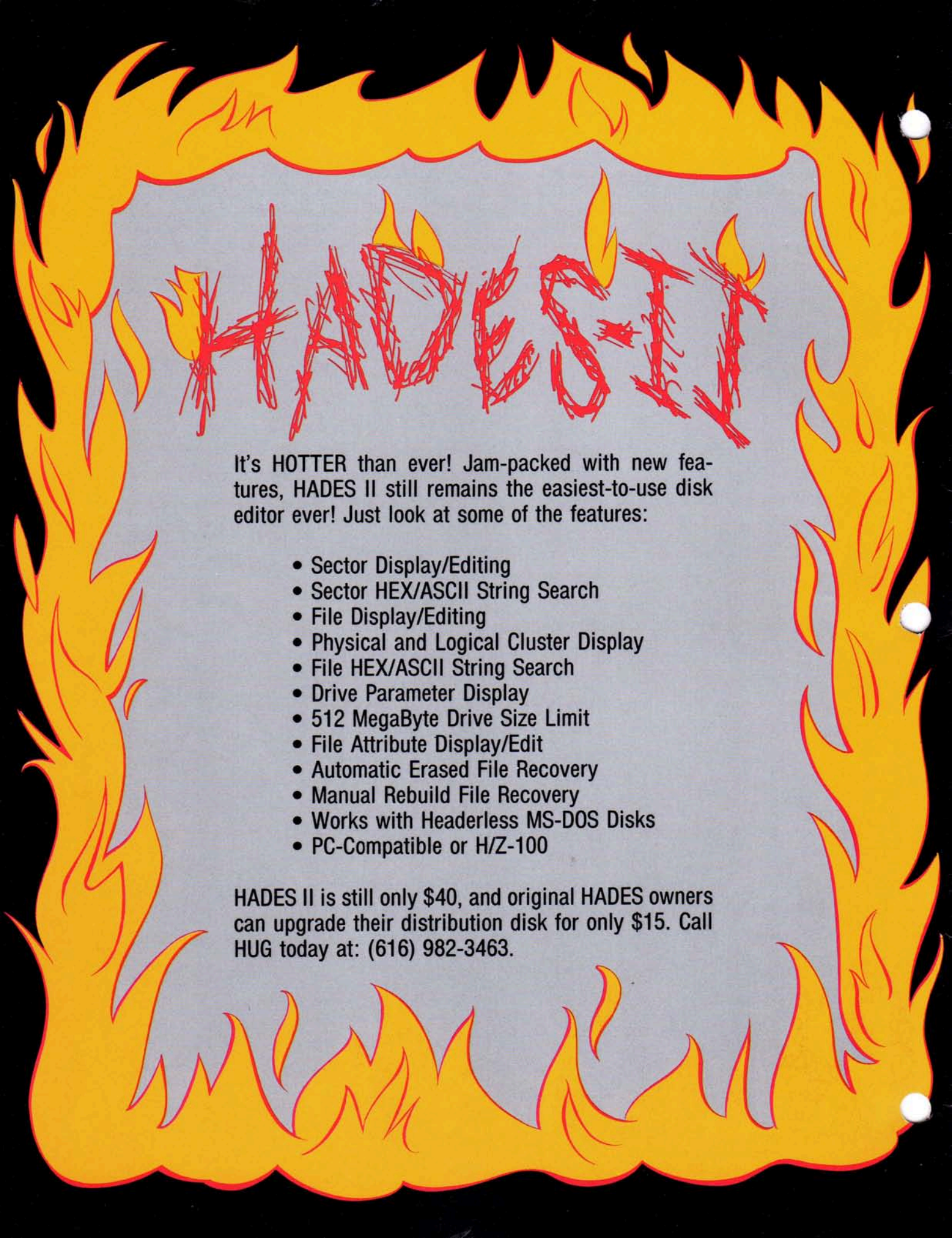
REMark®

January 1991

The Official Zenith Data Systems Computer Users Magazine



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.

January 1991

REMark®

The Official Heath/Zenith Computer Users Magazine

Resources

PC Compatibles

Z-100 and PC Compatibles

General

Managing Editor
Jim Buszkiewicz
(616) 982-3837

Software Engineer
Pat Swayne
(616) 982-3463

Production Coordinator
Lori Lerch
(616) 982-3794

Secretary
Lisa Cobb
(616) 982-3463

COM1 Bulletin Board
(616) 982-3956
(Modem Only)

ZUG Software Orders
(616) 982-3463

Hardware Questions
(616) 982-3309

Printer
Imperial Printing
St. Joseph, MI

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

	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 ZUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and include appropriate, additional cost.

Send Payment to: Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463

Although it is a policy to check material placed in REMark for accuracy, ZUG 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 Zenith Data Systems Computer Centers.

ZUG is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their usage of Zenith Data Systems equipment. As such, little or no evaluation of the programs or products advertised in REMark, the Software Catalog, or other ZUG publications is performed by Zenith Data Systems, in general, and Zenith Users' Group, in particular. The prospective user is hereby put on notice that the programs may contain faults, the consequence of which Zenith Data Systems, in general, and ZUG, in particular, can not 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 Zenith Users' Group, St. Joseph, Michigan.

Copyright (c) 1991, Zenith Users' Group

Software

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H8 - H/Z-89/90				
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 PKG	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	CHKBK 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 PKG	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 SECT SUPPORT PKG	885-1121	HDOS	UTILITY	30.00
HDOS PROG. HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUP UTILITY	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PROD TO 1982	9.75
HUGMAN & MOVIE ANIM	885-1124	HDOS	ENTERTAINMENT	20.00
INFO SYS AND TEL. & MAIL SYS	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DB	25.00
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 DEC '80	20.00
RUNOFF	885-1025	HDOS	TEXT PROCR	35.00
SCICALC	885-8027	HDOS	UTILITY	20.00
SMALL BUISNESS PACKAGE	885-1071-[37]	HDOS	BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PKG	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PKG	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUN & UTIL	20.00
TINY BASIC COMPILER	885-1132-[37]	HDOS	LANGUAGE	25.00
TINY PASCAL	885-1086-[37]	HDOS	LANGUAGE	20.00
UDUMP	885-8004	HDOS	UTILITY	35.00
UTILITIES	885-1212-[37]	CPM	UTILITY	20.00
UTILITIES BY PS	885-1126	HDOS	UTILITY	20.00
VARIETY PACKAGE	885-1135-[37]	HDOS	UTILITY & GAMES	20.00
WHEW UTILITIES	885-1120-[37]	HDOS	UTILITY	20.00
XMET ROBOT X-ASSEMBLER	885-1229-[37]	CPM	UTILITY	20.00
Z80 ASSEMBLER	885-1078-[37]	HDOS	UTILITY	25.00
Z80 DEBUGGING TOOL (ALDT)	885-1116	HDOS	UTILITY	20.00
H8 - H/Z-89/90 - H/Z-100 (Not PC)				
ADVENTURE	885-1222-[37]	CPM	GAME	10.00
BASIC-E	885-1215-[37]	CPM	LANGUAGE	20.00
CASSINO GAMES	885-1227-[37]	CPM	GAME	20.00
CHEAPCALC	885-1233-[37]	CPM	SPREADSHEET	20.00
CHECKOFF	885-8011-[37]	CPM	CHKBK 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 CAT UPDT #1	885-4501	VARIOUS	PROD 1983 TO 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
X-REFERENCE UTIL FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATIONS	20.00

Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H/Z-100 (Not PC) Only				
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	UTILITY	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
DUNGN & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSDOS	PRINTER PLOT UTIL	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	UTILITY	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BKGRD 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
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	ATRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PKG	885-3018-37	MSDOS	VARIOUS SPRDST	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
H/Z-100 and PC Compatibles				
ADVENTURE	885-3016	MSDOS	GAME	10.00
BACKGRD 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
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 CAT UPD #1	885-4501	MSDOS	PROD 1983 - 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
ICT 8080 - 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAG DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS' PC &Z100 UTILITIES	885-3052	MSDOS	UTILITIES	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
REMARK VOL 11 ISSUES 120-131	885-4011	N/A	1990	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				
CARDCAT	885-6006	MSDOS	CAT SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMUL	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOT UTIL	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' 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 UTIL	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00
YAUD (YET ANOTHER UTIL DSK)	885-6015	MSDOS	UTILITIES	20.00

This Software Price List contains all products available for sale. For a detailed abstract of these products, refer to the Software Catalog, Software Catalog Update #1, or previous issues of REMark.

Now Available!

ZUG 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 ZUG's MS-DOS software is available on 3-1/2" micro-floppies too! When ordering, just add a "-80" to the 7-digit ZUG 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, MasterCard, and American Express phone orders, telephone the Zenith Users' Group directly at (616) 982-3463. Have the part number(s), description(s), 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: Zenith Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. VISA, MasterCard and American Express require minimum $10.00 order. No C.O.D.s accepted.

Questions regarding your subscription? Call Lisa Cobb at (616) 982-3463.

BUGGIN' ZUG

Help for SupersPort 286

Dear ZUG:

Do any of your REMark readers or experts know of a way to use an enhanced 101-key keyboard with my Zenith Data Systems SupersPort 286 portable.

As you know, it was not originally equipped with a port or socket to connect a detachable keyboard. I bought it in the Fall of 1989 and am very pleased with the computer, but I now have a specialized application which requires that I be able to use a detachable enhanced keyboard.

I would appreciate any help the HUG experts or readers can give.

Sincerely,
Tom Holub
2904 Georgia Street
Vallejo, CA 94591

Looking for Z-170 Information

Dear ZUG:

Some time ago, I purchased a Heath/Zenith portable computer model number Z-170 with dual 260K drives, 640k memory, a video board and built-in modem.

I'm interested in expanding the unit, and would like to ask your users for help and information.

I would like to purchase 1.4mb/1.2mb drives for my unit. Would also like to hear from users who have added a hard drive or other expansion items. Need schematic and/or manuals.

Thank you in advance for your assistance!

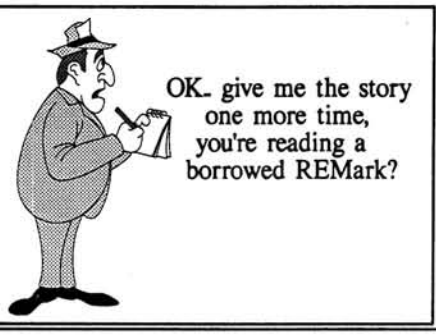
Sincerely,
George Kelm
P.O. Box 476
Pohnpei, FM 96941

Local Club Update

Dear ZUG:

The "Shore Heath/Zenith Users' Group (SHUG) will meet for the very last time in the H/Z store where they've been meeting (continuously) every month since July 1981. Because of a policy change by store management since acquisition by Groupe Bull, most clubs have left store premises and demised (disbanded). After the February 13th BLOW OUT FAREWELL Party, the SHUG will change its meeting place, but not the day or time (2nd Wed., 7:30pm); its address and telephone number will change, but not the ver dynamic membership of 45-50 computerists. The BBS (24 hrs.) 2400 baud n-8-1 at (908) 775-6705. Please respond if you'd like more information.

Rich Holst, President
Blake Berning, Secretary



Moving?!?

Don't miss a single issue of REMark!

Please let us know 3-4 weeks before you move. Call (616) 982-3463 or write:

Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217

Microsoft(R) Windows-PC 8.00

20 Meg. Hard Drives From 75.00

SURPLUS COMPUTER PARTS

- **Largest Surplus Electronics Dealer in Western Michigan**
- **In Business Over 40 Years**
- **Over 12,500 square foot Store and Warehouse**

Examples of Zenith Data Systems Salvage Products:

Z-100 Motherboards	From 20.00
Z-100 Power Supplies	15.00
Z-100 Winchester Controller & Data Separators	Ea. 35.00
Swivel Base for 1490 Monitor	15.00
Z-171 1/3 ht. drives	15.00
Laptop Keyboards	20.00
20 Meg. Tape Backups	From 25.00
8087 & 80287-6 Co Processors	75.00
Numeric Keypads for Laptops	20.00
Laptop Modems	From 5.00
PC & AT 84-key Keyboards	From 8.00
Z-148 Power Supplies	15.00
Zenith Data Systems 101-key Keyboards	From 15.00
Equipment Power Cords	2.00
Power Supply Fans	From 5.00
40 Meg. Hard Drives for 286 SupersPort	300.00
Laptop Replacement Batteries	From 6.00

Many Other Zenith Data Systems Salvage Parts Available
All Surplus/Salvage Sold *AS IS - WHERE IS*
- NO GUARANTEE -
No Catalog - Please Call for Quotes
We Ship U.P.S.-*C.O.D. (ONLY)* Sorry - No Foreign Shipments

Surplus Trading Corp.
THE HOUSE OF EVERYTHING "ALMOST"
2700 N. M-63, P.O. BOX 1082
BENTON HARBOR, MI 49022

CALL: LES - (616) 849-1800
FAX - (616) 849-2995 (Orders Only)

Reader Service #153

Desktop Publishing

Improving Your Image

Inexpensive Hand Scanners and Digitizers Can Add Pizazz to Ho-Hum Documents

Mark Haverstock
6835 Colleen Drive
Youngstown, OH 44512

Publishers and graphic artists tell us that too much text is boring and makes readability difficult. They maintain that we need graphics to improve the appearance and readability of our desktop publications. The old saying, "One picture is worth a thousand words," certainly applies, but where do we get the right pictures and illustrations when we need them?

Clip art is great, but often you can't find the exact picture you need. Using typical clip art from booklets is inconvenient because you have to leave blank spaces where you think the pictures will fit. Clip art is also available on disk, but a reasonably complete collection of can cost a bundle. What do you do when you need a custom picture or illustration? Consider a scanner or digitizer.

Hand Scanners

Scanners are designed to capture an image, such as line art or a photograph, and convert it into a format that can be used by one of the many drawing or publishing programs available. Most scanners that are used with PCs can produce images from 75 to 400 DPI (dots per inch) with up to 256 levels of gray. They can be set to read at various levels of brightness, darkness or resolution, usually from the accompanying software. This is about all any scanner does, be it an expensive full-size table model or a relatively inexpensive hand scanner.

Hand scanners, like their larger counterparts, contain basically the same parts: mirrors, lights, and motherboards. Figure 1 is a diagram of its basic components. Actually, these units can do just about everything a desktop unit can, except scanning a full 8.5 x 11 inch page in one pass. They have the advantage on price ($200-500), and take up virtually no desk space.

The hand scanner has a somewhat undeserved reputation for being difficult to keep steady while being pulled over a piece of artwork. Beginners who are not familiar with the hand scanner often voice this complaint. Ideally, the scanner should be moved across the picture with one smooth, steady stroke. But the scanner can still compensate for quite a bit of movement during a scan and produce good pictures. Scanners such as Logitech's ScanMan Plus include an indicator light to help you roll the scanner across your picture at the right speed. This helps to prevent distortion or vertical squeezing in your image.

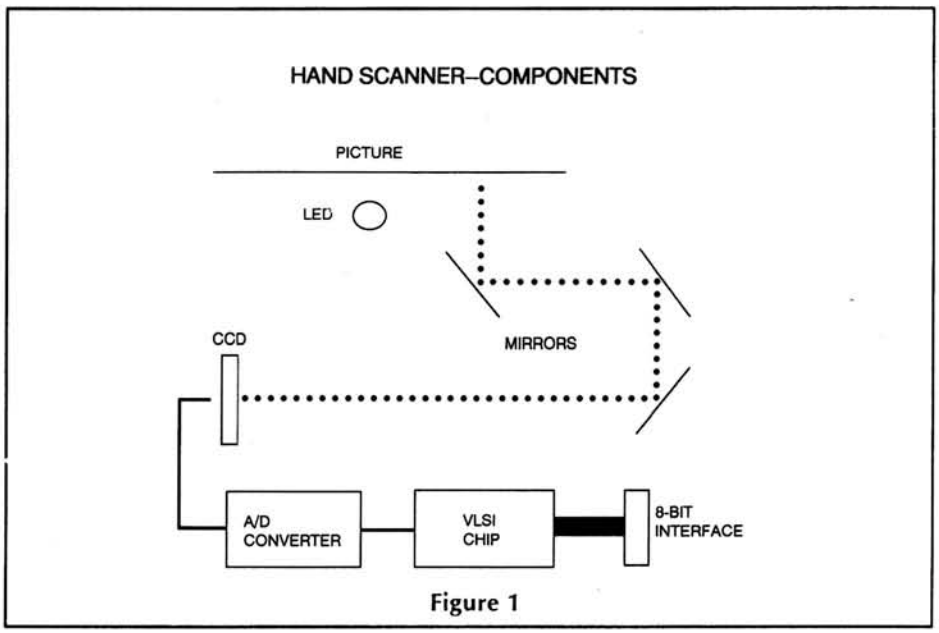


Figure 1

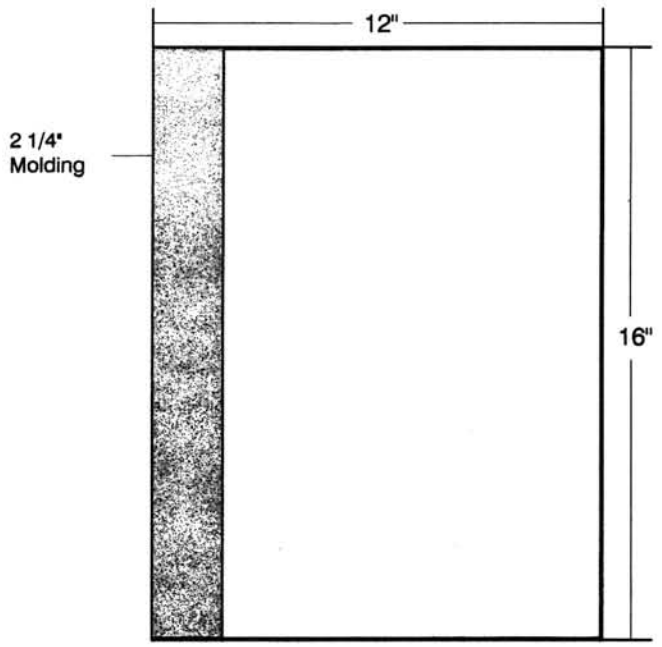
There are a few simple tricks to help you get the best possible scans with your hand scanner. One is to tape a ruler to the desk to act as a straightedge. I prefer to use a simple scanning table constructed out of a scrap sheet of plywood and some molding (Figure 2). It's easy to build and can be stored away when not in use. Difficult-to-hold images can be tacked or taped to the table. They also could be placed under a transparent sheet.

Scan more slowly for images with shades of gray or when using higher resolution settings. If you scan too fast, the scanner may not be able to receive all the incoming information. As a result, the image may be distorted or squeezed vertically.

If your original is too large, you can reduce it on a photocopier and then scan. Another approach is to scan a large original in several passes. Some scanner software will allow you to reassemble it later.

Color artwork, even line art, may not scan well. This is particularly true with art containing shades of red. To compensate, you can adjust the brightness control. Another approach is to make a photocopy of the original image, and then scan the copy.

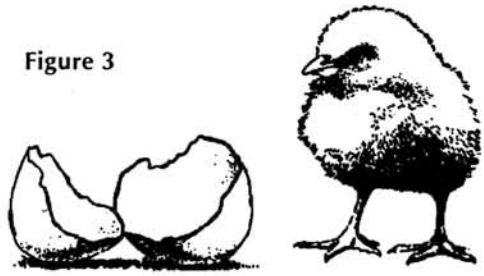
All pictures don't scan equally well. Line art usually works best. Most people who



SCANNING TABLE

Figure 2

Figure 3



LINE ART 200 DPI

DITHERED B&W PHOTO 200 DPI

Figure 4

use scanners will tell you that they use scanners primarily for line art rather than photos. A black-and-white photo with good contrast can be scanned with reasonable success. Color pictures, generally speaking, produce poorer quality scans. Figures 3, 4 and 5 are examples of scanned line art, a black-and-white photo, and a color photo done at 200 DPI respectively.

Unfortunately, none of today's scanners, including the high priced desktop units, can faithfully capture the gray tones that make up a photo. Conventional methods of printing pictures are still superior to scans. However, you can expect an acceptable halftone image, suitable for most newsletter use.

These halftones are produced through a process called dithering. Most hand scanners are capable of rendering at least 32 shades of gray. Dithering arranges the pixels in particular patterns to suggest gray shades, and most hand scanners have several dither settings ranging from coarse to fine. Generally speaking, the finer the dither pattern, the clearer the image appears. You may need to experiment with dithering to get the best results.

Scanning resolution determines how the image is digitized and displayed on the screen. Resolution is the number of dots per inch and ultimately determines the degree of detail in your scanned image. The more dots per inch, the better the resolution. Of course, you want the best resolution possible, but there are some other things to consider.

The resolution setting also determines how large the image appears on the screen. Compare the picture that was scanned at 200 DPI in Figure 6 with the picture in Figure 7 that was scanned at 400 DPI. When you double the scanning resolution, the surface area is multiplied by four.

When you increase the resolution, you

DITHERED COLOR PHOTO 200 DPI

Figure 5

200 DPI

Figure 6

400 DPI

Figure 7

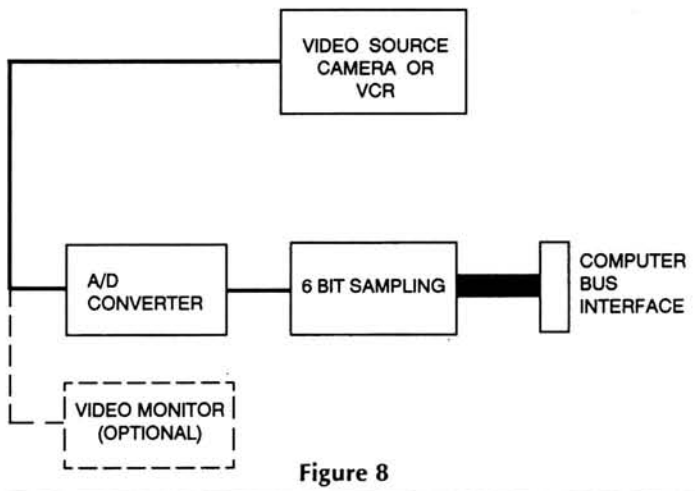


Figure 8

also increase the size of the file. This reduces the area you can scan, since you are limited by the available RAM memory in your computer. A good rule of thumb is to use the 200 DPI setting for general scanning, and 300 DPI for printing on laser or PostScript printers.

Several hand scanners are available at reasonable prices for IBM PC compatible computers. Four of these are listed below. Most come with bundled software to operate the scanner and manipulate scanned images. A graphics card and mouse are recommended for these scanners.

	Niscan	DFI	Mars 128	ScanMan
Memory required	640K	640K	640K	640K
Resolution (DPI)	100-400	100-400	200-400	100-400
Scan width (inches)	4.2	4.13	5	4.1
Software	Gem Scan	Halo DPE	Halo DTP	PaintShow Plus

Video Digitizer–Computer Eyes

There's a growing number of people who own video camcorders. Businesses, families, and schools routinely use them to record special events. Chances are you've taped many items that could be used as clip art or illustrations. Why not include some of those faces from your video in a newsletter or presentation? A video digitizer can change your video images into picture files..

One of the most widely available of these packages, ComputerEyes by Digital Vision, can easily capture images from any video camera, VCR, or even the Cannon Xap Shot digital camera. Plug the video output from any of these sources into the video accessory board and you're on your way to creating black and white images that you can manipulate with your favorite graphics or publishing program.

ComputerEyes works with any PC, XT, AT, 386 or compatible with a minimum of 384K RAM. It supports Hercules, CGA, EGA, MCGA, or VGA graphics adapters with up to 64 gray levels on the screen. The package includes a half-size video interface board and software on a 5-1/4 inch disk. Figure 8 is a block diagram of the components in a typical ComputerEyes system.

The best way to describe ComputerEyes is to compare it to the concessions you often see at fairs and carnivals that produce computerized picture T-shirts and posters. The subject sits in front of a camera, the equipment captures a halftone image and then prints it out on an iron-on transfer sheet. ComputerEyes, however, offers a more sophisticated range of features for capturing and editing video images.

The heart of the program is the capture command. This is the part that allows you to take the image from your video camera or VCR and display it on the monitor. During the capture, the image is not visible - it appears after the capture is complete. The capture command has two speed settings. The 6-second fast speed gets the image quicker, and is fine for lower graphics resolution modes. The 12-second normal speed gives somewhat better results with higher resolution.

Both manual and automatic calibration features are available. The automatic calibration usually produces the right combination of brightness and contrast. The operation takes about seven seconds to perform, and may be canceled at any time with the ESCape key.

Manual adjustment of the images can be made by adjusting the brightness or contrast controls. When you select either brightness or contrast, a slider control on the screen is activated, allowing you to adjust the image as needed. A restore calibration setting defaults to the last calibration setting.

When the capture is complete, the screen is painted from the image data. The image remains until you press any key, which returns you to the main menu. If you want, you can go back to the main menu to readjust the brightness or contrast levels and then rescan. Once you have an image to your liking, you can go back to the main menu to change the graphic mode or display style as needed.

ComputerEyes gives you several different graphics modes:

Hercules Mode - 720x348
CGA - 320x200 and 640x200
EGA - 320x200, 640x200 and 640x350
VGA - 320x200 and 640x480

After you've selected a graphics mode, you can choose a display style. There are up to four display styles available, depending on the graphics mode you select.

High Contrast - All areas of an image are set as black or white. This is perfect for line art or sketches.

Dithered Intensity - Dithering represents grays as areas of varying dot intensities. Even though some graphics modes can only display two or four actual gray levels, dithering gives the perception of many more gray levels.

Solid Intensity - In graphics modes that are capable of displaying multiple gray levels (EGA and VGA), intensity levels are

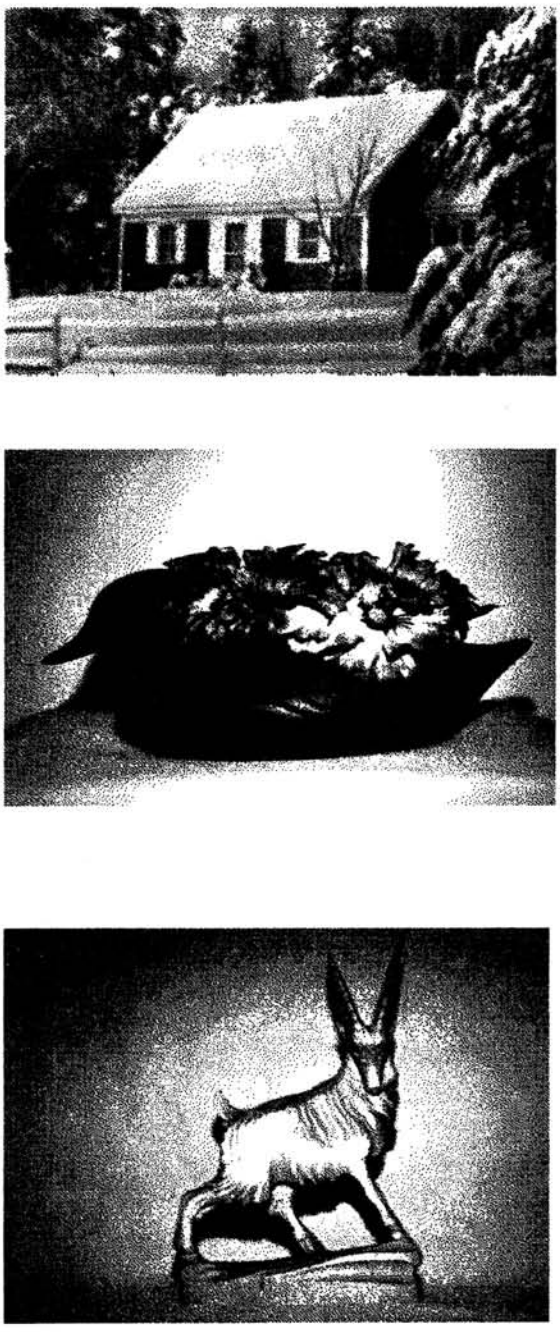


Figure 9

represented as solid shades of gray. When compared to dithering, fewer intensity levels are displayed, but their boundaries are more easily discernible.

False Colors - This style is only available with EGA. EGA is only capable of displaying 16 colors out of a palette of 64, and only four of these are shades of gray. The false color option uses all 16 of the EGA's displayable colors to represent the 16 intensity levels. The chosen become the default EGA palette. It is very easy to distinguish the boundaries between intensity levels. You may have seen similar techniques in medical imaging (CAT scans).

If you have a composite video monitor, ComputerEyes provides a view video input function. When this command is selected, the signal from the video source (camera or VCR) is sent directly to the monitor. This is especially helpful when you are trying to capture a single image from a video tape. As you step through the video tape in slow motion or freeze frame, you can watch and select the correct frame.

ComputerEyes files may be saved to disk in several formats. The first is the ComputerEyes Raw Data format with a resolution of 641 by 201 pixels. This is the data that is acquired when you capture an image. Although this format is not directly compatible with the other formats, it can be stored or readily converted to any of several widely-used formats. These formats include: TIFF (PageMaker), MSP (MicroSoft Paint), PCX (PC Paintbrush), IMG (GEM-based programs).

Once you've saved the captured images, they can be imported into drawing or desktop publishing programs and printed. Figure 9 contains examples of files saved in TIFF format that were printed using Aldus Pagemaker.

