REMark®

Volume 7, Issue 12 • December 1986

P/N 885-2083 Issue 83

$2.50

Official magazine for users of HEATH ZENITH computer equipment.

Announcing...

DOODLER V

Graphics Package

Available for the H/Z-100 and PC Compatibles
Requires: MS-DOS Version 2 or higher.

Features

- Runs on **both** the Z-100 and IBM-pc compatible computers.
- Data **transportability** between Z-100 and PC compatibles.
- Cloth covered binder with **slip case.**
- **130+ page manual** includes tutorial, reference section, and index.
- **Undo feature** allows quick correction of drawing errors.
- Complete support for MS-DOS 2.xx **path names and directories.**
- **Magnify mode** makes pixel-by-pixel editing easy.
- Playback capability for **slide shows** and continuous demonstrations.
- User defineable **macros** simplify repetitive drawing tasks.
- Font editor included with package allows **custom fonts** of any size.
- **Guaranteed support** for any dot matrix printer with bit-graphics capability.

Only $99

Makes a Great Gift!

Available at Heath/Zenith Computer Centers, or Order Direct from...

PAUL F. HERMAN Inc.

Software Graphics Tools
3620 Amazon Drive
New Port Richey, FL 33553

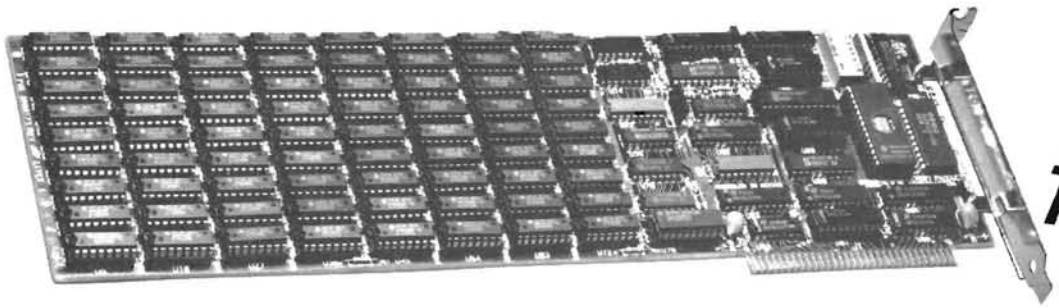
C.O.D. Orders Accepted

813•376•5457

Expand your ZENITH PC Memory with

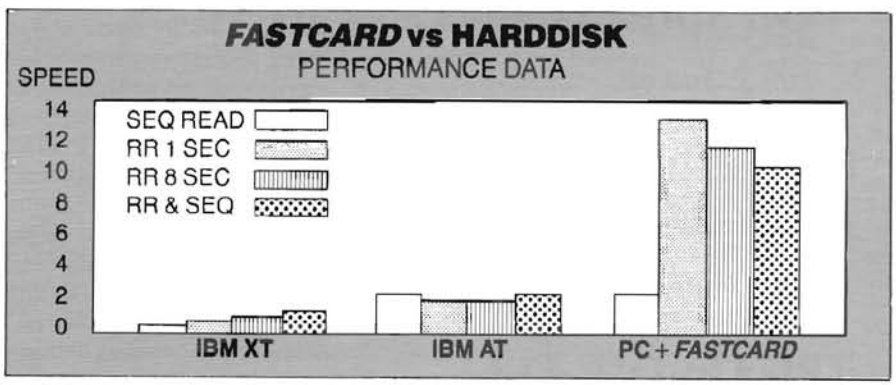
FASTCARD®

fully loaded with 2 *MBYTES* for just $295*



10 DAY FREE TRIAL!

FASTCARD speeds disk I/O by as much as tenfold, *as shown in the following chart.*



This graph shows benchmark test results of a PC equipped with *FASTCARD* versus a PC-XT and PC-AT. These tests were run by a major independent testing laboratory using their standard disk performance benchmark tests.

Specifically designed for compatibility with the 8 MHz Zenith bus, *FASTCARD* provides both extended and expanded memory plus the following standard features:

- Portable between all Zenith models
- Lotus/Intel/Microsoft compatible
- Up to 2MB with Split Memory Mapping to
  - Fill memory to 640K
  - Provide Expanded memory over 640K
- Unique Disk Caching
- Ram Disks (up to 8MB)
- Custom Password Security
- Print Buffering
- Built-in Diagnostics and Automatic Fault Tolerance
- Factory Installed and Tested DRAM s

*Each *FASTCARD III* comes with 2 MBytes of memory for $295. *FASTCARD IV*, available with 2 MBytes, includes serial/parallel ports, game port and a clock calendar for $295.

For additional information, call: PMi PERIPHERAL MARKETING, INC. **(602) 483-7983**
7825 E. EVANS RD., #600, SCOTTSDALE, AZ 85260

Staff

Manager ..... Bob Ellerton
(616) 982-3867

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

Bulletin Board and
Software Developer ... Jim Buszkiewicz
(616) 982-3463

Software Coordinator ..... Nancy Strunk
(616) 982-3838

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

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

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

Contributing Editor ..... William Adney

Contributing Editor ..... Joseph Katz

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

	U.S. Domestic	APO/FPO & All Others
Initial	$20.00	$35.00*
Renewal	$17.00	$30.00*

* U.S. Funds

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

Send Payment to: Heath/Zenith Users' Group
Hilltop Road
St. Joseph, MI 49085
(616) 982-3463

REMark®

Official magazine for users of HEATH ZENITH computer equipment.

Volume 7, Issue 12 • December 1986

Index of Advertisers

This index is provided as an additional service. The publisher does not assume any liability for errors or omissions.

On The Cover: *Shown is X-10 Powerhouse's Home Controller and Modules for your home. For information on these products and more, see Page 67.*

IN MULTIUSER COUNTRY, ALL ROADS LEAD TO SCO

SCO XENIX SYSTEM V

In the heartland of multiuser country, you're licensed for up to 16 users on IBM® PC AT® or PC XT®, respective compatibles, or AT&T PC 6300 Plus! Discover unprecedented price performance per user. Enjoy Multiscreen™ vistas in breathtaking **setcolor.** Explore DOS cross-development. Protect your investment with 8086/80286 and XENIX 3.0 applications compatibility. And bring along your widest range of peripherals!

SCO XENIX-NET XENIX-DOS LAN

Travel through SCO XENIX-NET and witness the future of office computing: networked DOS *and* XENIX PC workstations. Visit a unified environment that is both "multiuser" *and* "LAN." Watch XENIX and DOS systems serve each other and share resources — each doing what it does best — without sacrificing the strengths of either.

SCO uniPATH SNA 3270

Gateway to mainframe territory, SCO uniPATH is a must-see for both VARs and MIS/DP managers. Experience the real power of PCs in corporate distributed processing. Build customized applications with programmatic interfaces, and let your networked multiuser PCs process mainframe data – with session hold and multiple sessions per user, of course.

HOT-LINE SUPPORT
DOCUMENTATION
USER TRAINING
NEXT EXIT

SCO's multiuser traveler's services are unrivaled, and start with SoftCare™ Support, including a toll-free hotline! You can even enjoy 30 days of Introductory SoftCare with the purchase of each product – on us! SCO's excellent documentation is considered a standard for multiuser road guides. And a wide variety of training courses is available for all SCO products.

SCO applications available for all popular UNIX/XENIX systems.

The Santa Cruz Operation, Inc., P.O. Box 1900, Santa Cruz, CA 95061

SCO Professional 1-2-3 WORKALIKE

Take this exit, and you'll swear you're on Route 1-2-3® – until you notice enhanced features and multiuser access that Lotus® users only dream about! But don't worry about culture shock – if you like 1-2-3, you'll *love* SCO Pro. Read and write to 1-2-3 DOS files, enjoy full 1-2-3 functionality and macro support, and rest easy with file locking data security.

SCO FoxBASE dBASE WORKALIKE

As with SCO Professional, another *deja vu* awaits you on this route – this time, dBASE II™ – but once again, it's better, faster, and multiuser! Bring along your dBASE II data files, macros, and expertise – they all work here, too! You'll be sharing dBASE II-compatible data with friends, in record and file locking comfort, right out of the box.

SCO Lyrix WORD PROCESSING

Built especially for UNIX™, "what you see is what you get" SCO Lyrix is a favorite of both new and veteran multiuser travelers. Fully user-configurable, it's a natural attraction to foreign language visitors, as well as system administrators. You'll appreciate advanced WP features such as spelling corrector, and automatic footnotes and section numbering, as well as direct menu access to UNIX and other destinations.

SCO SALES - INFO
1-800-626-UNIX

Call us today to order any of these and other exciting SCO products such as Multiplan®, Informix®, languages, and business graphics. VARs, ISVs and other qualified customers can "mix and match" for cumulative discounts across the entire SCO product line. We look forward to booking your next multiuser tour!

SCO
THE SANTA CRUZ OPERATION

(800) 626-8649
(408) 425-7222
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info

BUGGIN' HUG

Letter To D. Bencivengo On His June '86 Article "H-150 Speed-up Modification"

Dear D. Bencivengo:

I have installed the H-150 speed-up using your circuit board using a 20 MHz crystal and the replacement DMA and BUS controllers with heat sinks per your recommendations with one exception: I could only locate the plastic version of the NEC 8237 (PN D8237AC-5). Although this chip works better than the original AMD chip, I am wondering if it could be the cause of my one remaining problem. At the 6.67 MHz clock speed, I am unable to copy floppy files using COPY. The error message is "DATA ERROR WRITING DRIVE B". The confusing thing is that other forms of disk operation work ok (executing programs, reading data from disk into memory and writing data from memory to disk). The error appears regardless of whether I am copying to drive A or B, so it is not the floppy unit itself. Any insights would be greatly appreciated.

