I would like to add a hard drive, but then that gets filled up too quickly. Another media was needed.

The Z-100 was the last of the Heath computers. The engineers designed a lot of growth into this device making it very hi-tech. The Heath engineers also designed the disk controller to work with both 360K and 760K 5.25" floppies by simply changing one jumper on the board. These, along with the included eight inch disk drive controller, provide a lot of removable storage at reasonable rates.

Last year I noticed a 5.5 Meg. floppy drive being offered by a clearance house. Because of other commitments, I was unable to purchase one and try to make it work. John Pierce of the Dayton Heath/Zenith Users Group and I talked about this and he decided to spring for a drive to see if it was possible to use on the Z-100.

He used the jumped cable I designed for the 1.2 Meg. drive as a starting point and modified it slightly for the Kodak 5 Meg. floppy. Because of the way it was configured, John found a way to daisy chain the 5 Meg. drives at the end of the 1.2 Meg. drive system I developed. Using this system, it now was possible to have six (actually eight but that follows) floppy drives on the Z-100 using the standard controller.

These six (eight) drives could hold up to:

Number	Drive	Total	
Drives	Size	Capacity	
2	360K	720K	5.25
2	1200K	2400K	5.25
2	5500K	11000K	5.25
2	720K	1440K	3.50

That's over fifteen and a half Meg. of removable storage media. The 360K and 1.2 Meg. floppies and drives are not expensive. The 5.5 Meg. Kodak drives are another story, but are not really that expensive either when compared to the capacity.

Kodak developed these floppy drives and marketed them under their Verbatim name. The drives look just like any other half-height 360K/1.2 Meg. floppy drive. The difference comes in the way they are used and the total number of tracks on a disk. Because the track density is so high, Verbatim preformats the disks. The disk has 384 tracks per inch and Verbatim inserts servo data between each block of data to keep the heads aligned correctly. This writes very small accurately placed tracks on the disk. You can only write using these disks in this drive. It is not possible to low-level format the disk. The low-level format writes the sector patterns. You must format the disk just like a hard drive or any other floppy. Figure 1 shows the front of the Kodak 5.5 Meg. drive and one of the 1.2 Meg. floppy drives. Note that the Kodak drive has a different looking front. When you insert a disk, instead of turning the locking lever you must push a lever down. Figure 2 shows a side view of the Kodak drive. Note that the drive is almost completely enclosed in a metal shield. Also note the two servos.

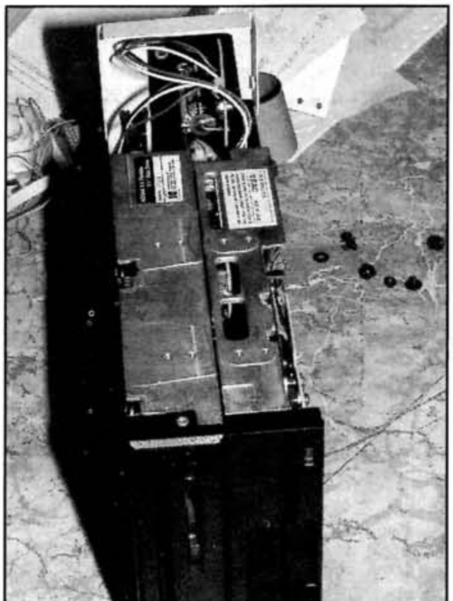


Figure 1
Kodak 5.5Meg Floppy Drive

John found that by modifying the operating system he could add a device driver to the system so that the Z-100 could use the Kodak drive. The drive is daisy chained off the eight inch drive on the Z-100 disk controller.

The first thing that is needed is the interfacing cable. I purchased the 50-pin and 34-connector IDSc at a local surplus dealer. I also purchased a short amount of

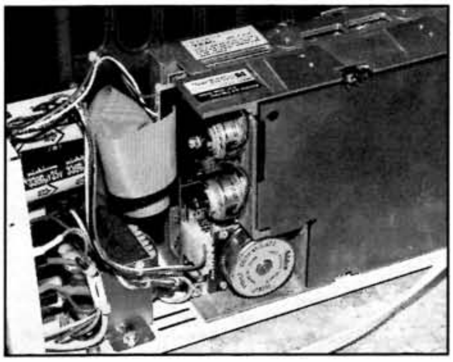


Figure 2
Kodak 5.5Meg Floppy, Side View

34-connector flat wire. The IDC connectors make it easy to build the required interface cable. I use a small vice to provide the even pressure along the entire connector that is desired. See Figure 2. I was building the cable for one 1.2 Meg. drive and one 5.5 Meg. drive although the same procedures can be used to make a cable to support four drives. There are at least four Z-100s in the Dayton Heath/ Zenith Users Group with this configuration.

A pair of wires should run between the two connectors. That is, the signal line between 18 and 4 should also have a ground between 17 and 3. Note that pins 9-10 and 21-22 are grounded on the 50-pin connector. This connector can be used with the 1.2 Meg. drive I discussed in an earlier article. The connections shown here assume that the drive you have has an ML jumper. If not, you will require a line between pin 18 on the 50-line connector and pin 16 on the 34-line connector. John covers these points in the documentation he provides.

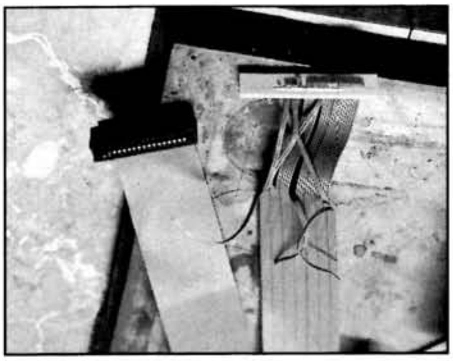


Figure 3
Drive Cable Assembly

It is recommended that you install the drives in an external drive enclosure. I added the new drive to replace one of the 1.2 Meg. drives that failed in my old system. I had the power in the case so there was no problem. The Kodak drives require more power than a normal floppy drive. They require 3.1A @ 12V and 1.4A @ 5V so mounting them inside the Z-100 may cause power problems.

After installing the drive in the case and attaching the interface cable and power, you are ready to start up the system. You must install the device driver in the system and make the required changes to CONFIG.SYS. After rebooting the system you are now ready to use the drive. You use the drive just like any other floppy. John provides the software driver and a special high-level formatter. The software driver places the Kodak drives after any other drives in the system. If you have a RAM disk, the Kodak drive will follow it in addressing.

You can purchase the floppies from John. The disk cost $9 for two disks or $20 for five. This sounds expensive, but then they do hold 5.5 Meg. and they are formatted. I can remember back in the H-89 days when a pack of 10 disk cost $30 and they held a big 90K each. John bought many of the available drives and is selling them for $225 including shipping in the U.S. Along with the drive is the driver and nine pages of installation instructions.

```
K>dir i:

 Volume in drive I is pierce-test
Directory of I:\

EKDRIVE  JCP     4693    12-12-89     9:27a
EKFORMAT DOC     1685     4-14-89     6:29p
EKFORMAT COM     2028     5-03-89     5:30p
EKVERIFY COM      286     5-03-89     5:01p
EKDRIVE  DOC     2501     2-12-90     3:33p
EKVERIFY DOC      693     4-14-89     6:30p
KODAK-G  NOT     2266     2-13-90     9:15a
KODAK-H  NOT     2772     2-13-90    10:22a
KODAK-S  NOT     6596     2-13-90    10:21a
INFO                 3747    12-09-89    12:06p

     10 File(s)      5513216 bytes free
```

Figure 4
Sample Directory Listing

```
K>chkdsk i:

Volume pierce-test created Mar 12, 1989 3:53p

 5552128 bytes total disk space
       0 bytes in hidden files
   38912 bytes in 10 files
 5513216 bytes available on disk
```

Figure 5
Check Disk Listing

The 5.5 Meg. drive can also be used in PC computers. If used in an XT, they conflict with COM port 2 or the hard drive controller. They work great in AT class machines. One of these drives was installed in a Z-248 without a problem. Kodak includes a disk controller with the drive. This is not necessary in the Z-100, but required in PCs. In addition, when using the Kodak controller, the drive will READ other formatted disks, but will not write them. It will read 360K, 720K, and 1.2 Meg. formats.