A color version, ComputerEyes Professional, has the same features as the B & W version, but adds color capture and adjustment capabilities. It requires MCGA, VGA or Extended VGA to operate.

Summary

Hand scanners and digitizers are two reasonably inexpensive ways of improving your image by capturing pictures as you need them. There's no need to leave big spaces in your desktop-published newsletters or flyers, only to later cut and paste photos or clip art. Let your desktop publishing program do all it can do, including the placement of your illustrations and artwork.

Another benefit of having a scanner or digitizer is the clip art library you're bound to collect over a period of time. Company logos and frequently used illustrations can be at your fingertips for a fraction of the cost of clip art or custom services.

Products mentioned

DFI $359
Diamond Flower Electric Instrument Inc.
5C Joanna Court
E. Brunswick, NJ 08816
(201) 390-2815

Niscan $299
Nisca Inc.
1919 Old Denton Road
Carrollton, TX 75006
(800) 245-7226

Mars 128 $299
Marstek Inc.
17785-A Skypark Circle
Irvine, CA 92714
(714) 833-7740

Scanman Plus $339
Logitech, Inc.
6505 Kaiser Drive
Freemont, CA 94555
(415) 795-8500

ComputerEyes
B&W Version $250
Color Professional Version $450
Digital Vision, Inc.
66 Eastern Avenue
Dedham, MA 02026
(617) 329-5400

Classified Ads

DESKJET PRINT CARTRIDGE REFILLED Black ink, $8.00. Send print cartridge and check to Altomare PC Solutions, 27 Cove, Alameda, CA 94501.

WANTED: HEATH PC1200 IEE-696 BOARD. Need not work but must have all documentation. Send asking price to Emil Rossdeutscher, 16 West Sixteenth Street 9LS, New York, NY 10011.

Z-248 512K RAM. CGA, 360K and 1.2mb Drives. $700. Al (301) 546-5573.

I Hate Progress!

Richard Kehlmeier
950 Shore Acres Road
Salisbury, NC 28146

As you read this article keep in mind that I am an Architect and not really computer oriented. Some of the things I write about will be elementary to some of you readers who may be more computer knowledgeable than I. However, there may be some who are more like me and might enjoy reading about my experiences.

Back in the computer dark ages of 1985 I decided I needed a PC or equivalent. Since I couldn't afford an assembled one (at least so I thought) I bought a H-151 Heath kit, with 256K memory, and assembled it. Much to my surprise it worked! Of course I realized that H/Z guaranteed that it would. As a matter of fact, a horizontal choke was furnished with the monitor kit, in leu of a vertical one. I could never get it to work and neither could the H/Z Dealer. As a result I still have the Zenith 122 amber monitor "loaned" to me by the dealer.

Back in those days a kit was really a kit and it seemed at that time there were millions of solder joints to make. However, today's kits are much easier to assemble. With two floppy drives and that much memory, I felt that should have met my needs forever.

I would not recommend that anybody deliberately buy any PC and add the things I am about describe. It has been suggested a new 286 machine could be purchased for less than a $1000.00 so why spend the money on improving a 151? On one hand, items were added as the need arose. On the other, I wonder if a new 286 could be had with a quality NEC CD Reader, Nec VGA multisync monitor and 50 meg. hard drive for less than a $1000.00. The purpose of this article is to show that no matter what computer you have there is always some way to come close to meeting your needs without buying a new one. By the way, I think if I did get a new machine, it would be another H/Z product. However, at this time I am really resisting a new one. Maybe I feel some affection for my old 151.

The first sign of progress was to find that there wasn't sufficient memory. By adding another memory board and additional RAM IC's the memory was raised to 640K.

This was great until McGraw-Hill (Publishers of voluminous construction catalogs called Sweets) offered a NEC CDR-77 SCSI Compact Disk Reader, at a good price, in order to use a search program for the catalogs. This couldn't be resisted. However, a 151 is pretty slow for a CD Reader so an AST Hotshot 286 accelerator board was added.

The AST accelerator is a plug-in card that has a ribbon cable connection to the existing CPU board that plugs in between the 8088 chip and it's socket. Not a difficult installation at all. Incidentally, someplace along the line I also installed math coprocessors. Both 8087 and 80287 chips were required to work with the AST accelerator.

There were a lot of problems installing the CD Reader and the 286 accelerator. It took a lot of exchanges of boards and writing special software by NEC and AST to make both devices compatible with the 151. At first it took 15 min. to boot. After the revisions, and toggleing the accelerator before autoexec.bat kicked in, boot time was reduced to 2 min. I must say that both AST and NEC were most cooperative in doing their best to iron out the problems.

I wish some one had suggested at that time to update the BIOS. There might have been fewer problems installing the above peripheral devices. It wasn't until we moved to North Carolina that the then dealer in Charlotte made such a suggestion. I say "the then" dealer because now they are a repair station only. As I understand it, H/Z has reorganized their sales methods and the nearest dealer to me now is in Raleigh which is a quite a distance from here. Since the BIOS update was installed, boot time has been reduced to seconds. The BIOS update is a very cost effective procedure. It is an easy plug-in replacement and reasonably priced.

To further facilitate the operation of the CD Reader, a Seagate ST 225 N SCSI 20 meg hard drive was installed. You may wonder why only a 20 meg. and a SCSI drive. At that time 20 meg. sounded like the world to me and being naive I thought I could daisy-chain from the Nec SCSI card. In addition, by now expansion slots were getting to be in short supply. However as I soon found out, SCSI is not standardized so a Seagate ST01 SCSI card had to be added after all, filling my last slot.

The CD Reader is used much more for a NIBS (National Institute of Building Sciences) CD disk than for the McGraw-Hill search program. This disk at an annual

subscription of $970.00 is really a bargain for those involved in construction. It has all the Government Specifications, some graphic illustrations, government and institutional standards and some CADD symbols to mention a few of the features. In my opinion, there is going to be more and more information on CD disks and a CD Reader may turn out to be a very worth while investment.

However, in order to use the NIBS graphics at least EGA is required. Since I didn't want to mess around with EGA, a NEC multisync GS2A VGA monitor and a Paradise Basic VGA card were added. I might not have had the courage to add any more non-H/Z peripherals if it were not for Robert Maskasky's article in the August issue of Remark about adding a Nec multisync VGA monitor and a Paradise card to a Z-150.

As I mentioned previously, all my slots were filled so you may wonder how VGA could be added. Payload (a Remark advertiser) markets a Z-150 Video Eliminator which piggy-backs the 16K video memory and other necessary functions to the CPU board. It is also an easy plug-in installation and allows the H/Z video board to be removed so the VGA card can be installed in it's place. It has worked so far (knock on wood). FBE (also a Remark advertiser) markets a similar device.

Has anybody else noticed that MSDOS dropped GDU and Compact in v4.0? I have not been able to get results using GDU but I attributed that to my inexperience. However I use Compact quite often. It does apparently improve access time even if it would do no more than defragment and I was surprised that such useful features would be dropped. Since they are stand alones, they can be copied from a previous version and will work.

Now a company called Eclat is Beta testing a catalog system similar to Sweets except it is all on a CD disk. No voluminous catalogs. The problem is, it requires Microsoft Windows to operate which in turn requires 8 meg. of free hard disk. My 20 meg. hard disk has been operating on the ragged edge for some time, even without the approximate 5 megs. another information program takes.

Remark to the rescue again! I might not have seriously considered a third drive had I not read Edouard Piche's article in the September issue about adding a 3.5" drive to an H/Z 158. Otherwise I might have just replaced the 20 meg. drive with a 40 or 60 meg drive, abandoning the 20 meg. drive. Anyhow, just now UPS delivered a second Seagate ST138N SCSI 32 meg. hard drive. Seagate, and the installation manual for the first drive, has assured me that one SCSI card will operate two hard drives. Of course that is important to me as there are no expansion slots left.

I am considering using expanded memory by adding a LIM150, sold by the previously mentioned FBE, to the main memory card. The disadvantage to this is that the LIM150 simulates LIM EMS v3.2. To fully utilize windows it requires LIM EMS v4.0. LIM EMS v3.2 will work but it will store data only. The other solution is to put all the memory on the main memory card using 256K RAM chips. An AST Rampage, AT or Intel Above Board could then be installed in place of the old second memory card (I think) but the LIM150 is a much less expensive solution. Decisions, decisions.

Incidentally, I have typed this using the old MS Word version 1.10 that originally came with the H-151 kit. Since most of my word processing is elementary, once Windows is installed, Windows Write will probably take care of my needs.

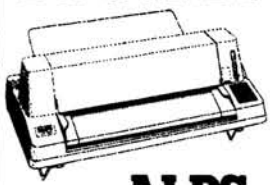
The above should meet my needs forever. **I hate progress!**

W S Electronics

(513) 376-4348 ** Since 1975 **** (513) 427-0287**

1106 State Route 380, Xenia, Ohio 45385

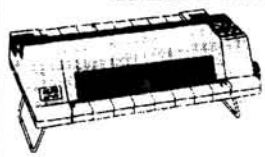
YOU AND YOUR BIG IDEAS.



* 300 cps draft speed
* Wide carriage
* 24 pin printing
* Front panel controls

ALPS *Allegro 500XT™*

THE ALPS ASP1600 PRINTER. COSTS VERY LITTLE, JAMS NOT AT ALL.



- Auto tear bar prevents paper waste. With the flick of a button, your fanfold paper advances to the perforation, then automatically returns to top-of-form position.
- Rugged 9-pin head delivers crisp output at 192 cps in draft mode, 38 cps in letter quality.
- Compact 9-lb. body makes for easy portability.
- Printer stand is built-in.
- Prints labels easily.
- Full Epson FX-85 compatibility.

ALPS AMERICA
Built by popular demand.

SPECIAL

Z-415 1.5 Meg Ram for Z-248	$100.00
Z-505 1 Meg Ram for Z-386	$200.00
Z-315 EMS Kit for Z-159	$ 10.00
Z-417 H. D. Controller Z-248	$100.00
Z-317 H. D. Controller Z-150	$ 75.00

Quanities limited to stock on hand

Attention: Federal Government Offices
We stock ALL ALPS Printer Models and we stock ALPS PARTS and RIBBIONS for all ALPS models including your P2000's and ASP 1000's

******Government Discounts Offered******

We are looking for good dealers.
ALPS Authorized Distributor and Service Center.

Reader Service #149

Getting Started With . . .

Personal Information Managers

Alan R. Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

We spend much of our day organizing schedules and keeping track of time — deciding what we must do and who we have to see. We worry about meeting deadlines and remembering important events. We write notes and reminders, jotting down appointments on a calendar or tablet.

In short, we spend our time trying to organize our time, a crucial commodity that can never be restored when lost.

Wouldn't it be nice if we had someone to manage our time for us? To remind us of appointments, let us know when we have conflicts, look up and dial telephone numbers for us? We need a PIM, a personal information manager that performs these very functions, keeping track of tasks, appointments, people, and notes.

Some PIM functions can be performed by desktop managers. Windows, for example, includes a pop-up calendar, notepad, communications program, and an alarm clock that can remind us of important times. These desktop tools can help organize our activities, particularly when used in conjunction with a database management program. Through multiple databases and indexes, we can link people, places, and activities, and print reports that double as reminders and management tools. However, even this combination of programs still lacks the comprehensive range of features we'd need to organize our entire schedule.

In this article, you'll look at PIMs, a new and exciting class of software. As examples, you'll see two PIMs, PrimeTime Personal, a standard DOS application, and IBM's Current, which operates under Microsoft Windows. In a future article I'll look in detail at Instant Recall, a DOS PIM from Chronologic Corporation.

For all of their power and utility, however, PIMs have not taken the computer world by storm and have had slow acceptance among middle and upper-level managers. This is unfortunate since, as you'll see, PIMs can contribute greatly to a manager's effectiveness.

PrimeTime Personal

As a TSR (terminate and stay resident pop-up program), Primetime Personal is always available. Not only can you pop it up when needed, but it works in the background keeping track of your appointments as you work with another application. You can access your schedule, check on projects, and even auto-dial your phone through a modem, in a few keystrokes.

Instead of simply listing features, let's go through some of the steps you'd take to manage your schedule. You'll see how convenient PIMs can be.

Managing Appointments

Let's start by checking on our appointments for the day

1. Load Primetime Personal, then press Ctrl-Alt to display the main screen shown in Figure 1.

The functions that you can perform, and the current date and time, are listed on the left. On the right is the reference calendar with the current date shown in brackets.

2. Press A to select appointments, then R for review, to show the appointment window for that day across the lower half of the screen (Figure 2). Notice that there is a budget meeting between 10:30 and noon, then a lunch meeting until 1:30. Use the up and down arrow keys to move through the schedule.

Adding Appointments

Now let's add a one hour appointment with the accountant, starting at 3 PM. The function keys to use with the window are listed across the bottom line of the screen.

1. Press the down arrow key to highlight 3 P.M., then press the F2 function key to display the update appointment form (Figure 3). The day's date is already in the form.

2. Fill in the form, as shown in Figure 4. In this case, we've set an alarm to go off 15 minutes before the meeting. Now whatever you're doing on the computer, at 2:45 Primetime Personal will sound an alarm and display a reminder message at the bottom of the screen.

3. Press F9 to save the data and return to the schedule, then ESC to display the main menu.

Now let's add an appointment two weeks from today.

1. Select Appointments, then Update to display the update appointments window.

2. Type 2W, for 2 weeks, then press Enter to display the date two weeks from the current day. Primetime Personal automatically calculates the date using W for weeks, D for days, M for months, Y for

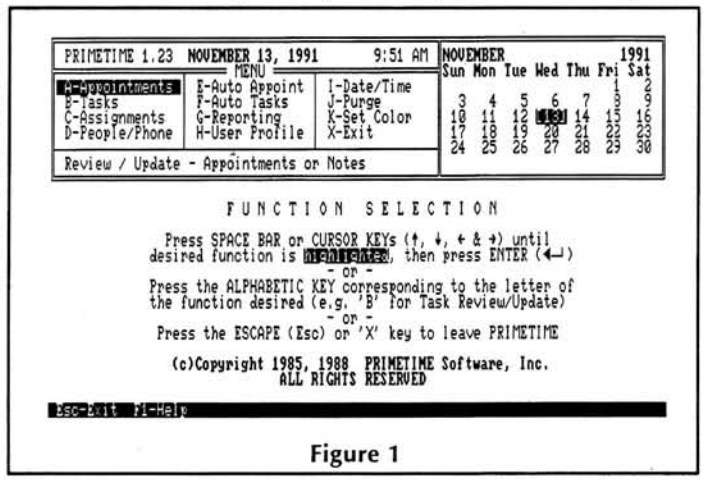


Figure 1

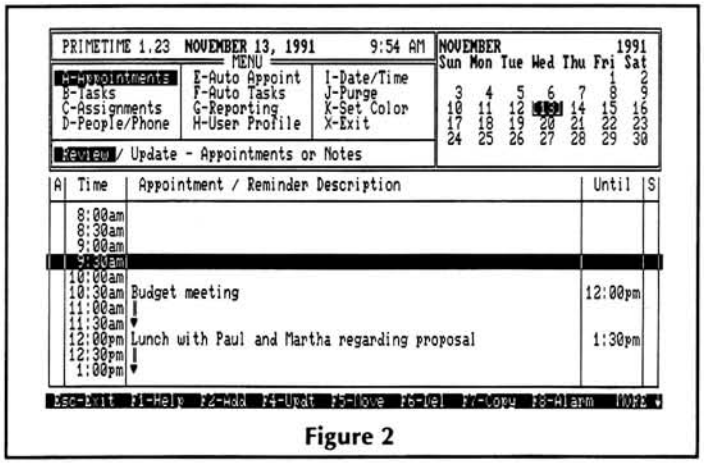


Figure 2

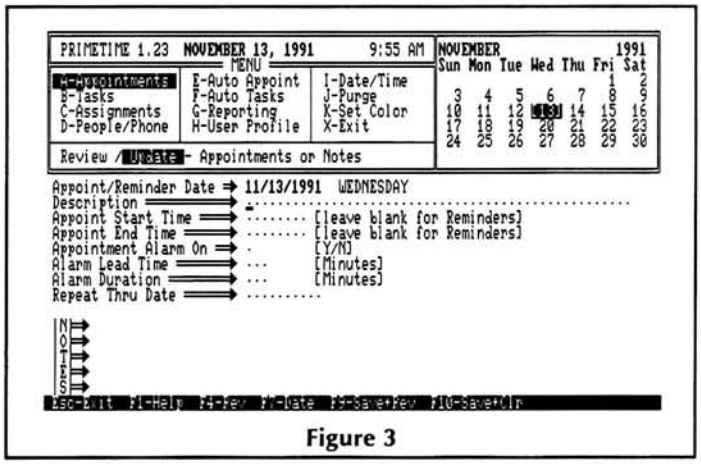


Figure 3

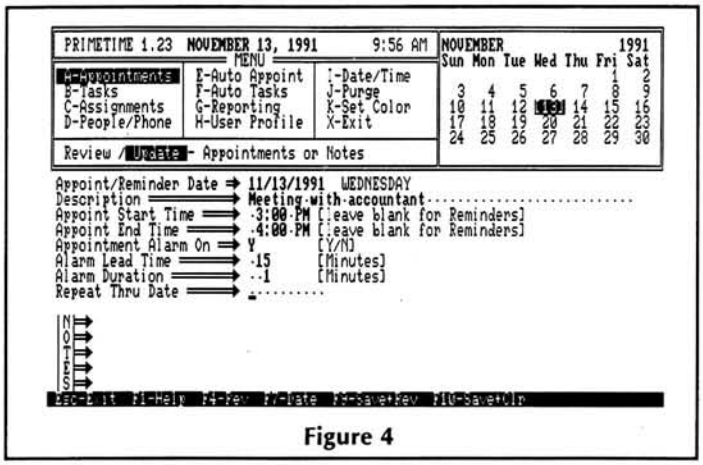


Figure 4

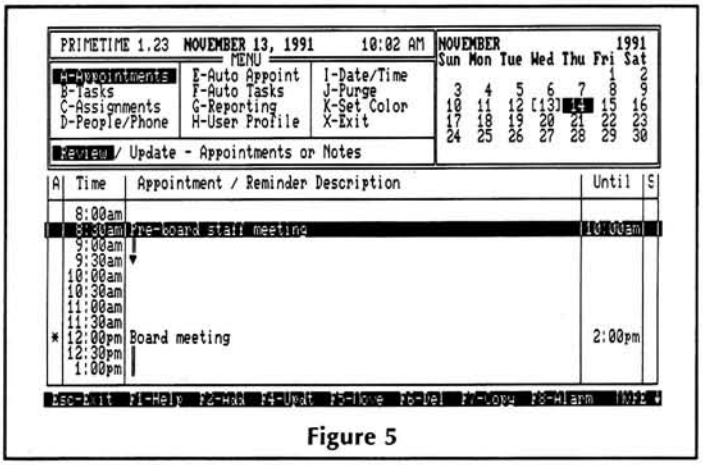


Figure 5

years, or N for next month. For example, enter 12N to set the date at the 12th of next month.

3. Enter the remaining formation for the appointment, then press F9.

Changing Appointments

When you first start Primetime Personal, the current date is shown in brackets and in reverse on the reference calendar. The reverse indicates the date pointer, the date that is referenced in schedules shown on the screen. To see appointments for another day, you simply point to another day in the reference calendar. The schedule shown will automatically adjust.

You can use this point-and-shoot method to check on the schedule for the next day.

1. Press Num Lock to change to the calendar mode. The keypad will now move the highlight a day (left and right arrow keys), week (up and down arrow keys), or month (PgUp and PgDn keys) at a time.

2. Press the right arrow key to point to the next day. The appointments for that day appear on the screen.

Now, reschedule a 10:30 meeting to start at 8:30.

3. Highlight the 10:30 meeting, then press F5.

4. Highlight 8:30, then press F5 again. The appointment moves to that time and the until column automatically adjusts (Figure 5).

Automatic Appointments

You might have regular appointments or dates to keep track of, such as weekly staff meetings, monthly board meetings, or yearly anniversaries and birthdays. You can quickly set up repeated appointments using the Auto Appointment option. Figure 6 shows the Auto Appointment menu set for a weekly Monday staff meeting. The dates of the appointment are determined by the settings in the Select-A-Date and Periodic sections. The X under M in the weekday prompt indicates a weekly Monday meeting.

By combining various options, you can create any regular schedule. For example, this combination schedules an appointment on the first Tuesday of every month:

```
DAY                          1 1 1
           1 2 3 4 5 6 7 8 9 0 1 2
Month
Week       X
Quarter
           S M T W T F S
Weekday      X
```

To mark a birthday, designate just the day and month, as in this setting for November 16th:

```
DAY        16                1 1 1
           1 2 3 4 5 6 7 8 9 0 1 2
Month                          X
Week
Quarter
           S M T W T F S
Weekday
```

Use the Periodic setting to schedule events that occur at regular intervals, such as this example of every ten days starting September 1, 1991.

```
Base Date          9/1/91
Period TypE        D              [D/W/M/Q/Y]
Interval           10
```

The automatic appointments will appear on the appointments schedule for the designated days.

Keeping Track of People

PIMs include a telephone directory for storing information about the people with which you work. The directory in Primetime Personal includes areas for the individual's personal data, their company or place in your organization, dialing instructions, and notes.

Use the directory to dial the phone if you have an auto-dial modem. You include dialing codes that tell Primetime Personal how to dial the individual based on protocols that you establish.

You designate a primary and alternate code for each person in your directory. For example, you can designate both a primary and secondary number for each individual in your list. For someone within your company, the primary number can be their in-house extension, the secondary number their home phone. For friends, the primary number can be their home phone,

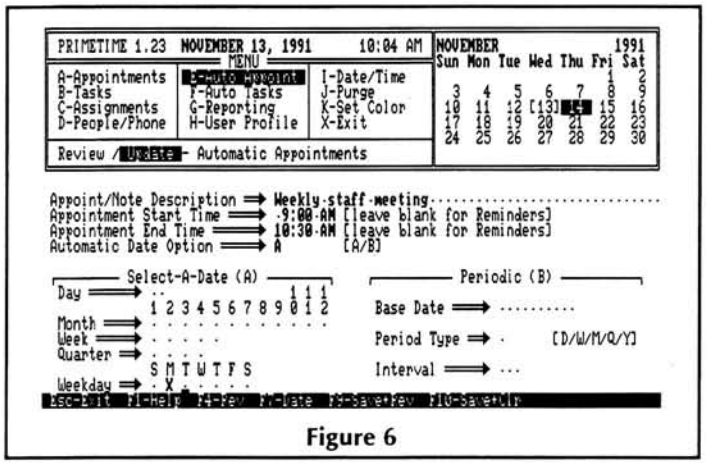


Figure 6

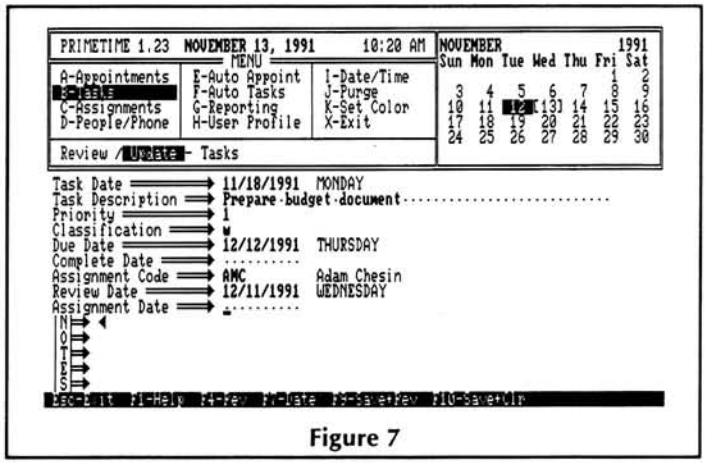


Figure 7

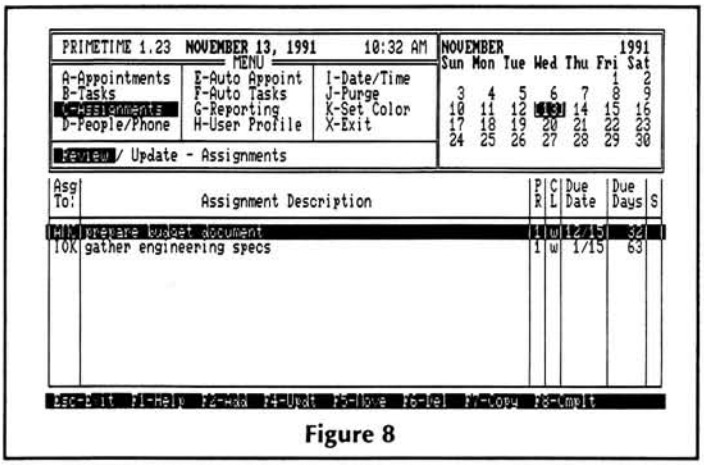


Figure 8

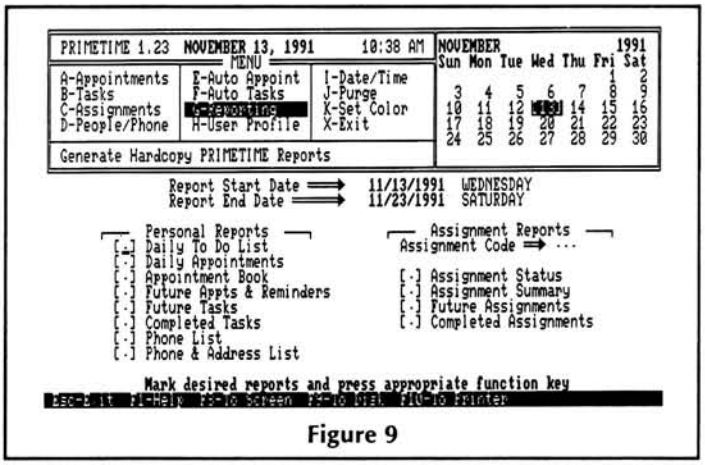


Figure 9

the secondary their office. To dial a number, you display a list of persons in your directory, highlight the name of the person you want to call, then press the F3 function key.

The personal directory also includes an assignment code for each individual. The code is a one- to three-letter abbreviation that you use to link them with tasks they have to perform.

Managing Tasks and Assignments

There's more to work than meetings and appointments, however. There are tasks that you have to perform and assignments that you've given others. You enter information about tasks and assignments in the Update Task menu, using the assignment code to link the task with the person performing it (Figure 7). When you enter the abbreviated code, the individual's full name automatically appears. You can quickly review assignments and the number of days remaining to the due date in the Assignments screen (Figure 8).

Tasks are activities that you have to perform yourself, such as phone calls to make, people to see, small jobs to perform. At the start of each day, display a task list to refresh your memory.

Reporting Progress

All of the data stored by the PIM can be displayed or printed for your review. Primetime Personal offers 12 different reports that you generate individually or in combination, either on the screen, to a disk file, or printed (Figure 9).

For example, you could print an assignment status report before a staff meeting, or a future appointments listing to see if you can schedule a well-deserved day off. If you have to leave your office, print an appointment schedule or phone list to take with you, or a list of the next day's tasks to review in the evening.

Current

IBM's entry in the PIM market is Current. As a Windows application, it takes full advantage of the graphic user interface by using visual images that represent everyday objects. For example, if you want to telephone someone, you can look up their number in a telephone book, as illustrated in Figure 10. The darker tabs indicate that persons are listed on those pages — click the mouse on the tab to display the page.

Current also allows you to include graphic images in your information files. Figure 11, for example, shows a Current window describing a meeting location. The layout was drawn using a painting program, then copied to Current using the Window's clipboard.

Current classifies your information into categories, or types of information. The built-in categories are appointments, companies, conference rooms, letters, notes, persons, phone calls, projects, tasks, to do's, and travel expenses.

Specific instances of categories are called items and are defined in item details. The item detail in Figure 11 is one instance from the conference room category.

You link items of different categories using connections. The diamond symbol at the Reserved for Appointment prompt in Figure 11 indicates a connection, in this case with one or more appointments. The first appointment is shown at the prompt. To see details of this or other appointments schedules for the room, click the filled-in half of the diamond with the mouse. A filled-in half means that there are additional items connected that are not shown in the item detail.

As an example of how Current works, let's review how you might keep track of a major project, starting with outlining its steps.

When you start Current, the appointment calendar for the day appears on the screen (Figure 12). You can scroll through the day, week, month, or year by clicking the mouse on one of the arrows. The time bar to the left of the daily schedule shows the entire day at a glance. In this case, there are scheduled appointments only between 10:30 and 1:30--the rest of the day is free.