Another matter concerns the applicability of this modification for H-150 owners that have installed a 5 MHz 8087 math co-processor chip. What is the track record of this chip at the higher clock rates? Can any additional heat generated on the chip be reduced effectively by a heat sink, or is heat not a problem with the 8087? Will the higher clock rates stress the 8087 all the time, or just when programs using 8087 op codes are being run, and what would be the anticipated impact to 8087 chip lifetime. Finally, what would be a good test to insure that the 8087 is functioning properly (my understanding is that this chip tends to fail gradually with increasingly frequent intermittent failures). Since the 8087 is an expensive chip, I am sure that many H-150 owners would want to trade-off the pros and cons of this speed-up with regard to their particular 8087 configuration, as well as use the above information in selection of the appropriate crystal frequency.

Sincerely,

Donald G. Harley
P.O. Box 24
San Martin, CA 95046

Is EasyPC Really Easy?

Dear HUG:

In the year I have been reading REMark magazine I have always found it to be a valuable resource and its articles to be generally free of misleading and biased reporting . . . until the August 1986 issue arrived. Of particular concern was LtColonel Somer's 'review' of the UCI EasyPC board for the Zenith 100 series computers. After reading the review, I question what is *easy* about the EasyPC if one has to make all of the modifications Somers indicates.

When it came time for me to choose between the UCI board(s) and the Gemini board(!), I chose, for my Zenith 118 and 768kb RAM, the Gemini. I use the Zenith side more than I use the IBM side and so I wanted to keep as many S100 slots as possible. Including the disk controller card, a winchester controller, and two UCI cards, only one S100 slot would remain available if I had chosen the UCI system. The Gemini board allows me to retain three slots.

While I also use an NEC V20 chip, I did not have to wait for a 'new ROM version', I merely plugged the chip into the Gemini board when I installed the board. Instead of getting an 8087 daughter board from UCI for $80, I can plug an 8087 right onto the original Gemini board. I didn't have to make any 'modifications' for highlighting for my C.Itoh RGB monitor.

Perhaps best of all is the revelation that D.E.L., makers of the Gemini board, is currently developing firmware to allow the use of IBM short cards. I don't intend to pay $395 for an Easy Memory board when the Gemini upgrade will allow me to use the $60 (or so) IBM memory cards and then I can spend my money on the chips to fill it. Besides, if Somers adds the Easy Memory card, all his S100 slots should be up and I will still have two.

The most frustrating item in his article is his conclusion that 'comparing the Gemini board to EasyPC is like comparing a Volkswagen to a Porsche'. Where, in his article, is *any* comparative study done? At least William Adney had access to both systems before he suggested the Gemini board. It should be the job of REMark editors to watch for such spurious suppositions from, even well-meaning, writers. As for the conclusion itself, I would rather use a dependable and affordable machine like a Volkswagen (and Gemini board), than spend my meager earnings for over-priced accessories and unexpected modifications.

Terry Carlisle
Box 676 USAF RMC
APO, NY 09220

Results Are Fascinating

Dear HUG:

I could not help but do a double-take upon reading Joseph Katz' 'Mainstream Computing' column in the August '86 issue. Mr. Katz advises not to use the NEC V20 on the "dual-speed H/Z-158, which uses an 8088-2 microprocessor". Well, the 8088-2 is just an 8 MHz version of the 8088. Replacing it with an 8 MHz V20 (-8) works just fine. I know, I have been running my H-158 with an NEC V20 for some time now, and although I haven't reached warp speed with it, the results have been "fascinating".

Yours truly,

E. L. Becquer
4338 SW 147 Court
Miami, FL 33185

Money Well Spent

Dear HUG:

When I received my first installment of REMark in July I was, to put it mildly, less than impressed. The magazine was full of all kinds of 'stuff', but I saw nothing that was of interest to me. Then came the August issue.

With one article I felt that the money for membership was well spent. Pat Swayne's 'Clock Watcher's Delight Revisited' (August '86) solved a problem I have had since I began using MS-DOS

based systems. Being relatively new to the micro environment (I started out on PRIME minis using the PICK operating system), I am not yet certain of where to look for the various utilities that make life so much more pleasurable. I enjoy programming very much, but am not real fond of re-inventing the wheel.

If the August issue of REMark is an example of future offerings, you can bet I will remain a loyal reader.

Sincerely,

Scott F. Rush
Firefighter/Computer Coordinator
P.O. Box 3827
Beaumont, TX 77704

Still Look Forward To Every Issue

Dear HUG:

Seems about time for me to offer some more comments for the column "Buggin' HUG". Having been a "HUGGY" since the very early issues of "REMark", I still look forward to every issue, though not quite as enthusiastically as a few years ago when the H-89 and CP/M dominated every issue. Times change, but I am not about to discard my old reliable and literally tons of software. My H-89 has really grown up: Four floppy drives, hard disk, interactive graphics, RAM disk, print spooler, clock board, 64k RAM, speech synthesizer, sound generator, universal parallel ports, graphics printer, modem, joysticks, etc. I'll continue to subscribe to REMark, BUSS and Sextant as long as each issue gives me at least one article of interest.

During the past year, a number of very interesting articles have appeared in REMark. Notable were the MBASIC articles by Dodgen in August 1984 and Harper in June 1985. I found Harper's article to be a literal gold mine with great serendiptous value to me. I have a number of programs (mostly adventure games) that when loaded into RAM, together with the MBASIC interpreter, use up just about every available byte of RAM. I would like to run these programs from RAM disk and eliminate all that disk action at every command. My RAM disk requires loading a couple of kbytes of driver and there I go — "Out of Memory". It occurred to me that Harper's PC.BAS might be the answer to compacting these long programs. After digging out several typos on Harper's PC.BAS and CRG.BAS, I was able to get both programs to work well on short programs, but found that CRG.BAS would abort on long programs where I really needed them. I wrote John Harper a letter explaining the problem and sent him a listing of one of my long MBASIC programs that I could not get to run under CRG.BAS. John wrote me a four page handwritten reply (his computer and printer were out of order). I am sure he spent hours figuring out why his program would abort on long programs and being handicapped with his own computer out of business. What a great guy! He made some suggestions to me that he thought would solve the problem. I tried John's approach and it worked. I found that another string variable also caused an abort. The same type of modification worked for this also.

For anyone wishing to use these programs, be sure to make the corrections noted in "Buggin' HUG" June 1986 and then add the following to CRG.BAS:

```
10035 ON ERROR GOTO 11220
11220 IF ERL=10800 OR 10590 THEN RESUME NEXT
11230 ON ERROR GOTO 0
```

These programs now work perfectly in greatly compacting those super-long programs. They have helped me get much greater utilization from my RAM disk and Dodgen and Harper have certainly helped me glean a better insight into the MBASIC interpreter. I have communicated with many of the contributers to REMark. I have found them all to be fine persons and always anxious to help their fellow computerist. We should all extend out appreciation to them at every ooportunity.

Sincerely,

S. K. (Tad) Magee
234 Oak Street
Audubon, NJ 08106

Adding Two 5-1/4" Disk Drives To The Z-100

Dear HUG:

Reference the letter in the August 1986 "Buggin' HUG" where Glenn Faini requested help to add two additional external 5-1/4" drives to his Z-100. Since I have added two to my Z-100, I believe I can be of assistance. There are three ways I know of to add floppy drives to the Z-100: 1) add 8" drives to the existing 8" port, 2) add 8" 'look-alike' 5-1/4" drives to the 8" port, and 3) add 5-1/4" drives as an extension to the existing 5-1/4" bus. The pros and cons of each approach are as follows:

1. 8" drives are readily available to 'plug in' to the existing 8" ports (most recent Z-100s will need the internal jumper cable from the controller card to the J16 port on the rear.) The controller card can support up to four 8" drives, however, the Z-DOS and MS-DOS software can handle only two drives. I have modified my MS-DOS 2.x I/O handlers to use four drives. 8" drives also provide greatly increased capacity (1.2 MB each), however, their relatively high cost and limited use (interface to other Z-100s) make them unattractive.
2. The 8" 'look-alike" 5-1/4" drives have the same capacity as the 8" drives (1.2 MB) and are generally available to plug in to the 8" port. They are less expensive than 8" drives, but require high capacity floppy disks and have very limited use (little interface to other Z-100s). These disks cannot be used in the existing internal 5-1/4" drives either.
3. The most attractive choice, I feel, is to add two DS/DD 5-1/4" drives to the existing 5-1/4" bus as I have done. The controller card can support four drives (two internal and two external), however, the Z-DOS and MS-DOS software can only support two drives. My modifications to the MS-DOS 2.x I/O handlers overcome this restriction and I am presently running four 5-1/4" floppy drives on my Z-100. In addition, since they are identical to the internal drives, they are completely compatible with all other Z-100 floppy disks — great interface capability! The price is lower than the other two options too.

I can supply a limited number of I/O handler modification programs and installation instructions to those requesting them. By the way, the I/O handlers can be modified to support more winchester partitions than the standard four, although I have not had the opportunity to verify the correct functioning since I do not have access to a winchester drive.

Frank K. Heath
1280 Kings Way
Redlands, CA 92373

Calendar Algorithms Unraveled

Dear HUG:

I was more than a little surprised to find my "Calendar Algorithms Unraveled" article in the September issue of REMark, given the

time lapse since it was submitted. Somehow an error seems to have crept into the publishing process, in Listing 5. The result is that, as printed, the OPEN.COM file produced will allow the entry of one escape sequence, if the escape character is the first character input from the keyboard, but the program terminates with the next escape character that is input, even though the escape characters are not consecutive.

Lines 10, 20, 30 and 40 of that listing should be changed as shown below:

```
10 DATA 180,1,205,33,60,27,117,250
20 DATA 205,33,60,27,117,244,205,32
30 OPEN "R", #1, "OPEN.COM", 16: FIELD #1, 16 AS W$
40 FOR I = 1 TO 16
```

With those changes, once the OPEN.COM program is invoked, it will remain active until two successive escape characters are input from the keyboard. Since many of the function and cursor keys emit a two character sequence when struck, with the first being the escape character, the program will also terminate if you strike ESC followed immediately by one of those function keys.