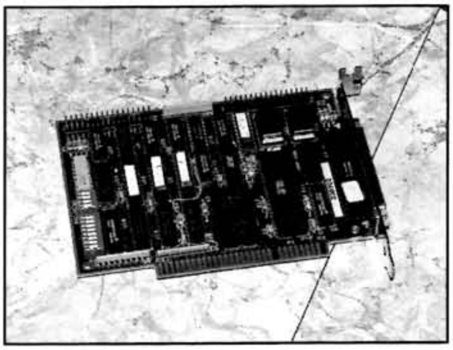


Figure 6
Kodak AT Interface Board

John continues to work with the Z-100 and has now added two 720K 3.5 floppies to his system. This required another minor change to the operating system. That's pretty good for a system that was not PC compatible. His system has eight floppy drives and two 64 Meg. hard drives. He runs the 3.5 inch drives off the 360K controller. The Heath engineers made it possible to do this. John modified the BIOS sightly to handle these drives. He did write a small program that permitted him to select either the 5.25" 360K drive or the 720K 3.5" drive.

The drives can be purchased from John Pierce. He can be reached at 3980 State Route 235, Fairborn, OH 45324-9728, Day Phone (513) 252-2402, Nite (513) 879-1332.

Implementing Postscript

Alan Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

In a previous article, I described the benefits of Postscript printing and the Postscript language. In this article, let's look at how to implement Postscript, even if you don't have a Postscript printer.

Postscript Printers

The most obvious way to print with Postscript is by using a Postscript equipped printer. The most popular Postscript printers are marketed by Apple Corporation, but a number of companies sell Postscript printers or those with clones of the official Postscript interpreter.

Two standard sets of fonts are usually supplied with Postscript. Printers compatible with the original Apple LaserWriter have 13 fonts: a symbol set of Greek characters, and Courier, Times, and Helvetica in normal, bold, italic, and bold italic. LaserWriter Plus and compatible printers have 35 fonts, those 13 plus Zapf Chancery Medium Italic, Zapf Dingbats, and four fonts each of Palatine, ITC Avant Garde, ITC Bookman, Helvetica Narrow, and New Century Schoolbook.

The fonts are supplied as outlines built into the printer itself. So you don't have to waste valuable disk space storing numerous font files.

There are two disadvantages of Postscript printers however — cost and speed. Postscript printers are generally more expensive than HP LaserJet models, the main competition to Postscript in the laser printer market. With the LaserJet IIP going for less than $1000 on the street, you really have to need the benefits of Postscript to justify spending four times the cost or more.

Postscript printers are also notoriously slow. Remember that all characters have to be created at print-time by rasterizing the font outlines. This process takes time.

There is one other possible disadvantage to Postscript, depending on your orientation. As long as you're using an application that has a Postscript driver, you'll do fine. But there are programs that don't communicate with Postscript, especially in the DOS world. All of the Epson or LaserJet drivers, the most popular in DOS applications, won't do you any good at all.

However, there are other ways to get the benefits of Postscript without having a Postscript printer.

Hardware Alternatives

A less expensive way to get the benefits of Postscript printing and still be compatible in the DOS world, is by equipping an HP LaserJet printer with Postscript.

A number of add-on boards are available that provide the LaserJet with the Postscript interpreter and Postscript outline fonts.

In most cases, the cost of the LaserJet and the add-on system is less than a comparably equipped Postscript printer. For example, you can purchase a LaserJet II for about $1600. Add to that a $2000 Postscript board and you're saving at least $400 from some of the less expensive Postscript printers.

But there are other benefits in addition to saving money. A Postscript equipped LaserJet is still a LaserJet. Not only do you have Postscript printing available, but a LaserJet as well. So if your DOS application doesn't have a Postscript driver, you can use the printer as a standard LaserJet — like having two printers in one.

In addition, many of the Postscript boards outperform Apple LaserWriters. One benchmark (PC Magazine, April 11, 1989) shows that six Postscript-equipped LaserJets printed text and graphics faster than an Apple LaserWriter IINT.

The boards, which contain the Postscript interpreter and fonts, are installed either in your LaserJet or in your computer. As far as your application is concerned, you have a Postscript printer, using the original Postscript driver supplied by your application program.

These systems work by bypassing the LaserJet's internal hardware that converts text and graphics into bitmaps. The rasterizing of text and graphics is performed on the add-on card, then transmitted directly to the LaserJet's video circuits.

Let's briefly look at two such add-on systems. Other vendors of Postscript cards are listed at the end of this article.

PS-388 Accelerator

The PS-388 Accelerator, from Princeton Publishing Labs, contains the Postscript interpreter, 35 resident outline fonts, and three megabytes of memory for processing fonts and graphics.

It consists of two circuit boards. One installs in a slot in your computer and is configured as parallel port LPT2. (This way you keep LPT1 configured to access your printer as a standard LaserJet.) The second card installs into the LaserJet's interface slot, with a special cable connecting the two cards. To use the card, you have to install several programs and a batch file on your hard disk.

Set up your applications for Postscript using LPT2, then print documents as you would normally. As far as your program is concerned, you have an Apple LaserWriter.

If you have an application that doesn't support Postscript, you can bypass the PS-388 Accelerator and use your printer as a standard LaserJet.

PS Jet

The earlier model LaserJet and LaserJet Plus do not have an interface slot. So PS Jet, from Laser Connection, takes another approach.

PS Jet actually replaces the entire LaserJet controller and interface connectors. It takes about an hour, but you have to install an entire new top section on your printer that contains the Postscript controller, fonts, and a new control panel.

The printer is connected to your computer's serial port, which must be set at 9600 baud.

For all practical purposes you now have a Postscript printer, 100% compatible with Adobe Postscript, complete with an interactive mode. While some applications may include a separate PS Jet driver, you can use those for the Apple LaserWriter or generic Postscript. Just select one of the Postscript drivers, then create and print documents.

For applications that do not support Postscript, PS Jet can be configured to emulate the LaserJet Plus, HP 7475A plotter, or Diablo 630 daisy-wheel printer.

When used as a LaserJet, up to 64 fonts can be resident and printed per page, and you can access the PS Jet's internal Courier, Times, and Helvetica outline fonts.

The 7475A plotter emulation lets you print Lotus, CAD and presentation program documents containing HP Graphic

Language commands, while the Diablo emulation can be used for printing text files from older applications having limited driver support.

PostScript On a Cartridge

A much simpler approach to hardware Postscript is by installing a font cartridge that contains the Postscript interpreter and fonts.

Pacific Data Products and Hewlett-Packard, among others, sell cartridges that provide Postscript capabilities by just being slipped into one of the cartridge slots in the printer.

There are definitely some advantages to this method. To begin with, the cartridges sell for about $1000, half of the cost of most add-on circuit card systems. In addition, you don't have to install any circuit boards in your computer, or make any permanent changes to your printer. Just slip the cartridge in and out when needed.

The Hewlett-Packard cartridge is designed for the LaserJet IID, giving you both portrait and landscape orientations of all fonts and two-sided printing.

Implementing Postscript Without a Laser Printer

Hardware Postscript solutions still require a laser printer, and still cost at least $1000 to implement.

By selecting a software solution, you cut your costs further and get Postscript even on dot matrix printers.

Instead of having the Postscript interpreter built into the hardware, software interpreters translate Postscript documents into your printer's native language. They not only work with HP LaserJet and other popular laser printers, but they provide Postscript capability to popular dot matrix and ink jet printers as well.

Just like "real" Postscript, these interpreters use outline fonts to scale type "on the fly." The fonts are name and width compatible to their Postscript counterparts, so you use your application's standard Postscript driver — even though you have an EPSON or IBM Graphics printer, for example.

For instance, say you have WordPerfect that supports Postscript fonts in almost any size, and in outline and shadow styles. You configure WordPerfect as if you have an Apple LaserWriter, create your document, but print your text and graphics on your dot matrix or ink jet printer.