Outline the Project

Using an Outline function, you break

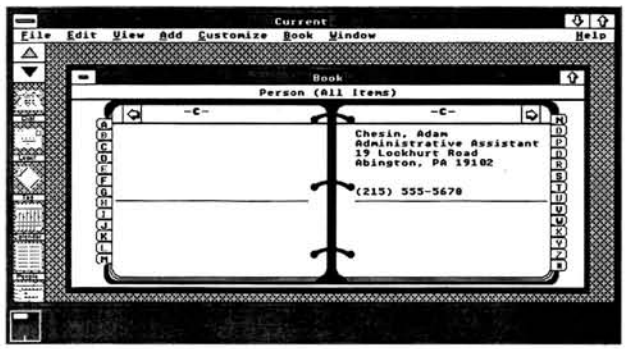


Figure 10

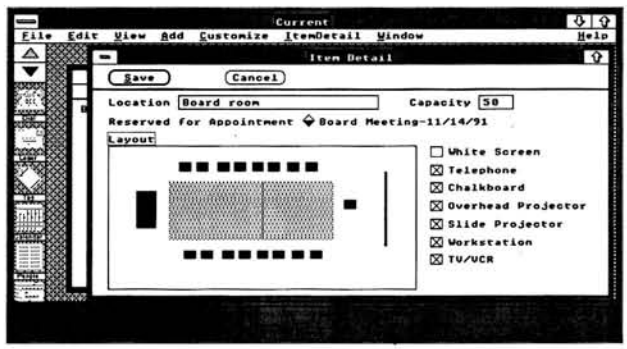


Figure 11

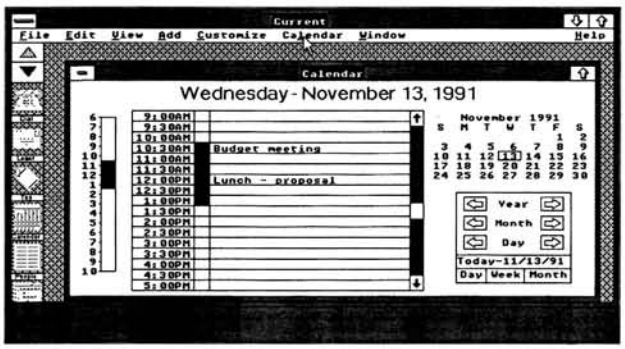


Figure 12

Figure 13

down the project into its major tasks and the persons assigned to each (Figure 13). Even though person names are stored last-name first in the phone directory, you can enter them first-name first elsewhere. Current will automatically locate the proper person no matter what order you enter it in later searches.

From the outline you describe the individual tasks and, if necessary, the individuals given the assignment. Current has an Outline Enter function that takes you step-by-step through creating item details for each task and individual not already in the system. As part of the description, you add connections to item details in other categories. The item detail for the task Distribute Requests, for example, shows a link to Adam Chesin and the project Network Purchase (Figure 14). The connection will also be shown in the individual's personal item detail.

Charting

Once the tasks and assignments have been entered, you can review the overall project by displaying a Gantt Chart (Figure 15). A Gantt Chart illustrates the start and end times of each tasks, and how they fit in with the overall project. On a color monitor, the Gantt chart symbols are color coded as either planned, in progress, or completed.

Updating Progress

During the course of the project, you update the item details to indicate when each task is completed. Display or print progress reports, arranged appointments, or record activities as additional item details.

Current has a built-in function for logging in phone calls and linking them with projects. Add an item detail for each call, then link the subject and person with a project or task item detail.

Connections

The concept of connections is very important to PIMs. Primetime Personal connects persons to tasks using the assignment code. Using the code, you can track an individual's progress or view their overall workload and efficiency.

Current includes a number of built-in connections. For instance, the project Network Purchase has four basic connections. It is associated with companies, assigned to persons, includes tasks, and incurs expenses.

You can customize Current, however, to suit your own needs and tastes. Figure 16, for instance, illustrates a custom-made category named authorization that includes personal information and the scanned image of a signature. Banks or other companies can keep a file of authorizations available to check signatures against the scanned version.

In addition to the category, two new connections were created. The connection Checked by Person in the authorization category records the name of the employee who obtained the initial signature. The employee can easily be identified and consulted if a signature mismatch occurs. A corresponding connection- -Authorization Checked- -has been added to the persons category. Use that connection to see what signatures a specific employee obtained.

Resources Guide

For further information on Primetime Personal, Current, and other PIMs, contact these companies.

Chronologic Corporation
5151 N. Oracle Street
Tucson, AZ 85704
(602) 293-3100
Program: Instant Recall

Chronos Software, Inc.
555 De Haro Street
San Francisco, CA 94107
(415) 626-4244
Program: Who-What-When

Futuresoft, Inc.
1331 N. I-10 Service Road
Metairie, LA 70002
(800) 327-8296
Program: Right Hand Man

Good Software Corporation
13601 Preston Road
Dallas, TX 75240
(214) 239-6085
Program: Arriba

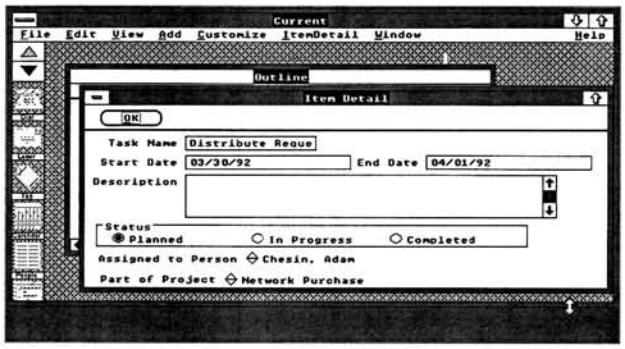


Figure 14

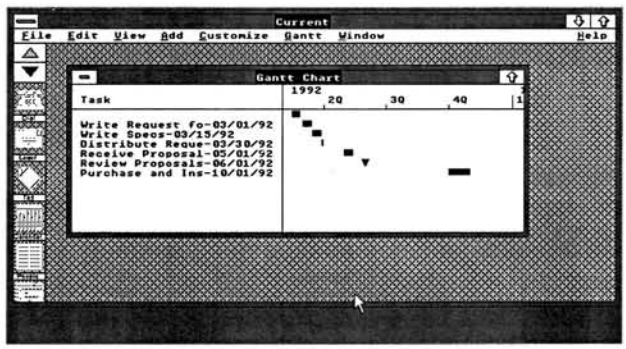


Figure 15

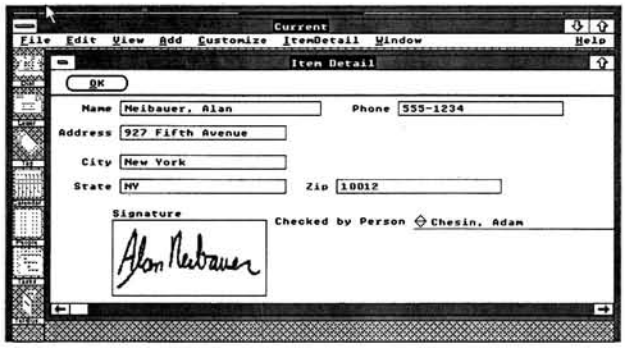


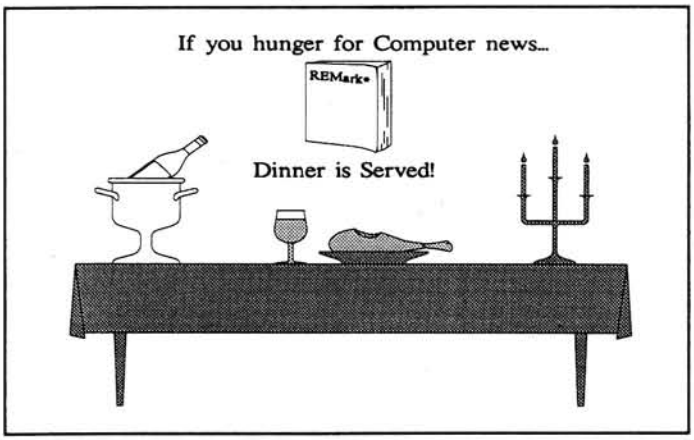
Figure 16

IBM Corporation
P. O. Box 1328-W
Boca Raton, FL 33429
(404) 956-4001
Program: Current

Polaris Software
613 West Valley Parkway
Escondido, CA 92025
(619) 743-7800
Program: Packrat

Primetime Software
P. O. Box 27967
Santa Ana, CA 92799
(714) 556-6523
Program: Primetime Personal

Richmond Software
Suite 420
6400 Roberts Street
Burnaby, British Columbia
Canada V5G 4C9
(604) 270-3311
Program: Maximizer



SEAGATE ST-251-1 42 MEG HARD DRIVES ON SALE

Seagate HARD DRIVES

MODEL	CAPACITY/FORMAT/SPEED/SIZE	DRIVE ONLY	XT KIT
* ST-125	21 MEG / MFM / 40 MS / 3.5"	$229.00	$277.00
* ST-138	32 MEG / MFM / 40 MS / 3.5"	$277.00	$325.00
* ST-138-1	32 MEG / MFM / 28 MS / 3.5"	$307.00	$355.00
* ST-151	42 MEG / MFM / 24 MS / 3.5"	$353.00	$401.00
* ST-157R	49 MEG / RLL / 40 MS / 3.5"	$286.00	$339.00
* ST-225	21 MEG / MFM / 65 MS / 5.25"	$199.00	$247.00
*** ST-250R**	**42 MEG / RLL / 70 MS / 5.25"**	**$248.00**	**$288.00**
*** ST-251-1**	**42 MEG / MFM / 28 MS / 5.25"**	**$289.00**	**$337.00**
* ST-4096	80 MEG / MFM / 28 MS / 5.25" FH	$582.00	$631.00
* ST-238R	32 MEG / RLL / 65 MS / 5.25"	$218.00	$271.00
* ST-277R-1	65 MEG / RLL / 28 MS / 5.25"	$348.00	$401.00
* ST4144R	122 MEG / RLL / 28 MS / 5.25" FH	$623.00	$671.00

* 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,	$281.00	
* ST-138/Z150 Kit	32 Meg, 40 MS,	$329.00	
*** ST-251/Z150 Kit**	**42 Meg, 28 MS,**	**$341.00**	**SALE PRICED**

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,	$352.00	
* ST-138/Z148 Kit	32 Meg, 40 MS,	$399.00	
*** ST-251/Z148 Kit**	**42 Meg, 28 MS,**	**$408.00**	**SALE PRICED**

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 $84.00 External Model $128.00 Cable for External Modem $8.50

⇨**MICROSOFT WINDOWS SOFTWARE** version 1.04 for PC (not Z-100)
⇨**FREE** with any order from this ad. Just ask for it and add $5.00 for shipping and ⇨handling. Includes 5 disks and 300+ page manual. Offer good for limited time.

WESTERN DIGITAL HARD DISK CONTROLLER CARDS

⇨WD XT GEN2+ 8 BIT, MFM, DUAL HARD DRIVES, XT COMPUTERS	$48.00
⇨WD1004-27X 8 BIT, RLL, DUAL HARD DRIVES, XT COMPUTERS	$53.00
⇨WD1006V-MM2 16 BIT, DUAL HARD, DUAL FLOPPY, 1:1 AT	$99.00
⇨WD1006V-SR2 16 BIT, RLL, DUAL HARD, DUAL FLOPPY, 1:1 AT	$109.00
⇨HARD DRIVE CABLE SETS, XT.. $3.50, AT .. $4.50, AT + Floppy	$6.50

PRINTERS

⇨STAR NX-1000 II 9 PIN, 180 CPS, 45 CPS NLQ,	$179.00
⇨SEIKOSHA SL-90 24 PIN, 240 CPS, 80 CPS NLQ, PAPER PARKING	$314.00
⇨PANASONIC 1180 9PIN, 192 CPS	$204.00
⇨PANASONIC 1124 24 PIN, 192 CPS	4335.00
⇨PANASONIC 1624 24 PIN, 192 CPS, WIDE CARRAGE	$422.00
⇨PANASONIC 1695 9 PIN, 288 CPS, WIDE CARRAGE	$458.00
⇨PRINTER CABLE	$12.00

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	$ 429.00
⇨CM1440	SEIKO VGA 1024x768 .25 dot	$ 569.00

VIDEO CARDS

⇨BASIC VGA	PARADISE AUTOSWITCH 640x480	$ 99.00
⇨VGAPLUS	PARADISE AUTOSWITCH 800x600	$ 146.00
⇨VGAPLUS16	PARADISE AUTO 16 BIT 800x600	$ 177.00
⇨VGA 1024-256k	PARADISE AUTO 1024x768	$ 194.00
⇨VGA 1024-512k	PARADISE AUTO 1024x768	$ 247.00

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	$68.00
⇨41256	256x1	100 ns	$2.65	SIM 1Mx9 80 ns	$69.00
⇨41256	256x1	120 ns	$2.30	SIM 256x9 80 ns	$24.00
⇨1Mbit	1Mx1	80 ns	$7.40	V-20 replaces 8088 CPU	$14.00

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 $207.00
⇨Dual Drive Unit $299.00 Bare drive and cable for internal mount $115.00

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

⇨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 8.25% state sales tax.

DISCOVER VISA MasterCard

PAYLOAD

15718 SYLVAN LAKE * HOUSTON TX 77062

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

PAYLOAD * PAYLOAD * PAYLOAD * PAYLOAD

Do You Want to Program?

Gil Hoellerich
2617 Country Way
Fayetteville, AR 72703

Have you used a word processor, spreadsheet, or database management system? If you have used one of these, you probably know that these were programs designed by one or more programmers. Perhaps you have been curious about how these programs were designed. If so, this article should be of interest to you.

We will design a very simple program. The first step in designing a program is to select the task that you wish the program to perform. We will select the task of printing a label.

Now we will divide the task of printing a label into three smaller tasks:

1. Print name on first line.
2. Print street address on second line.
3. Print name of city, state and zip on third line.

Next, we need to give the computer the appropriate command for that task. Unfortunately, the computer hardware doesn't understand english commands, but requires machine language. So we will not be able to give the command `print name` on the first line.

While we could write this program in machine language, we will use a high level language instead. In high-level languages, we use english-like commands which an interpreter can translate into machine language.

While there are many high-level languages, we will use BASIC because most computers are sold today with an operating system and a BASIC interpreter. So you probably have a BASIC interpreter with your software although you may not have noticed or used it.

You will need to find this BASIC interpreter. The name of the program file is either BASIC.EXE, GWBASIC.EXE, or BASICA.EXE. Look for these program files by using the DOS command:

```
DIR BASIC.EXE or
DIR BASICA.EXE or
DIR GWBASIC.EXE
```

You may not find the file in the root directory. So you may need to change to another subdirectory; probably the \DOS or \BIN subdirectory. Remember, to change to another subdirectory, you use the command cd \ followed by the name of the subdirectory. So if you want to change to the \DOS subdirectory, you would enter cd \DOS.

If you have found the BASIC interpreter and are in that subdirectory, enter the command BASIC, GWBASIC, or BASICA; choose the one which matches the name of your program file.

After a few seconds, you will see on the screen some words like this:

```
GWBASIC version 3.31
(c) Micro-soft copyright 1984,1986
 60300 bytes free
 Ok
```

The Ok is the important part of those words; this word is the prompt of the BASIC interpreter. A prompt tells the user that another command may be entered. This BASIC interpreter will allow us to enter english-like commands on the keyboard.

Now let's return to the three small tasks that we listed earlier. We need to convert these tasks to the english-like language that the BASIC interpreter can understand and we can enter on the keyboard; it will translate into machine language which the computer then understands.

In addition to converting the tasks to the english-like language of the BASIC interpreter, we must enter them in the correct sequence.

Our first small task was to print the name on the first line of the label. So, for my labels I would enter the command:

```
10 Print "Gill H. Hoellerich"
```

You need the number 10 for sequence numbering. You can use any positive number! So, you could have used 1; however, the statements or commands are usually numbered in tens. We will show the reason for this later.

Now enter the first line of your labels, but read this entire paragraph before you begin! You will use your name instead of mine. But don't forget to enter the double quotation marks. **Caution:** These are *not* two apostrophes: '' ! After you have entered the command, press the ENTER or RETURN key!! Don't worry if you made a mistake, just type the line correctly on the next line. *One exception:* If you entered a different number than 10, you will need to enter that number on the next line and press the enter key. Now enter your first line!

Our second small task was to print the street address on the second line. So the second line for my labels, I enter:

```
20 Print "2617 Country Way"
```

This is to print your street address. So use your street address in your label program.

The third line for my labels:

```
30 Print "Fayetteville, AR 72703"
```

because that is my city, state and zip. Enter your city, state and zip in a similar way. You now have a short program in the computer memory. If you want to see that program, enter (without a number) the command LIST. Don't forget to press the ENTER key. Your program will be displayed on the screen.

Now let's print your label on the screen. Enter (without a number) the command RUN. Don't forget to press the ENTER key!! Your label should be displayed on the screen!

Congratulations! You have written a small BASIC program.

Perhaps you wondered why we didn't have a number before RUN and LIST. If you want the command executed immediately, you don't use a sequence number; if you want the command executed later and placed in a program, you use a sequence number.

If you are interested, here are a few more things that you can do.

First, you will notice that the label was printed at the end of the listed program. So, let's clear the screen before printing the label. Enter:

```
5 Clear
```

Now use the command LIST and you will see that the interpreter added this command to the top of the program. (This is the reason we used tens, so we can place new commands in their correct place later.) This command clears the screen just as CLS does in DOS. Now enter RUN again. See the difference?

Perhaps the label would appear more appealing to the eye if we added a blank line between each of the lines. Enter:

```
15 Print
25 Print
```

Now enter LIST. Again, you see how the interpreter added the lines in the proper

Quality Enhancements!

EaZy PC Products

EZM-128: Expand 512K base memory to 640K. Simple, plug-in installation. $125.00

EZCLOCK: Calendar/Clock. Piggy-back add-on for EZM-128. $35.00

No Slot Clock/Calendar

FBE SmartWatch: Automatic date/time on bootup. Installs under BIOS/Monitor ROM. Ten year battery. Works with all Heath/Zenith MSDOS computers. For PC's $32.00, Z-100 $33.00 Module: $25.00

H/Z-148 Expansions

ZEX-148: Adds one full-size and one half-size expansion card slot. $79.95

ZP-148: PAL chip expands existing 640K memory to 704K. CGA/MDA only! $19.95

Configuration Control

CONFIG MASTER: Menu-select active CONFIG.SYS during bootup. Software for PC/Z-100 MSDOS. $29.95

H/Z-150 Items (Not for '157, '158, '159)

VCE-150: Eliminate video card. Install EGA or VGA card. All plug in. Includes circuit board, SRAM and RM-150. $49.95

ZP640 PLUS: PAL chip to expand standard memory card to 640/704K with 2 banks of 256K RAM chips (not included). $19.95

ULTRA-PAL: Three PAL chips: MR150 for 704K + 512K RAM Disk; MR150T for 640K + 512K RAM Disk; LIM150 for 640K + 512K (32 pages) of simulated v3.2 Lotus/Intel/Microsoft Expanded Memory. With software. Install on standard memory card. No soldering. Needs 45 256K RAM chips (not included) for maximum memory configuration. $39.95

COM3: Change existing COM2 to COM3. Put internal MODEM at COM2. Don't lose serial port. With software. $29.95

H/Z-100 Modifications

ZMF100A: Expand "old" motherboard (p/n 181-4917 or less) using 256K RAM chips (not included). No soldering. $65.00

ZRAM-205: Put 256K RAM chips on your Z-205 board. Get 256K plus 768K RAM disk. Contact us for data sheet before ordering. Without RAM chips. $39.00

H/Z-89 Add-Ons

H89PIP: Parallel printer 2 port interface card. With software. $50.00 Cable $24.00

SLOT4: Add fourth expansion slot to right-side accessory bus. $39.95

Order by mail, FAX, telephone, or see your dealer. UPS/APO/FPO shipping included. VISA/MasterCard. WA residents add 8.1% tax. Hours: M-F 9-5 PST. We return all calls left on our answering machine!

FBE

FBE Research Company, Inc.
P.O. Box 68234, Seattle, WA 98168
206-246-9815 Voice/FAX TouchTone Selectable

Reader Service #104

sequence.

Now enter RUN again. See the effect of the blank lines?

By now you are probably saying that a label printed on the screen is not very practical. And you are right! So let's change the program slightly so that we send the label to the printer.

Replace each PRINT word with LPRINT. If you need practice typing, you can retype 10, 20, and 30; if not, move the cursor to the letter P and then strike L and press Enter. It should place an L in front.

Now enter LIST again to make sure your program is correct. Then enter RUN. This should produce your label on the sheet of paper in the printer.

Suppose you want to print labels later using this program. You can save this BASIC program in a file. You will be able to give the file the name you wish if it meets the restriction for naming files in DOS. Let's call this program "label". The BASIC interpreter will add an extension of .BAS.

So enter the command:

```
SAVE "label.bas"
```

Now you are finished with the BASIC interpreter. How do you exit the BASIC interpreter? You enter this command:

```
SYSTEM
```

This should return the DOS prompt. Now use the DIR LABEL.BAS command to see if your program file is in the subdirectory.

Now you should be able to run this existing program very simply. At the DOS prompt enter the command:

`BASIC LABEL.BAS` or `BASICA LABEL.BAS`

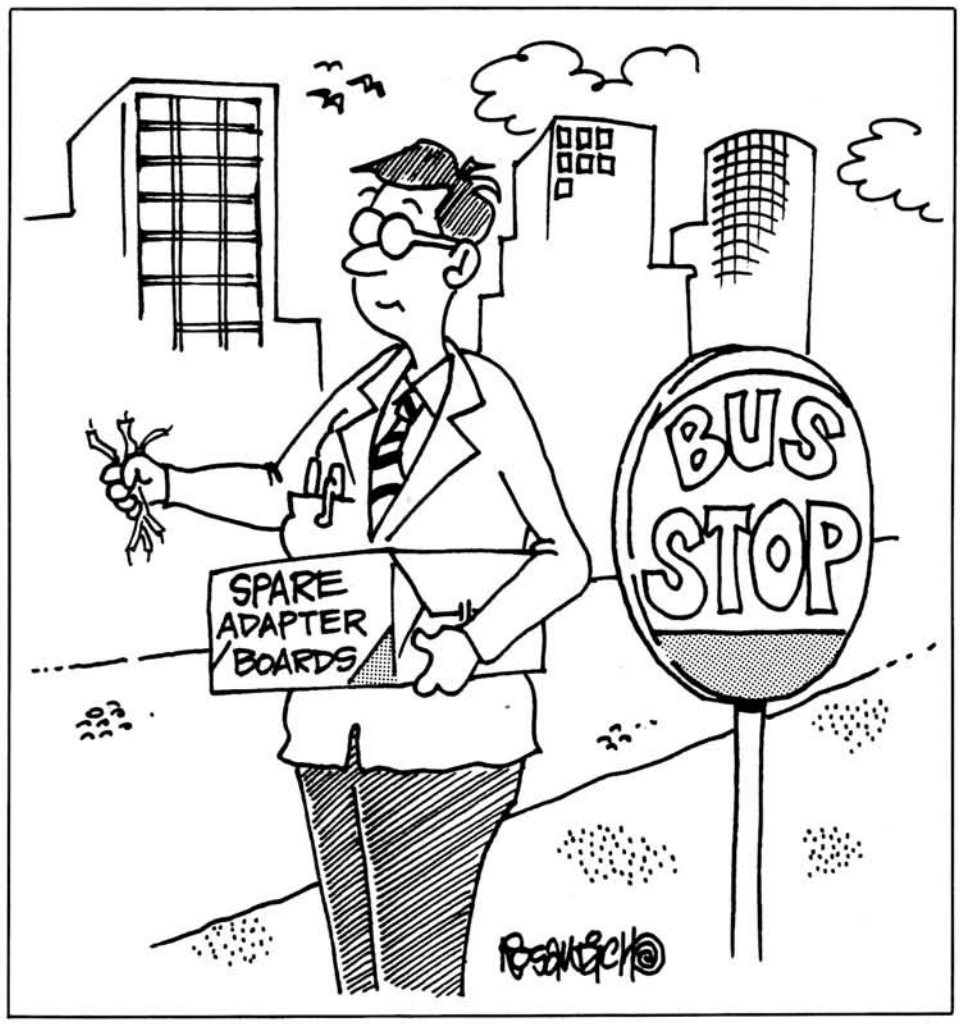
depending on whether your BASIC interpreter file was BASIC or BASICA.

Your label should be printed without any further input by you.

Summary

This article did not intend to make you an accomplished programmer. We only used one of the capabilities of the computer: Output. The computer can also input letters and numbers and then "massage" the input (change or calculate). I did intend to show you some of the things that a program can do and hope for some of you, create enough interest that you will pursue programming.

There are many languages which can be used to write programs: COBOL, RPG, FORTRAN, C. Also some programs have interpreters within the programs: WordPerfect, dBASE, AutoCAD. Each, of course, has its own set of acceptable english-like commands.



On the leading Edge

Computer Model Numbers

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

If this is your first introduction to our users' group, let me be among the first to welcome you. I have written this particular column to include specific information that is of special interest to new Zenith Users' Group members. I began writing this special column last year, and it turned out to be so popular that I will continue the idea. But before we get too far along, I think it might be a good idea to introduce myself again for the benefit of our new members.

An Introduction

Regardless of what my byline says, my friends call me Bill, and I would be pleased if you would do so. I have over 23 years' experience in various computer-related and data processing functions ranging from systems analysis and design to management. During this time, I have worked for various companies like Honeywell, McDonnell Douglas, TransAmerica, and Atlantic Richfield Company (ARCO). My educational background includes a B.S.E.E. (Electrical Engineering) degree from Purdue University and an M.B.A. from West Coast University in Los Angeles, California. And I have written over a dozen books or courses as well as several hundred computer-related articles of one kind or another.

I am currently a Senior Consultant with TAP, Incorporated (Total Assets Protection) in Arlington, Texas. TAP provides just about all kinds of services for computer systems ranging from the design/build of major data centers to communications to the analysis and development of information security and disaster recovery programs to protect data. My personal specialty is the analysis and development of information security and disaster recovery programs for large mainframe data centers. As a result of this consulting experience, I have also worked with all major brands of microcomputers and software as well as a number of Local Area Networks (LANs) and Wide Area Networks (WANs). This has given me the opportunity to use and compare a lot of computer systems from major manufacturers (including Heath/Zenith, IBM, Compaq, and Tandy), add-on boards, and a wide array of word processing, spreadsheet, and data base software. Regardless of what hardware I have used, I still think that the Heath/Zenith systems are the best, and those are the ones that I personally chose to buy and write about.

As you will see on the masthead, I am a Contributing Editor and have been writing this column for *REMark* since 1983. I have been a ZUG member since 1982. During this time, I have seen a major shift in the ZUG membership from primarily hobbyist-oriented interests to the business-related use of computers. Many new ZUG members are not necessarily interested in all of the technical details of their computers, but at least some knowledge is required to successfully operate and use these systems. You can find nearly all of the information you will need to begin using your computer in the documentation provided with your computer and the *Powering Up* books available from ZUG.

What's the Purpose of ZUG?

The Zenith User's Group, usually called ZUG, was established over 12 years ago. As published in each and every *REMark*, ZUG is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their use of Heath/Zenith equipment. In other words, the whole concept of ZUG is to help members exchange ideas on the use of Heath and Zenith Data Systems computers. And the whole idea of *REMark* is to provide a medium where members can write about, and exchange information on, various topics that can help other ZUG members.

Articles published in *REMark* are written by ZUG members for ZUG members. If a ZUG member (including the ZUG staff) does not write about it, then there is no way you will see it in this magazine. If you have a specific question or suggestion for a topic that you need additional information on, I suggest you write to ZUG or directly to me with that idea. Most of the topics I write about are a direct result of questions or suggestions in the letters I receive. Or, if you have an idea for an article that would be of interest to other ZUG members, consider the idea of writing about it yourself. Information about compensation for published articles is included elsewhere in this issue.

There are occasional suggestions that articles in *REMark* should have more of "this" and less of "that". For example, I received one letter that suggested that information about all ROM releases for ALL Heath and ZDS computers should be published in this magazine. On the surface, that sounds like a good idea. In fact, it is such a good idea that I checked it out several years ago with the idea of writing about it. Unfortunately, there is no one person who knows everything about all ROMs for all ZDS computers (including discontinued models). And wading through hundreds of pages of technical information to get one little tidbit of important information is not really my idea of a good time. Besides, the real answer to this suggestion is that NO major manufacturer – not ZDS, not IBM, and not Compaq – is willing to generally release this kind of highly confidential and proprietary information about changes made in each ROM release. For ZUG members, there is another way to find this kind information.

The Purpose of this Column

Since it began, the purpose of this column has always been to help ZUG members keep "On the Leading Edge" for information about Heath and ZDS computers. That includes various topics, such as new ZDS or Heath hardware, operating systems, and third-party products which can be used on these systems. When I find out about new items of general interest, such

as ROM changes that may be required to use new software, that information has been and will be published here. When a new release of ZDS MS-DOS is available, I include general information about it to help you decide if you should upgrade or not. Sometimes it takes a little time for me to explore new features so that I can provide you with ideas on how to intelligently use them. In other cases, I may not use a new feature on a regular basis, and I may not write about it unless I receive a letter which indicates that the subject may be of general interest to all ZUG members.

Some members have suggested a continuing series of articles that includes specific information about ZDS computers and MS-DOS. In short, the purpose of this column is exactly that. Now let's move on to some other topics of particular interest to new ZUG members.

Compatibility

The most common question that we receive at ZUG has to do with the various model numbers of Heath and ZDS PC compatible computers. Perhaps the most common question of all is something like: "Why don't you have any articles about *MY* computer?" That kind of question is either preceded or followed by a statement about which model that user has. Sometimes the question is phrased in such a way that it is clear the user does not know what the general "type" (e.g., PC compatible) category the system falls into. Before we get into specific model and series numbers, it is important to define what we are talking about when the term "PC Compatible" is used.