It should be of interest that on an IBM-XT machine in which the ANSI.SYS file has been installed, OPEN.COM (produced by the BASIC program corrected as above) can be used to execute, from the keyboard, all of the ANSI.SYS escape sequences. With the availability in ANSI.SYS of keyboard key reassignment sequences, this enables OPEN.COM to serve as a keyboard enhancement tool. Lengthy strings, greater in length than the IBM's usual 16 byte type-ahead buffer, can be input by one of the seldom used keys, such as CTRL-F10. Presumably, this same effect could be had on the Zenith IBM compatibles. These escape sequences can also be executed by use of the PROMPT command, however, they are simpler to execute by OPEN.COM because they then do not require the entry of any characters other than the escape sequence itself.

Another error (my own, admittedly) occurs in the next to last paragraph of the article. In each of the two places where it refers to 128 bytes, the proper number would be 512, or more, depending on how the disk was formatted. I had just reported the same number of bytes that the DIR command reports, whereas the file will actually occupy one or more sectors on a disk.

I have typed in all of the other listings in the article, from the published text, and find no other errors. In Listing 2, lines 110, 340 and 360 make use of escape sequences that are not available on the IBM-compatibles. Listing 2 can be fixed, for them, by deleting line 360 and changing line 340 to:

```
340   IF DAYS <> DESIRED(N) THEN PRINT "---".  GOTO 370
```

Listing 3 can be improved somewhat by adding line 165 and changing 170 and 220, as shown:

```
165     Q$ = MID$(D$(I),A,1)
170     IF Q$ > "/" AND Q$ < ":" GOTO 220
220     V = (V * 10) + VAL(Q$)
```

By that change, any character other than a numeral will be treated as a delimiter, and the algorithm will even cope with such wild delimiter errors as, for example, 8-n__a*22n%n 1986.

Since the purpose of the article was to give users of these, and comparable, calendar algorithms a basis for confidence in them, and since some readers may gain confidence by the use of the algorithms by large and prestigious corporations, I might add that the user's handbook for the Hewlett-Packard HP-25 programmable calculator shows a version of the algorithm of Listing 1. It uses the constant 30.6, and the same strategy for dealing with February, lacking only the automatic inclusion of 100 year and 400 year leap-day adjustments.

Keep up the good work with an interesting and helpful REMark.

Robert G. Brasfield
303 N. 175th Street
Seattle, WA 98133

A Particularly Ropy (I mean, Knotty) Problem

Dear HUG:

Perhaps a member of HUG can help me with a particularly knotty problem with a Heathkit H11-A computer which I bought in 1979, and an Okidata Microline printer that I bought in 1984. I had been and am still using a Heathkit H14 printer that works just fine with serial interface at 4800 baud. No matter what cable configuration and dip switch settings I use with the Okidata, the best I can do is print 27 lines. I then lose a bit of data before it prints more. This indicates that the handshake signal isn't working. I called the Okidata hot line and they gave me the connections for the DEC PDP11. I tried it and it doesn't work. I also modified the Serial Card H-11-5 according to REMark, Issue 8, 1979 page 20-21 by Randy Borchardt, and that didn't help either.

I would surely like to be put in touch with someone who has interfaced an H-11A computer with an Okidata Microline 92 using serial interface RS-232C. I could then learn the correct cable configuration and printer DIP switches to make it work.

I was a member of HUG but resigned after Heath discontinued the H-11A and there was no further activity in HUG on this computer.

I would appreciate your help.

Sincerely yours,

Frank Y. Speight
2011 Jackson Heights Drive
Sebring, FL 33870

Could Use Some Help

Dear HUG:

I am trying to get some information on a utility program to interface an NEC P2/P3-2 Pinwriter printer with the IBM graphics program. I'm trying to get the printer to print the graphic displays appearing on my IBM monitor without success. I can use some help. If anybody has any ideas, please let me know.

Yours truly,

Edward Iannuzzi
USA Eng Div Europe
EUDED-MH
APO NY 09757-5301

Line Clipping In BASIC

Jim Tursa
2001 Gemini #1302
Houston, TX 77058

Introduction

Although ZBASIC is a powerful graphics language, one aspect of the LINE statement has always frustrated me. Lines are not automatically clipped (as they are in PC BASIC), but are "translated" instead. For example, see Figure 1. The statement LINE(0,-100)-(200,100) will produce the line segment labeled A instead of segment B. For nearly all of my purposes I would rather have segment B. This article will develop a line clipping subroutine in BASIC and then present a couple of small programs that demonstrate its use. In particular, the subroutine can be used to develop graphics "windows" on either a Z-100 or a PC. For this reason the example programs include code for both machines. A general knowledge of algebra and geometry is assumed.

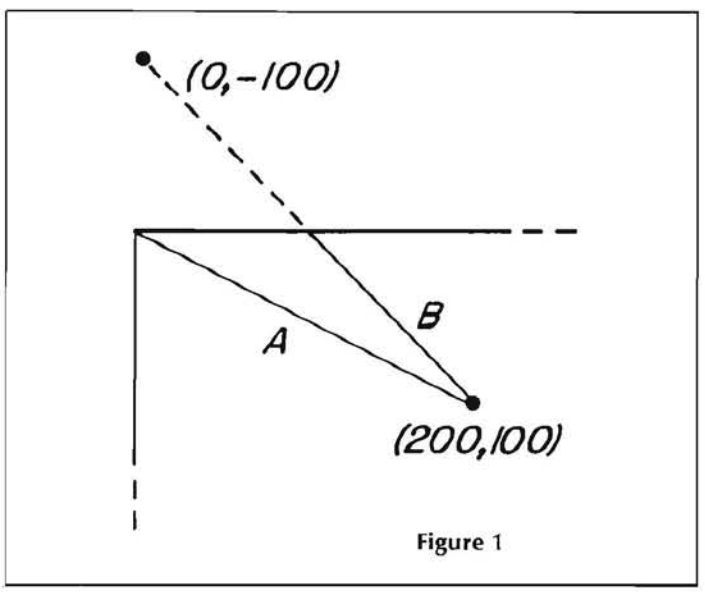


Figure 1

Similar Triangles

Similar triangles? Wasn't this article supposed to be on line clipping? Yes and yes. To develop the algorithm we will first review some geometry. Consider Figure 2. The large triangle has sides of lengths A, B, and C, while the small triangle has sides of lengths a, b, and c. The two triangles are similar since all three angles of the small triangle have corresponding equal angles in the large triangle. (You can consider this as a definition of similar triangles for our purposes.) Notice that this will always be the case if all three sides of the small triangle are parallel to the corresponding sides of the large triangle. Similar triangles also have the following property: The ratio of any two sides of the small triangle is equal to the ratio of the corresponding sides of the large triangle. For instance, (b/a)=(B/A) and (c/b)=(C/B).

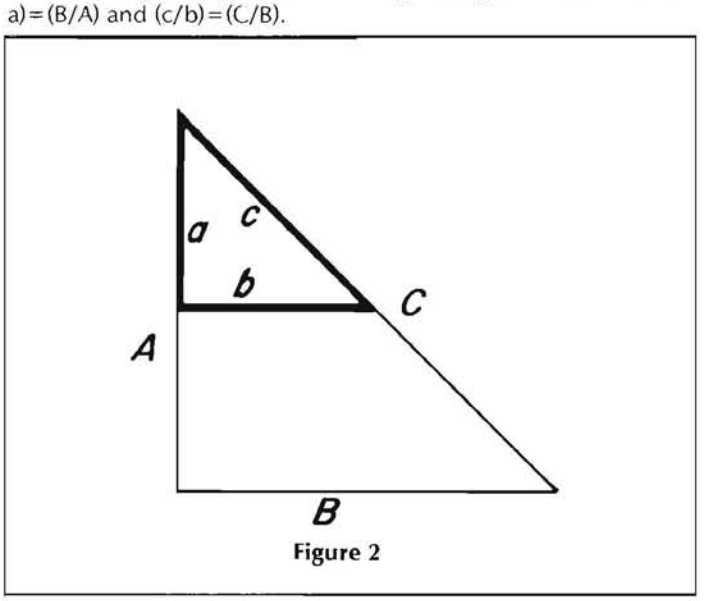


Figure 2

Line Clipping Development

Consider Figure 3. We would like to pass the coordinates (X1,Y1) and (X2,Y2) to a subroutine and have a line drawn from (X3,0) to (X2,Y2). To simplify matters for now, we will assume that X1>0, Y1<0, X2>0, and Y2>0. We will now find the value of X3. Compare Figure 2 with Figure 3 and notice that we have a pair of similar triangles in Figure 3. We will use the property (b/a)=(B/A). The following conditions apply:

```
b=(X3-X1), a=(0-Y1), B=(X2-X1), A=(Y2-Y1)
```

Plugging these values into our similar triangle property yields:

```
(X3-X1)    (X2-X1)
-------  = -------
( 0-Y1)    (Y2-Y1)
```

We will now solve this for the value of X3: Multiplying both sides of the equation by (0-Y1) yields

```
(X3-X1) = (0-Y1)*(X2-X1)/(Y2-Y1)
```

Adding X1 to both sides yields

```
X3 = X1 + (0-Y1)*(X2-X1)/(Y2-Y1)
```

Now that we have the value of X3 our subroutine can simply draw a line from (X3,0) to (X2,Y2). I have left the 0 in the expression (0-Y1) for a specific reason. To make our subroutine more versatile, we can specify an arbitrary top side value, YTOP, and rewrite our solution for X3 as:

```
X3 = X1 + (YTOP-Y1)*(X2-X1)/(Y2-Y1)
```

I would note at this time that it is possible to use the triangles in Figure 4 to obtain the following equation:

```
X3 = X2 + (YTOP-Y2)*(X2-X1)/(Y2-Y1)
```

At first glance this looks different than our previous solution for X3, but in fact it is algebraically equivalent. The following transformation demonstrates this:

```
X3 = X2 + (YTOP-Y1-(Y2-Y1))*(X2-X1)/(Y2-Y1)
X3 = X2 + (YTOP-Y1)*(X2-X1)/(Y2-Y1) -
     (Y2-Y1)*(X2-X1)/(Y2-Y1)
X3 = X2 + (YTOP-Y1)*(X2-X1)/(Y2-Y1) - (X2-X1)
X3 = X1 + (YTOP-Y1)*(X2-X1)/(Y2-Y1)
```

As you can now see, our second solution for X3 is really the same as our first solution. In fact, the solution for X3 turns out to be the same regardless of whether X1>X2 or X2>X1.