Depending on your system, printing Postscript with software interpreters can be a one- or two-step process.

Unless you have extended or expanded memory, most software systems require two steps. You first set your application for Postscript, create the document, and save the output to a disk file rather than to the printer. Then you exit the application and run the interpreter. The interpreter reads the Postscript file, scales the fonts, and prints both text and graphics.

It's not an ideal way to implement Postscript if it is your primary method of printing. But it is fine for occasional jobs, or to see if Postscript is really for you.

If you have extended or expanded memory, and an AT class computer, many of the interpreters let you print directly from the application.

Let's look at two software Postscript solutions.

UltraScript PC

UltraScript PC, from QMS, comes in two versions: a $195 package with 25 scaleable outline fonts, and UltraScript PC Plus, a $445 version with 47 fonts. Both versions require a hard disk drive with at least 4 Mb of space, a high-density floppy, and an AT-class machine or above.

QMS has been in the Postscript business for some time and was the first to offer a true Abobe Postscript add-on card for the LaserJet. (Laser Connection, which markets the PS Jet is a QMS company.) The UltraScript PC interpreters also provide officially licensed Adobe Postscript fonts and 100% compatibility with hardware Postscript.

The program supports the HP LaserJet, as well as the HP DeskJet and PaintJet; Canon LBP-8II, and Bubble Jet BJ-130; Epson FX, LX, and LQ printers; the NEC 5200 series; and the IBM Graphics and Proprinter X24 printers. Any printers that emulate those listed can also be used. Version 2 of UltraScript can also print documents in full color from applications that support color PostScript printers.

Laser printers must have at least 1 MB of memory to print a full page at 300 dots per inch. If you have less memory, you can still use the program but at lower resolutions, or for less than a full page of graphics at high resolution.

I've used the program successfully on an older LaserJet Plus without extra memory. I just couldn't print a full page at 300 dpi resolution.

UltraScript can work as either a one-step or two-step system. In either case, you first load the interpreter, then a special program called PCAPTURE which is assigned to its own LPT port, such as LPT 2 or LPT 3. You set up your application for a Postscript printer and assign it the same port used by PCAPTURE. When you print to that port, PCAPTURE saves the printer output in a spool file.

If you have at least 800k of extended memory, UltraScript prints the file while you remain in your application, swapping out memory if needed. So it is just like printing directly from the application.

Without extended memory, you have to exit the application, then print the file using one of several printing programs provided.

Since PCAPTURE is assigned its own LPT port, you can easily bypass UltraScript and print in your printer's native mode. Just change the application to output to LPT 1 and use the printer's normal driver.

While you can print from within your application or as a two-step process from the DOS command line, UltraScript includes its own control program for selecting printer options and printing files. I had a little trouble running the program on my Zenith Data Systems 386/16. DOS reported trouble finding drives E and F. I kept pressing I to ignore the situation and the program finally ran.

UltraScript lets you batch print multiple files, and set the resolution and page size. It also has an interactive mode in which you can enter Postscript commands directly to the interpreter. This is an excellent way to experiment with the Postscript language and to print special effects that your application cannot handle.

One nice extra feature of UltraScript is a program that converts captured Postscript programs to TIFF or PCX graphics files. You can then load the file into your application or edit it with a drawing program.

GoScript

The GoScript Postscript interpreter also comes in two versions: a $195 package with 13 typefaces, and GoScript Plus, with 35 fonts for $395. Both versions require a hard disk and use the two-step process — save your Postscript document to a file, then exit the application.

The program supports the HP LaserJet, as well as the HP DeskJet and PaintJet; Canon SX and CX, LBP-8II, and Bubble Jet BJ-130; Epson FX and LQ printers, including 360 dpi for LQ-950 and LQ-2550; NEC Pinwriter; Toshiba 24-pin dot matrix printers; IBM Proprinter, Quickwriter, and Quietwriter; Fujitsu DL Series; and Panasonic KX 24-pin dot matrix printer.

Before printing, however, you have to know how your application handles the preamble, the collection of routines used by your application in its Postscript programs.

If the preamble is included with the script as one file (such as with WordPerfect), you log onto the Goscript directory, then enter GS followed by the name of the print file. GoScript builds the bitmap for the entire page in your computer's memory, or in temporary disk files, then prints the document.

If you're using a program with a separate preamble file (such as Microsoft WORD), you have to specify its name before the Postscript file. For instance, to print a WORD Postscript print file called REPORT.PS, you would enter this command from the DOS prompt:

```
GS C:\WORD5\POSTSCRP.INI C:\WORD5\REPORT.PS
```

GoScript would then use the WORD

Continued on Page 46

Powering Up
Volume 2

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

Connecting a Printer to Your Computer

In the last article, the details of connecting a modem to the serial port on a computer were discussed. That article included a description of the various signals available in the RS-232C serial interface and will not be repeated in this article. Therefore, it is important to read the previous article for a complete description of the serial interface (usually referred to as COM1) before attempting to connect a serial printer as discussed in this article.

This article also includes information on connecting a printer to the parallel port, LPT1, using the Centronics interface. And we will also look at the ZDS CONFIGUR command in terms of how to use it to set up both serial and parallel printers. Let's begin by looking at what a null modem really is.

The Null Modem

There has always seemed to be a lot of confusion about what a null modem really is, and I suppose that is a result of its name. Actually, it is quite simple. A *NULL MODEM* is nothing more than a special cable or "adapter" which allows one DTE device to communicate with another DTE device *without* using a modem. Since you know the general rule that like serial devices (i.e., a computer and a printer) cannot communicate with each other (e.g., DTE to DTE), we can effectively "change" one DTE device (printer) so that it looks like a DCE device by rearranging some of the wires. Because so many people have missed this point, I will stress it again — A modem cable will NOT properly connect a DTE printer to your computer. And a serial printer cable will NOT properly connect a modem to your computer. For additional information on DCE and DTE devices (including how to connect a modem to your computer), be sure to read the previous article.

Although I knew that there were a wide variety of null-modem configurations before I began writing this article, my research uncovered so many different ones that even I was surprised. As a result, I am including only the three most common configurations in this article, but you should be sure to check the Owner's Manual for your specific printer to verify which configuration is correct for your computer/printer combination. In order to identify these configurations, I have identified them as the IBM-specific Configuration, the Compatible Configuration, and the Computer Communications Configuration. But before we get into that specific discussion of each one, there is one point that must be covered to keep the wiring configurations clear. That point is how to convert the 9-pin serial connections found on a laptop or an 80286 or later desktop to a standard 25-pin configuration.

9-Pin to 25-Pin Adapter

This point was mentioned in the original *Powering Up* book on page 57, and Photo 3 on that page shows a cable with a DB-9 female connector at one end and a DB-25 connector at the other. You can also buy a small "adapter" that has the appropriate connectors at each end, and the adapter is about the same size as the Gender Changers shown in Photo 4 on page 58 of the same book. My personal preference is to use a cable adapter because it requires less space behind the computer, and that can be important in some installations.

Many people have asked me why a serial port on one computer has a DB-25 pin connector and a different computer only has a DB-9 connector. My usual answer is that is the IBM standard, but I do not know why that was done. To be specific, the DB-25 connector was used on the IBM Asynchronous Communications Adapter card (for a serial port) that was available as an option for the IBM PC and XT which used an 8088 CPU. As a result, most of the manufacturers of compatible computers, including Heath and Zenith Data Systems, also used a DB-25 connector for the serial port, except that it was STANDARD (not an extra-cost option) for most non-IBM computers. When IBM released the 80286-based AT computer, it included a serial port as standard, but it had a 9-pin connector. One can guess that IBM used a 9-pin connector for the serial port because of space limitations or to avoid confusion with the 25-pin connector for the parallel port. It could have also been a cost consideration because a DB-9 connector is less expensive than a DB-25 connector. I don't know, except that that was the IBM "standard." And once the DB-9 connector was used on a desktop, it is quite probable that it became a standard on laptop computers because of size/space limitations. In any case, the use of the DB-9 connector became a standard on 80286 and later computers, and virtually all manufacturers, including Heath and Zenith Data Systems, followed that standard. Let's begin by looking at the cable configuration required to convert a 9-pin connector to a 25-pin connector as shown in Figure 1.