IBM Compatibility

What the term "PC Compatible" means is really determined by the context in which it is used. As used by ZUG and in *REMark*, the term "PC Compatible" refers to *SOFTWARE* compatibility. That is, software generally listed as "PC Compatible" will run on Heath and ZDS computers as well as other compatibles, such as IBM, Compaq, etc. In other words, it is generally safe to assume that any software which will run on an IBM PC (where PC compatibility is essentially defined) will also run on a ZDS or Heath PC compatible computer. For purposes of HUG software, the term "PC Compatible" means that the software will run on any Heath or ZDS PC compatible computer, and most other compatible systems too. If you read this paragraph carefully, you will note that I said "generally safe to assume", which of course means that there are some exceptions. To understand why that is true, it is necessary to review a brief bit of history on the development and "compatibility" of IBM computers.

The first IBM microcomputer was called the IBM Personal Computer, or IBM PC for short. The original IBM PC featured a whopping 64 kilobytes of memory (later increased to 128 K), SINGLE-sided/double density 5.25-inch disk drives (180 K) that were later changed to double sided/double density (360 K), a cassette tape interface, a single parallel port for a printer (a serial port was optional at extra cost), an 8088 CPU running at 4.77 MHz, and a 65-watt power supply. Most early PCs were equipped with a CGA video card and monitor. In the hardware sense, true PC compatibility was essentially defined as this kind of configuration. Because of several interesting problems with this specific configuration, IBM introduced the IBM XT computer.

The XT (eXtended Technology) was essentially a "reworked" PC with one very important difference: it had a larger power supply. As I recall, it was rated at something around 90 watts or so. A beefed-up power supply was required because the other significant difference in the XT was that it included a hard drive, usually on the order of 5-10 MB. In other words, the PC was so underpowered that it did not have a big enough power supply to support internal electronics (including memory), two floppy drives, *AND* a hard drive; otherwise the technical specifications were about the same, including the 8088 CPU and the 4.77 MHz clock speed. Through the normal development process, IBM introduced the AT.

The original AT (Advanced Technology) contained an 80286 CPU and ran at a screaming 6 MHz clock speed with one wait state. This AT came equipped with a whopping 512 KB of system memory and a 5.25-inch high density (1.2 MB) floppy disk drive. This was the "standard" configuration, but most ATs were also equipped with a hard drive. Later, another AT was released, and it had a clock speed of 8 MHz with one wait state. These two computers essentially define what "AT compatible" really means; however, advances in this technology caused all kinds of problems. And general software and hardware compatibility with other manufacturer's systems were the biggest.

Copy Protection and Compatibility

In the early days of computer software, many paranoid software manufacturers developed software with a copy-protection feature. At least one form of copy protection checked a computer's clock speed, and many irate users found out the hard way that true PC compatible software would not run on their new AT systems because of the increased clock speed. In particular, many games checked to see if the computer was running at 4.77 MHz, and if it wasn't, the game program would not run. Many users viewed this approach as a rather shabby marketing technique to force them to buy new software. And many software manufacturers at the time would not even give a legally registered software user a discount for an update to a new version. This particular problem is one reason that ZDS introduced the dual-speed (4.77 MHz and 8 MHz) Z-158 computer system.

Although some copy-protected software still exists today, most enlightened software manufacturers have removed it. Lest you believe they did that because of a user rebellion against it (which may be a small part of the reason), most manufacturers dropped copy protection for the very good business reason that it was causing incredible problems (especially installation) for their technical support groups. In other words, copy protection was causing lots of problems and costing them lots of MONEY — so much so that it was more cost effective to remove copy protection than to support it. Some software manufacturers might have you believe that they removed copy protection for altruistic reasons, but I am more than a little skeptical of those claims.

There are many other forms of copy protection — some of them use one or more "hidden" files (from the DIR command), a few actually change the "format" of a disk, and one form of copy protection actually has a hole punched in a disk (usually by a laser) that prevents it from being copied by DOS. Another form of copy protection requires a "key disk" that must be inserted in a floppy drive to run a program, even if the software is installed on a hard drive. The latest form of copy protection is in the form of a "hardware key" —called a DONGLE— (yes, that's really the name of it) that plugs into a serial or parallel port and is checked by the software during operation. A dongle looks like, and is about the size of, a gender changer for a serial or parallel port. It usually has a DB-25 connector on both ends, so that you can plug it into the computer and then plug in a peripheral, such as a printer or modem, directly into the dongle. It is claimed that the dongle does not interfere with any peripheral in any way.

Hardware Compatibility

A discussion of hardware compatibility is difficult because it means different things to different people. Unfortunately, many people have the mistaken idea that a "compatible" computer from any manufacturer, including Zenith Data Systems, is 100% compatible for both hardware and software. That is very often not true, because of some things that hardware and software manufacturers do. Speed-sensitive software that was mentioned earlier is one of those problems, but there are two other common problems.

The first problem is that some software is not compatible with other software, regardless of the general hardware compati-

bility features of a system. Memory-resident programs, like device drivers and many desktop utility programs (e.g., SideKick), sometimes have conflicts with each other. In many cases, it is at least difficult, if not impossible, to try to predict what software will produce conflicts unless it is actually tested. Some manufacturers still use non-standard programming techniques that are used to improve speed, functionality, or both. Windows 3 is an example of that.

The second problem that is directly related to hardware compatibility is that some manufacturers, including ZDS, have built better systems than IBM. I think the best example of that is when ZDS introduced the AT-compatible 6 MHz Z-241, which could not use some of the memory boards developed for the IBM AT. As it turned out, ZDS used better engineering to produce a computer with zero wait states (memory) which was demonstrably faster than its IBM counterpart. Unfortunately, many of the early memory boards were developed using the IBM "standard" of one wait state, and these memory boards were simply not "fast" enough to keep up with a ZDS computer using a zero-wait-state configuration. If you are interested in knowing more about a computer's memory, the chapter on "Adding More Memory to Your Computer" in *Powering Up* goes into considerably more detail on the information you must know before you buy anything.

The issue of hardware compatibility goes even deeper. In the true hardware sense, I don't think ZDS has ever produced a "real" *PC compatible* desktop computer. The first PC compatible, called the Z-151, was actually an XT compatible because its larger power supply COULD support a hard drive, which is really the major difference in hardware between a PC compatible and an XT compatible. That's why you sometimes see the term "PC/XT compatible" when you look at software.

As time goes on, you will probably continue to see the term "AT compatible" which usually refers to a computer with an 80286 Central Processor Unit (CPU). In many cases, 80386 and even the 80486 systems are also compatible, at least from a software point of view. Whether or not a system is really AT compatible depends on how the definition is established by a manufacturer. In general, though, "AT compatible" means that SOFTWARE that runs on an IBM AT will run on an 80286-based system as well as an 80386 and 80486 system. If you examine most of today's software, you will find that it is clearly labeled as being "PC, XT, and AT compatible." Of course, you must be careful in reading the label because a lot of current software now requires a high-resolution video display (e.g., EGA or VGA), so the fact that it is generally compatible with the non-video system hardware is not enough. Some software requires a minimum of an EGA video system, so be sure to check that before you buy anything. If you need to know more about your computer's display system, you may find the *Powering Up* chapter on "Understanding Video Hardware" helpful.

Advances in technology have produced the 80386 CPU and the 80486 CPU. From a software perspective, Zenith Data Systems computers (e.g., the Z-386/16, the Z-386/25, and the Z-386/33) are still AT compatible. From a hardware perspective, the Z-386 systems are *generally* AT compatible, except for the fact that at least one system (i.e., the Z-386/16) must currently use ZDS 32-bit memory cards for memory expansion because of the proprietary bus. That brings up one other point.

What type of bus do you have in your computer. A *BUS* can be defined as the set of electrical connectors on the backplane or motherboard inside the computer. Simply stated, the bus contains the expansion "slot" connectors that are used for the boards you can plug into the computer, such as memory boards, video boards, and so on. In general, you will find four types of bus architectures.

The first type is the standard PC/XT and AT buses which are commonly known as ISA, which means Industry Standard Architecture. The second type is a modified ISA bus, which is generally proprietary to a specific manufacturer, and was required to support the 80386 and later systems because the ISA standard could not support those CPUs. This bus was developed independently by several manufacturers, including Zenith Data Systems and Compaq. This particular bus is used in the Z-386/16 and other similar computers.

Because the ISA bus could not support 80386 CPUs, a new bus or at least modified standard was required. The third bus architecture was created by IBM for the new PS/2 series computers, and it is called the Micro Channel Architecture or MCA bus. At the time IBM developed this bus, they decided to get REAL finicky about licensing, royalties, patents, and copyrights associated with this bus. As a result, many manufacturers decided NOT to use this IBM standard bus, and a group of nine major computer manufacturers created a standard bus called the Extended Industry Standard Architecture or EISA. The EISA bus systems are the fourth and last type.

The real distinction between the MCA bus (in the IBM PS/2 series) and the EISA bus is generally important only when one is looking at add-on boards for the system. Many boards developed for the ISA standard will work just fine on an EISA bus because there are still ISA slots available. That is also true of virtually all manufacturers modified ISA bus designs. My research indicates that is NOT generally true for the MCA bus in the IBM PS/2 series because the system was developed for an entirely different design layout. MCA boards, for example, will simply not plug into a slot in an ISA or EISA bus. I should also note that you will occasionally see software labeled as PS/2 compatible, but it will generally say that it is PC/XT/AT compatible as well. If software is ONLY PS/2 compatible, be sure to check that it will run on your system by asking someone.

The whole point of the discussion so far is that you really must have some kind of technical knowledge about your computer system. As you can see, the whole issue of compatibility gets very involved very quickly, and it frequently depends on whether you are talking about hardware or software. Hardware compatibility most often depends on the type of CPU a computer has, the type of bus used, the resolution of its video display system, and sometimes its clock speed and number of wait states. In some cases, it also depends on exactly which Zenith Data Systems computer model you have in terms of how much built-in expandability the system has. For example, the eaZy PC (all models) hardware configuration cannot be upgraded by using the usual add-on cards, such as memory or video, because it was designed to be a low-cost system. More on that in a minute.

Heath/Zenith Computer Classes

There are at least three things that all computer users should know about their systems: the CPU, the type of bus, and the type of video display. In some cases, knowing about the system clock speed can also be critical. However, knowledge of the exact CPU used in your computer is becoming more and more critical when selecting software, and the discussion of Heath and ZDS computer models will be based on the type of CPU that each system has. Today, there are four classes of Zenith Data Systems computers: the 8088, the 80286, the 80386, and the 80486. To find out which class of computer you have, all you need to do is look at the "Specifications Page" in your Owner's Manual. Let's begin with the 8088-based computer models which will generally be discussed in the order of release dates.

8088-based Computer Models

The very first compatible computer released by ZDS was called the Z-151 which was basically an XT compatible running at 4.77 MHz. In addition, ZDS also released the Z-161, which was a "portable" (I call it "luggable" because it was heavy) version of the Z-151, except that it had a self-contained CRT.

When a faster 8088 CPU was available, ZDS released the Z-158, which looked the same as the Z-151, except that it contained a switch that allowed a user to choose

between the 4.77 MHz and 8 MHz clock speeds. As far as I know, ZDS was the first major manufacturer to release a production system that used the fast clock speed that is now commonly known as a "turbo" system. All of these systems were generally expandable and had at least four slots for add-on cards. A Z-159 was also released, and all of these I have seen have featured a 101-key keyboard and EGA video support. One model that received little attention was the Z-157, which was limited in the way it could be expanded (using a daughter board), and this system was later placed in a different "box" and called the Z-148. All of the Z-15x computers are generally known as the Z-150 series computers, even though the Z-157 was quite different in its internal hardware.

The Z-148 is a low-cost system with very limited expandability. Although it will accept some add-on cards (half size), it requires the purchase of an additional board, called a daughter board, to add a card. The traditional use of this expansion capability was to add a hard drive and controller. There were several different "versions" of this computer produced (some with floppies only and one with a hard drive), but it is generally referred to as the Z-148 or the Z-140 series. The Z-138 portable is essentially the Z-148 system hardware in a luggable box, and its expansion capabilities are also limited. This computer is sometimes discussed as the Z-130 series.

The Z-171, sometimes called the Z-170 series, was a "lunchbox-style" computer that was one of the first of the true portables, weighing in at about 15 pounds. It was not really a "laptop", unless your lap just happens to fit its shape.

ZDS has developed a number of high-quality and highly-rated laptops that are generally known as the 180 series. These include the Z-181, Z-183, and Z-184. The model number generally defines what features were available on the system that was primarily related to whether the system had a hard drive. In the model number series, the Z-184 actually refers to the SupersPort computers, and there are two different SupersPort models: the original SupersPort (with an 8088 CPU) and the SupersPort 286 (with an 80286 CPU). This is one example of where the actual series includes two different classes of computers.

Small, light computers are the current rage, but a few people may remember the ZP-150 laptop that was released a number of years ago. Its ROM contained some programs, like word processing (and of course MS-DOS), but it was not very popular, probably because its display was very difficult to read under most conditions. And of course the new ZDS MinisPort computer is the latest in the super-small and light laptops. Although I suppose it is inevitable that I will miss one, Figure 1 is intended to include all of the ZDS models. If you have one that I missed, I apologize in advance.

Before we go on, it is important to remember that these systems are generally compatible with all software which have a label of "PC or XT Compatible." Depending on the system, the level of hardware compatibility varies as previously discussed. There is one important exception that is listed in Figure 1, but does not actually have an 8088 CPU.

The eaZy PC

From the letters I receive, the eaZy PC seems to be the most controversial system that ZDS has ever released. The entire eaZy PC series (EZ-1, EZ-2, and EZ-3) was discontinued some time ago. Unfortunately, it does not have the expansion capabilities of the more expensive desktop systems, and many eaZy PC owners have only learned about that AFTER buying one. My guess is that some discount firm must have obtained a rather large inventory of the discontinued eaZy PCs because I get a lot of mail with questions about them. The worst part of this particular unit is that whoever is selling them can't help and support the buyers, probably because the seller doesn't know anything about computers either. If you have an eaZy PC, you might be interested in my comments about it in the "On the Leading Edge" column that was published in the April 1988 *REMark*. Back issues are generally available from ZUG, and I suggest you call to check on availability. But here are some things you must know about the eaZy PC, if you have one.

First, the eaZy PC contains a NEC V40 CPU, which is compatible with the 8088 CPU systems listed in Figure 1. The eaZy PC contains its own monochrome CRT (with 16 shades of gray) which provides CGA-compatible video resolution (640 x 200), but it cannot be upgraded to color because the video circuitry is part of a single board inside the computer.

The eaZy PC is completely limited in its expansion capabilities to those manufactured by ZDS: a 128 K Memory Expansion Module (to increase memory to 640 K from the original 512 K), a Serial Port Module (required to support any serial device, including a printer), and a Modem Module. You may find one or more of these accessories as a single unit, if you are able to find them at all. I recommend that you buy the Memory Module, and the Serial Port Module may be important because the eaZy PC's built-in serial-type port is designed to only support a mouse, not a printer or external modem. Other than that, you cannot expand or add to the eaZy PC in any other way. You cannot add a color monitor, nor can you add ANY disk drives beyond what was included in the original system. Now let's move on to the more advanced systems.

80286-based Computer Models

The ZDS 80286-based systems began life as the Z-200 series, and many people (including me) still refer to any system that contains an 80286 CPU as a Z-200 system. I have already mentioned the 6 MHz Z-241 and 8 MHz Z-248 as AT compatibles, but sometime after these were released, ZDS changed the model number designations to the point that they are confusing to everyone. Not to be outdone, Heath also came up with some incredibly odd model numbers to describe the same computer available in kit form. I won't even pretend that I know how to decode all of these strange numbers, but there is still an easy way for you to figure out what kind of system you have. As mentioned previously, you need to refer to the "Specifications Page" in your ZDS Owner's Manual to determine the type of CPU in your system.

Sometime around the beginning of 1988, ZDS began introducing several different "packages" containing an 80286 CPU. The first one that I found was the ZW-286-25 which was an 8 MHz 80286 system (Z-286/8) that appeared to use the same cabinet as the Z-150 series. Then, a 10 MHz "Compact AT Compatible" appeared that reminded me of the Z-148 cabinet with a different front plate and a strange model number of ZCF-2326-EY. Heath released the same computer as a kit which, for some reason, was called the HS-40. Depending on when you purchased this computer, it may be called the Z-286 LP (Low Profile) or perhaps the Z-286/10 (10

Series	Model/Description
Z-130	Z-138
Z-140	Z-148
Z-150	Z-151, 157, 158, 159
Z-160	Z-161 (usually included as the Z-150 series)
Z-170	Z-171
Z-180	Laptop. Z-181, 183, 184 (except SupersPort 286)
eaZy PC	EZ-1, EZ-2, EZ-3
ZP-150	Laptop
MinisPort	Advanced Laptop

Figure 1
8088 CPU PC Compatible Systems

MHz). All of these systems were somewhat less expensive than the large Z-241/248, and they had fewer expansion capabilities and slots too. Their biggest advantage is that they used SIMMs (Single Inline Memory Modules) for memory expansion, and problems with having to use ZDS memory boards were a thing of the past.

As 12 MHz 80286 chips became available, ZDS and Heath released the "2500 series" which was a large desktop unit. As a Heathkit, it was available as the HS-2526. As an assembled ZDS computer, it was available in various models such as the ZBO-2503-EK, ZBF-2526-EK, and the ZBF-2527-EK. This computer may also be called a Z-286 or a Z-286/12, depending on the documentation. The cabinet is virtually identical to the Z-386 system with four drive bays.

The SupersPort 286 is a popular and well-designed 80286-based laptop that runs at 12 MHz. It has CGA-compatible resolution, but it provides a high-resolution display (640 x 400) when used with its own LCD screen. The SupersPort 286e is similar, but has VGA resolution. Figure 2 summarizes the various ZDS and Heath 80286-based systems.

Now for the latest and greatest in the 80386-based and later systems.

80386-based and Later Computer Models

In about 1988, the original Z-386 was released, and it contains an 80386 CPU running at 16 MHz. Because of its zero-wait-state technology, it can still process as fast as other 80386 systems with higher clock speeds — the Compaq 386/20 for example. As more powerful and faster computers are developed, it is becoming more important to understand the relationship between the computer's clock speed and the number of memory wait states. When *Byte* magazine reviewed the Z-386 that was featured on the cover, the 16 MHz Z-386 was just a few hundredths of a second slower in processing speed than the Compaq 386 that was running at 20 MHz with one wait state. In short, the processing-speed penalty for using slower (and cheaper) memory chips becomes significant even with adding just one wait state.

In 1988, ZDS released the 25 MHz Z-386 (Z-386/25), and a 33 MHz system (Z-386-25) with an EISA bus was released in 1989. The 25 MHz version is also available as a Heathkit called the HS-3629. And of course the 12 MHz TurbosPort 386 with the page-white CGA double-scan screen was ZDS's entry into the high-powered laptop market. The SupersPort 386SX laptop was also released in 1989, and it features an 80386 SX CPU with a VGA screen. Figure 3 summarizes the ZDS 80386 and later systems.

The Z-1000, in its base configuration, contains two 80386s and may be configured to include up to six. As I recall, its base price is somewhere in the $40,000-50,000 range, but I have included it here to be complete. By the time you read this, there will probably be at least one new model that I have not listed here.

Are Model Numbers Important?

Sometimes, but not often, at least in terms of the articles in *REMark* and most other magazines that cover personal computers. How often have you seen an article about the model 99 or the model 339 computer? How about the model 30 or the model 50, or the model 70? What you do see are articles about the IBM PC, AT or the PS/2 series in general, but very few articles (unless they are hardware comparisons) will refer to the specific model number or even mention a series number. For Heath and ZDS computers, you will occasionally see model-specific articles in *REMark*, such as the ROM installation on my SupersPort 286 (November 1989), but virtually all *REMark* articles are written by people who have done something specific with a Heath or ZDS computer. For IBM computers, the model number usually doesn't matter, so long as the computer can run the software. For ZDS and Heath computers, the model number is also not that important in many cases because the computers *ARE* compatible.

I mentioned IBM model numbers specifically because I have noticed that many new ZUG members have a preoccupation with computer model numbers for some reason. I suppose that many new members expect to see a specific article about the Z-159, Z-148, eaZy PC or whatever, but I continue to be puzzled by the fact that some computer users spend a wad of money on a computer and apparently don't understand that these are compatible systems. It's true that each model has some unique features, such as the Z-386/33E with the EISA bus, and some have specific limitations (e.g., the eaZy PC), but these systems are still compatible and will still run compatible software. As I mentioned earlier, I think it's unfortunate that many new computer buyers only find out after the purchase that a computer has some kind of limitation, such as the eaZy PC. Perhaps the only good approach is to say that that's the "cost" of education and chalk it up to experience.

In short, the exact Heath or ZDS computer you have is not generally too important for most articles you will see in this and other similar magazines. The whole idea of buying a Heath/ZDS compatible computer in the first place is to have reasonable assurance that you can use a wide variety of software on it, regardless of the exact model number you have. Computers that cost less, such as the Z-148 and eaZy PC, don't have the hardware flexibility and expansion capability that the more expensive systems have, but that should not be a surprise either. The cost of computers, like cars, is based on the type and number of standard features. A Cadillac costs more than a Chevrolet, but both will still get you from here to there. A Z-386 costs more than an eaZy PC, but both will do about

Model/Series	Description
Z-241	Large, desktop unit. 6 MHz.
Z-248	Large, desktop unit. 8 MHz.
ZW-286-25	Smaller desktop unit. 8 MHz. (Z-286/8).
ZCF-2326-EY	Compact desktop unit. 10 MHz (Z-286/10 or Z-286 LP). Same as Heathkit HS-40.
2500 Series	Large desktop unit. 12 MHz (Z-286 or Z-286/12)
5100 Series	Small desktop unit. 12 MHz.
SupersPort 286	Laptop with CGA display. 12 MHz.
SupersPort 286e	Laptop with VGA display. 12 MHz.

Figure 2
80286 CPU AT (Z-200) Compatible Systems

Model/Series	Description
Z-386	Large, desktop unit. 16 MHz. (Z-386/16)
Z-386/25	Large, desktop unit. 25 MHz. (Z-386/25)
Z-386/33	Large, desktop unit. 33 MHz. (Z-386/25)
TurbosPort 386	Laptop. 12 MHz.
SupersPort 386SX	Similar to SupersPort 286, but with VGA display.
3600 series	Large 80386 desktop unit. 25 MHz. (Z-386/25).
HS-5100-X	Small desktop unit with 80386SX CPU. 16 MHz.
Z-486	Small, desktop unit with 80486 CPU.
Z-1000	Multiple 80386 CPUs.

Figure 3
80386 CPU and Later Systems

the same job even though eaZy PC is a LOT slower. For the most part, virtually all of these systems will still run compatible software, regardless of which CPU is used. All of the computers listed in this article will run ZUG software that is listed under the "PC Compatibles" or "H/Z-100 and PC Compatibles" headings.

Give Your Computer a ZUG!

Membership of course. With a Zenith Users' Group membership, you get a year's subscription to *REMark*. Articles include information that is useful to the beginning, as well as the advanced computer user ranging from "how-to" discussions (e.g., the *Powering Up* series) to programming in various languages. Some articles include specific hints and tips on how to use various application software and utilities. And this *On the Leading Edge* series includes information about new hardware, software (including new releases of ZDS MS-DOS), and other information that is specific to Heath and ZDS computers.

If you have a question about your Heath or ZDS system, I strongly encourage you to write to us. Although the ZUG staff does not have every single Heath and ZDS computer model, we can usually help solve a problem because of our experience with these systems, including ZDS MS-DOS. If you do have a question or a problem, be sure to include the information mentioned at the end of this article so that we can provide a specific answer to help you. You also get access to the COM1 bulletin board, which has megabytes of down-loadable software, a message system, and a special "Bargain Centre" area.

If you have found something that you think needs to be shared with other Heath and ZDS computer users, I also encourage you to let us know about it. It may be something that solves a problem, like a new ROM version, or it may be about a software problem or bug. At ZUG, we do not use or test every possible software and hardware combination, and there may be some little-used software (especially networking-related software) or hardware that has a conflict with something in a system. When I know about these kinds of things, they usually appear in this column with a credit (by name) to the provider of the information.

Powering Down

If this particular issue of *REMark* is your first as a new member, welcome to ZUG. I think you will find that the articles published here will help you learn about and use your Heath and ZDS computer system more effectively. To help you find out more about Heath and ZDS computer products, I also suggest that you write to Heath Company at the address shown at the end of this article for a free catalog. This may help you learn more about the different kinds of computer equipment and accessories available for your system. Although it is my usual practice to list the models and prices at the end of each article, they are not included for the general discussion of model numbers because many of the older systems mentioned have been discontinued.

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

If you have questions about anything in this column, or about ZDS 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

Powering Up (885-4604) $12.00
Heath/Zenith Computer Centers
Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (HUG Software only)

FREE Catalog — Write to:
Heath Company
Benton Harbor, MI 49022

All Checks must be made out to
Zenith Data Systems

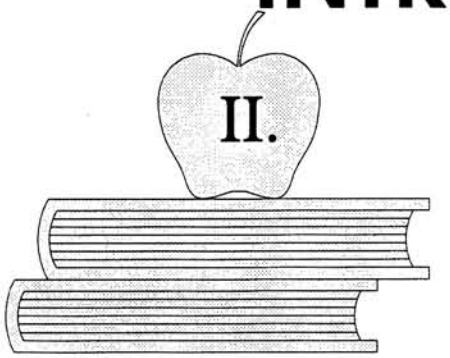
January 1, ■ 1991

Zenith Data Systems ■ 00.00

Zero dollars and zero/100

SAMPLE

INTRODUCTION TO C++



Second Installment

by
Lynwood H. Wilson
2160 James Canyon
Boulder, CO 80302

The Second Installment

In "The C Programming Language", Kernighan and Ritchie say "Our aim [in this first chapter] is to show the essential elements of the language in real programs, but without getting bogged down in details, rules, and exceptions."

That is my aim in this series as well. I will also include news and opinions and reports on relevant resources, books, software, etc. For example...

Resources

There is a new book from AT&T, the heart of C++ land. It is The Annotated C++ Reference Manual" by Margaret Ellis and Bjarne Stroustrup.

According to Bruce Eckle, in an article on C++ from the August 1990 issue of Dr. Dobbs Journal, Release 2 was intended to be the stable version of C++ from now until the ANSII C++ committee produces a standard. In the course of writing the Annotated Reference Manual, Stroustrup found a few things that he could not resist changing and so the language and the book drifted into Release 2.1. This book should be the definitive reference on the standard version of the language. At least for a while. I will report further when the book comes.

In the same article, Mr. Eckle has high praise for Borland's new Turbo C++.

If you need more background on C, I recommend "Microsoft C Programming for the PC" by Robert Lafore. Make sure you get the revised edition which conforms with ANSII C. I have used this as a text in the C classes I teach for some time, and it is also a good book for self study. After you finish it you will be ready for "The C Programming Language" by Kernighan and Ritchie.

It's useful to read some of the other magazines, particularly when you're involved with a new and changing language such as C++. You need the magazines to keep you up with changes and new products and show new ways people are using the language. The books are usually six months to a year out of date.

Dr. Dobb's Journal was originally called Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, with the subtitle "Running Light Without Overbyte". It is one of the oldest computer magazines and has been at the leading edge ever since the leading edge was their first tiny basic for microprocessors. They keep well up with developments in C++, as in Bruce Eckle's article mentioned above. Irreverent but/and good.

Computer Language Magazine is a bit more formal. They are a good source for algorithms and examples of code. They print a lot of C and C++ code. They also have an annual review of C compilers (May) and have promised a review of C++ systems for this October.

I like them both. (Remember, I promised you plenty of subjective opinion.)

And now, this month's lesson.

Output

Here is a program very similar to the one we wrote last class, but with a slightly more complex output statement.

```
// CP2.CPP  a little more output
#include <stream.hpp>

main()
{
   cout << "C++ will improve my programming ";
   cout << 30;
   cout << "%.";
}
```

The difference is that this program has several output statements. The first and third output statements just write strings to the screen, like last month's program. The second, however, lacks the quotes. This is because the constant number 30 is automatically interpreted as a number. It does not need the quotes to make it a literal. The number 30 and the character string "30" are not the same thing, but they both print the same.

You can add 7 to the number 30, but not to the character string "30".

There is a shorter and simpler way to do the same thing.

```
// MOREOUT.CPP  a little more output
#include <stream.hpp>

main()
{
   cout << "C++ will improve my programming "
<< 30 << "%.";
}
```

This produces the same output as the program above. The stream output operators (<<) may be stacked and each passes its own message to the left and then passes on anything it gets from the right.

As in C the input and output are not part of the language itself but are done by library functions.

Notice that the << operator does different things depending on the kind of data it gets. This is an example of an overloaded operator.

An overloaded operator is one which does different things depending on the data you send it. This phrase may be new to you, but you have seen the thing itself before. For instance, the plus sign, "+". If you add 3 and 4 (3 + 4) the actual work done is quite different from what happens if you add 1.23 and 4.5E3. This is the same addition from our point of view, but the actual operations performed by the computer are different, depending on whether we give it two integers or two floating point numbers. More on this later.