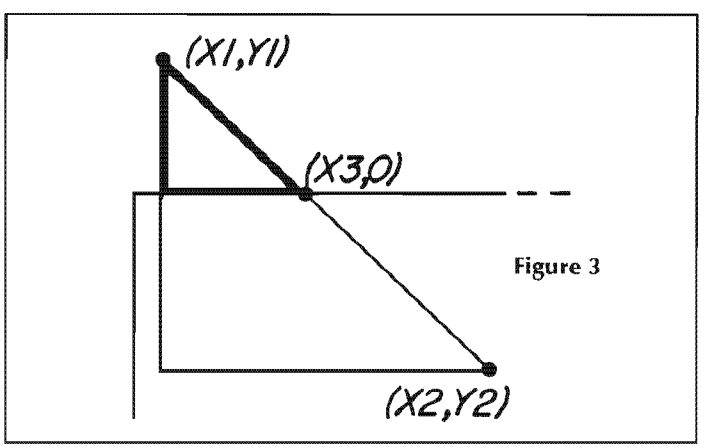


Figure 3

Line Clipping Subroutine

We can use similar triangles to develop formulas for clipping on the bottom, right, and left. All of these equations are presented in lines 1060-1130 of Listing 1. The variable NOLINE is used to indicate whether or not a line was drawn on the screen, but it is not a necessary part of the subroutine. To avoid changing the values passed to the subroutine, the contents of X1, Y1, X2, and Y2 are copied into U1, V1, U2, and V2. The clipping corresponding to our development in the previous section is actually done in line 1060. Let's examine this line in detail.

```
1000 ' LINE CLIPPING  **********************************
1010 '
1020 NOLINE=0   U1=X1   V1=Y1 . U2=X2 : V2=Y2
1030 XLEF=XLEF(WIND)-.5   XRIT=XRIT(WIND)+.5
1040 YTOP=YTOP(WIND)-.5 . YBOT=YBOT(WIND)+.5
1050 '
1060 IF V1<YTOP THEN
        IF V2<YTOP THEN NOLINE=-1   RETURN
                   ELSE U1=U1+(YTOP-V1)*(U2-U1)/(V2-V1)
                   V1=YTOP(WIND)
1070 IF V2<YTOP THEN
        IF V1<YTOP THEN NOLINE=-1 : RETURN
                   ELSE U2=U2+(YTOP-V2)*(U1-U2)/(V1-V2)
                   V2=YTOP(WIND)
1080 IF V1>YBOT THEN
        IF V2>YBOT THEN NOLINE=-1 : RETURN
                   ELSE U1=U1+(V1-YBOT)*(U2-U1)/(V1-V2) :
                   V1=YBOT(WIND)
1090 IF V2>YBOT THEN
        IF V1>YBOT THEN NOLINE=-1 : RETURN
                   ELSE U2=U2+(V2-YBOT)*(U1-U2)/(V2-V1) :
                   V2=YBOT(WIND)
1100 IF U1<XLEF THEN
        IF U2<XLEF THEN NOLINE=-1 : RETURN
                   ELSE V1=V1+(XLEF-U1)*(V2-V1)/(U2-U1) :
                   U1=XLEF(WIND)
1110 IF U2<XLEF THEN
        IF U1<XLEF THEN NOLINE=-1 : RETURN
                   ELSE V2=V2+(XLEF-U2)*(V1-V2)/(U1-U2) :
                   U2=XLEF(WIND)
1120 IF U1>XRIT THEN
        IF U2>XRIT THEN NOLINE=-1 : RETURN
                   ELSE V1=V1+(U1-XRIT)*(V2-V1)/(U1-U2) :
                   U1=XRIT(WIND)
1130 IF U2>XRIT THEN
        IF U1>XRIT THEN NOLINE=-1 : RETURN
                   ELSE V2=V2+(U2-XRIT)*(V1-V2)/(U2-U1)
                   U2=XRIT(WIND)
1140 '
1150 LINE(U1,V1)-(U2,V2),LCOLOR : RETURN
```

Listing 1

First there is the test V1<YTOP. If this test fails, then there is no clipping at the top for the point (U1,V1) and the routine skips to line 1070. If the test passes, then another test is made, V2<YTOP. If this test passes, then both endpoints are above the top line and, therefore, no line will be drawn. Otherwise, the point (U1,V1) is clipped and the variables U1 and V1 are replaced with new values. There is a possible problem here since we are dividing by (V2-V1). . . we do not want this to be zero. However, we are guaranteed (algebraically) that this will not be the case since V1<YTOP and V2>=YTOP must be true before the divide takes place. If a problem still exists in a particular application, we can always round coordinates to the nearest integer before passing them to the subroutine.

You have probably noticed by now that the program is set up to use any of several graphics "windows" by use of the variable WIND. Example programs using this feature are shown in Listings 2 and 3 and are discussed later.

Lines 1030-1040 represent a "quick fix" for a problem that occurs when a line is nearly parallel to a boundary. Consider Figure 5 and recall that your screen display is pixel oriented. Each square in the figure represents one pixel. Our formulas draw only pixel A, since

all of the line is above the horizontal YTOP line (assumed to pass through the center of pixel A) except pixel A. It would be nice if all of the pixels horizontally level with pixel A that the line crossed were drawn. If we expand the window edges by .5, then this will be accomplished. The statements resembling V1=YTOP(WIND) keep the routine from drawing outside the edges of the window. I have used the phrase "quick fix" because this method will not produce an exact reproduction of the line segment produced by a true line clipping routine. In defense I would mention that this is impossible to do without virtually rewriting a line drawing algorithm at the pixel-by-pixel level.

Well, if it's not exact, then how good is it? Listing 2 is a program that demonstrates how well our routine works. An exact line clipping routine would eliminate all of the pixels within the box window. As you can see, some of the line segments are eliminated entirely while others are not. I have experimented with several variations of my "quick fix" that improve the performance of Listing 2 to some degree, but the amount of improvement did not justify the complications introduced into the code. Listing 3 is a program that demonstrates the use of six simultaneous windows, each with a different color. It also demonstrates how well the routine joins line segments that cross window boundaries. Some are joined very well while others are slightly wavy at the boundaries.

Although the routine is not exact, I have found it to be close enough to be very useful in several of my application programs. In particular I use it in SOLARSIM, where only parts of constellations are drawn on the screen. Depending on your needs, you may be able to find it useful as well. Happy computing!

```
100 ' PROGRAM:  TESTCLIP
110 DIM XLEF(1),XRIT(1),YTOP(1),YBOT(1)
120 NWIND=1
130 DATA 220,420,62,162     'Z100
140 DATA 110,210,50,150     .'PC
150  FOR I=1 TO NWIND : READ XLEF(I),XRIT(I),YTOP(I),
     YBOT(I) : NEXT I
160 SCREEN 1    ·'PC
170 X1=320   Y1=112 . F=200   LCOLOR=0 : WIND=1 . CLS  .
    'Z100
180 X1=160 : Y1=100 : F=100 : LCOLOR=0 : WIND=1 : CLS  :
    'PC
190 LINE(XLEF(1),YTOP(1))-(XRIT(1),YBOT(1)),2,B
200  FOR A=0 TO 6 STEP .25
210  X2=CINT(X1+F*COS(A)) : Y2=CINT(Y1+100*SIN(A))
220  LINE(X1,Y1)-(X2,Y2),2 : GOSUB 1000
230  NEXT A
240 END
```

Listing 2

Using The Programs

To use the programs, run the BASIC interpreter of your computer and type in Listing 1. You can space past the 80th column and let the cursor wrap around to the next line to get the columns to line up nicely. After you have done this save it using the A option, for example: SAVE "LINECLIP",A. Then type in Listing 2, omitting lines that don't apply to your particular machine. For example, Z-100 owners should ignore all lines that have :'PC attached to the end. Then type MERGE"LINECLIP and the line clipping subroutine will be ajoined to the end of your program. Then type RUN. A similar procedure can be used to run the program in Listing 3.

HUG Software Engineer's Note: The new GW-BASIC for the H/Z-100 series (Heath catalog no. MS-5163-13) clips lines like PC BASIC, and does not "translate" them as ZBASIC does.