Figure 1 shows the pin connections as they are required for a straight-through adapter, which is what I recommend to keep things simple. In other words, Figure 1 only converts the signals from a DB-9 connector to the standard DTE configuration. Despite the way Figure 1 "looks," it is *NOT* a null-modem configuration. It will work with a DB-25 connector for your modem (if you have the modem cable described in the previous article) or printer with the proper null-modem cable (described in this article or in your printer's Owner's Manual). I should also mention that the adapter configuration shown in Figure 1 will *NOT* support a serial printer

Computer DB-9 Pin	Output DB-25 Pin	Description
1	8	Carrier Detect (CD)
2	3	Receive Data (RD)
4	20	Data Terminal Ready (DTR)
6	6	Data Set Ready (DSR)
7	4	Request to Send (RTS)
8	5	Clear to Send (CTS)
9	22	Ring Indicator (RI)
Case	Case	CHASSIS GROUND (CG) to cable shield

Figure 1
DB-9 to DB-25 Adapter

on the eaZy PC's 9-pin port. With that caution, let's take a look at each of the null-modem configurations.

Null-Modem Configurations

The first one — IBM-specific — is generally for IBM printers, although it may work with other brands of printers as well. Again, be sure to check the Owner's Manual for your printer (or check with the printer manufacturer) to see what the specific cable requirements are. Figure 2 shows the IBM-specific cable configuration.

Computer Pin #	Printer Pin #	Description (Computer/DTE)
2	3	Transmitted Data (TD) --> Received Data (RD)
3	2	Received Data (RD) <-- Transmitted Data (TD)
4	8	Request to Send (RTS) — Carrier Detect (CD)
5 & 6	20	Clear to Send (CTS)/Data Set Ready (DSR) — DTR
7	7	SIGNAL GROUND (SG) <--> SIGNAL GROUND (SG)
8	4	Carrier Detect (CD) — Request to Send (RTS)
20	5 & 6	Data Terminal Ready (DTR) -- CTS/DSR
Case	Case	CHASSIS GROUND (CG) to cable shield

Figure 2
IBM-Specific Configuration for DB-25

During my research, I managed to find a manual for the HP LaserJet Series II printer, and it is interesting to note that the recommended cable is quite similar to Figure 2. The manual shows (on page 3-37) that the recommended cable only connects pins 2, 3, 5 & 6, and 7 on the computer side to the pins shown above on the printer side. In other words, you only need to use four wires to use the serial interface on the LaserJet Series II as opposed to all seven wires (excluding shield) shown in Figure 2.

The second configuration, which I call the Compatible Configuration, is slightly different as shown in Figure 3.

My research indicates that the compatible configuration shown in Figure 3 will work with a number of different brands of computers and printers. One source told me that this cable will also work with Apple computers and printers, but I have not tested or confirmed that.

Aside from connecting a serial printer to your computer, a null-modem cable can also be used to connect two computers together so that you can transfer files quickly between systems — with the proper software (e.g., HyperACCESS/5) of course. The third configuration — the Computer Communications Configuration — is generally used for that purpose and allows for additional hardware handshaking between the two computers. Figure 4 shows this configuration.

Figure 4 shows a computer connected to a general DTE device which may be either another computer or a printer. In Figures 2, 3, and 4, you will note that the Ring Indicator (RI) for Pin 22 has been intentionally omitted (and should not be connected) because it does not apply when connecting two DTE devices.

As you can see, there are a variety of ways to construct a null-modem cable, and the exact configuration depends primarily on what hardware you are connecting and to a lesser extent, the purpose of the connection. If you compare any of these cable configurations to the "modem" cable discussed in the previous article, it is obvious that a null-modem cable is considerably different than the "regular" configuration. More importantly, you must know which cable is which because you might have two null-modem cables: one for your printer and one for connecting two computers together for file transfer.

Computer Pin #	Printer Pin #	Description (Computer/DTE)
2	3	Transmitted Data (TD) --> Received Data (RD)
3	2	Received Data (RD) <-- Transmitted Data (TD)
5, 6 & 8	20	Clear to Send (CTS)/Data Set Ready (DSR)/CD — DTR
7	7	SIGNAL GROUND (SG) <--> SIGNAL GROUND (SG)
20	5, 6 & 8	Data Terminal Ready (DTR) -- CTS/DSR/CD
Case	Case	CHASSIS GROUND (CG) to cable shield

Figure 3
Compatible Configuration for DB-25

Now that you know what each kind of cable configuration looks like, you should be able to find a ready-made cable or build one of your own. In general, you can find all of the parts you will need at any large electronics store.

Once you have the proper cable, you can connect two DTE devices. If you are connecting a printer to your serial port, be sure to double-check the Owner's Manual to verify the appropriate cable configuration listed here will work. If there are any differences, be sure to follow the recommendation in the printer manual. If

Computer Pin #	DTE Pin #	Description (Computer/DTE)
2	3	Transmitted Data (TD) --> Received Data (RD)
3	2	Received Data (RD) <-- Transmitted Data (TD)
4	5	Request to Send (RTS) — Clear to Send (CTS)
5	4	Clear to Send (CTS) — Request to Send (RTS)
6 and 8	20	Data Set Ready (DSR)/Carrier Detect — DTR
7	7	SIGNAL GROUND (SG) <--> SIGNAL GROUND (SG)
20	6 and 8	Data Terminal Ready (DTR) — DSR/CD
Case	Case	CHASSIS GROUND (CG) to cable shield

Figure 4
25-Pin Computer to 25-Pin DTE

you are connecting two computers together for transferring of data, then you will need to refer to your communications software manual for the proper procedure. If you are connecting a printer to your computer, there are several other things you must do before the printer will work properly.

Getting a Serial Printer to Work

Like a modem, most of today's printers also have DIP switches that must be set correctly for a printer to work properly. As usual, you should check the printer manual for the correct settings, and in most cases, you will find that the factory default settings will work just fine. Once you have the printer connected to your computer with the proper cable, the next thing you should do is check the manual and printer for the correct DIP switch settings. Be sure to do this before you power up anything.

The next thing to do is to establish the proper system configuration for the printer. You must check your printer's manual to find out what the correct parameters are, and you will need to have the following information: baud rate, type of parity, number of data bits, and number of stop bits. Note that this is the same kind of information required when you are setting up a modem. The major difference in setting up a printer is that you must enter these parameters in a different way so that your computer can talk to the printer. For initial testing, I usually recommend establishing the parameters with the MODE command and using the PRINT command to see if everything works.

For purposes of this discussion, I will assume that you want to connect the printer to COM1, the usual case. Let's say your printer manual tells you that the printer parameters are: 4800 baud, No parity, 8 data bits, and 1 stop bit. To set those parameters, you would enter the following command:

```
MODE COM1: 4800,N,8,1
```

As shown, this command sets COM1 for 4800 baud, No parity, 8 data bits, and 1 stop bit as assumed for this "example" printer. If everything is set up properly, you should see an information message like: "Resident portion of MODE loaded" or a similar confirmation message. If you don't, go back and check everything including the cable, printer switches, and of course, make sure the printer is powered on and has paper.

Now you can use the PRINT command to see if your computer can basically communicate with the printer. Because the PRINT command only works properly for printing an ASCII file, I recommend using it to print the AUTOEXEC.BAT and/or CONFIG.SYS files for testing. I highly recommend using these files for initial testing because they must be ASCII files (required for the PRINT command) and they are usually small, so you will not waste a lot of paper if something is not quite right. Of course, you can use any ASCII file for this test, so long as you are absolutely *CERTAIN* that it IS an ASCII file. Data files created with many word processors and virtually all spreadsheets will NOT be printed properly with the PRINT command unless you have used some option that specifically stores data in the ASCII format. If you find that the PRINT command does not correctly print the file, you will need to check the printer manual for possible solutions. Most printer manuals have a "Troubleshooting" section.