One more thing to add to the stream output. If you have a long string to write to the output, you can continue it on the next line. For example:

```
cout << "I expect that C++ will improve my programming by "
      "about 37%.";
```

As long as these strings are separated by nothing but white space (carriage returns, spaces, tabs) they will be concatenated and sent out as one. (I indented the

second line just for readability.) The line written to the screen will be:

```
I expect that C++ will improve my programming by about 37%.
```

You cannot, however, break the string without the quotes like this:

```
cout << "I expect that C++ will
improve my programming by about
37%.";// WRONG
```

That is because a carriage return inside a string is illegal.

> Editor's note: The rest of the listings presented in this article are restriced to the width of the column. Some lines may be broken that should not be. For example, quoted strings may be broken in conflict with the rules presented above.

If you want to send a carriage return to the screen, you use \n. The backslash makes the next character into a control char and the n control char is the newline. This is called an escape sequence, and represents a single character. Other common ones are \t for tab, \b for backspace, \" to print a double quote without ending the string, and \\ to print a backslash. For example,

```
cout << "I expect that C++ will
        improve my programming\nby "
"about 37%.";
```

will print:

```
I expect that C++ will improve my
        programming
by about 37%.
```

Variables

All variables must be declared. In C they must be declared at the head of each function, but in C++ they may be declared anywhere. Some of the C++ books suggest declaring variables just before they are used the first time. I think that this would aid readability if the variable was only used in that one spot, but if the variable is used over a lot of code it seems it would be nice to be able to look to the head of the function or block to see its declaration.

I like to include a brief comment on the meaning and use of the variable on the same line as the declaration, if the variable will be used far from the declaration.

I also recommend descriptive variable names. This is a great help toward readable programs. Most modern compilers let you use quite long variable names.

And while I am on the subject of readability, I like to leave a blank line between blocks of code which do different things. For example, a blank line between the declarations and the beginning of the program code lets the eye know that they are two different things.

```
// VARS.CPP  a little more output with
variables
#include <stream.hpp>

main()
{
   int improvement;// the improvement
        in percent

   improvement = 43;
   cout << "C++ will improve my
        programming " << improvement
        << "%.";
}
```

In this program the variable "improvement" is interpreted as its value which is 43 and sent as output to the screen just as if we had put the number 43 in the statement. The difference is that the variable can stand for many different things.

The line:

```
int improvement;
```

is the declaration of improvement as an integer variable. On the PC, integer (int) variables can represent numbers between -32768 and +32767. (This is machine dependent.)

Other data types available include:

```
short    short integer
long     long integer
float    floating point number
double   double precision floating
         point number
char     a single character, usually
         one byte
```

Assignments

The next line,

```
improvement = 43;
```

assigns the value 43 to the variable "improvement". Before the assignment the value of the variable is whatever happened to be in that portion of the computer's memory, which is to say junk. If you'd like to see, you can prevent the assignment statement from executing by making it into a comment:

```
// int improvement;// the improvement
        in percent
```

and recompiling the program.

This is a common technique for debugging in a case where you'd like to know if a particular line is causing trouble, but you don't want to remove it permanently in case it isn't.

If you do this, the value you get when you print the variable is whatever data happened to be in that memory location.

There is another way of initializing a variable. You can do it in the same statement as the declaration.

```
int improvement = 43;
```

This line serves as both declaration and initialization and is exactly the same as:

```
int improvement;
improvement = 43;
```

In general the assignment operator, the equals sign, causes the value of whatever is on the right to be assigned to whatever is on the left.

Whatever is on the left must be able to accept the assignment. It is illegal (fortunately) to do something like:

```
38 = 41;
```

because we cannot change the value of 38. This will get you an error message with reference to L- values. An L-value is anything which is legal on the left side of an assignment.

An R-value is anything which is legal on the right side, that is anything which evaluates to a value. As we will see later, almost everything in C and C++ evaluates to a numerical value.

A Bigger Program

The next program uses a bit more of the language to calculate and print a list of numbers with their squares and cubes. This is what the output should look like.

```
The square of 0 is 0 and the cube is 0
The square of 2 is 4 and the cube is 8
The square of 4 is 16 and the cube is
        64
The square of 6 is 36 and the cube is
        216
The square of 8 is 64 and the cube is
        512
The square of 10 is 100 and the cube
        is 1000
The square of 12 is 144 and the cube
        is 1728
The square of 14 is 196 and the cube
        is 2744
The square of 16 is 256 and the cube
        is 4096
```

And here is the program which produces it.

```
// Print numbers and their squares and
        cubes
#include <stream.hpp>
main()
{
   int index, lower, upper, step,
square, cube;
   lower = 0;
   upper = 16;
   step = 2;
   index = lower;
   while(index <= upper) {
      square = index * index;
      cube = square * index;
      cout << "The square of " <<
         index << " is " << square;
      cout << " and the cube is " <<
         cube << "\n";
      index = index + step;
   }
}
```

The form of the program is a single function just like the last one.

Variables

The six integer variables are all declared on one line. This type of declaration can be spread over several lines if you want to comment each variable.

```
int index, // the current number
    lower, // the lower bound
    upper, // etc.
    step,
    square,
    cube;
```

The compiler sees no difference between these two declarations. The spaces,

tabs, carriage returns and comments are ignored by the compiler. Of course, you could also declare each of the variables separately, like this:

```
int index; // the current number
int lower; // the lower bound
int upper; // etc.
int step;
int square;
int cube;
```

The next section of the code, neatly separated by blank lines, initializes three of the variables.

Why these three and not the others? Because the others are given values in the workings of the program before they are used and these three are not.

The next line assigns the value of "lower" to "index".

This could well be considered an initialization just like the three above and grouped with them instead of with the body of the code. The difference between this and the first three assignments is this: "index" is getting a starting value, but its value will change many times as the program runs, while the first three variables are being given the only values they will ever have. They are used in this program like constants. (And would be declared as constants if only we knew how.)

The While Loop

Since each line of the output is calculated and printed the same way, I did it with a loop. The while loop repeats its body, the code it controls, as long as the test, the statement in parentheses following the word "while", is true.

In this case the test is (index <= upper), so the loop will repeat as long as the value of the variable "index" is less than or equal to the value of the variable "upper".

The body of the loop is the code enclosed in curly braces and indented below the "while". The "while" expects a single statement for its body. If you have more, as in this example, wrap them in curly braces. Then they look like one statement from outside.

There are several common conventions regarding the placement of the curly braces. Here is another common way of doing it:

```
while(index <= upper)
{
   square = index * index;
   cube = square * index;
   cout << "The square of " <<
      index << " is " << square;
   cout << " and the cube is " <<
      cube << "\n";
   index = index + step;
}
```

Consistency seems more important to me than any particular style.

When the line beginning with "while" is executed, the test code is evaluated to see if it is true. In this case, the test "index upper" is true, since "index" is 0 and "upper" is 16. Since the test is true, the body of the loop is executed. Then the test is performed again. When the test is finally false, the body of the loop is not executed and control passes to the next statement after the end of the loop body.

In this program, there is no code after the body of the loop, so the program ends.

The Body of the Loop

The body of the while loop is the code that does the real work. The rest of the program is just support.

The body of the loop is indented. Generally if one or more lines are controlled by another, they are indented from it. The compiler doesn't notice, but it is a lot easier for mere humans to read.

The first statement multiplies the value of "index" by itself and assigns the result to "square". Then that value is multiplied by "index" again, and the result is assigned to "cube".

Next the square and cube are printed to the screen with some appropriate text.

Last, "index" is increased by the value of "step". Then control returns to the test at the head of the loop. If the test is true the body of the loop is repeated, else the loop (and the program in this case) ends.

The variable "index" is the control variable for the loop. It is the thing which is tested to decide whether to run the loop again or to terminate it. The code in the body of the loop must change the control variable, must move it closer to the value which will cause the test to fail, or the loop will run forever. (Referred to in some circles as an endless loop, and in others as a dynamic halt.)

Different Strokes

As we saw when we first used variables in an output statement, a variable of the right data type can replace a literal such as a number. It is also true that any expression can replace a literal if it evaluates to the same data type as the literal, and almost everything in C evaluates to something. Thus we can replace the variable in the output statement from this program with an expression which produces the value.

```
// Print numbers and their squares and
cubes
#include <stream.hpp>

main()
{
   int index, lower, upper, step;
   lower = 0;
   upper = 16;
   step = 2;
   index = lower;
   while(index <= upper) {
      cube = square * index;
      cout << "The square of " <<
         index << " is " << index *
         index;
      cout << " and the cube is " <<
         index * index * index
         << "\n";
      index = index + step;
   }
}
```

The output lines are wrapped around on the page because they are too long to fit. The compiler won't notice.

Notice that this version lacks the two variables "square" and "cube", and the values are computed right in the output statements. The output statement evaluates the data and then decides how to send it out. It doesn't care whether it is evaluating a literal or a variable or a complex expression. This principle also works in many other situations, as we will see later.

True and False

There are many other constructs in C++ which use a test similar to the one in the while loop. This test is true or false, the loop executes or ends, based on whether the expression inside the parentheses is true or false. But what is truth? There is no Boolean data type in C++, so it uses this simple scheme: 0 is false, all else is true. For instance, this while loop will never execute:

```
while(0)
   cout << "You'll never see this.";
```

This one, however, will run forever:

```
while(1)
   cout << "You will see this 'til it
        bores you.\n";
```

and this is one of the common ways to write an endless loop (which is sometimes useful, as irritating as one can be when you don't want it).

If the test is based on zero and nonzero, what about the test in our program above, which didn't seem to have numbers in it at all? It turns out that the relational operators such as "<" or ">=" evaluate to a zero if the relationship described is true and a 1 if it is false. This statement will print a 1 to the screen:

```
cout << (3 <= 7);
```

and this one will print a 0:

```
cout << (4 == 8);
```

Here are the six relational operators used in C++.

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

Note that the test for equality is TWO equal signs and the assignment operator is one. It is easy to forget and use one equal sign when you mean the equality test, especially if you are familiar with Pascal or BASIC, and the compiler cannot help you. For example,

```
while(x = y)
   do something
```

will run as long as y is not 0. Most expressions return a value, and the value returned by an assignment expression is the value being assigned.

```
int x, y, z;
x = 7;
y = (z = x);
```

The variable y now has the value 7 since the expression (z = x) evaluates to 7 (because y equals 7 and that is the value being assigned). The parentheses are not necessary, I added them for clarity.

When Dennis Ritchie was writing C he examined a large amount of code and found that the assignment operation occurred much more often than the test for equality. So he made the assignment operator one character and the test for equality two, or so the legend goes. It makes a good way to remember which is which, and a pretty good philosophy for building a language, too.

The ! indicates negation when used with the equal sign, as in != for not equal. In general it negates whatever follows it. If the expression following is 0, the ! interprets it as 1. If the expression is non-0, the ! interprets it as 0.

```
(5 < 3)   \\ evaluates to 0
!(5 < 3)  \\ evaluates to 1
```

The ! does not change the value of the thing following it, it only changes the way it looks from outside. For example, if x has the value 11, !x has the value 0 but x still has the value 11.

These logical tests can be made on numbers as well as the relational operators. Remember, 0 is false, all else is true.

```
// Print numbers and their squares
        #include <stream.hpp>
main()
{
   int i = 7;
   while(i) {
      cout << i << " squared is " << i
         * i << "\n";
      i = i - 1;
   }
}
```

This program will print the numbers 7 through 1 with their squares and then end when i equals 0.

This is a very common idiom in C and C++. Many loops are organized so the loop control variable counts down rather than up and terminates the loop on 0. This often produces faster code since testing a var for 0 is usually faster than comparing it with another value.

We've been over enough of the language that you can do some interesting experimenting between now and next month. You will find that you can pick up new parts of the language easily on your own because they will fit with what you already know, and will seem to work the way you expect. C has an underlying unity and coherence which I think comes from being written by one man.

Sources

The Annotated C++ Reference Manual
Bjarne Stroustrup and Margaret Ellis
Addison-Wesley, 1990.

Microsoft C Programming for the PC
Robert Lafore
Howard W. Sams, 1989

Turbo C++	$199.95
Turbo C++ Professional	$299.95

(Includes debugger, profiler and assembler.)

Borland International Inc.
P.O. Box 660001
Scotts Valley, CA
95067-0001
1-800-331-0877

New and Confused Subscribers

More often than not, I receive many phone calls requesting the same information, concerning everything from subscriptions, to the ZLink-COM1 Bulletin Board. This is an effort to console ALL, including those who never had the time to ask some of these simple questions.

I get many requests for technical information on every computer model Zenith Data Systems ever manufactured. I am happy to help if I'm able, but my knowledge is extremely limited. The rest of the Staff at REMark, have so many other duties to perform, that it prevents them from being "Tech Consultation". Zenith Data Systems, at this time, does not have a Technical Consultation group, thus any Zenith Authorized Service Center or the ZLink-COM1 Bulletin Board is the answer! You may find your nearest service center, through 1-800-877-7704 (in the U.S. only).

The secretary's duties are various;

- Including taking phone orders (616) 982-3463 for new and renewal subscriptions and software orders with VISA, MASTERCARD, and AMEX. (software and subscription sales may also be placed through the ZLink-COM1 BBS with 20% discounts).
- Intercepting questions about the bulletin board, sign on's, configurations and capabilities. I don't have information on when the next sale is and if an item is in stock and of course, I won't give that kind of information out to callers. As a rule, you are all on your own to check the bulletin board daily.

To back up a little and expand on the new subscription, once I've received a method of payment, it generally takes 10 working days to process the order and update the database. One year from that date is your expiration date.

Since we have moved over to the Zenith Data Systems side of business, Heath Company no longer offers any discount on computer purchases, thus, we no longer print ID cards. The information about the ID number is on the mailing label of your magazine. A welcome letter for new subscribers will follow with a comformation of your address and ID#. Once you have received your ID#, at that time you may sign on to the ZLink-COM1 Bulletin Board. The first time you sign on to the board, it will ask for your first and last name, and the ID#. After that you'll need to start a new account and answer a few simple questions, (24 hours later you will have full access).

For those new and present subscribers who do not have access to a modem, all transactions may be placed on the phone, or you may mail in a check or money order, again for the subscriptions or software available in the magazine. All checks and money orders must be made out to Zenith Data Systems, and sent to our P.O. Box 217 in Benton Harbor, MI 49022.

I have tried to think of any possible questions you may or may not have, if I have failed to answer your questions, please feel free to call and ask.

One last note, I'm sure you are all well aware of the delay in the magazine, the only reason is that we are short handed, and hopefully, this is soon to be corrected! Thanks again for your patience. The Staff at REMark hope you will enjoy the magazines now, and in the future.

Best Wishes for the new year,

Lisa Cobb,
Secretary-REMark®

Assembly Language

Part 9
Inputting from the Keyboard

Pat Swayne
ZUG Software Engineer

Those of you who have been following my series on Assembly Language have probably wondered where I have been. Well, we have some new equipment here at REMark headquarters, and I have been learning how to use Ventura Publisher (did this article on it), among other things. But I guess it's time to settle down and start writing these things again.

Check Your Homework

In Part 8 of this series, I presented a sample program that prints HELLO THERE on the screen when you run it. I gave you a homework assignment, which was to modify the program so that every other letter in the message would blink when it was run. Figure 1 shows a version of the program which does that.

This program is the same as the last one except that the attribute code for every other character in the message has 128 (80 hex) added to it. This has the effect of setting the blinking bit (bit 7) for those characters.

Keyboard Input Methods

As with screen output, there are different ways to input characters from the keyboard. At the MS-DOS level, there are two functions that read an entire line of characters at a time, and four functions that read a single character at a time. Of the two methods for reading an entire line, the easiest to work with (or so it seems at first) is also the newest. It was added to MS-DOS with version 2, and it uses something called the "standard input device". It is an MS-DOS function (function 3F hex) that can also be used to read disk files, as you will see later in this series).

In Figure 2 is a program using function 3F similar to one that I used in my old 8-bit assembly language series. This program asks you to type in your name, and then prints it out 10 times.

The first part of this program uses DOS function 9 to print a prompt for your name. Then it uses function 3F (hex) to read what you type at the keyboard. This is a general function that can be used to read from any "device". To indicate to DOS that the program wants to read from the keyboard, it loads a 0 into the BX register, which is the "handle" number of the keyboard. We will discuss handles in more detail later. The program loads 40 into the CX register to indicate the maximum number of characters that will be accepted. It loads the address of where to store the characters in the DX register.

Normally, when function 3F is used for keyboard input, it accepts characters from the keyboard until Return is pressed. It stores all of the characters typed plus carriage return and line feed characters in the designated buffer and then returns control to the calling program. Upon return, the AX register will contain the total number of characters stored, including the return and line feed. However, if the user types the same or more than the number of characters specified in the CX register, only a number of characters equal to the

```
        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             ;CALL BIOS ROUTINE
        LOOP    PMLOOP          ;LOOP UNTIL DONE
        INT     20H             ;EXIT TO DOS

MSG     DB      'H',137,'E',10,'L',139,'L',12,'O',141
        DB      ' ',13,'T',142,'H',15,'E',142,'R',13
        DB      'E',140,13,0,10,0
MSGSIZ  EQU     $-MSG

CODE    ENDS
        END     START
```

Figure 1. The homework program.

CX count are stored, and no return or line feed are stored. If the user types one fewer characters than the CX register count, those characters plus a carriage return are stored, but no line feed character. In both of these last two cases, the AX register will contain the count from the CX register.

Because of these "quirks" with function 3F, the sample program will not work properly if more than 40 characters are typed in. To correct this problem, add the lines in Figure 3 before the line with the comment TERMINATE NAME.

This modification checks the count in the AX register, and if it is the same as the limit supplied in the CX register (40 in this case), the program will supply a line feed if the last character is a return, or else both a return and line feed.

Type in the sample program with the additional lines (call it NAME10.ASM), and assemble, link, and EXE2BIN it to get NAME10.COM. To see how the program works under normal conditions, run it and type in a name of 38 characters or less. One of the nice things about using function 3F for input is that DOS command line editing is supported. To illustrate this, run the program normally as instructed above. Then press F3 to recall your previous command line entry (NAME10), press Return, and when the program prompts for your name, press F3 again. You will see the name you typed previously appear.

Now run the program again, but this time enter a string of about 50 characters for "your name". You will notice that the program only prints out the first 40 of the characters. Run the program once more, and this time it will not even wait for you to enter a name, and it will print the remainder of the 50 or so characters you typed the first time. This illustrates another "quirk" of function 3F, which can be a problem in "real" programs. You can make the program absorb the extra characters by adding the lines in Figure 4 before the label SHONAM.

Function 3F seems easy to work with at first, but after you add the code necessary to check for overtyping and the code to absorb extra characters, it gets to be a mess. However, if you are not interested in limiting the number of characters input, you can just supply a large number in the CX register, and supply a large enough buffer in case the user types a lot, and you probably will not need the extra code.

The other method (function 0A hex) for reading an entire line is a carry-over from the old 8-bit CP/M operating system. Figure 5 shows a version of NAME10 that uses this method.

Function 0A requires that you put the limit count in the input buffer, as the first byte. After you execute the function, the first byte of the buffer will still contain the limit count. The second byte will contain the actual count of the characters typed, including a carriage return character at the end (but not a line feed), and the characters will follow this byte.

Type in the lines listed above (name the file NAME10A.ASM), and assemble them to get NAME10A.COM. Run the program and try to type in more than 40 characters for "your name". You will find that after you have typed 39 characters, the computer will just beep if you try to type more. So the only problem with this method of input is that the count you specify (as the first byte in the buffer) is one less than the number of characters you can type. If you really want a maximum name length of 40 characters, you must set the count to 41. Like function 3F, function 0A supports command line editing, so for most purposes you may find it better than function 3F.

Notice that the message at MSG2 in the second example has return and line feed codes added to the beginning. That

```
;       TYPE MY NAME 10 TIMES

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

START:  MOV     DX,OFFSET MSG1
        MOV     AH,9
        INT     21H             ;PROMPT FOR NAME
        MOV     BX,0            ;GET STANDARD INPUT "HANDLE"
        MOV     CX,40           ;READ 40 CHARACTERS MAX
        MOV     DX,OFFSET NAMBUF ;PUT INPUT HERE
        MOV     AH,3FH
        INT     21H             ;GET USER'S NAME
        CMP     AX,2            ;ANY NAME ENTERED?
        JBE     START           ;IF NOT, RE-PROMPT
        MOV     BX,OFFSET NAMBUF
        ADD     BX,AX           ;POINT TO END OF BUFFER
        MOV     BYTE PTR [BX],'$' ;TERMINATE NAME
        MOV     DX,OFFSET MSG2
        MOV     AH,9
        INT     21H             ;SAY "NAME IS"
        MOV     CX,10           ;SET A COUNTER
        MOV     DX,OFFSET NAMBUF ;POINT TO NAME BUFFER
NAMELP: MOV     AH,9
        INT     21H             ;SHOW NAME
        LOOP    NAMELP          ;DO IT 10 TIMES
        INT     20H             ;EXIT

MSG1    DB      'What is your name? $'
MSG2    DB      'Here is your name 10 times:',13,10,'$'
NAMBUF  LABEL   BYTE            ;NAME BUFFER

CODE    ENDS
        END     START
```

Figure 2. "Type My Name 10 Times", first version.

```
        CMP     AX,40           ;ALL 40 CHARACTERS USED?
        JNZ     SHONAM          ;NO
        CMP     BYTE PTR -1[BX],13 ;LAST CHARACTER CR?
        JZ      ADDLF           ;IF SO, ADD LF
        MOV     BYTE PTR [BX],13 ;ADD CR
        INC     BX
ADDLF:  MOV     BYTE PTR [BX],10 ;ADD LF
        INC     BX
SHONAM:
```

Figure 3. Modification to allow long names.

```
        PUSH    BX              ;SAVE END OF NAME
        MOV     DX,BX           ;PUT EXTRA INPUT HERE
        INC     DX              ;AFTER END OF NAME
        MOV     BX,0            ;GET STANDARD INPUT "HANDLE"
        MOV     CX,40           ;READ 40 CHARACTERS MAX
        MOV     AH,3FH
        INT     21H             ;GET EXTRA CHARACTERS
        POP     BX
```

Figure 4. Modification to absorb extra characters.

is because DOS does not issue a line feed after you type Return while it is executing function 0A, but it does with function 3F.

If you type Control-C while either version of NAME10 is prompting for input, the program will exit and return you to MS-DOS. A special interrupt called the Control-C interrupt is generated when you type Control-C during input via function 0A or 3F, and if your program is not set up to process that interrupt, control is returned to MS-DOS. I may cover the Control-C interrupt in a later installment.

Inputting Single Characters

If you are writing a program that requires single character input (with no Return after an input), there are four DOS input methods to choose from (functions 1, 6, 7, and 8). Each function waits for a single key to be typed, and then returns with the key value the AL register. Function 1 also echoes the key typed on the screen, and both functions 1 and 8 process Control-C the way functions 0A and 3F do. Function 6, called the "Direct Console I/O" function, can be used for input or output. If the byte in DL is anything other than FF (hex), that byte is output to the screen as a character. If DL contains FF, then function 6 will return a character code in AL and clear the Zero flag if a character has been typed, or else it will set the Zero flag. Function 6 does not check for Control-C, so it can be input as a character like any other. Function 7 (Direct Console Input) is like the function 8 except that it also does not check for control-C.

Since it is nice to see what you have typed on the screen, you may be wondering why there are functions that accept input but do not echo it on the screen. They are provided so that you can check the input before you echo it. Figire 6 shows a little program that accepts only the letters Y, y, N, or n, and echoes the input if it is one of those, or beeps if it is not.

You will notice that this program only checks for Y and N, and not for the lower case y and n. Instead, it ANDs the character you input with 5F (hex), which will not affect a capital letter, but will cause a lower case letter to be capitalized. That is because the ASCII values of the lower case letters are the same as the capital letters plus 20 (hex). The value 5F, which is 1011111 in binary, will "strip" the extra 20 (hex), which is 100000 in binary. When you run the program and enter a lower case y or n, it will echo an upper case Y or N.

This program illustrates (in a simple way) how keyboard input can be checked and processed. If neither Y or N are typed, it will beep and wait for another character. If you type either Y or N, the program will print one of two messages and then exit.

Figure 5. "Type My Name 10 Times", second version.

```
;       TYPE MY NAME 10 TIMES

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

START:  MOV     DX,OFFSET MSG1
        MOV     AH,9
        INT     21H             ;PROMPT FOR NAME
        MOV     AH,0AH
        MOV     DX,OFFSET NAMBUF ;POINT TO NAME BUFFER
        INT     21H             ;GET USER'S NAME
        MOV     BL,NAMBUF+1     ;GET SIZE OF ENTRY
        OR      BL,BL           ;ANY ENTRY?
        JZ      START           ;IF NOT, RE-PROMPT
        XOR     BH,BH           ;BX = SIZE OF NAME
        ADD     BX,OFFSET NAMBUF+2 ;POINT TO END OF NAME
        INC     BX              ;POINT AFTER CR
        MOV     BYTE PTR [BX],10 ;ADD LF
        INC     BX
        MOV     BYTE PTR [BX],'$' ;TERMINATE NAME
        MOV     DX,OFFSET MSG2
        MOV     AH,9
        INT     21H             ;SAY "NAME IS"
        MOV     CX,10           ;SET A COUNTER
        MOV     DX,OFFSET NAMBUF+2 ;POINT TO NAME BUFFER
NAMELP: MOV     AH,9
        INT     21H             ;SHOW NAME
        LOOP    NAMELP          ;DO IT 10 TIMES
        INT     20H             ;EXIT

MSG1    DB      'What is your name? $'
MSG2    DB      13,10
        DB      'Here is your name 10 times:',13,10,'$'
NAMBUF  DB      40              ;MAX CHARACTERS ALLOWED

CODE    ENDS
        END     START
```

Figure 6. Single character input with character checking.

```
;       ACCEPT Y OR N

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

START:  MOV     DX,OFFSET MSG
        MOV     AH,9
        INT     21H             ;PROMPT FOR Y OR N
REGET:  MOV     AH,8
        INT     21H             ;INPUT A KEY
        AND     AL,5FH          ;CAPITALIZE
        CMP     AL,'Y'          ;IS IT "Y"?
        JZ      CHAROK          ;YES
        CMP     AL,'N'          ;IS IT "N"?
        JZ      CHAROK          ;YES
        MOV     DL,7            ;ELSE, GET BELL CHAR
        MOV     AH,2
        INT     21H             ;MAKE A BEEP
        JMP     REGET           ;AND GET ANOTHER KEY
CHAROK: MOV     DL,AL
        MOV     AH,2
        INT     21H             ;ECHO REPLY
        MOV     DX,OFFSET GOOD  ;ASSUME Y
        CMP     AL,'Y'          ;IS IT
        JZ      PMSG            ;YES, PRINT MSG
        MOV     DX,OFFSET BAD   ;ELSE, GET N MSG
```

Checking the Keyboard Status

Sometimes a program needs to check if a key has been typed, but it must continue some kind of processing rather than waiting for the key. There are two MS-DOS functions that can be used for this. One is function 6, mentioned previously, and the other is function 0B (hex). Function 0B is a status check only, whereas function 6 is both a status check and an input function. Function 0B also checks for Control-C, while function 6 does not.

To illustrate a use of checking the keyboard status, Figure 7 contains a program that continuously prints HELLO on the screen until a key is typed.

Rather than setting or clearing the Zero flag to indicate status, as function 6 does, function 0B returns a value in the AL register. It returns FF (hex) if a key has been pressed, or 0 if no key was pressed. This program prints HELLO in a "sawtooth" pattern on the screen until function 0B indicates that a key was pressed. Then function 8 is used to "absorb" the key. If the program did not do this, then the character representing the key that was typed would appear at the MS-DOS prompt after the program exited.

To modify this program to use function 6, replace these lines

```
MOV     AH,0BH
INT     21H     ;CHECK STATUS
OR      AL,AL   ;ANY CHARS?
```

with these lines

```
MOV     AH,6
MOV     DL,0FFH ;SIGNAL INPUT
INT     21H     ;ANY CHARS?
```

and then remove these lines

```
MOV     AH,8
INT     21H     ;ABSORB KEY
```

Since function 6 is both a status and input function, a separate function call to absorb the key is not needed.