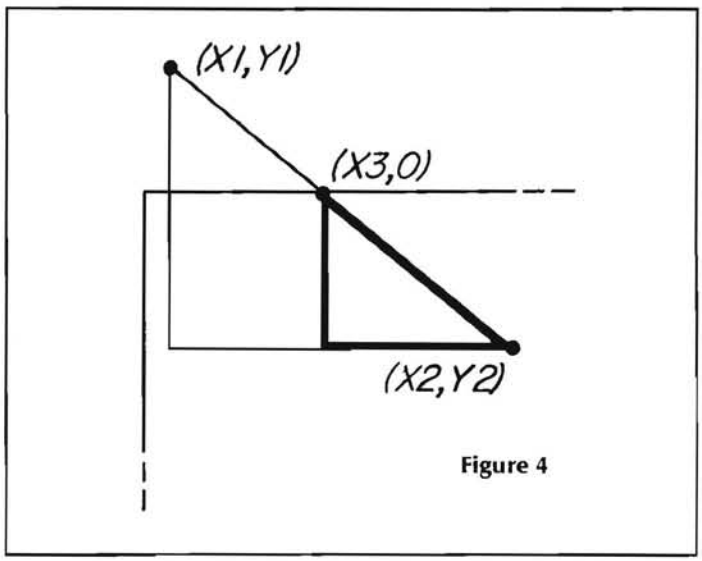


Figure 4

```
100 ' PROGRAM:  RANDLINE
110 DIM XLEF(6),XRIT(6),YTOP(6),YBOT(6)
120 NWIND=6 : C1=1 · C2=2 · C3=3 · C4=4 : C5=5 : C6=6
    :'Z100
130 NWIND=6 . C1=3 . C2=2 : C3=1 : C4=3 : C5=2 . C6=1
    :'PC
140 DATA 0,213,  0,112,   214,426,  0,112,   427,639,  0,112
    :'Z100
150 DATA 0,213,113,224,   214,426,113,224,   427,639,113,224
    :'Z100
160 DATA 0,106,  0, 99,   107,213,  0, 99,   214,319,  0, 99
    :'PC
170 DATA 0,106,100,199,   107,213,100,199,   214,319,100,199
    :'PC
180  FOR I=1 TO NWIND : READ XLEF(I),XRIT(I),YTOP(I),YBOT(I)
     : NEXT I
190 SCREEN 1                                          :'PC
200 CLS
210  FOR I=1 TO 50
220  X1=RND*839-100 : Y1=RND*324-50 : X2=RND*839-100
     : Y2=RND*324-50  :'Z100
230  X1=RND*419-50    Y1=RND*249-50   X2=RND*419-50
     : Y2=RND*249-50  .'PC
240  WIND=1 : LCOLOR=C2 : GOSUB 1000
250  WIND=2 : LCOLOR=C4 : GOSUB 1000
260  WIND=3 : LCOLOR=C6 : GOSUB 1000
270  WIND=4 · LCOLOR=C3 : GOSUB 1000
280  WIND=5 : LCOLOR=C5 : GOSUB 1000
290  WIND=6 : LCOLOR=C1 : GOSUB 1000
300  NEXT I
310 END
```

Listing 3

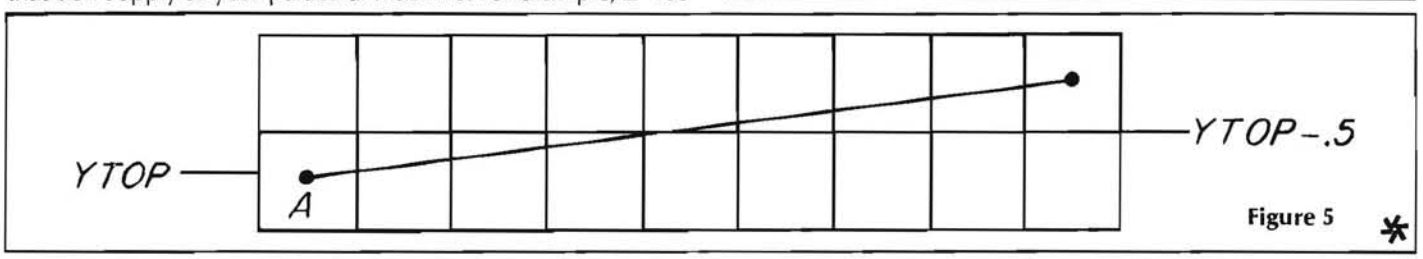


Figure 5

The H89 *SPEED* Center

4MHz mod — $45

An easy to install plug-in module. No trace cutting or soldering. Speed may be toggled with software. Includes a replacement Z80A (4MHz). Includes CP/M software support for Heath, CDR Systems and Magnolia. Call or write for info on HDOS support. Specify disk format.

6MHz mod — $59

Similar to our 4MHz modification, but increases the CPU speed to 6MHz. Requires some soldering on the CPU board. Includes a replacement Z8OB (6MHz). May require replacing additional parts. Some technical knowledge is recommended for installation. CP/M support only. Specify disk format.

CDR Super RAM 89

Make your H89 a speed whiz with this board and the included RAM drive software. With our 6MHz mod and the RAM drive, the Heath BIOS assembles (using MAKEBIOS) in less than 1½ minutes! Note our discount prices!

Main board w/o RAM - specify disk format $175
Expansion board w/o RAM $69
Main board w/512K RAM $239
System w/MEGABYTE RAM $369
Clock option $45
SCSI option $85

CDR Controllers

Double density disk controllers. Includes board, instruction manuals, ROMs and software. Allows control of 5 and 8 inch drives. Includes BIOS source code and some useful utilities. Please include your CP/M s/n when ordering.

For the H89 FDC-88OH $349

REP3 - Automatic Key Repeat

Stop wasting time - make full use of the repeat function just like the Z100. Simple plug-in installation on your keyboard assembly. Hold any key down for half a second and the key begins repeating. Combine this with our 4MHz mod and make WordStar fly! Provision for a defeat switch.

Kit $35
Assembled $45

TIM2 - Real Time Clock

This circuit installs in the left hand expansion slots of the H89 or H89A. Can provide application programs with time and date information. Includes battery backup for continuous timekeeping when the computer is turned off. Program access to the clock is through an I/O port. The port address is user selectable by means of a jumper. Requires soldering 4 wires to the CPU board. A prepared ribbon cable is included.

Kit $65
Assembled $75
Software on disk - specify disk format $10

DATESTAMPER

Product of Plu*Perfect Systems. Provides automatic time and datestamping for CP/M 2.2 files. Works with many real time clocks, including our own TIM2 product.

CP/M - specify disk format $45

WSPATCH — $20

Adds H19/H89 function key patches to versions 3.0 or 3.3 of WordStar. Key functions similar to the PIE editor. Includes provision for redefining the keys by the user. Also includes a printer driver for the Epson MX80/FX80 printers.

EMULATE

Allows the H89 to read/write to the following disk formats.

Actrix	Eagle II	Morrow MD	Superbrain Jr
AMPRO	Epson QX-10	NCR DecMate 5	Televideo
Beehive Tpr	Fujitsu CP/M86	NEC PC-8001A	TRS80-1 CP/M
CDR	IBM CP/M86	Osborne 1	TRS80-3 CP/M
Cromemco	IMS 5000	Otrona	TRS80-4 CP/M
DEC VT180	Kaypro II	PMC MicroMate	Visual 1050
DEC Rainbow	Magnolia	Royal/Adler	Xerox 820
		Sanyo 1100	Zorba

Now includes 44 formats! Uses a modified BIOS which is included with the program. Allows reading of 40-track disks in an 80-track drive.

For H37 with Heath CP/M $59
For CDR BIOS 2.91 . . . . . $49
Check for Magnolia version.

ZCPR3

We are licensed by ECHELON, Inc. to distribute the Z-System for the Heath market. This includes the ZCPR3 system with supporting software utilities, the ZRDOS-plus BDOS replacement and applicable documentation. Comes already installed and includes a bootable disk ready to run.

ZCPR Gives you many features found in UNIX and MSDOS. The ZRDOS is not in the public domain and gives you automatic logon of disks. No more "BDOS ERROR ON B:" messages just because you forgot to do a ^ C after changing disks.

Available for Heath BIOS 2.2.03 and .04 and for CDR BIOS. Check for availability of other BIOS configurations. Call or write for a brochure on this excellent product.

ZCPR3 - specify format and hardware $98

ANALYTICAL PRODUCTS (805) 688-0826

213 Teri Sue Lane Buellton, CA 93427

Specify Disk Format on Software

CALL OR WRITE FOR CATALOG PRICES SUBJECT TO CHANGE

Terms: Check or Money Order - VISA/MC - C.O.D.
Add $3 per order for shipping and handling
California residents add 6% tax

Interfacing H/Z-100 Assembly Language Graphics Routines To Turbo Pascal

Randy DeBey
4316 N. County Road 13
Ft. Collins, CO 80524

As with many H/Z-100 owners, the first language that I programmed with was ZBASIC. The editor environment made it easy to write and debug small programs quickly, and it had built-in graphics functions. For me, the ZBASIC graphics functions were so easy to use that in many programs, I would plot the results on the screen, then dump the screen to the printer. As my programs increased in size and required more structuring, ZBASIC became less acceptable for program development. Turbo Pascal has become a capable alternative.

Turbo Pascal has received much praise, and is available at a reasonable cost, but the generic MS-DOS version that we H/Z-100 owners must use has no graphics capabilities. In the beginning, this lack of graphics caused me to avoid using Turbo. There have been several graphics procedures written in Turbo Pascal for the H/Z-100 (Frank Dreano, REMark, April 1985), but they work so differently than the ZBASIC functions that they seem clumsy.

Eventually, I decided to write my own set of fundamental graphics subroutines. My requirements were that they perform as much as possible like the ZBASIC functions and use similar syntax, they should be fast and not occupy a lot of disk space, and they should be easy to include in any Turbo program. There is a section in the Turbo manual which explained adequately the use of external subroutines and parameter passing, which convinced me that assembly language was the way to go.

This article is not intended to be a tutorial on 8088 assembly language programming. It's primary intention is to provide some fast, easy-to-use graphics procedures to users of Turbo Pascal. You don't need to know any assembly language to make use of these subroutines. For those who may be interested in the Turbo/Assembly interface or the fundamental pixel manipulations, these subroutines may serve as examples for more elaborate endeavors.

What The Graphics Routines Do

There are only four basic graphics routines described here. Each performs a function similar to the function by the same name in ZBASIC.