The process of getting a printer to work correctly is quite easy to describe, but it is sometimes more difficult in practice, so let's review the basic steps. First, make sure that the DIP switches are set correctly, and that usually means to make sure they are set to the manufacturer's recommendation. Resist the temptation to change any setting until you get the printer working satisfactorily.

Second, make sure that you have the correct serial cable for the printer. When you buy a printer, I recommend that you buy the recommended cable for it because this will help minimize problems and troubleshooting time. The cable configurations shown in Figures 2 and 3 will generally work with most serial printers, but you should check the Owner's Manual provided with your printer to make sure that you follow the manufacturer's recommendation for the cable. ALWAYS follow the manufacturer's recommendation for best results. Now that you have verified that the hardware (DIP switches and cable) is set up correctly, you can think about powering up the system.

Third, use the MODE command to install the proper parameters for the printer. The details required for each parameter are found in the Owner's Manual for the printer. I recommend using the MODE command for initial set up and testing because it is faster and easier to change. Remember that you will need to re-enter the MODE command every time you power up or reboot your system until you make a permanent change with the ZDS CONFIGUR command.

Fourth, use the PRINT command to print an ASCII file, such as AUTOEXEC.BAT or CONFIG.SYS. If the printer does not seem to work correctly, double check the parameters you entered for the MODE command. If those are correct, you will probably need to refer to the Owner's Manual to check the settings of the DIP switches. It is extremely important to test the printer with a simple file using the DOS PRINT command because if that does not work correctly, you will not be able to print anything correctly on the printer.

And finally, it is time to see if your word processor works with your new printer. When you get a new printer, it is extremely important to remember that you *MUST* change or "install" that printer for each and every one of your application programs. In particular, you must install the correct printer driver in each of your application programs, especially for a word processor like Word Perfect or WordStar. Some spreadsheets, such as Quattro Pro, also have printer drivers, and you must make sure that the correct driver is installed for every application like this. I usually recommend testing with a word processor because it can exercise virtually all of a specific printer's features, such as bold and underline. Many of today's word processors include a special file that contains a test document to check out the available features on a printer. And be sure to check each software manual (and "readme" file on the distribution disk), just in case there is a change required to a specific printer's DIP switch configuration.

Now that you know the printer works with the basic printer communication features of DOS, it's time to set it up "permanently" with the ZDS MS-DOS CONFIGUR command. I recommend using CONFIGUR because it makes changes to the BIOS and does not require any additional memory like the memory-resident MODE command does. Since you already have the basic parameters for your printer that have been tested with the MODE command, you can enter that information by following the CONFIGUR program menus. For the newer versions of DOS, you may be able to use the factory default (with no changes) of the "Compatibility Mode" as shown in the box at the top of the CONFIGUR menu on the first screen. Note that the factory default will only work if you have a 2400 baud printer that has parameters of No parity, 8 data bits, and 1 stop bit. Otherwise, you will need to select the CONFIGUR menu's "Configure a COM device" option, select the port to be configured (normally COM1), and you may find your printer is already listed in the CONFIGUR menu. If it is not, just select the "User Defined" option. Answer *N* to both questions about stripping parity, answer *N* to both questions about mapping to uppercase, select the baud rate for your printer, select the number of stop bits (normally 1) for your printer, select the correct parity ("Even" is a good starting choice if you do not know) for your printer, select the word length ("8 bit words" is a good starting choice if you do not know), select the handshaking ("Compat." is a good starting choice for virtually all of today's printers), enter *0* for the number of pad characters and 255 for the time out value, and save the changes to memory and disk. By the way, all of these suggestions are simply that, and you may have to experiment to get these parameters set correctly for your particular printer. And if your printer manual lists different parameters, by all means follow the manufacturer's recommendations. And be sure to keep notes on what parameters you entered so that it will be easy to make the same changes in CONFIGUR to future DOS versions. I recommend rebooting the system at this point.

Now that you have set the printer configuration up, check it out again with the PRINT command. If you used the tested parameters and recommendations noted above, everything should work fine. If it does not, rerun the CONFIGUR command, and re-enter the correct parameters. Be sure to save all changes to memory and disk as before.

Once you have this set up, now you can check it out with your application software, such as a word processor or spreadsheet. This is a step that many people miss, and it causes a lot of headaches, both for computer users and hardware manufacturers. This step involves using some kind of installation or change program (supplied with the application) to tell the application to use a specific port address (e.g., COM1, in this example) and what printer driver to use for that port. If you omit this step, you will probably find that your printer will not work at all with that application or it may only print garbage. More importantly, you *MUST* "install" the printer for each and every application program you have so that the printer will work properly. How this is done depends on what applications you have, and you should check each of your manuals for detailed information. One clue that you have not followed this procedure correctly is that you can print an ASCII file with the PRINT command, but your printer will not print correctly with an application program, such as a word processor.

The Centronics Connection

The parallel or Centronics (developed by Centronics Corporation) interface is generally easier to implement because it is basically standard for virtually all computers. With one or two exceptions, it is really much easier to add a parallel printer to a computer's LPT1 port because there is really only a single cable configuration. It turns out that the Centronics interface was designed to be a low-cost, easy-to-use interface specifically for a printer. Aside from the fact that a parallel printer is easier to set up because of the single cable configuration, its data transmission capability is much faster as you will see in a moment. But there is also a significant disadvantage.

The disadvantage is that a parallel cable cannot be very long. Many manuals suggest that the maximum length of a parallel cable be six feet or less. I have successfully used ten-foot cables with no problem, but that may be because I use a high-quality shielded cable with good connectors. The reason that parallel cables have a limited length is because the interface is designed to operate at low-voltage levels that are available inside a computer, which is one of the reasons for its low cost. In particular, the Centronics interface operates at TTL (Transistor-to-Transistor Logic) signal levels which are limited to five volts. A logic 0 (false) is defined as a voltage between 0 and 0.8 volts, and a logic 1 (true) is represented by any voltage between 2 and 5 volts. A parallel interface utilizes a MAXIMUM of 5 volts, and a serial interface may utilize up to a maximum of *25 volts*. Because all cables have some resistance to current flow, a long cable for a parallel transmission may reduce the voltage of a logic 1 signal so that is looks like a logic 0 signal by the time it gets to the printer. In other words, a printer may not print anything or it may print garbage if the cable is too long. That is an important point to remember when troubleshooting a problem on a Centronics parallel interface. If everything else seems to be set up correctly and you are still having problems, then try using a different cable that is shorter than the original. Now let's take a look at the details of the Centronics parallel interface.

Figure 5 shows the cable configuration for the standard DB-25 connector on a computer to the standard 36-pin Centronics connector on a printer.

If you compare the cable configuration in Figure 5 with any of the null modem (serial) configurations, you will note that there are eight data lines from the computer's parallel port (pins 2 through 9), but there is only a single output line from the computer's serial port (pin 2). A serial interface transmits one bit at a time on a single line, but a parallel interface can transmit a complete *BYTE* (8 bits) of data simultaneously because there are eight data lines. For that reason, a parallel interface can transmit data faster than a serial interface, which explains why much of the high-speed data transfer software (e.g., LapLink) uses the parallel port to transfer data between two computers very quickly.

In general, the computer sends a SELECT IN (computer pin 17) signal to the printer to "allow" it to print, and the printer verifies that it is selected by sending a SELECT OUT (computer pin 13) signal to the computer. Many older printers have a DIP switch that can be set to permanently "select" the printer, and you may want to check your Owner's Manual for a DIP switch setting if you cannot get a printer to print anything. Virtually all Epson printers send a 5 volt signal on the SELECT OUT line to verify that the printer is selected, so that is generally no problem.

The computer sends a timing pulse, called STROBE, to tell the printer when all 8 bits (one byte) of data are transmitted (simultaneously) on the data lines (DATA0 to DATA7). The printer returns an ACK pulse to acknowledge receipt of a byte of data.