BIOS Input Functions

The BIOS uses interrupt 16 (hex) for keyboard input and status checking. Interrupt 16 can perform several functions, and it uses the AH register to supply a code that selects a particular function. For this discussion, we are only interested in function 0, get character code, and function 1, check keyboard buffer (or status). Function 0 works like the MS-DOS single character input functions except that in addition to the ASCII value of the key in AL, it returns the "scan code" for the key in the AH register. This makes INT 16 useful for reading function keys, arrow keys, etc. I may discuss that more later. Function 1 works a bit like the DOS direct console I/O function except that it does not absorb the key when it detects that one is pressed.

```
GOOD    DB      13,10,'Good answer!$'
PMSG:   MOV     AH,9
        INT     21H             ;PRINT MSG
        INT     20H             ;AND EXIT

MSG     DB      'Input Y or N: $'
BAD     DB      13,10,'Too bad!$'

CODE    ENDS
        END     START
```

Figure 7. Keyboard status check with DOS function 0B.

```
;       PRINT HELLO UNTIL KEY PRESSED

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

START:  MOV     CX,1            ;SET A COUNTER
BIGLP:  PUSH    CX              ;SAVE IT
PRTSP:  MOV     DL,' '
        MOV     AH,2
        INT     21H             ;PRINT SPACES
        LOOP    PRTSP
        POP     CX              ;GET COUNTER
        MOV     DX,OFFSET MSG
        MOV     AH,9
        INT     21H             ;PRINT "HELLO"
        MOV     AH,0BH
        INT     21H             ;CHECK KEYBOARD STATUS
        OR      AL,AL           ;ANY CHARACTERS TYPED?
        JNZ     EXIT            ;YES, EXIT
        INC     CX              ;INCREMENT COUNTER
        CMP     CL,40           ;AT 40?
        JNZ     BIGLP           ;NO
        JMP     START           ;ELSE, RESET COUNTER
EXIT:   MOV     AH,8
        INT     21H             ;ABSORB KEY
        INT     20H

MSG     DB      'HELLO',13,10,'$'
CODE    ENDS
        END     START
```

Figure 8. Keyboard status check with BIOS interupt 16.

```
;       PRINT HELLO UNTIL KEY PRESSED

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

START:  MOV     CX,1            ;SET A COUNTER
BIGLP:  PUSH    CX              ;SAVE IT
PRTSP:  MOV     DL,' '
        MOV     AH,2
        INT     21H             ;PRINT SPACES
        LOOP    PRTSP
        POP     CX              ;GET COUNTER
        MOV     DX,OFFSET MSG
        MOV     AH,9
        INT     21H             ;PRINT "HELLO"
        MOV     AH,1
        INT     16H             ;CHECK KEYBOARD STATUS
        JNZ     EXIT            ;KEY PRESSED, EXIT
        INC     CX              ;INCREMENT COUNTER
        CMP     CL,40           ;AT 40?
        JNZ     BIGLP           ;NO
        JMP     START           ;ELSE, RESET COUNTER
```

However, it does return the key value.

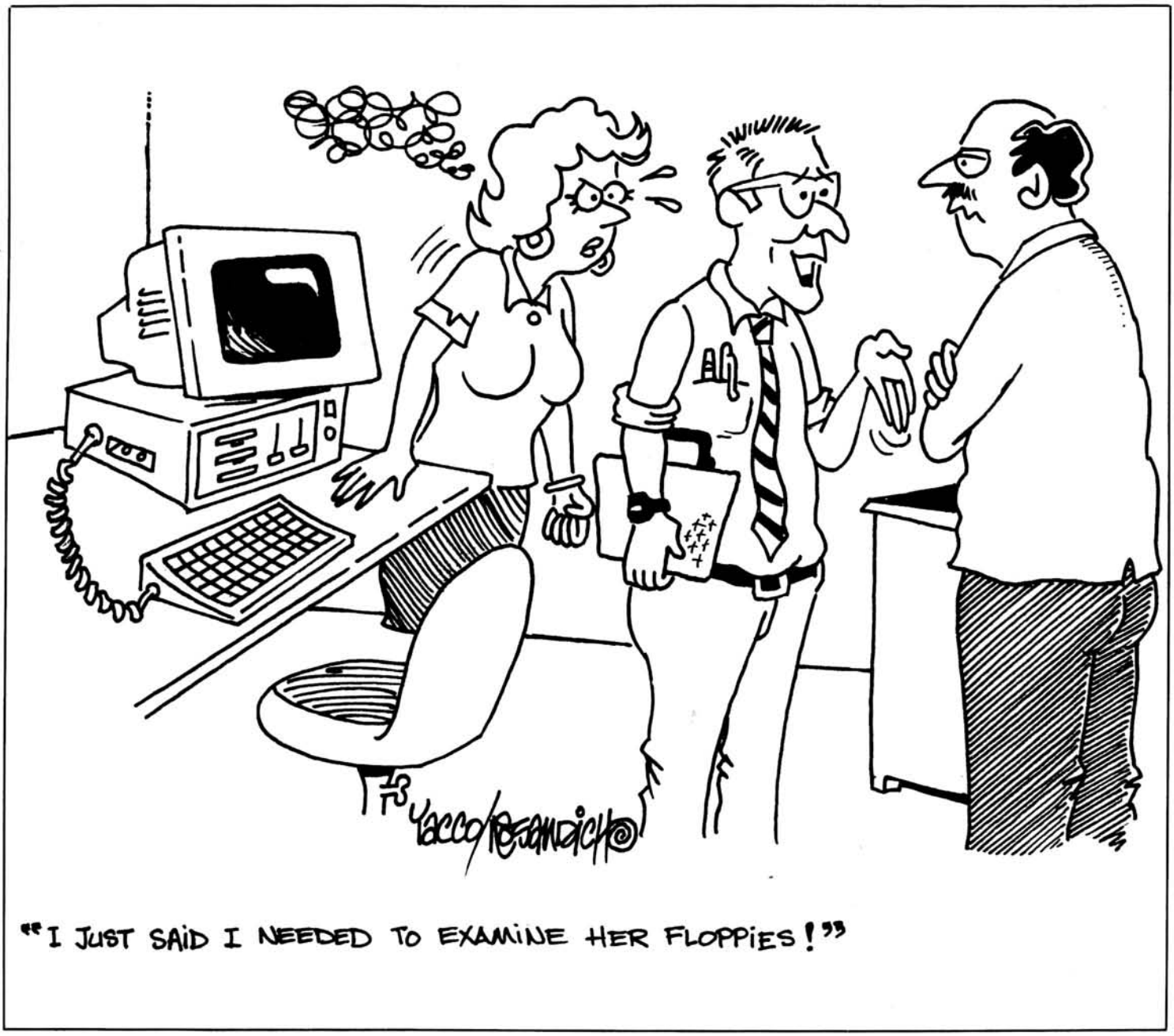
Figure 9 shows the previous example modified to use interrupt 16.

There is one other method for reading the keyboard that I will not cover, at least not for now. It involves capturing the keyboard interrupt, which is generated each time a key is pressed. This method of reading the keyboard is used by Terminate and Stay Resident programs (TSR's) and by action games that must update video rapidly and monitor the keyboard at the same time. It is also used when a program wants to detect a key that cannot be read by the BIOS or MS-DOS (for example, the keypad 5 key when Num Lock is off.

```
EXIT:   MOV     AH,0
        INT     16H             ;ABSORB KEY
        INT     20H

MSG     DB      'HELLO',13,10,'$'
CODE    ENDS
        END     START
```

Well, that's it for this installment of Assembly Language. Next time I will begin a discussion of disk I/O.



Powering Up Volume 2

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

How to Use ANSI.SYS

In the last article, we review-ed most of the standard device drivers that have been provided with DOS, and the ANSI.SYS device driver was purposely omitted because it has so many features and uses. But, the ANSI.SYS device driver is not quite as important as it once was. In the early-to-mid 1980s, ANSI.SYS was more important because it provided some compatibility features, primarily for screen control, that were important when there were a number of DOS-based computers that were not hardware compatible. For example, the famous and truly innovative Heath/Zenith Z-100 computer ran under the MS-DOS operating system, but it was not at all hardware compatible with the IBM PC. To help avoid writing different programs to support all of the wide variety of hardware, many software vendors used the ANSI.SYS standards so their programs would run on most DOS-based systems. Today, the ANSI.SYS device driver is still useful, and you will find one of its direct uses in the next article: Powering Up the PROMPT Command. But before we get into the details of ANSI.SYS, let's take a look at some of the background.

Some Background

The term ANSI is an acronym for the American National Standards Institute. Its purpose is to act as a coordinator for various standards in all kinds of technical areas, but it is perhaps best known in its role of helping standardize many computer-related areas. ANSI relies on voluntary industry cooperation to accomplish its goal of establishing standards, and it is composed of a number of members provided by supporting hardware and software vendors. One of ANSI's more important successes, relatively speaking, was the establishment of the industry-wide ANSI-74 COBOL standard for the COBOL programming language. Today, you will find that ANSI has considerable influence on standards for the C programming language, which is a significant advantage for both users and vendors.

As you can guess, the ANSI.SYS device driver is simply the implementation of one area of the ANSI standards. It isn't perfect, but any standard is better than leaving many of these things to different manufacturers who always seem to come up with "better" ideas.

Types of Device Drivers

One of the particular points that I intentionally omitted from the last article (due to space) was a discussion of the two types of device drivers: character and block. As mentioned in the previous article, all manufacturers of microcomputers provide default device drivers as part of the operating system, and the default device drivers are contained in the program code for the Basic Input Output System — normally referred to as the BIOS. Manufacturers develop this code to support the hardware that they supply to the users. In today's compatible systems, this is normally a hidden, system file called IBMBIO.SYS, although some specific information may also be contained in the ROM BIOS.

In the previous article, we have already reviewed some of the default device drivers for some microcomputers, such as COM1 and PRN. These device drivers provide the operating system interface to CHARACTER (or serial) type devices such as the console (and keyboard), modems, serial printers, and so on. It is also significant to note that FCB's (File Control Blocks) or file handles can be opened to perform input and output on character devices.

The previous article also noted that device drivers are used by the operating system to communicate with disk drives. These are known as BLOCK device drivers because they transfer a block of data (normally 512 bytes at a time, which is the same as the standard disk sector size in DOS). Block device drivers cannot be opened directly through an FCB or file handle. They are "mapped" to drive letters using a table located in memory when DOS is booted. That is precisely what happens when a drive letter is assigned to a virtual disk when the VDISK.SYS device driver (see the previous article) is installed in the CONFIG.SYS file.

It also happens that the contents of this table can actually be changed by a user with the ASSIGN command. The ASSIGN command actually changes the table which defines the correspondence between the hardware assigned (performed through a "programming" jumper or switch physically located on the disk drive) and the drive letter. Unfortunately, some programs actually "bypass" this table for one reason or another, so this technique will not always work.

One important characteristic of block device drivers is that they are responsible for one or more disk drives or physical units. For example, the first block device driver is responsible for two floppy drive units (drives 0-1 or letters A and B), and the second device driver is responsible for two hard drive units (drives 0-1, normally letters C and D). The position of the device driver in the list of device drivers determines which units correspond to which drive letters. That's also how the VDISK device driver determines which letter to assign when a virtual disk is installed in memory. If you have a two-drive floppy disk system, for example, you will find that the VDISK is assigned as drive letter C. If you have a hard drive with a single partition (i.e., C), you will find that the virtual disk is installed as drive D. And so on.

In summary, there are two types of device drivers. Character device drivers transfer information on a character-by-charac-

ter basis. Block device drivers transfer data in blocks of 512 bytes. For practical purposes, it is sufficient to recognize that block device drivers allow the operating system to communicate with disk drives whether they are floppies, hard disks or RAM disks. Character device drivers essentially perform all other interfaces required between the operating system and the physical computer hardware. At this point, you can probably see that the ANSI.SYS device driver is a character device driver, so let's take a look at what its "commands" are.

The ANSI Escape Sequences

Part of the ANSI standards includes some escape (or control) functions for controlling hardware functions. For those of you who have a copy of the 544- page MS-DOS FlipFast that I wrote, you will find the information similar to that presented here beginning on page 477.

The ANSI.SYS device driver really provides four separate types of functions: cursor functions, erase functions, mode functions, and key/string reassignment. Figure 1 is the list of cursor functions provided by ANSI.SYS.

Name Function	Description
CUU ESC [l A	**Cursor Up**: Move cursor up l lines without changing column. If the cursor is already on the top line, the command is ignored.
CUD ESC [l B	**Cursor Down**: Move cursor down l lines without changing column. If the cursor is already on the bottom line, the command is ignored.
CUF ESC [c C	**Cursor Forward**: Move cursor forward c columns without changing the line number. If the cursor is already at the right-most column, the command is ignored.
CUB ESC [c D	**Cursor Backward**: Move cursor backward c columns without changing the line number. If the cursor is already at the left-most column, the command is ignored.
CUP ESC [l;c h	**Cursor Position**: Move cursor to line l and column c.
HVP ESC [l;c f	**Horizontal and Vertical Position**: Move cursor to line l and column c. Note that HVP is essentially identical to CUP above.
DSR ESC [6 n	**Device Status Report**: Requests a Cursor Position Report (CPR) sequence (see below) from the console driver.
CPR ESC [l;c R	**Cursor Position Report**: Reports the current cursor position through the standard input device (STDIN) by line l and column c. Note this function is returned by the DSR function.
SCP ESC [s	**Save Cursor Position**: Saves the current cursor posi tion. See Restore Cursor Position (RCP) below.
RCP ESC [u	**Restore Cursor Position**: Restores the cursor to the ine and column values established by the Save Cursor Position (SCP) function.

Figure 1
ANSI.SYS Cursor Functions

For the most part, the cursor functions shown in Figure 1 are used primarily by programmers. The Name shown in Figure 1 is a name that can be used by programmers as a mnemonic, and it is not particularly useful otherwise. As you may have noticed, the Function may contain italic letters l for lines and c for columns. When the l and c parameters are valid, the default is always 1, and the values must be entered in the ASCII decimal value (not hex). With the exception of the l and c parameters, the Function shown in Figure 1 is the ANSI control sequence that must be used exactly as shown because uppercase and lowercase letters are very important.

The erase functions are used in a similar way, and they are shown in Figure 2.

Name Function	Description
ED ESC [2	**Erase Display**: Erases the entire screen and moves cursor to the home position in the upper left corner of the screen.
EL ESC [K	**Erase Line**: Erases from the cursor to the end of the line, including the cursor position.

Figure 2
ANSI.SYS Erase Functions

How useful the functions shown in Figure 2 really are depends on what you are doing, and again, these are primarily useful to programmers.

The ANSI mode functions are shown in Figure 3.

The mode functions shown in Figure 3 are some of the most interesting functions that you can use. How to use them with the PROMPT command to change screen colors will be shown in the next article.

The key/string reassignment functions are shown in Figure 4.

All of the keycode values shown in Figure 4 are just the defined ASCII values. How to use the functions shown in Figure 4 is not obvious, so I will jump ahead to the next article a little and use the PROMPT command to illustrate how these functions can be used.

You can use the key reassignment function to change the values of various keys, and the PROMPT command provides an easy way to do this as shown below:

```
PROMPT $E[65;81p
PROMPT $E[81;65p
PROMPT $E[97;113p
PROMPT $E[113;97p
PROMPT $N $Q$Q$G
```

The $E in the PROMPT command is the control sequence used for the ESCape in the ANSI function. The first line changes the capital letter A (decimal 65) to the capital letter Q (decimal 81). The second line changes the capital letter Q to the capital letter A. If you want to fully remap those two keys, you must also remap the lowercase letters too, so the third line changes the lowercase a (decimal 97) to the lowercase q (decimal 113), and the fourth line changes the lowercase q to the lowercase a. When you use the PROMPT command like this, you will also need to reset the command line prompt as shown on the fifth line and described in detail in the next article.

There are lots of examples one can choose to illustrate the string assignment function, but my favorite is to "program" the F10 Function Key to perform the DIR command. That is even easier when using the PROMPT command, which is:

```
PROMPT $E0;68;"DIR";13p
PROMPT $N $Q$Q$G
```

As before, the $E represents the ESCape character (27 decimal) in the PROMPT command. The 0;68 is the ASCII keycode for the F10 Function Key. DIR is the string value as shown in Figure 4 and must be enclosed in quotes ("). And 13 is the decimal value for a Carriage Return (CR), which is the same as the ENTER key that is required to execute any DOS command. If you did not add the 13 after the string, then pressing the F10 key would only display the DIR command, but would not execute it. As before, you must enter the PROMPT

Name Function	Description
SGR ESC [n;...;n m	**Set Graphics Rendition**: Sets character attributes such as bold, underscore, reverse video, and screen colors, depending on the values of n as shown in the following list. 0 = All attributes OFF 1 = High intensity (bold) on 2 = Low intensity (faint) on 3 = Italic on 4 = Underscore on (MDA only) 5 = Normal blink on 6 = Rapid blink on 7 = Reverse video on 8 = Hidden on 30 = Black foreground 31 = Red foreground 32 = Green foreground 33 = Yellow foreground 34 = Blue foreground 35 = Magenta foreground 36 = Cyan foreground 37 = White foreground 40 = Black background 41 = Red background 42 = Green background 43 = Yellow background 44 = Blue background 45 = Magenta background 46 = Cyan background 47 = White background
SM ESC [= n h	**Set Mode**: Sets the screen width or type as specified by n as shown in the following list. 0 = 40 x 25 b/w (black/white, monochrome) 1 = 40 x 25 color 2 = 80 x 25 b/w 3 = 80 x 25 color 4 = 320 x 200 color 5 = 320 x 200 b/w 6 = 640 x 200 b/w 7 = Enable word wrap 14 = 640 x 200 color 15 = 640 x 350 b/w 16 = 640 x 350 color 17 = 640 x 480 color 18 = 640 x 480 color 19 = 320 x 200 color
RM ESC [= n l	**Reset Mode**: Resets screen width or type as specified by n (same values as Set Mode above). Note the lowercase l in this function.

Figure 3
ANSI.SYS Mode Functions

Function	Description
ESC [keycode-old;keycode-new p	**Key reassignment**: Replaces the keycode-old with the keycode-new.
ESC [keycode;"string" p	**String Assignment**: Assigns a string value (i.e., a literal) to a keycode. The string must be enclosed in quotes (") as shown.

Figure 4
ANSI.SYS Key/String Reassignment Functions

command again because the command line prompt will be reset to its default by this command. But you are probably wondering where I came up with the 0;68 that represents the F10 key in this example. Actually, the 0;68 is an ASCII keycode, but it is more accurately known as an extended keycode.

Keycodes

Generally speaking, a keycode is just the decimal number associated with a specific character. Because the standard ASCII character set only defines 128 characters from 0 to 127 decimal, and there are some odd keys on a compatible keyboard that were NOT originally defined; an extended keycode was required to use these keys, such as the Function Keys and ALT-key combinations. An extended keycode is easy to identify because it always begins with a zero (0) followed by a semicolon (;). Keycodes and extended keycodes are sometimes difficult to find, and I have included them here as Figure 5.

As you can see, not all keys have defined keycodes, and the values listed in Figure 5 are quite similar to those available in most versions of the BASIC programming language. Perhaps the most important point is that these are the keycodes that can be used in most current versions of ANSI.SYS. Older versions of ANSI.SYS will not interpret the keycodes associated with some of the new keys added on the 101-key keyboard, such as F11 and F12. Specifically, all MS-DOS version 2 releases did not include the F11 and F12 function keys because the 101-key keyboard was not available until after MS-DOS version 3.1 was released. So, don't expect to use some of this information if you are using an older MS-DOS release with an 84-key keyboard.

Also, I should note that I have not tested every single one of these ANSI commands with every single release of MS-DOS, so you may find that something mentioned in this article does not seem to work as described. If you find that situation, and you have checked everything to be sure you are using the correct command and syntax, you have most likely found a bug in the ANSI.SYS device driver that you are using. Perhaps the best known bug in ANSI.SYS occurred in the Z-100 MS-DOS version 3.1 where not all of the standard ANSI functions were implemented correctly or at all. If you find that problem, you may be able to use ANSI.SYS from another DOS version, but you may have to experiment to find one that has the functions you want.

How ANSI.SYS Works

Now that you know something about the ANSI functions and control sequences, it is useful to take a brief look at how the ANSI.SYS device driver works. In general, it provides a "translation table" between the command Functions listed here and a hardware interrupt that is used in your computer. That interrupt is called the Video Input/Output (I/O) interrupt or INT 10H in technical terms.

The Video I/O Interrupt controls the CRT display and various attributes related to the display such as blinking characters, character intensity (normal, high, low) , page scrolling, and colors as well as the various

Main Keyboard

Keytop	(Lowercase) Normal	(Uppercase) SHIFT+Key	CTRL +Key	ALT +Key
A	97	65	1	0;30
B	98	66	2	0;48
C	99	67	3	0;46
D	100	68	4	0;32
E	101	69	5	0;18
F	102	70	6	0;33
G	103	71	7	0;34
H	104	72	8	0;35
I	105	73	9	0;23
J	106	74	10	0;36
K	107	75	11	0;37
L	108	76	12	0;38
M	119	77	13	0;50
N	110	78	14	0;49
O	111	79	15	0;24
P	112	80	16	0;25
Q	113	81	17	0;16
R	114	82	18	0;19
S	115	83	19	0;31
T	116	84	20	0;20
U	117	85	21	0;22
V	118	86	22	0;47
W	119	87	23	0;17
X	120	88	24	0;45
Y	121	89	25	0;21
Z	122	90	26	0;44
TAB	9	0:15	n/d	n/d
1 or !	49	33	n/d	0;120
2 or @	50	64	n/d	0;121
3 or #	51	35	n/d	0;122
4 or $	52	36	n/d	0;123
5 or %	53	37	n/d	0;124
6 or ^	54	38	n/d	0;125
7 or &	55	94	n/d	0;126
8 or *	56	38	n/d	0;127
9 or (	57	42	n/d	0;128
0 or)	48	40	n/d	0;129
- or _	45	95	n/d	0;130
= or +	61	43	n/d	0;131

Function Key

Keytop	(Lowercase) Normal	(Uppercase) SHIFT+Key	CTRL +Key	ALT +Key
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
F5	0;63	0;88	0;98	0;108
F6	0;64	0;89	0;99	0;109
F7	0;65	0;90	0;100	0;110
F8	0;66	0;91	0;101	0;111
F9	0;67	0;92	0;102	0;112
F10	0;68	0;93	0;103	0;113
F11	0;133	0;135	0;137	0;139
F12	0;134	0;136	0;138	0;140

Keypad Keys

Note: Keypad keys shown with Num Lock OFF (i.e., 7/Home = Home, not 7)

Keytop	Normal	SHIFT+Key	CTRL +Key	ALT +Key
Home or 7	0;71	55	0;119	n/d
Up or 8	0;72	56	n/d	n/d
PgUp or 9	0;73	57	0;132	n/d
Left or 4	0;75	52	0;115	n/d
(blank)/5	n/d	53	n/d	n/d
Right or 6	0;77	54	0;116	n/d
End or 1	0;79	49	0;117	n/d
Down or 2	0;80	50	n/d	n/d
PgDn or 3	0;81	51	0;118	n/d
Ins or 0	0;82	48	n/d	n/d
Del or .	0;83	46	n/d	n/d

n/d = Not defined in ANSI.SYS (see text)

Figure 5 (Cont'd. on next column)

Figure 5
ASCII Keycodes

video modes (i.e. color, monochrome, and display resolution).

The Video I/O Interrupt performs a wide variety of video I/O operations which can be implemented by assembly language programmers using register AH. Since virtually all these video functions require some knowledge of assembly language and the associated registers, it is not very convenient for programming in BASIC and some other high level languages that do not provide a convenient interface to low level bit functions.

What Does ANSI.SYS Do?

As previously mentioned, ANSI.SYS acts as a "translation table" between programs and commands (like the PROMPT command, which you will see in the next article). It translates the ANSI Function commands into the appropriate Video I/O Interrupt function. But more importantly, it provides a standard interface between the programmer and the hardware (in this case, the ROM BIOS). This means that if a program (or command) uses the standard ANSI Escape functions, it will work on almost ALL compatible hardware. This is true even for the H/Z- 100 if the ANSI mode is entered, and the ANSI.SYS device driver is implemented.

It should be noted that ANSI.SYS is a DEVICE DRIVER which has been previously discussed in detail. It is implemented by including the command line like DEVICE=ANSI.SYS in the CONFIG.SYS file which will be discussed later.

The ANSI driver only works under certain conditions. That is, a program must be "well behaved" meaning that it must use the appropriate MS-DOS system calls to perform various functions. In particular, the ANSI.SYS driver is called to check (intercept) all characters read or written to the screen when the following MS-DOS function calls (Function calls are initiated by INT 21) are issued:

```
01H - Read Keyboard and Echo Character on
      the CRT
02H - Write Character to CRT
06H - Direct Console I/O
09H - Write String to CRT
```

These functions are typically used by assembly language programmers to perform most video I/O operations. While I recognize that much of this discussion is very technical, it is not my intent to provide you with an intimate knowledge of assembly language which is beyond the scope of this discussion. I am only trying to explain the "why" and "what" of ANSI.SYS. Assembly language programmers will be able to use this information to make some decisions about programming.

ANSI.SYS
The Good News and the Bad News

ANSI.SYS provides the standard user level interface between the operating system and hardware, mostly the hardware. If the standard ANSI Escape sequences are used, there are many advantages for the software developer. Probably the most important is the idea that software using these video I/O sequences is (or should be) hardware independent. That is, the software will run on a Z-100 or any DOS-based computer, assuming that proper programming has been done.

For example, Borland's Turbo Lightening program used to require that ANSI.SYS be implemented to use the program. Similarly, it is also true that, since the ANSI driver acts as a "translation

table", it usually does not interfere with the manufacturer's standard Escape sequences or hardware control functions. Remember that the ANSI.SYS driver, when installed, checks ALL the characters read or written when used with the MS-DOS function calls. The bottom line is, that with careful programming and "well behaved" programs which do not use hardware dependent interrupt functions, programs can be developed which are compatible with all MS-DOS operating systems and manufacturer's hardware. If that is true, then why is it that many programs developed for the compatible do not run on all the MS-DOS supported hardware like the H/Z-100?

The bad news is that programs using the standard MS-DOS function calls (with or without the ANSI.SYS driver) can be very SLLOOOWWWW as compared to programs that use hardware interrupts such as the Video I/O Interrupt that we previously discussed. That means the performance and response of those programs, as compared to programs that use hardware interrupts, can be poor from the user perspective. Smart software developers have recognized that users will not buy programs that exhibit poor performance and significant delays during processing. Furthermore, since compatible hardware has become the de facto standard, most software developers do not find it profitable to deal with most of the other hardware. And that is the point...hardware functions can be performed MUCH faster than software functions.

Implementing ANSI.SYS

Fortunately, the implementa-tion of ANSI.SYS is very easy. All you need to do is add a command line to (or create) the CONFIG.SYS file. Figure 6 shows a general form of the CONFIG.SYS file.

In this example, it is assumed that the ANSI.SYS device driver is located in the \DOS subdirectory on drive C, and you will have to change the path if your system is different. The FILES= and BUFFERS= commands shown in this example are included only for a complete CONFIG.SYS file listing, and the values required for your system may be different.

Powering Down

ANSI.SYS is a special device driver that provides a standard way to implement some of the video I/O functions. It essentially provides a "translation table" between the ANSI Escape sequences and the video control of the hardware. For well-behaved programs, it provides a standard interface between hardware-capable systems which run MS-DOS. The use of this standard interface has the significant disadvantage in that it is very slow from a performance and user response perspective. However, it does provide a way for users to implement various features of their machines with standard commands like the PROMPT command. When these video features are used in the PROMPT command, ANSI.SYS must be implemented to access the video I/O features.

```
FILES=25
BUFFERS=30
DEVICE=C:\DOS\ANSI.SYS
```

Figure 6
Implementing ANSI.SYS

Now that you have some of the important information about the ANSI.SYS driver, we will take this one step further in the next article on: Powering Up the PROMPT Command.

If you have any questions about anything in this article, 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.

NEW! EGAD for Z-100

Enhance your screen printouts with EGAD. Can your current print-screen utility do this?

* Print any part of the screen - crop box pops up when Shift-F12 is pressed. Use arrow keys to select region.
* Enlarge graphics 1-4 times, or Reduce.
* Print in color (color printers) or black & 8 grays
* SET program selects printer colors, other options.
* Callable from user programs.

Supports: **HP** laser and inkjet, **Epson** & compatibles, **Epson** & Compatible Color, **NEC-8023** and **C. Itoh 8510**, MPI series, others. Order # 271, **$35.00** postpaid.

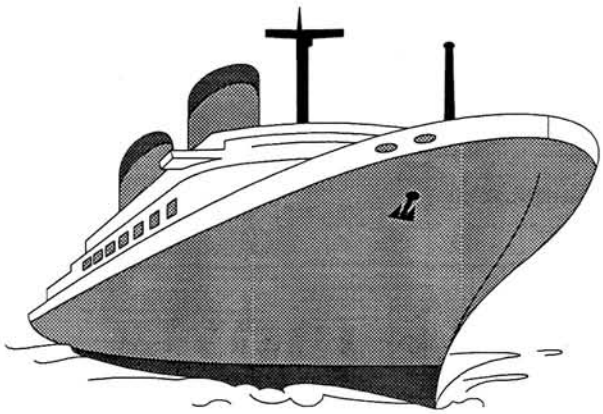
PC89LINK

Transfer files between: PC and/or H/Z-100 (MSDOS); H/Z-89/90 or H-8 (HDOS or CPM). Wildcards supported. Send binary or text files. Baud rates to 38400. Uses a simple 3-wire cable. Specify any two: 5.25" MSDOS; 3.5" MSDOS; hard or soft HDOS&CPM disk. **$25.00** postpaid.

Lindley Systems—4257 Berwick Place, Woodbridge, VA 22192-5119 (703) 590-8890
Call or Write for Free Catalog. Ask about our FORM FILL-R package and Printer Drivers for HDOS & CPM.