CLS — Besides clearing the screen, this procedure enables the 25th line and enables access to all three planes of video RAM. The ClrScr procedure of Turbo Pascal (same as ESC E) doesn't clear the 25th line, so it wasn't very useful for full screen graphics. This procedure uses the method described in the Z-100 Technical Manual page 4.45, to clear all VRAM. Enabling the three planes of VRAM is necessary in order to use the following routines in TURBO Pascal. The syntax is simply:

```
Cls,
```

Listing 1

```
, procedure Cls, external 'CLSPAS',
; A procedure which clears the screen, enables all VRAM
; planes, enables the 25th line, and homes the cursor
; To be called from TURBO Pascal programs

FUNC    SEGMENT
        ASSUME          CS:FUNC

START   PROC            NEAR
        MOV     AH,2            ;DOS call for video output
        MOV     DL,27           ;ESC character
        INT     21H
        MOV     AH,2
        MOV     DL,'x'
        INT     21H
        MOV     AH,2
        MOV     DL,'1'
        INT     21H             ;25th line now enabled
        MOV     AL,0FH          ;disable VRAM
        OUT     0D8H,AL
        IN      AL,0DBH
        AND     AL,0F7H         ;clear bit 3
```

```
        OUT     ØDBH,AL
        IN      AL,ØD9H
        AND     AL,ØF7H
        OUT     ØD9H,AL         ,turn CLRSCRN on
        MOV     CX,1Ø4FH        ;do 4175 loops (for 16.7 ms)
DELAY:  NOP                     ;3 cycles
        LOOP    DELAY           ;17/5 cycles
        OR      AL,Ø8H
        OUT     ØD9H,AL         ,turn CLRSCRN off
        MOV     AL,78H
        OUT     ØD8H,AL         ;enable all VRAM planes
        MOV     AH,2            ;'Home' the cursor
        MOV     DL,27
        INT     21H
        MOV     AH,2
        MOV     DL,'H'
        INT     21H
        RET

START   ENDP
FUNC    ENDS
        END     START
```

PSET — This routine will be used to turn on a screen pixel at a point specified by horizontal and vertical coordinates. The color of the point will be selectable by a third parameter. Although not exactly the same as in ZBASIC, the syntax will be very similar, as follows:

```
Pset(X,Y,Color);
```

The horizontal coordinate, X, will range from 0 at the left edge of the screen, to 639 at the right edge. The vertical coordinate, Y, will range from 0 at the top to 224 at the bottom of the screen. Color can be a value from 0 to 7. The color 0 of course is black, and will blank the pixel.

Listing 2

```
, procedure Pset(x,y,color:integer); external 'PSETPAS';
, A pixel setting procedure to be called from TURBO
; Pascal programs.

FUNC    SEGMENT
        ASSUME          CS:FUNC

START   PROC            NEAR
        PUSH    BP
        MOV     BP,SP
        MOV     AX,[BP+6]       ,get Y value
        CMP     AX,ØØEØH        ;clip Y coordinate at 224
        JC      DONE
        CMP     AX,Ø            ;and at Ø
        JL      DONE
        MOV     BL,9
        DIV     BL              ;divide Y by 9 to find row
        MOV     BL,AH           ;save remainder
        MOV     AH,Ø            ;clear upper byte of quotient
        MOV     CL,11           ;use CL to count left shifts
        SHL     AX,CL           ;multiply quotient by 2Ø48
        MOV     BH,Ø            ;clear upper byte of remainder
        MOV     CL,7            ;use CL to count left shifts
        SHL     BX,CL           ;multiply remainder by 128
        ADD     BX,AX           ;add products, result in BX
        MOV     AX,[BP+8]       ;get X value
        CMP     AX,Ø27FH        ;clip at 639
        JG      DONE
        CMP     AX,Ø            ;and at Ø
        JL      DONE
        MOV     CL,8            ;the divisor
        DIV     CL              ;divide X by 8
        MOV     DL,AH           ;save remainder in DL
        MOV     AH,Ø            ;clear upper byte of quotient
        ADD     BX,AX           ;add quotient(column#) to addr
        MOV     DI,BX           ;move addr to index reg
        MOV     CL,7            ;subtract remainder from 7
        SUB     CL,DL           ,result in CL
        MOV     DH,1            ;set low bit of display byte
        SHL     DH,CL           ;shift left to correct bit
                                 position
        MOV     DL,DH
        NOT     DL              ;DL contains complement of
                                 Pset byte
        MOV     AX,[BP+4]       ;get Color value
        CMP     AX,7            ,limit color to 7 or less
        JG      DONE
        CMP     AX,Ø            ;and Ø or greater
        JL      DONE
        MOV     BX,ØB8ØØH
        MOV     ES,BX           ;ES holds VRAM plane seg addr
                                 ;DH contains Pset byte
                                 ;DI contains offset addr
                                 ;AL contains the color value
        MOV     CX,3            ;CX contains loop count for
                                 3 planes
COLOR:  MOV     BX,ES
        ADD     BX,1ØØØH        ;adjust ES for next color plane
        MOV     ES,BX
        SHR     AL,1            ;shift Color bits into carry
                                 flag
        JNC     OFF             ;clear pixel if that color
                                 bit was Ø
        OR      ES:[DI],DH      ;set pixel in VRAM
        JMP     COLOR1
OFF:    AND     ES:[DI],DL      ;clear pixel in VRAM
COLOR1: LOOP    COLOR           ;do another color plane
DONE:   POP     BP
        RET     6

START   ENDP
FUNC    ENDS
        END     START
```

PRESET — This procedure blanks a pixel specified by the horizontal and vertical coordinates. It is somewhat redundant to the PSET procedure in that it has the same action as PSET when Color is zero. The syntax is:

```
Preset(X,Y),
```

Listing 3

```
; procedure Preset(x,y:integer); external 'PRSETPAS';
; A pixel clearing procedure to be called from TURBO
; Pascal programs

FUNC    SEGMENT
        ASSUME          CS:FUNC

START   PROC            NEAR
        PUSH    BP
        MOV     BP,SP
        MOV     AX,[BP+4]       ,get Y value
        CMP     AX,ØØEØH        ;clip Y coordinate at 224
        JC      DONE
        CMP     AX,Ø            ;and at Ø
        JL      DONE
        MOV     BL,9
        DIV     BL              ;divide Y by 9 to find row
        MOV     BL,AH           ;save remainder
        MOV     AH,Ø            ;clear upper byte of quotient
        MOV     CL,11           ;use CL to count left shifts
        SHL     AX,CL           ;multiply quotient by 2Ø48
        MOV     BH,Ø            ;clear upper byte of remainder
        MOV     CL,7            ;use CL to count left shifts
        SHL     BX,CL           ;multiply remainder by 128
        ADD     BX,AX           ;add products, result in BX
        MOV     AX,[BP+6]       ;get X value
        CMP     AX,Ø27FH        ;clip at 639
        JG      DONE
        CMP     AX,Ø            ;and at Ø
        JL      DONE
        MOV     CL,8            ;the divisor
        DIV     CL              ;divide X by 8
```

```
        MOV     DL,AH       ;save remainder in DL
        MOV     AH,0        ,clear upper byte of quotient
        ADD     BX,AX       ;add quotient(column#) to addr
        MOV     DI,BX       ;move addr to index reg
        MOV     CL,7        ;subtract remainder from 7
        SUB     CL,DL       ;result in CL
        MOV     DH,1        ,set low bit of display byte
        SHL     DH,CL       ;shift left to correct bit
                             position
        NOT     DH          ;complement the bit pattern
                             for ANDing
        MOV     BX,0B000H
        MOV     ES,BX       ;set for segment below blue plane
                            ;DH contains PRESET byte
                            ;DI contains offset addr
                            ,ES contains seg addr
        MOV     CX,3        ;CX contains loop count for
                             3 planes
COLOR:  MOV     BX,ES
        ADD     BX,1000H    ,adjust ES for next color
                             plane
        MOV     ES,BX
        AND     ES:[DI],DH  ;clear bit
        LOOP    COLOR       ;do another color plane
DONE:   POP     BP
        RET     4

START   ENDP
FUNC    ENDS
        END     START
```

POINT — This integer function returns the color value of a pixel specified by the horizontal and vertical coordinates. The value returned will be in the range 0 to 7 unless a coordinate was out of range, in which case the value -1 will be returned. Here is an example of the syntax:

```
Color := Point(X,Y),
```

This function might be useful for screen dumps.