The BUSY signal (computer pin 11) from the printer tells the computer to stop sending data. Virtually all of today's computers have a small amount of memory (called a buffer) that is used to store data before printing. A printer will typically send a BUSY signal of 5 volts to the computer when that buffer is full or when an error is encountered (e.g., printer off

Computer Pin #	Signal Direction	Printer Pin #	Description
1	-->	1	STROBE — Timing for printer
2	-->	2	DATA0 — 1st data line
3	-->	3	DATA1 — 2nd data line
4	-->	4	DATA2 — 3rd data line
5	-->	5	DATA3 — 4th data line
6	-->	6	DATA4 — 5th data line
7	-->	7	DATA5 — 6th data line
8	-->	8	DATA6 — 7th data line
9	-->	9	DATA7 — 8th data line
10	<--	10	ACK (Acknowledge)
11	<--	11	BUSY — Printer busy
12	<--	12	PAPER EMPTY — Printer out of paper
13	<--	13	SELECT OUT — Printer verifies selection
14	-->	14	AUTO FEED EXT — Automatic Line Feed
15	<--	32	ERROR — Printer error
16	-->	31	INIT — Initialize printer
17	-->	36	SELECT IN — Printer is selected
18	n/a	n/a	Not connected
19-25	n/a	19-25	Ground
Case	n/a	Case	Chassis Ground to cable shield

Figure 5
25-Pin Computer (LPT1) to 36-Pin Printer

line or other error). A "low" signal in the 0-0.8 volt range is sent when the printer is "not busy" and can accept data.

The AUTO FEED EXT (computer pin 14) signal is rarely used today, but the computer (and software) can use this signal to control what a printer does when it receives a Carriage Return (CR) character. When this signal is low (0 volts from the computer), a Carriage Return simply returns the print head to the first column on the left side of the paper. When the signal is high (5 volts), the printer performs an *automatic* Line Feed (LF) in addition to the Carriage Return (CR). In general, this signal should be at the 5 volt level so that a CR is treated only as a CR character, not a CRLF combination. Many of today's printers, such as my Epson FX-850, have a DIP switch that can permanently ensure that a CR is printed only as a Carriage Return. And most of today's software expects to have total printer control so that it will send a CR and/or LF whenever it is required. A symptom of an incorrect signal or DIP switch setting usually appears as an extra blank line on a printout, and it can usually be corrected by setting a printer's DIP switch correctly. If that does not work, you will then have to check the appropriate software manual or "readme" file to see if your software has a specific requirement for a different setting on your printer.

The printer sends an ERROR signal (on computer pin 15) in the 0-0.8 volt range indicating that an error condition exists, such as paper empty or off line. A 5-volt signal indicates the printer is operating normally.

The last two signals may cause problems with some computers and printers, and they are the exceptions mentioned in the beginning of this section. The first signal, INIT (Initialize printer), is sent by the computer, and the signal "resets" the printer to the default configuration and clears the printer buffer. The INIT signal is not really necessary for many printers because most software has printer drivers that will send necessary commands, including initialization, to a printer. The computer will typically send an INIT signal to a printer when the computer is powered on or rebooted with CTRL-ALT-DEL. If your printer is powered on, you can always tell when this happens because most dot matrix printers will make some noise, which is mostly resetting the print head to the first available print column.

The PAPER EMPTY signal (computer pin 12) can cause real problems, and I have run into this one a couple of times. Although most printers have a feature that allows you to disable the printer's response to a "paper empty" condition when you use single-sheet paper, the printer may still send a PAPER EMPTY signal back to the computer. I have had particular problems with this when using a sheet feeder on a dot matrix printer, but it is quite easy to fix if you know how the signal works. A printer will send a PAPER EMPTY signal of 5-volts to the computer indicating an out of paper condition. To *ALWAYS* keep the signal at 0 volts (indicating the printer is NOT out of paper), I opened both of the connectors and cut the pin 12 connection on both. When I cut the pin 12 wire on the computer connector (DB-25), I left enough wire connected to the pin so that I could solder it to the connector case, which permanently grounds that wire at 0 volts so I will never get a PAPER EMPTY signal. By the way, this trick does NOT disable the feature that allows a printer to beep when it is really out of paper or the paper in a sheet feeder does not feed correctly.

Getting a Parallel Printer to Work

People rarely have much difficulty in getting a parallel printer to work, and the steps installing it are easier than a serial printer. The first step is to make sure the DIP switches are set correctly, and that usually means to make sure they are set to the manufacturer's recommendation.

Second, make sure that you have the correct cable for the printer. When you buy a parallel printer, I recommend that you buy the recommended cable for it because this will help minimize problems and troubleshooting time. Although cable problems for a parallel hookup are rare (except if the cable is too long), it is best to buy the proper cable when you buy the printer.

The third step is actually to test the printer, and this is where the implementation steps begin to differ from a serial printer installation. There is seldom any need to use the ZDS CONFIGUR command to set up a parallel printer, especially with the newer DOS versions such as 3.3 Plus or 4.0. These newer ZDS MS-DOS versions are shipped with all defaults set up to correctly run virtually all IBM-compatible printers, and you should not use CONFIGUR unless you find some problem with the default parallel setup. The test process is quite simple: use the PRINT command to print an ASCII file, such as AUTOEXEC.BAT or CONFIG.SYS. If the printer does not seem to work correctly, then you will probably want to run the ZDS CONFIGUR command after checking the printer parameters in the Owner's Manual. You also may need to refer to the Owner's Manual to check the settings of the DIP switches. Again, it is extremely important to test the printer with a simple file using the DOS PRINT command because if that does not work correctly, you will not be able to print anything correctly on the printer.

And finally, it is time to see if your application programs, such as a word processor or spreadsheet, works with your new printer. Remember that you *MUST* change or "install" that printer for each and every one of your application programs. The correct printer driver must be installed for each of your application programs, especially for a word processor like Word Perfect or WordStar. Some spreadsheets, such as Quattro Pro, also have printer drivers, and you must make sure that the correct driver is installed for every application like this. As before, I usually recommend testing with a word processor because it can exercise virtually all of a specific printer's features, such as bold and underline. And don't forget to use any test document that is usually available with many of today's word processors to check out the available features on a printer. And be sure to check each software manual (and "readme" file on the distribution disk), just in case there is a change required to a specific printer's DIP switch configuration.

Powering Down

Although this point was mentioned in the previous article, it should be obvious in this article that the RS-232C standard for the serial interface is not even close to being standard when you are connecting a serial printer to your computer. The fact that there are so many serial cable configurations is because a given printer manufacturer decided to ADAPT the RS-232 standard to support a specific printer, and not all printer manufacturers implemented that standard in the same way. If you are having difficulty installing a new serial printer in your system, then I recommend that you read your printer's Owner's Manual and/or check with the printer manufacturer for help, especially for cable configuration and DIP switch settings.

If the printer does not seem to work correctly with one specific application program, such as a word processor, then you should check that program's Owner's Manual and/or check with the software manufacturer for help. This applies to both serial and parallel printers. For example, Microsoft Windows has had a problem in correctly printing anything on an HP LaserJet printer, and I suppose that Microsoft will have that corrected by the time you read this.

Although someone at a Heath or ZDS dealer may be able to help you with advice on installing a serial or parallel printer, it is important to recognize that any installation problem you have is not likely to be directly caused by Heath, ZDS or ZDS MS-DOS (or any other DOS, such as PC-DOS). Virtually all printer installation problems are caused by incorrect DIP switch settings, bad or incorrect cable configurations, incorrect parameter definition in MODE or CONFIGUR, or by using an incorrect or invalid application program's printer driver for an application program. Of course, there is always the possibility that some kind of hardware failure inside the computer (or printer) can cause a problem, but a hardware failure is usually quite obvious because the printer works fine with everything and then suddenly stops working correctly.

The next article will help you in Choosing a Programming Language if you are interested, an overview of some of the popular languages available for microcomputers. This article will discuss some of the advantages and disadvantages of these languages including: Assembler, C, Basic, and Pascal. And because of the popularity of data bases, I will also include a brief discussion of dBase compatible programming.