Reader Service #136

Port-O-Call:



COM1

Laura White
759 Polfus
Benton Harbor, MI 49022

This month in Port-O-Call — COM1, I thought it would be helpful to discuss, in detail, some of the problems huggies have been having with Windows 3.0, even though it's been on the market for awhile. For those of you who are new to Windows 3.0, which has an incredibly powerful potential, this new version opens up with an icon-run, or Graphical User Interface (GUI). This feature makes working with application software an absolute dream — especially if you have a mouse hooked up to your system.

Windows 3.0 runs in three possible modes: real, standard and Enhanced. You get the Real mode with at least 256K and less than 1MB of available extended RAM. In this mode, you can run Windows 2.xx and 3.xx applications and exploit expanded memory if it is installed. With a 286 or better system, and at least 640K of RAM, Windows will run in the Standard mode. All memory is treated like one block, which improves performance and allows you to run DOS programs. You cannot, however, multi-task these programs with other applications. (Getting beyond the standard mode seems to be the problem most of our huggies have had.) Windows 3.0 is at its best in the 386 Enhanced mode. This happens when you have a 386 or 486 with 2MB Ram or more. With this setup you can multi-task multiple DOS programs and Windows' applications simultaneously.

When Windows 3.0 is first installed in your computer it checks to see what memory is available. For all practical purposes, it puts itself in either the real, standard or enhanced mode depending on how your memory is set up and what purpose it's serving. If you have a Z-386/16 MHz computer, have verified that you should be running in the enhanced mode by clicking on "help" and then "About Program Manager" in the Program Manager screen, and have found that you are not running in the enhanced mode, you might have to fix this situation by installing a new ROM. This ROM, which costs approximately $15, must be purchased from your local Heath Dealer and it's part number is 444-536-5. This update will correct some memory allocation problems previously inherent in the Z-386/16. When running correctly, Windows automatically converts the portion it needs of extended memory to expanded, and uses the rest for smart drive.

For the most part, complications have occurred on Zenith Data Systems 286, 386 or better when using a generic version of Windows 3.0. The reasons are many, but if it had to be explained in simple terms, they use memory differently than anyone else—IBM, Compaq, etc. The Z-286/386 machines are engineered to use a portion of the extended memory between 640K and 1Mb for increased speed and enhanced graphics. There is however, "only so much room at the inn", and Windows is also designed to use some of the same memory to run in its enhanced mode.

One huggie called in and stated that Windows "didn't like his video driver." What was happening was that he had a problem with "blotches" or "mouse trails" being left behind on the screen when moving the mouse around. There are two distinct situations and solutions where this occurs. First, if you are using a Z-449 video card, you must select "EGA" instead of "VGA" as your video driver. This is the case because the Z-449 card is not a true "VGA" video card.

The second and more difficult problem, as one huggie was experiencing, occurred with a Z-525 cache card. If the CPU and cache card do not match, you may need a new cache controller PAL, part number 442-572-1. A quick way to determine if this problem exists, is to remove the cache card and run windows again. If the blotches go away, you probably need the new PAL; if not, try the new ROM mentioned earlier. If you already have a copy of Windows 3.0, not distributed by Zenith Data Systems, the rest of this article will help you to get your program operational on your computer. The following details about this version of Windows should give you a better understanding of how a Z-286 or Z-386 system differs from other IBM compatibles.

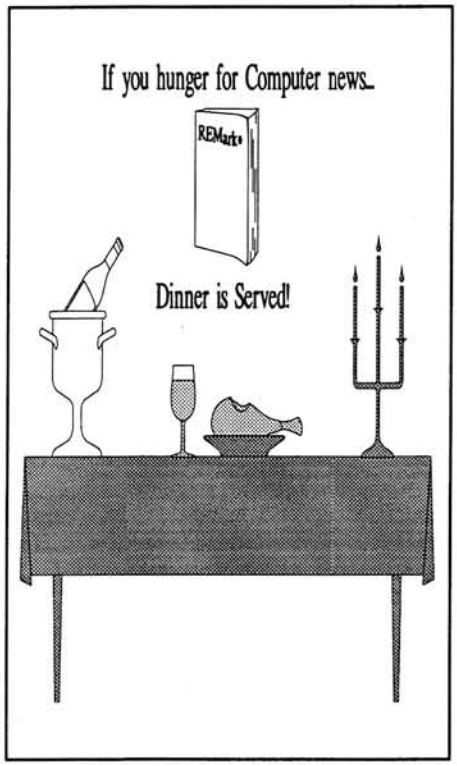
Zenith Data Systems' version of Windows 3.0 differs in a number of ways. First of all, in addition to the copyright notice, their version of Windows 3.0 has included a series of commands which are kept in the initialization files (.INI) to tell the program how to be configured and to stay away from that part of the expanded memory that holds Zenith Data Systems' "slushware." In addition, their version of Windows 3.0 has taken care of specific Zenith Data Systems configuration commands by including them right in the SETUP program. This is what Windows 3.0 can and does do for other machines, but not in Zenith Data Systems machines due to the advanced engineering regarding faster speed and better graphic quality. There is also a modified SWAPFILE.EXE program in Windows 3.0 which creates a permanent virtual memory file on the hard disk for program swapping, but has been modified to allow recognition of MS-Dos 3.3+ disk partitions which are greater than 32 MB.

Also, their version of Windows has been adjusted to compensate and support the proper grey-scale shading when using a VGA portable based on Chips and Technologies 82C455 VGA/LCD video controller used in the SupersPort 286e, SupersPort 386SX and the TurbosPort 386e. If necessary, in the laptops, you can select the standard microsoft VGA display driver instead of the VGA/LCD driver by requesting "VGA" for the "DISPLAY" in SETUP. Of course for laptops without a VGA display,

you will need to select "CGA" as the "DISPLAY" in the windows SETUP program.

Another adjustment in the Zenith Data Systems version of Windows lies in SETUP itself. The SETUP program has been modified to identify three more classes of Zenith Data Systems computers and performs various configurations depending on which type is selected. All 80386-based Zenith Data Systems computers get the line "EMMEXCLUDE =E000-EFFF in the [386 ENH] section of the SYSTEM.INI file. This line instructs Windows 3.0 to avoid the E000-EFFF area of memory for its EMS paging because this memory range is used for video RAM. All portables get the KBDZDS.DRV configured as their keyboard driver for handling the unique requirements of the function keys (FN-SLOW,FN-FAST, AND FN-CRT/LCD). I should point out that if it is ever necessary to change back to the standard Microsoft keyboard driver, the line in Windows .INI should be changed to read KBD.DRV=KBD.DRV. If the change is made, you must copy the keyboard driver from Microsoft distributed disks to your Windows sub-directory using the EXPAND.EXE program located on disk number two. It must also be pointed out that SETUP, with all the modifications, is equipped to automatically identify the 84-key, AT keyboard.

With these various hardware and software changes, your Z-286, Z-386 or better, should be at its maximum utilization level in Windows 3.0. If you don't have Windows 3.0, please contact your nearest Zenith Data Systems dealer. Low-cost upgrades are available if you are a registered user of an earlier version of Windows. So what are you waiting for, the Database area on COM1 BBS already has a section devoted solely to Windows Shareware applications. Once again, if it hadn't been for some useful information coming from fellow huggies, REMark readers wouldn't have had a chance to benefit from it. I wish to extend a special thanks to Larry Bollman, Phil Curtis, Kurt Ehrardt, Richard Haertle, Warren Kotaitshek, Pat Swayne, and Anthony Tontodoneto for contributing to this article. Remember, if you have any other questions or comments, you know which Port-O-Call to use! ✱



Quality Heath/Zenith upgrades

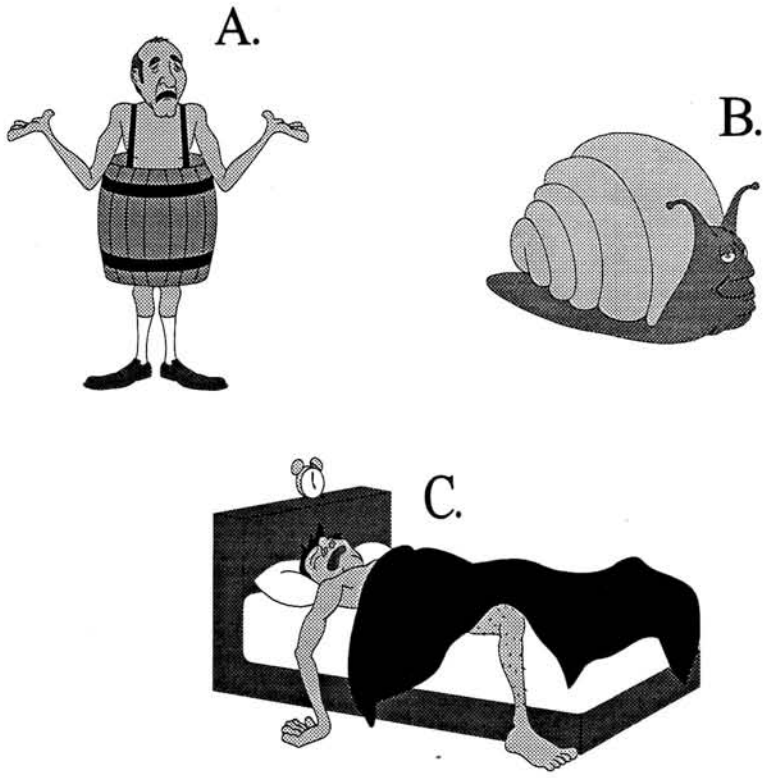
- 150 Speed Mod $34.95, MT PAL 704k RAM $19.95, Super PAL 1.2 meg RAM $29
- SmartWatch clock for H/Z Computers $29
- H89: WIN89 20 meg Hard Disk $329, WIN89 Interface $149; Speed Mod $34.95
- NZCPR $59, MT Modem (CP/M,HDOS) $15
- Complete Line of EVEREX Systems.
  - * Z-248 3 meg EMS RAM Card $99
  - * Z-150 2 meg EMS RAM Card $69
  - * ViewPoint VGA (8/16 bit) $199
  - * Serial (2 Opt), Parallel, Game Card $69
- EVERFAX - 24/9600 baud Modem/FAX $259!
- H-100 Speed Mod (7.5 or 8 MHz) $37
- 148 Exp Buss $69, MT148 704k PAL $19.95
- Call or Write for out FREE Catalog.

Micronics Technology

(205)-244-1597 BBS: (205)-244-0192
Suite 159, 54 Dalraida Road
Montgomery, AL 36109

Reader Service #114

Is this how
you've pictured
the REMark staff lately?



Answer:
D. (All of the Above)

Sorry 'bout the late issues!

Getting To Know QUATTRO PRO
(You CAN Try This at Home)

Richard J. O'Connor
848 Fenske Drive NE
Olympia, WA 98506

Introduction

Did you ever watch a desktop computer user reach past a calculator in order to use the keyboard to painstakingly enter a small table of numbers into a spreadsheet to add them together? Or build a wonderful 8-color pie chart that depicts the relative numbers of PC's, PC-XT's, AT's, 386 machines and Macintoshes in their company, which is then printed on a 9-pin dot matrix printer in glorious black-and-white? If so, you've probably witnessed an electronic spreadsheet in action; the marvel of the 1980's. You may have also harbored serious concerns about the mental state of such users.

Spreadsheets to consultants are a lot like liver-and-onions, in that there aren't too many neutral observers. People like me either love to see them in use or groan inwardly at the sight of yet another runaway model mired in the tangles of circular recalculation. Are spreadsheets the Heroes of the Average User, who knows no programming languages, and has no other way to build applications? Or are they simply a plot to sell more RAM chips, numeric co-processors, and after-market "Your Spreadsheet Explained!" books at $25 a pop?

Yes. They most certainly are. All this, and much more!

Spreadsheets...Love 'Em? Leave 'Em?

I've had a love-hate relationship with spreadsheets since the beginning. It was several years after Lotus 123 came out before I finally used a spreadsheet for anything really "worthwhile." Judging from some of the creations I've helped debug as a consultant, I'm not alone here. I first used spreadsheets because they were fun to play with; as a programmer, I had other ways to build applications to solve "real" problems. Yet, once I built a spreadsheet to post monthly expenditures for the research projects I manage, I realized that I would never go back to manual means. And the entire model was up and running far sooner than I could have built it in any language I know.

For initial analysis, summarization, and trend detection in small-to-medium sized datasets, modern spreadsheets are the tool of choice. Beyond this scope, it wouldn't hurt to have your local expert help you with a reality check on the problem you need to solve. And if you have the opportunity to choose your own spreadsheet tool, you should strongly consider the offerings from Borland International, Quattro (version 1) and Quattro Pro.

Where We're Going

In this article, I'd like to show you how to get started using Quattro Pro, a modern spreadsheet with some surprising capabilities. I'll start with a quick look at the history of the Quattro products, with special emphasis on the dual marketing strategy Borland has adopted for the Quattro Line. Then, I'll summarize some of the improvements that added so-called "next generation" features to Quattro Pro. Finally, I'll describe in detail a typical (?) home application to show you how to put Quattro Pro to work. Everything that follows was developed on a 12 MHz Z-248 with 1 MB RAM, a 40 MB hard disk, ZCM-1492 VGA monitor, and an Epson FX86e 9-pin dot matrix printer.

Borland, Quattro and Quattro Pro

Borland International joined the desktop computer spreadsheet business in 1987 with Quattro, a Lotus 123-compatible spreadsheet that was both faster and more agile than Lotus. Their aim was to go Lotus "one better," which no doubt inspired the name. Quattro enjoyed immediate success, but incremental improvements by Microsoft to their Excel spreadsheet and Lotus to their own product sent Borland's programmers back to the drawing boards in order to regain the advantage Quattro once held.

In June, 1989, Borland purchased the rights to a spreadsheet called "Surpass," a product sporting such technological breakthroughs as multiple windows, spreadsheet links, and intelligent recalculation. With these capabilities in hand, Borland was able to quickly incorporate the improvements they had envisioned for the successor to Quattro, Version 1.

In December, 1989, Quattro Pro was introduced, following an advertising blitz that was strong even by Silicon Valley standards. Test mailings were made to current owners of Borland software, including some who never bought Quattro, but had purchased one of the Turbo Language products. I'm married to a serious Turbo Pascal fan, and when Cathy received her offer to buy Quattro Pro (a $495 retail product) for $99, we jumped at the chance! Other Borland mailings inundated current owners of competing spreadsheet products, who were given a chance to swap their copies of Lotus, SuperCalc, or Excel (along with $99) for a copy of Quattro Pro.

With this introduction, Borland divided potential spreadsheet users into two camps, which they term "retail" and "corporate". The original Quattro, which lacks "next generation" features such as Quattro Pro's spreadsheet publishing capabilities, is still on the market, targeted for the price-sensitive home user, while Quattro Pro is positioned to attract those who need high-end graphics, file consolidation and database capabilities, presumably in business environments.

I can't help but think of the pair as Quattro Pro and Quattro Lite. It works for beer companies; will it work for Borland? Actually, I endorse the concept heartily. One reason I remain a steadfast user of PC WatchWord for my word processing and programming needs is the current absence of a "WordPerfect Lite", with a third less commands than the regular (Version 5.1) release. Be honest; don't you sometimes dream of manuals that were less filling, menu trees that were designed with better taste?

Major Strengths

What does Quattro Pro do so well? It hides the details of computation, data dependencies, and on-screen format concerns from the user who just wants to get

a few numbers organized in a way that makes sense visually. All modern spreadsheets do an impressive job of this, but Quattro Pro has a unique combination of features that you will enjoy nearly as much as Borland's marketing staff does! For example:

- presentation-quality graphics with slide show capability
- spreadsheet linking and consolidation
- compatibility with Lotus 1-2-3 Versions 2.01 and 2.2
- user interface with pulldown menus and mouse support
- ability to read and write Paradox, Reflex and dBase files

In the example I describe here, we didn't make use of all of these features, but remember; our purpose is "Getting Started!"

Typical Home Applications

Some of the things we've done with spreadsheets at home won't surprise you. Spreadsheets are great at ordinary arithmetic, and we've used one to keep a simple checkbook balance, as well as to post box scores for the coed softball team we play for. Simple forecasting is possible as well; we've projected household budget categories and various loan payoff schedules in order to help us plan major purchases. (Note that they have yet to build the spreadsheet that can KEEP you on a budget, once you concoct one.) You can also use graphics to explore things such as spending patterns or the decline in your batting average as the summer progresses!

For today's example, I'll describe an application we used Quattro Pro for recently which used both expense posting and visual displays in order to shed light on what we termed . . .

The Case For Constant Power Company Income

Have you ever noticed that you seem to get notices of increased billing rates just as you enter the months when you consume less power? A friend had a theory for this, which he termed The Case For Constant Power Company Income. He felt that power companies had a strong need for dependable income, and so they tied electric power rates inversely to demand in order to insure a steady flow of cash. How else, he would claim, can we explain the fact that the more energy we conserve, the more the price goes up?

We got a good laugh from this, but then it occurred to me that it might be interesting to look at three years of power usage at our place, and see if the total cost (including all those seasonal adjustment factors) was in sync with actual usage. Were there any times when power costs diverged from usage? That might be fun to bring up at a utility company board meeting! But before I make a fool of myself in public, I generally like to get my facts straight and in some sort of presentable form. This is where we decided to experiment with Quattro Pro.

The first step involved locating three years worth of bimonthly power bills. (Thanks to Cathy's strong organizational skills, that only took a few minutes!) We found records from 1987 through 1989, and began to build a table.

Quattro Pro lets you choose the interface you'll work with. We prefer the default, with horizontal pulldown menus at the top of the screen, but you can also use a Lotus-like interface, or one that resembles the Quattro 1 vertical menu style. Entering Quattro Pro, once you've run through the installation procedure, is done with a Q <CR>.

Building Data Tables

Since this was to be a joint learning operation, Cathy and I took turns at the keyboard and manuals. As it turns out, the one with the manuals has a lot less work to do! We started with a title POWER USAGE AND COST, which we typed into cell C1. You move through this spreadsheet with the arrow keys, just like you would expect. We dropped down to D4 to enter the year 1987, and then over to place headings like "Jan-Feb" in cells B5 through G5. We decided to place side titles "KWH" (Kilowatt hours) and Cost in cells A6 and A7, and then entered the power figures in B6 through G6, and the costs in B7 through G7. Costs should be dollars and cents, so we changed the format of those numbers to Currency style, two decimal places with the /SNC2 command applied to range B7 through G7. The original font we chose (Bitstream Swiss 12 point Black) was fine for the table body, but we picked a larger font (Bitstream Swiss 18 point) for the title. Again, the Style menu comes in handy; with a simple /SF we found the list of six fonts available, so we chose the larger font and indicated the cell to change. Time for a first look; we had already configured Quattro Pro for our printer, so we used the /Print menu to indicate the block to be printed and the destination. At this point, our table looked like Figure 1. This seemed like a good time to save the initial work, using the familiar /FS from the /File menu.

The figures for 1988 and 1989 were added below the 1987 figures in much the same way. As in Lotus 1-2-3, you can use a Copy command from the /Edit menu to duplicate the cell titles in B5 through G5 to the ranges B10..G10 and B15..G15 to save typing. We added the same side titles, reformatted the cost figures into Currency style for the two additional years, adjusted the heading, and printed the table you see in Figure 2.

So far, this was all very standard work. But Cathy wanted to explore some of the aspects of Quattro Pro that make it different. Yes . . . we were ready to enter the bold, new world of "spreadsheet publishing," in which you create tables that don't look like they came from a typewriter. With Quattro Pro, you can draw boxes of various styles and sizes around any of your data blocks for emphasis, and even add shading. Note that we haven't done any graphics yet; this is simply dressing up a spreadsheet table with visual improvements.

Boxes turned out to be easy. We entered the line-drawing section of the /Style menu with /SL, and chose a Top Single line for B4..G4, a Bottom Single line for B5..G5, a Left Single line for B4..B5, and a Right Single line for G4..G5 to box in the 1987 headings. We repeated this pattern for B9-G9, B10-G10 and B14-G14, B15-G15 to get the same effect for 1988 and 1989. For further clarity, we tried shading the data for Mar-Apr, Jul-Aug and Nov-Dec, which alternated clear and shaded data blocks for each year. The /SSG command chooses Gray Shading from the /Style menu, which we then applied to C5..C7, E5..E7, G5..G7 (as well as rows 10-12 and 15-17). Note that you can choose Black shading if you like, but I would only recommend that to those with data to hide! Figure 3 shows the final table, ready to bring to the meeting.

POWER USAGE AND COST

			1987			
	Jan-Feb	Mar-Apr	May-Jun	Jul-Aug	Sep-Oct	Nov-Dec
KWH	2550	2150	2120	1780	2190	2490
Cost	$124.76	$107.63	$105.18	$87.61	$105.88	$118.90

Figure 1

POWER USAGE AND COST
(1987-1989)

	Jan-Feb	Mar-Apr	May-Jun	Jul-Aug	Sep-Oct	Nov-Dec
			1987			
KWH	2550	2150	2120	1780	2190	2490
Cost	$124.76	$107.63	$105.18	$87.61	$105.88	$118.90
			1988			
KWH	3200	2720	2350	2130	2190	2630
Cost	$156.27	$132.67	$115.39	$108.50	$110.88	$132.74
			1989			
KWH	3190	2370	2150	1890	1800	2420
Cost	$165.07	$124.21	$108.90	$92.39	$90.00	$124.89

Figure 2

POWER USAGE AND COST
(1987-1989)

	Jan-Feb	Mar-Apr	May-Jun	Jul-Aug	Sep-Oct	Nov-Dec
			1987			
KWH	2550	2150	2120	1780	2190	2490
Cost	$124.76	$107.63	$105.18	$87.61	$105.88	$118.90
			1988			
KWH	3200	2720	2350	2130	2190	2630
Cost	$156.27	$132.67	$115.39	$108.50	$110.88	$132.74
			1989			
KWH	3190	2370	2150	1890	1800	2420
Cost	$165.07	$124.21	$108.90	$92.39	$90.00	$124.89

Figure 3

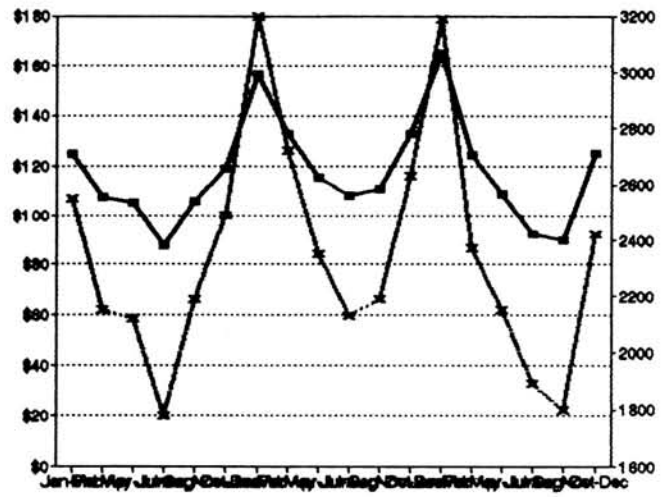


Figure 4

POWER USAGE AND COST
(1987-1989)

	Jan-Feb	Mar-Apr	May-Jun	Jul-Aug	Sep-Oct	Nov-Dec
			1987			
KWH	2550	2150	2120	1780	2190	2490
Cost	$124.76	$107.63	$105.18	$87.61	$105.88	$118.90
			1988			
KWH	3200	2720	2350	2130	2190	2630
Cost	$156.27	$132.67	$115.39	$108.50	$110.88	$132.74
			1989			
KWH	3190	2370	2150	1890	1800	2420
Cost	$165.07	$124.21	$108.90	$92.39	$90.00	$124.89

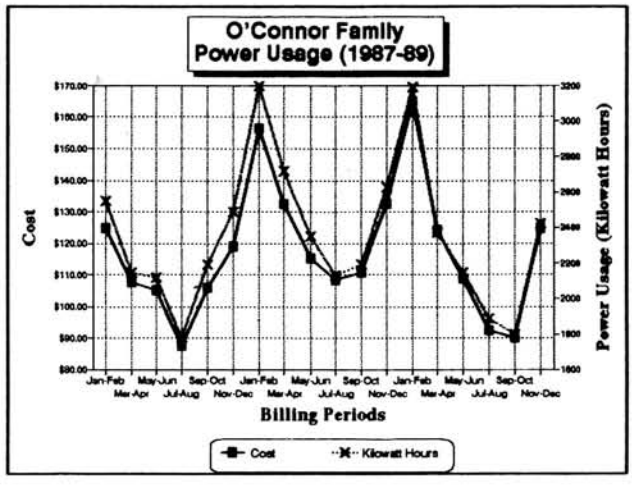


Figure 5

Note that it was necessary to extend the print block to column H in order to get the G column right side lines to print.

Now For the Pictures . . .

Remember that the point of this work was to search for a trend, and many people find it easier to find trends in pictures, rather than in neat tables of numbers. By using line graphs to plot power usage over the three-year time period, along with the associated costs, we hoped to notice any unusual pattern that emerged.

When you plot a series of numbers using a spreadsheet, you need to arrange all of the numbers in a consecutive range of cells. Spreadsheets can't gracefully plot discontinuous ranges, which means the layout we chose for maximum visual effect (Figure 3) couldn't be used to generate a three-year plot. How do you handle this? Use the Oldest Trick in the Theater; "hide the props off-stage". We used the /ECopy command again to copy the three sets of monthly headings into columns I5..Z5, the power usage figures into I6..Z6, and the costs into I7..Z7. Now it was a simple matter of giving Quattro Pro enough information to draw a plot using these numbers.

We chose a Line graph with the /GGL command from the / Graph menu. The series of numbers to be graphed were defined with the /GS command; we defined the costs as the first series, and the power usage figures as the second series. We assigned these series to the Y-axes of the plot by using the Customize command from the / Graph menu. The first series was defined as Primary (left Y-axis) and the second was

defined as Secondary (right Y-axis). The X-axis was assigned the "Jan-Feb" headings from I5..Z5. We tried a cost range of $0-$180 with $20 increments for the left Y-axis and a usage range of 1600-3200 KWH with 200 KWH increments for the right Y-axis just to see what Quattro Pro would produce. At this point, Cathy pressed the <F10> key to View the graph, and Figure 4 was the uninspiring result.

The major problem with Figure 4 is the lack of descriptions. However, sharp-eyed readers will notice the difficulty in comparing costs and usage because of the scaling we chose. While power usage values cover the entire height of the plot, the scale chosen for costs forces all values into the upper half of the graph. There are doubtless other refinements you might wish to make. It was time to roll up the sleeves and polish this picture!

To fix the cost values, we changed the total range to $80-$170, and the increment to $10 with /GYS Manual, choosing Low value=80, High value=170, Increment=10, and then backed up in the menu to the Format command to change the displayed values to Currency with 2 decimals. The power values look fine. The only problem with the X-axis headings is that they're unreadable. The use of alternating tick marks should eliminate this problem; /GXAlternate ticks Yes was all we needed.

Now to describe the picture a bit. The Text command from the /Graph menu has many options for adding descriptions to a graph. We entered the first title line "O'Connor Family", and the second line "Power Usage (1987-1989)", along with an X-axis title "Billing Periods", a Primary Y-axis title "Cost", and a Secondary Y-axis title "Power Usage (Kilowatt Hours)". In order to tell the two different lines apart, we chose Legend entries for the 1st data series "Cost", for the 2nd data series "Kilowatt Hours", and positioned the Legend at the bottom of the graph.

But Wait! There's More!!

At this point, you can also customize the fonts (including size, typeface, style and color) used to display this text on your screen by using /GTF and making choices. Why have a color VGA monitor and not make use of it? So we chose 24 point Swiss Bold in Magenta (!) for the two titles lines, 20 point Dutch Bold Blue for the X- and Y-axes, 12 point Swiss in Red for the Legends, and 10 point Swiss in Black for the Data and Tick Labels. Sorry, I can't show you what that conglomeration looked like in the pages of REMark, but you could always try it at home!

Line style choices were next, and we used the Customize Series option to set the line color for the 1st series to Light Blue and for the 2nd series to Light Red, after some experimentation. We chose a Heavy Solid line style with Filled Square markers for the 1st series, and a Heavy Dotted line style with X markers for the second series. The Formats option at that point in the menu allowed us to plot both lines and symbols, and we added a background of both horizontal and vertical grid lines for reference with the /GOGB command. At this point, the Graph View key <F10> was getting considerable use, but we were starting to like the results!

While in the /Graph Overall menu for Grid lines, we noticed an Outlines submenu that allowed us to place a shadow box around the title; at this point, we were getting hooked on special effects (I'm such an easy sell)! Finally, we found a way to place a box around the whole graph; we chose a Thick Line box, and saved the current version, bells, whistles, and confetti.

The Finished Product

At this point, Cathy looked up from the manual and announced that she had found a way to get both the graph and the table of figures finished earlier printed on a single page. We named the graph with the /Graph Name Create command, inserted it into the spreadsheet below the table with a simple /Graph Insert A19..G45, and the project was complete (see Figure 5). On a single sheet, we now have figures presented in two different ways (for the left-brainers as well as the right-brainers) with a style reminiscent of laser-printer output. Not bad for a 9-pin printer!