Listing 4

```
{ function Point(x,y:integer):integer; external 'POINTPAS';
{ A function which returns the color value of the screen
; pixel specified by coordinates x and y   To be used with
; TURBO Pascal programs

FUNC    SEGMENT
        ASSUME          CS:FUNC

START   PROC            NEAR
        PUSH    BP
        MOV     BP,SP
        MOV     AX,[BP+4]  ;get Y value
        CMP     AX,00E0H   ;clip Y coordinate at 224
        JG      DONE
        CMP     AX,0       ;and at 0
        JL      DONE
        MOV     BL,9
        DIV     BL         ;divide Y by 9 to find row
        MOV     BL,AH      ;save remainder
        MOV     AH,0       ;clear upper byte of quotient
        MOV     CL,11      ;use CL to count left shifts
        SHL     AX,CL      ;multiply quotient by 2048
        MOV     BH,0       ;clear upper byte of remainder
        MOV     CL,7       ;use CL to count left shifts
        SHL     BX,CL      ;multiply remainder by 128
        ADD     BX,AX      ;add products, result in BX
        MOV     AX,[BP+6]  ,get X value
        CMP     AX,027FH   ;clip at 639
        JG      DONE
        CMP     AX,0       ;and at 0
        JL      DONE
        MOV     CL,8       ;the divisor
        DIV     CL         ;divide X by 8
        MOV     DL,AH      ;save remainder in DL
        MOV     AH,0       ,clear upper byte of quotient
        ADD     BX,AX      ;add quotient(column#) to addr
        MOV     DI,BX      ;move addr to index reg
        MOV     CL,7       ;subtract remainder from 7
        SUB     CL,DL      ;result in CL
        MOV     DH,1       ;set low bit of display byte
        SHL     DH,CL      ,shift left to correct bit
                            position
        MOV     CX,0B000H
        MOV     ES,CX      ;ES contains color plane addr
                           ;DH contains display byte
                           ;DI contains offset addr
        MOV     CX,3       ;CX contains loop count for
                            3 planes
        MOV     DL,0       ;DL will contain the color value
COLOR   MOV     AX,ES
        ADD     AX,1000H   ,adjust ES for next color plane
        MOV     ES,AX
        MOV     BH,DH      ;copy display byte
        AND     BH,ES:[DI] ,AND it with the VRAM byte
        JZ      COLOR1     ,if result zero then pixel
                            not set
        MOV     AX,CX      ;save loop count
        MOV     BL,1       ,initial color value
        MOV     CL,3
        SUB     CL,AL      ;subtract loop count from 3
        SHL     BL,CL      ;shift color bit appropriately
        ADD     DL,BL      ;add to previous color value
        MOV     CX,AX      ,restore loop count
COLOR1. LOOP    COLOR      ;do another color plane
        MOV     AL,DL      ;move color value to AX for
                            return
        MOV     AH,0
        POP     BP
        RET     6

DONE:   MOV     AX,0FFFFH  ;return -1 if x,y out of range
        POP     BP
        RET     6

START   ENDP
FUNC    ENDS
        END     START
```

WRITING THE ASSEMBLY ROUTINES

Parameter Passing

In Turbo, you have the choice of passing parameters to external procedures and functions in either of two ways. You can pass by reference or by value. Passing by reference means that the address of the first byte of a parameter is passed to the procedure. Passing by value means that the actual value of the parameter is passed. If you have no need to alter the value of a parameter, it is easier to pass it by value. For these graphics routines, all of the parameters will be passed by value.

When a parameter is passed by value, the number of bytes it occupies is pushed onto Turbo's stack. Integer, Boolean, and Char types are transferred as two bytes. Reals are transferred as six bytes, and strings occupy a number of bytes equal to the number of characters in the string plus one. In the case of these graphics procedures, only integer values are required.

If the external routine is a function, space is also reserved on the stack for the function result. Although Turbo reserves the space, the space is actually only used if the function type is real, in which case six bytes are reserved. Other scalar types must be returned in the AX register. The one graphics function described here will be an integer function, so it returns its value in AX.

When an external procedure or function is called, the parameters are pushed onto the stack in the order that they are listed in the statement. In the case of a function, before any parameters are pushed onto the stack, the stack pointer is first decremented

by the number of bytes corresponding to the function type. Lastly, the return address is pushed onto the top of the stack. Although, the Turbo manual doesn't say so explicitly, external calls are short calls, so the return address is only two bytes and it is the offset for the code segment. The assembly routine must explicitly be defined as a NEAR procedure so that the assembler will generate the proper RET instruction.

Saving Registers

Only the BP, CS, DS, and SS registers must be returned unaltered from an external routine. If you need to use one of these registers, you must save it's value on the stack before you alter it, and then restore its value before returning. The only one of these that we will need is the BP register in routines where parameters are passed. Consequently, the first instruction in those routines will be PUSH BP.

Accessing Parameters

The parameters passed are in the Stack Segment. Since the Stack Segment is the default segment for the Base Pointer register, the Base Pointer can be used to access the parameters. The locations of the parameters are known relative to the Stack Pointer, so by copying the Stack Pointer to the Base Pointer, we can easily access parameters using a displacement from the Base Pointer. An example is, MOV AX,[BP+4]. Here, a word is transferred to the AX register from memory. The memory address is formed from the combination of the SS register, the BP register, and the additional displacement 4. This instruction could be used to access the last integer parameter listed in the calling statement. The displacement of 4 is necessary to jump over the contents of the BP register and the return address which are the last two items on the stack.

VRAM Address Calculation

This is simpler than it first appears. The first step is to determine which character row the pixel is in. There are 9 pixel lines in each character row, so divide the vertical coordinate (Y) by 9. The integer part of the quotient is the character row — multiply it by 2048. The remainder part of the quotient gives the pixel line within the character row — multiply it by 128. Add those two products.

Now you have to determine which character column the pixel is in. Each character position is 8 pixels wide, so divide the horizontal coordinate (X) by 8. The integer part of the quotient should be added to the previous sum of products to give the complete VRAM offset address. When used in combination with the segment address of the VRAM plane desired, it will access the proper byte.

Changing The Right Pixel

The pixel position is obtained by setting a byte register equal to 1, then doing a number of left shifts equal to 7 minus the remainder of the previous division, X/8. The result is a byte with only one bit set at the location of the pixel you want to change. It can be ORed with the byte in VRAM to set the pixel. Its complement can be ANDed with the byte to clear the pixel.

Returning To Turbo

Before returning to Turbo, if BP was PUSHed, it must be POPped back. Then a RET # instruction must be executed. The # is the number of bytes that must be 'popped' from the stack in the process. These bytes aren't actually transferred to any registers, the Stack Pointer is simply increased by this number. If three integer parameters were passed to a procedure, the return instruction will be RET 6. If two integer parameters were passed to an integer function, RET 6 will also be used — two bytes for each parameter, and two bytes for the integer result (which is actually returned in AX).

Specifics Of The Routines

CLS is a procedure with no parameters. The first thing it does is to enable the 25th line of the screen. This is done by repeatedly using DOS's video output function call to send ESC x1 to the terminal. The print string function would have been more efficient, but it requires the string to be in the data segment and Turbo doesn't allow the data segment to be used by external procedures. Next, the screen is cleared using the method described in the technical manual. The 16.7 millisecond delay required for this is created by executing a dumb loop 4175 times. If your H-100 runs at 8 MHz, you will have to change the loop counter to 1A18H (6680 times). Lastly, CLS enables all of VRAM by writing 78H to port D8H, and homes the cursor with ESC H.

PSET is a procedure which has three integer parameters passed to it. Each parameter is checked to be sure it is within the allowable range before it is used. If any is out of range, the routine is abandoned without affecting VRAM. Then, the VRAM offset address is calculated and the bit pattern is obtained. A loop is executed three times, once for each color plane. Inside the loop, the segment address in ES is incremented to point to the appropriate plane. The color value is shifted right into the carry flag, and the carry flag is tested to determine whether to set or clear the pixel in that plane. After the loop is finished, the RET instruction pops 6 bytes for the three integer parameters.

PRESET is a procedure with only two integer parameters. As with PSET, the parameters are checked and the VRAM offset address and bit pattern are determined. In the three color loop, however, no testing is done and the VRAM byte is simply ANDed with the complemented bit pattern to clear the pixel in each plane. The RET instruction pops 4 bytes for the two integer parameters.

POINT is an integer function with two integer parameters. Again, the parameters are checked, but if one of them flunks, a different route is taken in returning. The value of -1 (0FFFFH) is placed in AX before this return. Otherwise, the VRAM offset and bit pattern are determined, and it goes into the color loop. Here ES is incremented as usual, but VRAM is not altered. The bit pattern is ANDed with the VRAM byte to determine if the pixel is on in that plane. If it is on, a bit is set in a color value register at a position depending on the color plane, with bit 0 representing the blue plane, bit 1 representing the red plane, and bit 2 representing the green plane. This color value is placed in AX before returning. Six bytes are popped for the two integer parameters and the integer result.

Listing 5

```
program Grafprx; { Demonstration of the use of external
                   graphics routines }

var
   x,y,color :integer;

procedure Pset(x,y,color:integer); external 'PSETPAS';
procedure Preset(x,y,:integer); external 'PRSETPAS';
function Point(x,y:integer):integer; external 'POINTPAS';
procedure Cls; external 'CLSPAS';

begin   {main}
   cls;
```

```
    for x:=0 to 639 do      [ Draw a border ]
    begin
       pset(x,0,7),
       pset(x,224,7),
    end;
    for y:=0 to 224 do
    begin
       pset(0,y,7);
       pset(639,y,7);
    end;
    for color:=0 to 7 do    | Set some pixels and read them back ]
    begin
       pset(100,14+color*9,color);
       gotoxy(2,2+color);
       writeln(' Point',color,' = ',point(100,14+color*9));
    end;
    readln,
    for color:=0 to 7 do [ Change their colors and read them back ]
    begin
       pset(100,14+color*9,7-color);
       gotoxy(2,2+color),
       writeln(' Point',color,' = ',point(100,14+color*9));
    end;
    readln;
    for color:=0 to 7 do [ Clear the pixels and read them back ]
    begin
       preset(100,14+color*9);
       gotoxy(2,2+color);
       writeln(' Point',color,' = ',point(100,14+color*9));
    end;
    readln,
    cls,
    for y:=-32767 to 32767 do |Fill screen with random dots]
       pset(random(639),random(224),random(7)),
end     |main]
```

Summary

These assembly code routines must be assembled using MASM .EXE, linked using LINK.EXE, and then converted to .COM files using EXE2BIN.EXE. Be sure to erase the .EXE versions of the graphics routines after you get the .COM versions. You will end up with four .COM files occupying a total of only about 400 bytes. They may be used with any Turbo Pascal program by first declaring them as external procedures or functions. The declaration statements that should be used in your programs are given as the first comment line of the assembly listings. These procedures can be given names other than Pset, Preset, Point, and Cls, simply by changing the names in the declaration statements. You may want to give them the names Turbo uses in the IBM-PC version, Plot for Pset, and GetDotColor for Point. A short Turbo program is listed which demonstrates the use of these routines. Don't forget that you must use the CLS procedure before the other routines will work.