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

Products Discussed
HUG Software

Powering Up (885-4604)	$12.00

Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (HUG Software only)

Continued from Page 28

tute for block move.

<CTRL-F5>, which is labeled INTERPRET above, brings up a menu that lets you select what kind of macro you're working on. If it's a WordPerfect version 5 macro, select WPM. For macro editor macros, select MEM. Other choices pertain to version 4.2 and Shell macros. The current active selection appears at the bottom left of the screen. This selection determines what file extension is added when you save your macro, and it determines what commands are entered into the macro in FUNCTIONS mode. If you select MEM, you'll get a {Switch} command in your macro when you press <F9>. If you select WPM, <F9> gives you an {Underline} command (not an underline code).

The difference between a command and a code is important. Codes appear only in documents; commands only in macros. An underline code causes text to print with an underline. If you reveal codes in a document, you see the following:

```
[UND]...text...[und]
```

An underline command in a macro enters underline codes into the document when the macro is executed.

Going into the FUNCTIONS mode with <CTRL-F10> enables you to directly type any command that has a keystroke equivalent. Commands which do not have keystroke equivalents, like {IF} and {CASE}, can not be entered this way. Neither can you type the individual characters that make up these commands. If you do so, they are treated as text not commands.

In order to enter non-keystroke commands, you must first press <CTRL-PgUp>. This pops up a window containing a menu of commands.

Move the light bar cursor to the desired command and press <ENTER>. The highlighted command is entered into the macro text at the cursor position. Instead of moving the light bar cursor with the cursor arrow keys, you can begin typing the letters that make up the desired command. As you type, the light bar will move to the first command that matches the letters you have typed. When the desired command is highlighted, stop typing and press <ENTER>. After the command is inserted, you must type any text that follows it, including tildes.

Save completed macros by pressing <F10>. When prompted for the macro file name, do not press <ALT-letter>. If you do so, the program attempts to run a macro editor macro of that name. You can save a word processor macro that will run at the press of <ALT-A> by typing <A><L><T><A> at the prompt. Note that you must type the individual characters, not hold down <ALT> and press <A>. It's best to specify the path to the word processing directory, as well as the file name here. Otherwise, the macro may not end up where you want it. Once I found some lost macros in my root directory instead of my WP directory. The extension (WPM or MEM) is added automatically during SAVE.

Conclusion

The WordPerfect macro programming language is very powerful, but very fussy about syntax. You can use it to create your own prompts and menus and do many other things that you can not do with keystroke macros. If you already program in some high level language, it won't take long to master MPL. Just don't forget the tildes.

Continued from Page 36

seller, and I can certainly see why. Since I live in Smyrna, just outside Atlanta, I checked with three local discount software houses which carry CCPLUS. Their prices ranged from $29.00 to $39.95. Naturally, I went with the least expensive, which was Micro Center. Micro Center's only retail locations at present are here (Marietta, GA) and Ohio (Columbus and Westerville); their mail-order operation seems to be limited to supplies like floppies and paper. Other retailers such as Egghead and Software Etc. are also likely to have it.

I like CCPLUS because it fully automates the tedious part of keeping calendars straight, such as remembering holidays and printing up-to-date calendars. Consequently, it frees human beings to do the creative part — entering events and keeping them updated. The ease with which new calendars can be printed and the wide range of output choices make it much more likely that people will make the slight effort needed to manage their calendars effectively.

Technical And Ordering Information

Program

Calendar Creator Plus, Version 3.0	$49.95

(Check Micro Center or other local stores for price and availability.

Sources

Power Up!
Channelmark Corporation
P.O. Box 7600
San Mateo, CA 94403
Order line: 800-851-2917
(800-223-1479 in CA)

Continued from Page 40

preamble file POSTSCRP.INI to interpret the print file. If you always plan to print files from that application, however, you can add a line to GoScript's configuration file. Then whenever you print a document with GoScript, the Word preamble file will first be interpreted, so you don't have to type it from the DOS prompt.

GoScript also allows you to print multiple copies and has an interactive mode for entering commands directly to the interpreter.

All of the methods described here give you the capability to print Postscript documents. While hardware Postscript is faster, the software interpreters provide an inexpensive way to harness Postscript for almost any printer.

Products Mentioned

PS-388 Accelerator
Princeton Publishing Labs, Inc.
19 Wall Street
Princeton, NJ 08540
(609) 924-11563

PS Jet
LaserConnetion
1 Magnum Pass
Mobile, AL 36618
(205) 633-4300

Pacific Page
Pacific Data Products
6404 Nancy Ridge Drive
San Diego, CA 92121
(619) 552-0880

GoScript
LaserGo, Inc.
9235 Trade Place, Suite A
San Diego, CA 92126
(619) 530-2400

UltraScript
QMS Corporation
2650 San Tomas Expressway
P. O. Box 58101
Santa Clara, CA 95052-8101
(408) 986-9400

Other PostScript Boards
ConoDesk 6000
Conographic Corp.
16802 Aston Street
Irvine, CA 92714
(717) 474-1188

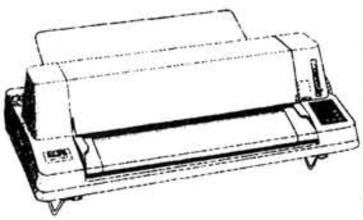
EiconScript
Eicon Technology Corp.
2196 32nd Avenue
Montreal, Quebec
Canada H8T-3H7
(514) 631-2592

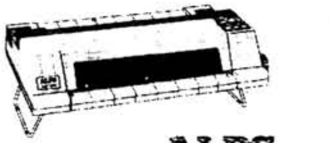
Pagestyler
Destiny Technology
300 Montague Expressway, Suite 150
Milpitas, CA 95035
(408) 262-9400
(800) 874-5653

Don't get HYPER or CROSS because your modem software is too complex to use, get *HUGMCP* for the no-hassle modem connection.
MoC

Don't Miss A Single Issue!

Let us know 3-4* weeks before you move!





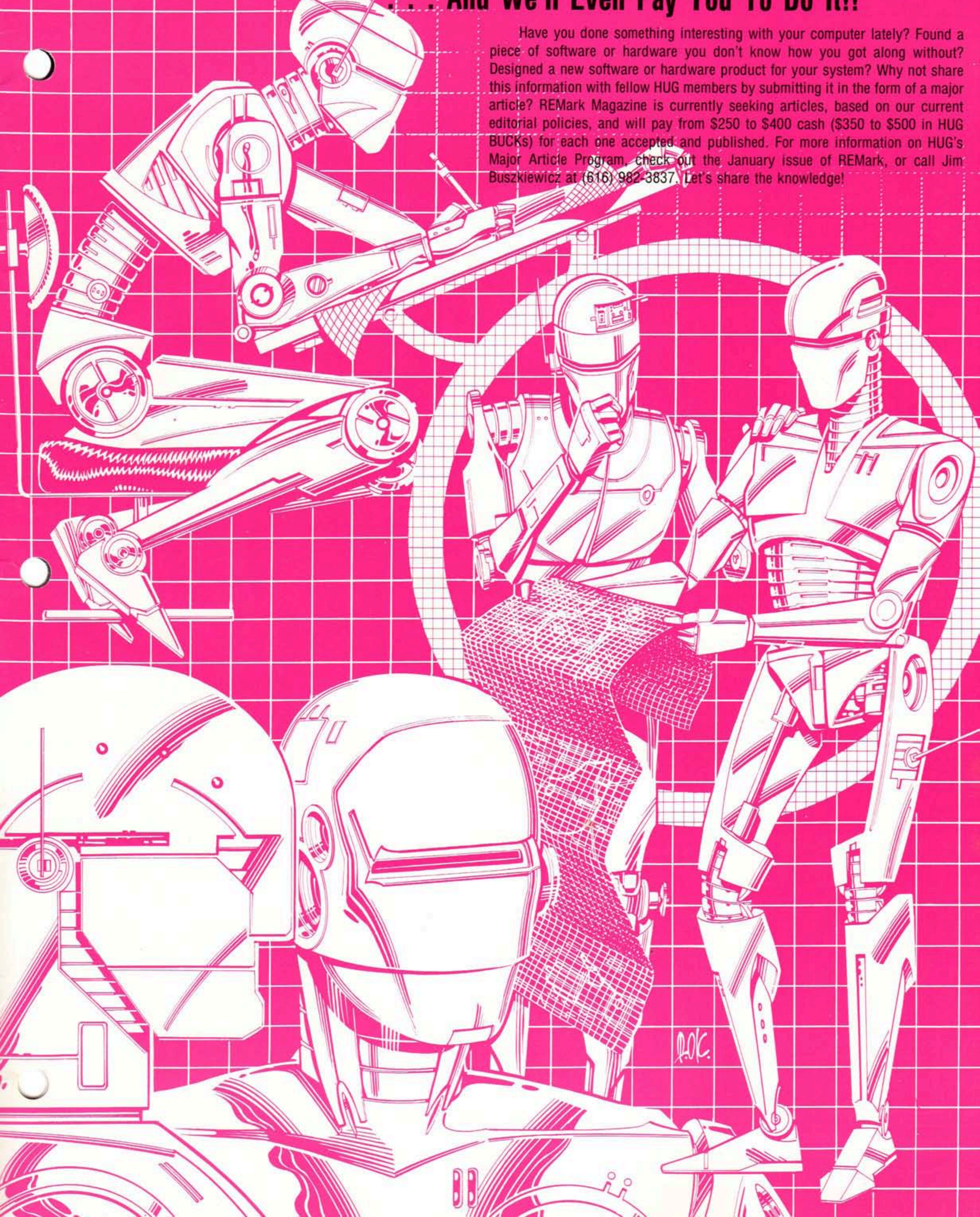
Don't Miss A Single Issue!
Let us know 3-4* weeks before you move!