Wrapup

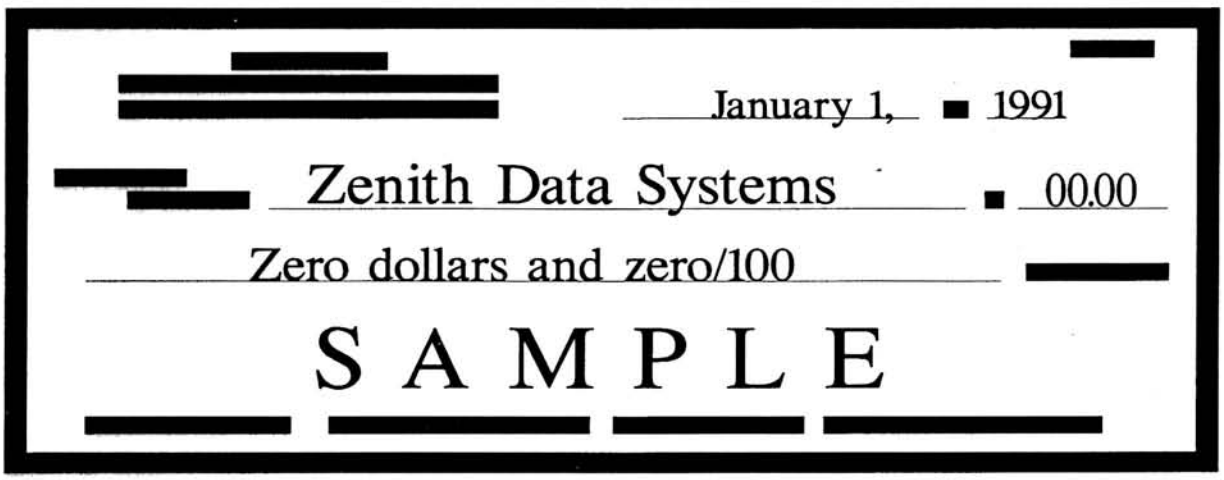
When I looked at the results, I realized I didn't have much of a case for shenanigans on the part of the power company, so I never did attend that meeting. But if I had, at least I would have gone in with a reasonable-looking handout. If nothing else, Quattro Pro can help you get the most out of your humble dot matrix printer. And until laser printers become as cheap as microwave ovens, that will have to do for me!

The logical consistency of Quattro Pro's menu/submenu interface makes it easy to learn, while offering a lot of optional power down some of those menu trees. It may be designed for the power user and targeted for the "corporate" market, but if you can afford it, don't let that keep you from "trying this at home".

Products Mentioned

Quattro, Version 1
Quattro Pro
Borland International, Inc.
1800 Green Hills Road
P.O. Box 660001
Scotts Valley, CA 95066-0001

All Checks must be made out to
Zenith Data Systems



QUIKDATA - 14 YEARS OF H/Z SUPPORT!

YOUR H/Z RAM & DRIVE EXPERTS!

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. Our EXP12 **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. 150ns 256K DRAMs are $1.79 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.

EAZYRAM - $95 Upgrades EaZy PC from 512 to 640K

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!!

Z248/12, Z286LP RAM UPGRADE Z605-1 consists of 2MB SIMM 80ns RAM kits to upgrade your H/Z systems.
Z605-1 - $179

Z386/25, Z386/33, Z386 EISA 2MB SIMM 80ns UPGRADE to add increments of 2MB to these systems.
38633M-2 - $295

SIP1X9-8 - $79 1 megabyte x 1 SIPP module, 80ns

WINCHESTER UPGRADE KITS

PCW20 - $249 Complete MFM winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes **21 meg** formatted half-height Segate ST-125 37ms drive, WD (DTC for Z148) controller, cable set, doc.

PCW30 - $295 32 meg with 38ms Segate ST-138.
PCW40 - $319 42 meg with 28ms Segate ST-251-1.
PCW80 - $595 80 meg with Segate ST-4096 full size drive.

We also have the DTC controllers ($59) and daughter board expansions ($65) to place a hard drive in the **H/Z148** computers.

BARE WINCHESTER DRIVES/CONTROLLERS

ST-125 - $219 21 meg 37ms autopark 3.5" drive in 5" frame
ST-138 - $249 32 meg 38ms autopark 3.5" drive in 5" frame
ST-251-1 - $289 42 meg 28ms autopark HH drive
ST-4096 - $559 80 meg 28ms autopark FH drive
WDCON - $59 PC/XT hard drive controller board
WDATCONF - $129 1:1 interleave HD/floppy controller for AT's
ST-2182 - $995 **160MB ESDI** half-height drive
DTC-6280 - $149 ESDI HD/floppy controller
IDE hard drives and controllers for AT computers:
ST08 - $54 IDE controller for 2 hard drives and 2 floppies
ST-1090 - $495 80mb 3.5" 15ms IDE hard drive
ST-126 - $595 111mb 3.5" 15ms IDE hard drive
ST-157A - $279 40MB 3.5" 15ms IDE hard drive

FLOPPY DRIVES

T28M - $159 2.8MB 3.5" floppy
MF501 - $71 5" 360K DS/DD drive
MF504 - $81 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 80 trk DS refurb (H8/H89 type)
TM100-2R - $69 40 trk DS refurb (H8/H89 PC type)

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

ADD ANY FLOPPY TO ANY LAPTOP OR PC WITH A PARALLEL PORT easily and inexpensively. With Backpack, you simply connect the external unit to your parallel printer port (do not lose printer function), install software, and away you go! No expansion boxes needed for laptops, and no slots required. Too easy to be true, but it is. Want a 2.8mb/1.4mb/720K floppy on your MinisPort? Want to add a 1.2 meg to your laptop? Want to add an additional drive to your desktop? Plug it in and go. 2.8MB 3.5" version will read and write 2.8 megabyte format, 1.4 meg format and 720K format. 1.2 meg version will read 1.2 meg and 360K floppies, and write 1.2 meg format.
BPACK2.8 - $319 2.8 Mb version
BPACK1.4 - $269 1.4Meg/720K version
BPACK1.2 - $269 1.2/360K version
BPACK360 - $269 360K only version

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 BIOS ROM upgrades for most H/Z PC/XT/AT computers, spike protection filters, backup power supplies, tape backup units, modems, printers, cables and ribbons, disk drives and diskettes of all types, external hard drive and floppy drive enclosures, cables and connectors, laptop batteries, CMOS batteries, video monitors and video cards, memory cards, memory chips and ICs, joysticks, accessory cards, serial and bus mouse, a variety of useful and most popular software and much more! **Need a PC/XT/AT computer?** Tell us what you want and we will quote you a price on one of our custom assembled QD computers made up to your exact specifications! Price is lower than you think for a high quality system.

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

The Alps Allegro Printer Series:

A User's View

Tom Bing
2755 Carolyn Drive
Smyrna, GA 30080

Laser and inkjet printers have center stage now. They produce eye-catching output that is camera-ready for publication. They are speedy and silent when producing graphics images, quite unlike their dot-matrix predecessors. Why would anybody want a dot-matrix printer any more?

There are two main reasons: cost and versatility. While inkjet printers of 300 dpi quality cost over $600 and lasers go for $1,000 and up, a high-quality 24-pin dot matrix that produces draft and letter-quality text can be had for about $300. What's more, the dot-matrix can handle continuous paper, labels, single sheets, and envelopes. Quite often, the paper handling add-ons for other types of printers cost more than an entire dot-matrix printer! Also, the dot-matrix printer can produce quite acceptable graphic images if you don't need scads of pictures per day. Admittedly, they're not laser quality and they tie up your printer for a few minutes per page, but they're far better than you might expect from an impact printer.

If you need one printer to do it all and you don't have large-volume graphics requirements, I think a printer like the Allegro deserves serious consideration. This article is based on the Allegro 24. However, Alps introduced the Allegro 500 in May 1990, offering additional features for the same price. I own and use an Allegro 24; I know the Model 500 only from the manufacturer's spec sheets. I'll try to point out some of the differences in features between the 24 and the 500 as I go along. There is another new model called the 500XT which can take wide (16-inch) paper, but we'll focus on the Model 24 and the Model 500.

In particular, why did *I* go out and buy an Allegro 24? First of all, I wanted Epson LQ compatibility. I write primarily about PC software, and I wanted a printer that would print screen images accurately. With the LQ-type printers, double-line boxes on the screen come out as double lines on the paper. Some 9-pin printers "fudge" and print just a single line. Even with inkjet and laser printers, Epson compatibility is often available only as an extra-cost cartridge or TSR program.

Second, I needed one printer to "do it all". In addition to screen dumps, I needed to print letters and envelopes and graphics from programs like Calendar Creator Plus, and to do labels as well. I wanted all this capability in an "everything included" product, without a lot of extra cost options. My old Diablo 620 daisywheel does a beautiful job on text-only documents, but it knows nothing about the IBM graphics character set, and changing from continuous paper to single sheets is a pain. Also, the Diablo uses friction feed instead of a forms tractor to pull the paper through. As a result, the paper gradually creeps sideways on documents of several pages. I also own an old Okidata Microline 82A that has served me well for six years, but it is a 9-pin unit that's not Epson compatible. Both of these are also fairly noisy in operation.

A Novel Design

I suppose that buying by mail is something of a gamble. However, I had seen Alps ads in major PC magazines for several months, and I was intrigued by the unique design of the Allegro 24. As you can see from Figure 1, the printer stands on legs which raise it about two inches off the table. The legs also fold down to allow it to rest directly on the table. This design allows some variations in paper placement and makes a separate printer stand unnecessary. I also liked the straight-through paper path, unlike the roll (platen) used by conventional printers. After seeing a sample of the Allegro's print quality in an article in *PC Week*, I decided to buy one. I made the rounds of dealers near my home, but I have yet to see an Allegro on display in a store around here. Consequently, I ordered one from Midwest Micro-Peripherals in Fletcher, Ohio (phone 800-423-8215). Within eight days, my printer arrived. Midwest's price was $298.00 plus shipping.

Price and Features

The price raises an interesting point: what does the future hold for prices and features in the 24-pin market? I think printer makers will compete by upgrading features rather than cutting prices. For instance, Alps has given the Model 500 more standard features than the Model 24, yet they sell for the same price. In local discount stores, comparable printers have been in the $290 – $315 range lately. I expect manufacturers to attract customers by focusing on feature differences, leaving price competition to distributors. Perhaps Brand A will offer a few more fonts, more convenient controls, or a bigger print

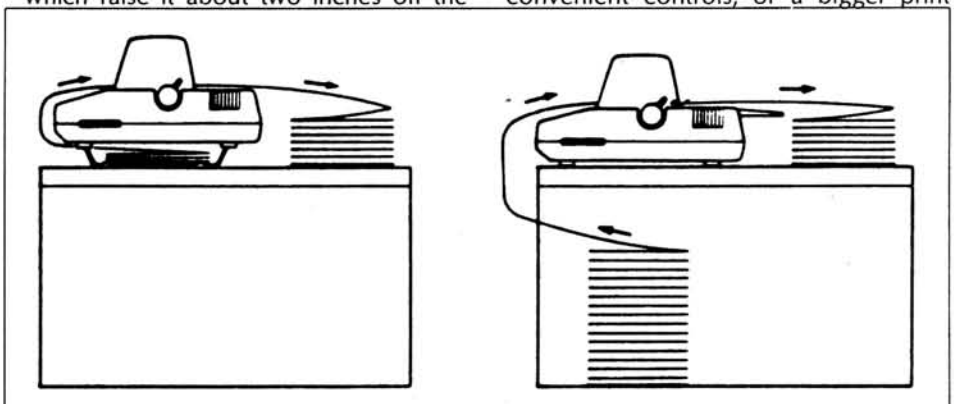


Figure 1

buffer, than Brand B. In my own case, I chose the Allegro because I was impressed by its print quality, straight paper path, and affordable price. The fact that I wouldn't need a printer stand or paper tray was also a plus. The Model 500 has seven resident fonts versus four in the Model 24; the three added fonts are Gothic Bold, Script, and Prestige Elite. Also, the 500 has a 12K print buffer, versus 7K in the older model. The graphics resolution in dots per inch is 360 by 180 for the Model 24, and 360 by 360 for the Model 500. There are other improvements in the newer model, but these three stand out.

First Impressions

When the Allegro arrived via UPS, the first thing I noticed was the handle on the box, allowing it to be carried like a suitcase. That little handle made a positive first impression which was reinforced as I set the printer up and began to use it. Unpacking was a breeze. The manual was thorough and clear in describing the initial setup. It's easy to find things in the manual because of its detailed index and table of contents. The printing is crisp and clear. There are several easy-to-follow illustrations for such tasks as installing the ribbon cartridge (Figure 2). This cartridge provided the only challenge during setup, but that was minor; it took a little patience to seat it correctly. I suppose the "doughnut style" ribbon cartridges of other printers are easier to handle, but this one seems to be designed for long life; only the next few months will tell. The printhead is mounted vertically and the pins fire straight down. The top cover completely shrouds the printhead, and the foam-rubber liner muffles a good deal of the noise.

Paper Handling

One area where the Allegro 24 really shines is paper handling. Many 24-pin printers feature "paper parking", which allows quick alternation between continuous forms, single sheets, or envelopes. The Allegro's paper parking is excellent. Instead of removing continuous pages to print an envelope, I simply press "SELECT" on the front panel (Figure 3) to take the printer off-line, then "PARK". The Allegro backs out the continuous paper, but doesn't turn it loose. By flipping the right-hand control knob from tractor to friction feed, the machine will accept and print on single sheets or envelopes (Figure 4). After the last single sheet is printed, I flip back from friction to tractor feed and press "PARK" again. The continuous paper then returns to its "Top Of Form" position, ready to print; no forms or single sheets have been wasted. Even if I do slip up occasionally with continuous paper by pressing the "Form Feed" once too often, I simply save the blank sheets and use them as described above to print one-page letters.

Another use for paper parking is in resetting top-of-form (TOF). If the paper is an inch or two beyond TOF when the printer is turned on, the correct top-of-form can be set by parking and "unparking" the paper. This default TOF setting will leave about one line (1/6th inch) blank at the top edge of the paper. This position can be adjusted with the "SET TOF" button. First, I press SELECT to take the printer off-line (the SELECT light goes off). After that, I press the SET TOF button. This causes the paper to advance, placing the current top-of-form line next to a marker on the input paper tray. Using the positions of the marker and the paper, the TOF position can be adjusted in very fine increments (1/180th inch) in either direction. When the top-of-form setting is complete, pressing "SELECT" returns the printer to on-line mode and the paper to its newly set TOF position. With the printer on-line, the SET TOF functions as a manual linefeed, as shown on Figure 3.

If you are printing a document on continuous paper, there's no need to waste the sheet after the last printed page just to remove the document. By pressing the TEAR BAR button while the SELECT light is on, the paper will advance so that the bottom perforation is lined up with the edge of the top cover, which serves as a tear bar. The paper will come apart cleanly when pulled up against the tear bar. Pressing TEAR BAR again puts the paper back in the top- of-form position.

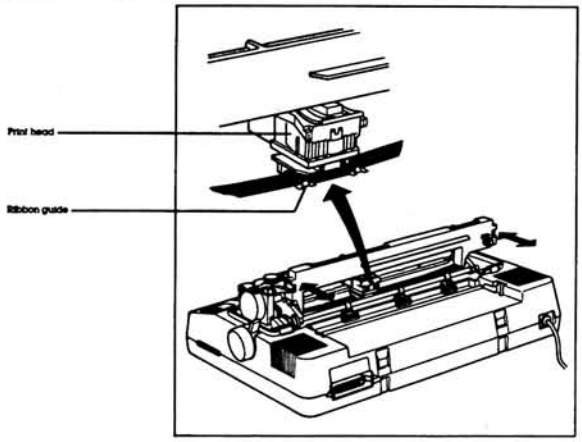


Figure 2

The main problem with paper parking occurs when the single sheets or envelopes are much thicker than the continuous sheets. Then the paper thickness lever inside the printhead cover has to be reset. Otherwise, a thick envelope will push the printer ribbon aside and nothing will print on the envelope. If the single sheets and continuous paper are about the same thickness, no adjustment is needed. This is a minor drawback, and if the user keeps this lever set correctly, the alternation between continuous paper and single sheets should work very well. In fact, my only suggestion for an improved mechanical

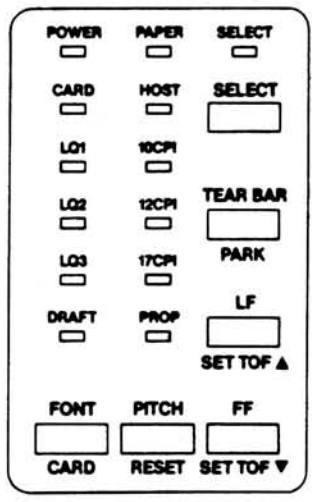


Figure 3

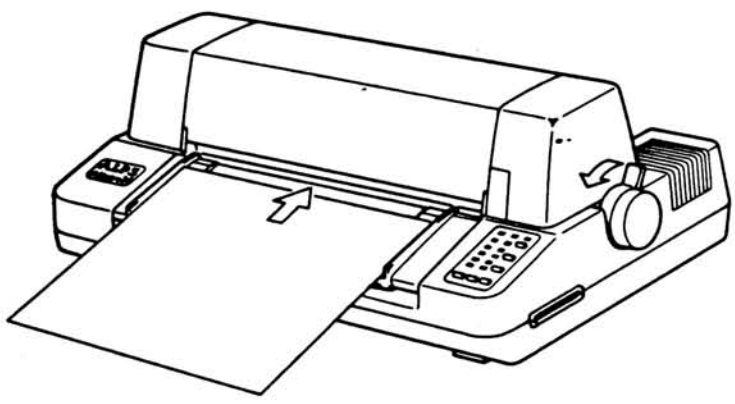


Figure 4

design of the Allegro would be a slide control or thumbwheel that allowed the paper thickness to be changed without removing the cover. Ideally, the area next to the control could be marked and erased to allow changes from thin paper to thick envelopes without guesswork. But even without this enhancement, I've printed letters, envelopes, single sheets, and labels with very good results.

Fonts and Other Choices

The ability to control all printer configuration simply by pressing buttons on the front panel is really a plus. There aren't any DIP switches involved and you don't even have to open a cover. Any of the first three fonts in Figure 5 can be selected by pressing a button, as can horizontal character spacings of 10, 12, or 17 characters per inch (cpi). Three more fonts (Prestige, Orator, and OCR-A) can be used on the Allegro 24 when an IC font card (a $39.00 option) is plugged into the printer. The IC font card fits into a slot on the right side of the printer near the operating panel. The various fonts and spacings can also be selected via escape codes sent from the computer. In contrast, font changes on daisywheel printers require you to open the cover and change printwheels. If WordPerfect is your WP software, additional large-letter fonts are available; see Figure 5. WordPerfect 5.1 includes double-high, double-wide (5 cpi) versions of the Alps letter-quality fonts. WordPerfect also supplies a supplemental font diskette for $10.00, which will add italic and proportional fonts to the standard ones. In Figure 5, fonts which are available via WordPerfect are marked with an asterisk (*). There is also a draft-mode font for high-speed printing (not shown). While it may be possible to generate these big fonts with escape sequences only, I haven't been able to do so. Besides, changing fonts in WordPerfect is simple; just press Control-F8, then 'F'. The IC cards for font and memory expansion are for the Allegro 24 only. Allegro 500 memory expansion is explained further on.

The power-on default settings of the printer are controlled by the 'Memory Mode' settings. The Memory Mode menu is displayed by holding down the SELECT button when the printer is turned on. This menu allows access to 19 default parameters, such as the default font (draft, LQ1, etc.) and whether numeric zeros are printed with a slash or not. You can step through the different menu items by pressing the LF button, and you can change the default value of the item by pressing the FF button. It's as easy as resetting the time on a digital watch. Each change you make will be confirmed by printing the new value. When you're satisfied with the changes, press SELECT and the printer is ready to print. What's more, it will retain these new defaults even when it's turned off. If you buy an optional serial interface for the Allegro, the Memory Mode menu gives you control over the serial data parameters, such as parity and number of stop bits.

Figure 6 is an example of the quality of graphic images which can be printed using the Allegro 24. This particular picture is from the clip art library supplied with WordPerfect 5.1. Of course, the quality of graphic output depends on two things: the resolution obtainable with the printer and the ability of the graphic software to make full use of the printer's capabilities.

The Allegro 24 comes with a 7K print buffer, and an IC memory card can be purchased for $75.00 that increases the buffer size to 39K. MS-DOS allows the size of the buffer for the PRINT command to be as large as 16K. If there's a recurring need to print really huge files, I'd recommend a print spooler program, which will set aside part of your computer's RAM (more than 16K) as a print buffer. The spooler program will probably cost less than the IC memory card. The standard 12K buffer on the Allegro 500 is also expandable to 44K. The Allegro 500 does not use IC cards at all; all fonts are built-in, and memory expansion is by means of a chip set installed in a door in the back of the machine.

Parts and Service

The Allegro comes with a one-year manufacturer's warranty. I asked Midwest Micro about authorized Alps repair centers in my area, and they gave me the names and phone numbers of two companies in metro Atlanta, TRW and Americom. Ribbon cartridges for the Allegro cost $9.95 each. The printhead has a rated life of 100 million dots per pin. The printhead costs $143.87 and can be replaced by the user; instructions are in the manual. I know this sounds expensive, but I checked the prices on printheads for other 24-pin printers, and they ranged from $135 to $225.

Conclusion

I have enjoyed using my Allegro. It has proven to be versatile, well-designed, and reasonably priced. It works well with WordPerfect and handles my needs for Epson LQ-compatible graphics. If I were buying today, of course, I'd buy the Allegro 500, since the price is the same.

1. This is Tiempo (LQ1)
2. This is Gothic (LQ2)
3. This is Courier (LQ3)
4. This is Tiempo proportional*

5. Gothic 5 cpi Double-high*

*6. Courier 6 cpi Double-high Italic**

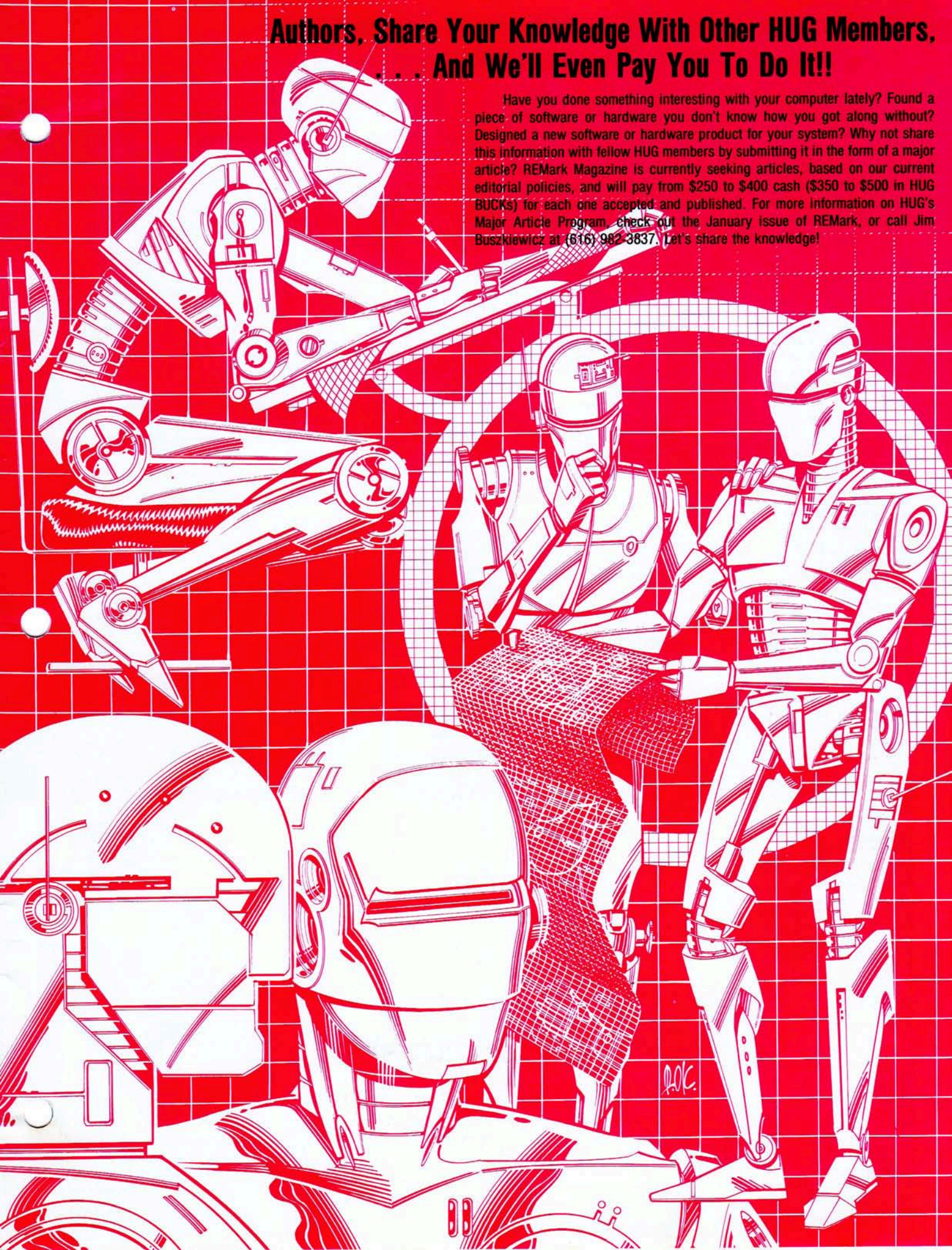
7. Tiempo 5 cpi Double-high*

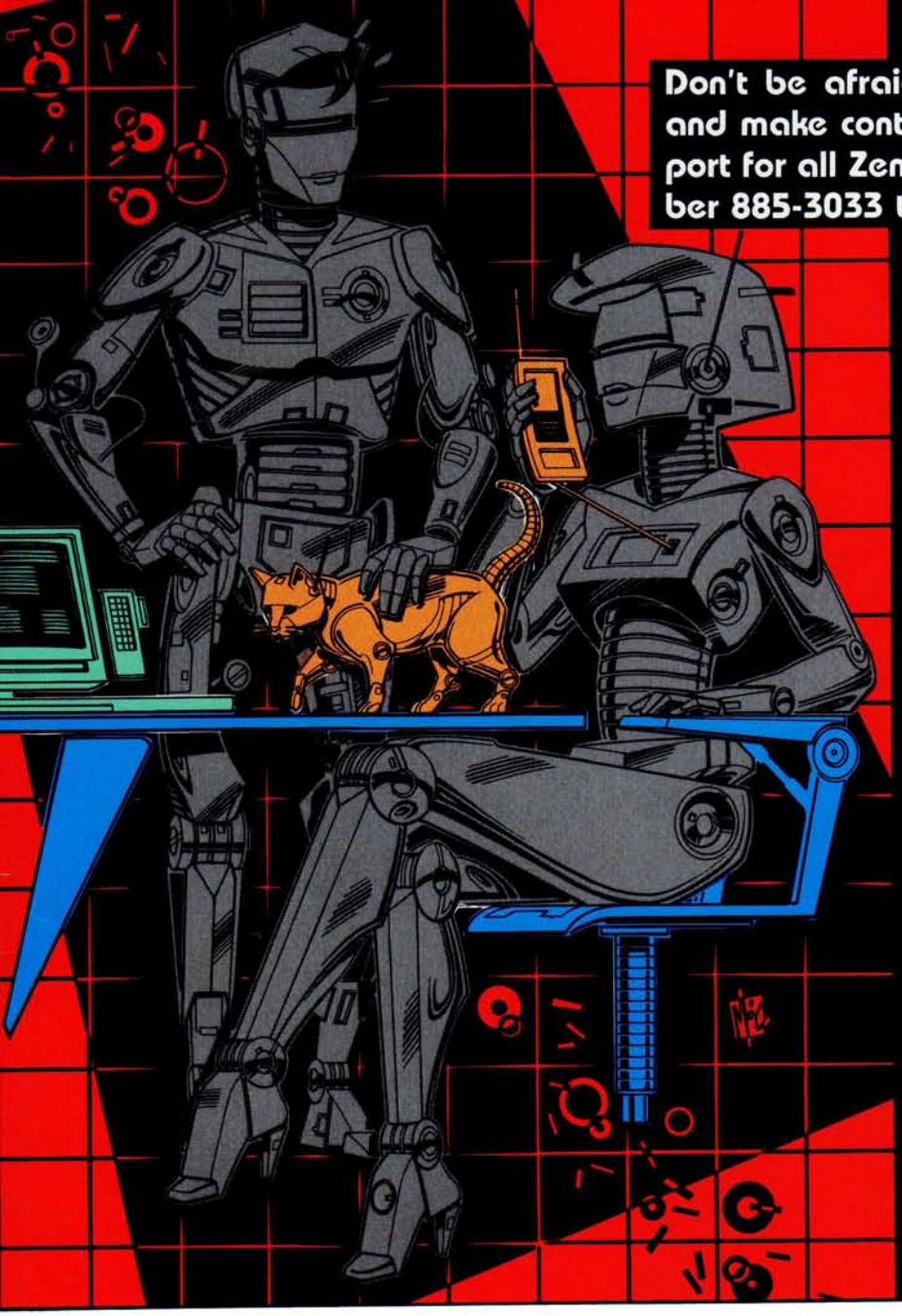
Figure 5

Figure 6

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.

Heath Users' Group

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

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

POSTMASTER: If undeliverable, please do not return.

$2.50
P/N 885-2132