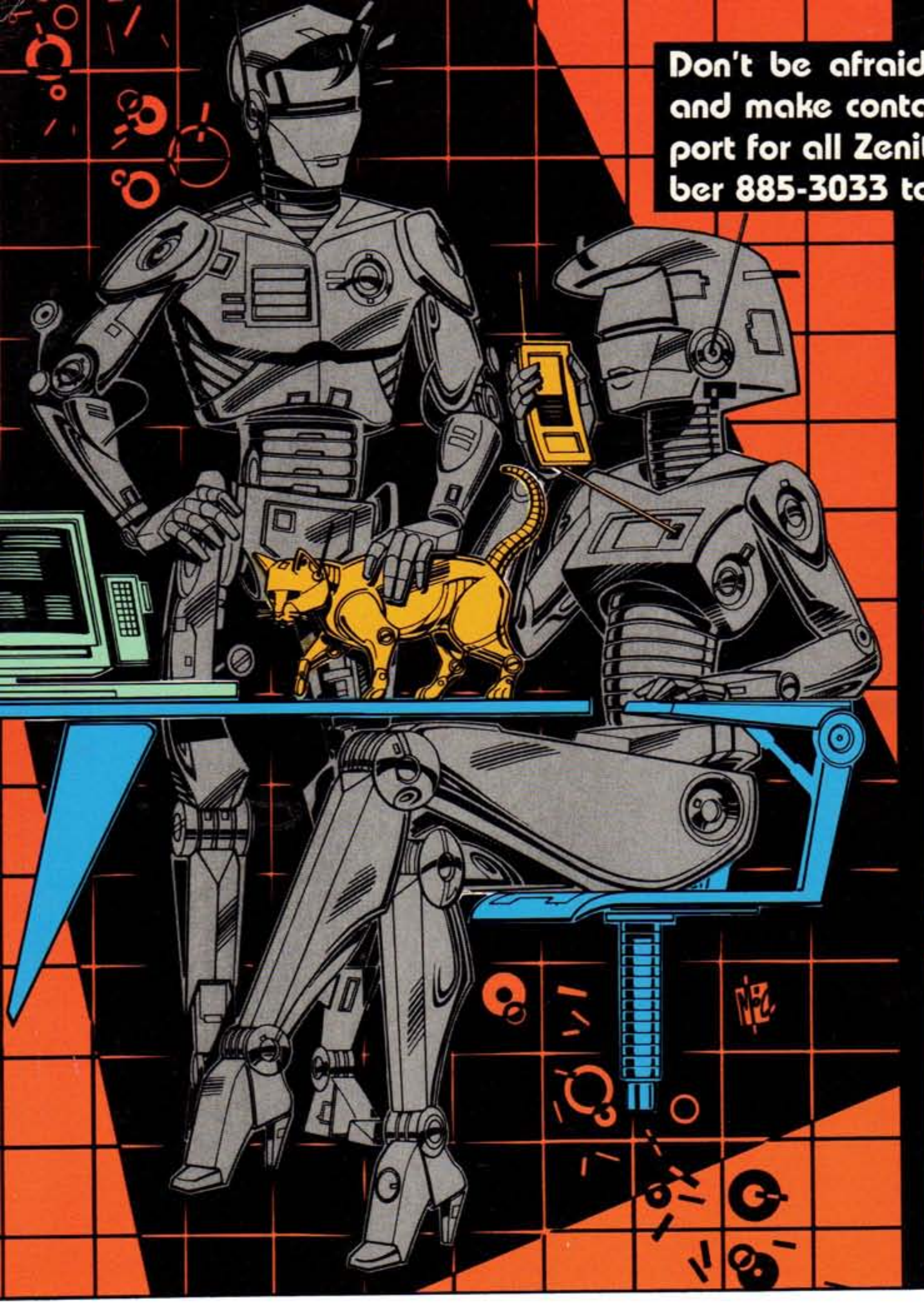
Classified Ads

Z-100 FOR SALE: Low profile 768K RAM, full color video, ZVM-135 color monitor, 8 MHz clock speed, 2 - 5.25" drives, manuals and software included. Best offer over $500 takes all. Will sell computer/software only. Call (703) 455-3114 or write 8100 Lake Pleasant Drive, Springfield, VA 22153.

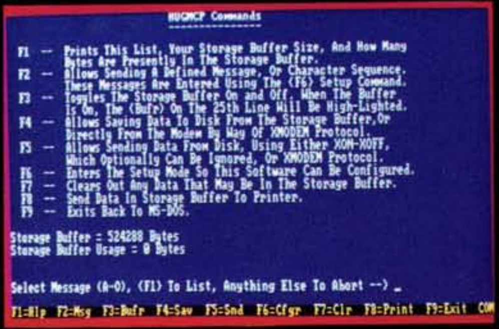
Authors, Share Your Knowledge With Other HUG Members, . . . And We'll Even Pay You To Do It!!

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!

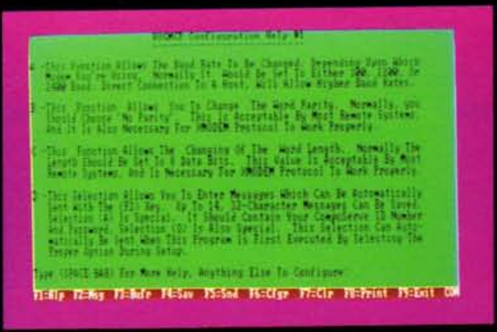




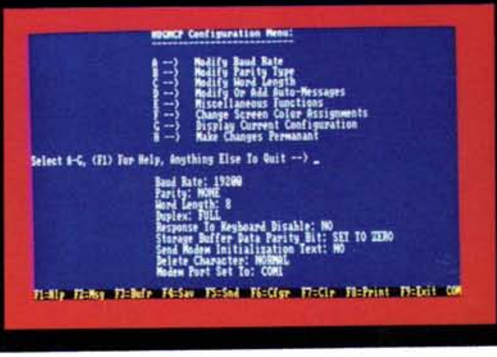
Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all Zenith Laptops, order HUG Part number 885-3033 today.



```
                         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, 32-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
Users'
Group

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

$2.50

P/N 885-2127