These routines can be used alone if you're doing ordinary line drawings or plotting graphs. You can also use them with some of the other graphics procedures, such as Line and Draw, which have been written in Turbo Pascal for the H/Z-100. Performing the basic pixel operations with this Pset procedure should speed them up significantly. A comparison between this Pset procedure and the one written in Turbo Pascal by Frank Dreano (REMark, April 1985), showed that the assembly routine operates about 2.6 times faster. *

Super-Cheapcalc And Mini-Tutorial On Spreadsheets

Luis E. Suarez
P.O. Box 66994
Caracas 1061-A
VENEZUELA

Some years ago, in REMark issue 37, it was published an article and MBASIC spreadsheet program authored by B.L. McFarland and named Cheapcalc. The program was an adaption for the H-89 of an original one, coded by William V. Smith for the Apple II. Later the HDOS version was debugged by Clifford Lundburg in REMark issue 40 and once again by Mr. McFarland in issue 44, who presented also a CP/M version. Then after, the program was placed in the HUG Library with additional modifications by Pat Swayne. No further modifications or enhancements were published since then. By the way, I have not seen the version by HUG. Enough for presentation and proper crediting.

After reading the article published on February 1983, I keyed it on my H-89 but was disappointed for the bugs I found (most due to my poor typing). Of course, I must confess, that my appreciation was biased for the fact that I had a commercial spreadsheet at my office for my daily engineering work and really had no need for a spreadsheet at home.

Three years later, this past March, I was at home and really did need a spreadsheet for some personal decisions based on income, taxes, inflation and so on (I live in an oil exporter country, you know.) It was a weekend and I remembered the Cheapcalc. After some searching on all my disks, I finally got the program and the articles mentioned above in my REMark's collection. I had the program on HDOS and since all my daily work is done under CP/M, I first converted it for this operating system. After debugging, I got a workable program and hence solved my financial problem.

The interpreted BASIC version of Cheapcalc was desperately slow and, as it was recommended in the articles and common sense dictates, compilation was mandatory (see REMark issue 63 page 39.) After all, computer means speed and a spreadsheet's primary reason is speed, too. I did it, and certainly the resulted improve was so significant that it was worth the 8 minutes plus it takes the program to compile. At that moment, I realized that Cheapcalc is really a tool for occasional spreadsheet work, but don't think it will replace Lotus 1-2-3 or Multiplan. The difference is just the price or "You get what you pay for." However, I thought that it would be possible to make some enhancements and . . . perhaps a couple of refinements and . . . an increase in speed and . . . better error recovering and. . . and . . . humm!

To make the story short, enough is to say that in fact, I planned a two day's job devoted to modifying the Cheapcalc . . . Actually, I spent 3 weekends and some couple of hours daily after dinner. The result was rewarding and certainly it was worth the time it took to accomplish it. I hope you will enjoy the new version as much as I did modifying it.

Getting Speed

First of all, I studied the program to understand how it works. Then I did some recall of what is a good spreadsheet. On the speed department, I remembered an article on this subject published some time ago in Sextant magazine. The article contains some benchmark results run on some popular commercial spreadsheets. Unfortunately, the Lotus was under development during those days, and hence no mention is done of the popular "1-2-3", let alone "Symphony", of recent appearance. While modifying the program, I projected to use those benchmarkings for final comparisons. Keep reading. You will be surprised!!!

In my outline, I preserved the original structure of the program including all the original features. For that reason, I couldn't use all the tricks I know to speed up a program. Even I kept untouched the original code, for the sake of clarity, where improved coding would do much better. There are several good articles on this subject in the REMark collection, in case you readers are interested. Anyhow, as a must, to speed up any program, no matter if it is intended to be compiled or not, I use the following tools.

- Avoid IF statements as much as possible. Use ELSE after the IF statement since any time an IF is found, the interpreter searches for an ELSE.
- Use multiple statement lines as much as possible. Each new line is a waste in memory and execution time.
- Avoid repetitive GOTO, GOSUB, RESUME, RESTORE, etc. within last lines of the program. Any time the interpreter reads such statements, it starts searching the referred line number from the first line coded. Obviously, a tenths line is found earlier than a hundredths line.
- For the same reason mentioned, the most used subroutines must be placed earlier in the program.
- Use short variable names. A$ is better than ANOTHERVARIABLENAME$.
- Avoid using ON ERROR GOTO statements. A waste of memory during compilation.
- If necessary, use ON ERROR GOTO just once at the beginning of the program, avoiding the repetition of this statement at each subroutine. Then group all error trapping lines in a single subroutine.
- Avoid RESUME NEXT and prefer the use of RESUME (line number). This will save luts of memory too, during compilation. If RESUME NEXT is used, you must compile with the X switch, otherwise, the E switch is enough. The former requires less memory.
- Do not use ';' for strings joining when unnecessary. Use PRINT AB"STRING" if accepted by your interpreter.
- Avoid unnecessary long strings like, "*** Hello, this is your smart computer and slave!!! May I help you, lovely boss? ***" Nuts!!
- Do not compile if the program does a lot of (repetitive) math crushing like, (SQR(ATN(B*C5)/COS(ATN(1)*4)-FIX(Z*X/2))). Instead, do compile if the main job is related to logical decisions like, IF A$=INFLATION GOTO THE HELL ELSE IF YOUR WIFE GIVES PERMISSION BUY A NEW CAR ELSE THINK IN DIVORCE.
- Delete all REM and ' statements. Please don't refer GOTO's to REM lines. This is an advice for authors publishing programs. A good practice is to use odd line numbers for remarks and coding just the even lines. However, if the program is to be compiled you may include the remarks.
- Use same variable names for utilities in all subroutines. The less variables are declared the less memory is used.

As I said, for the sake of preserving the original structure of the Cheapcalc program, not all above recommendations are met in my program modification.

The Old Look

The original Cheapcalc had the following functions:

- Arithmetic: Add (+), Substract (-), Multiply (*), Divide (/), Exponentiation (^), Summation (&).
- Editing: CTRL-U, Backspace, Delete.
- Clear Cell: (#)
- Define Format: Dollar and cents (:$), Integers (:I), Floating (:F), Graphics (:*)
- Recalculation: !
- Help Command: /
- Copy Cell: IL
- Replicate Cell: DL
- End Session: CTRL-D
- Select Column Width
- Print Sheet
- Save File
- Load File

After modification, Cheapcalc has all of the above, plus the following functions:

- 18 Math functions: @[function](argument). See Table 1.
- MAX-MIN function: @[function](argument)
- Prompted Input: ?(string)
- Define Form: Percent (:%)
- Global Column Width: (nn.)
- Erase Calculation: ERASE
- Fast Page Moving: GOTO <CR>
- Justification: Left (<), Center (^), Right = default
- Fill the Cell With Characters: (\[character])

Additionally, I thought that the program deserved a better interface with the operator. Hence, I included some cosmetics and also revamped the program output to get the following:

- Self-contained three-level-help command.
- Function key availability.
- Self-contained help.
- Better error recovering.
- Better display signaling.
- Direct access of sub-commands.

At this level of modification, I was almost "boiled" with my achievements. I couldn't stop or even reduce my speed. Instead, I rushed into a sprint resulting in the following additions:

- Selectable Angle Function.
- Formula-Form Printout.
- Selectable Printer Character Width.
- 1000 Cells Available.
- Function or String Storage Buffer.
- Left Justified Graphics.

Then, it happened!! While compiling, I got an "out of memory" advice . . . And once again, I found myself cleaning up the coding and recovering byte-by-byte the necessary room to compile. I was able to freed some 700 plus bytes after compilation and still there are possibilities to get an additional 1K or so, after further optimization. This will bring enough room for future enhancements or modifications. Anyhow, the effort was rewarding indeed, since the resulting modified program was a Super-Cheapcalc!!!

And certainly, any resemblance with a commercial product is absolutely intentional!!!

Limitations

As a rule, all computer programs must have limitations and Super-Cheapcalc has limitations, too. Stay away from programs "without limitations", those have more bugs than time you will spend on debugging.

To begin with, the Super-CheapCalc was written under CP/M and MBASIC version 5.21. This program is intended to be compiled, as explained earlier. If you don't have a Compiler, I recommend you buy one, or find someone to compile the program for you.

All the new 18 Math functions follow the conventions and limitations of MBASIC. See your documentation and also take a look at the excellent article on Integers in REMark issue 73, page 71, by Kenneth Mortimer.

The math functions accept as argument the content of a cell, a numerical value or an arithmetic function only. Thus, @COS(A1) or @TAN(1) or @SQR(30-A2) or @LON(A1*A2-B3) are valid inputs. @SQR(A1)*2, returns only the square root of A1 cell content. Not nested operation are allowed as @SQR(@ATN(1)).

The Fast Page Moving is incremental. It is not a real "next-page" command. The sequence of the incremental moving is A, B, D, H, L, etc. or C, F, J, N, etc. Where the letters are columns. Besides, vertical next-page is not provided either.

Customizing The Program For Your Printer

The routines affecting your printout start at line 2380. The lines used to select the printer character-width are 2450, 2460 and 2470. The codes are those required for my Epson FX-80. If your printer is different, you have the choice of either deleting the printer formatting routines or modifying the code to meet your printer requirements. The three mentioned lines should contain the code to print at 80, 96 and 132 characters per line. Should your printer have the option of hardware character-width selection, then your choice may be deletion of the mentioned lines, and replacing line 2430 by a RETURN in order to disable the character selection. Line 2560 also has a statement affecting the printer. LPRINT CHR$(27)"@" is used by Epson to reset the printer.

@SIN	SINE	
@COS	COSINE	Table 1
@TAN	TANGENT	Math Functions
@ATN	ARCTANGENT	
@ASN	ARCSINE	
@ACS	ARCOSINE	
@SQR	SQUARE ROOT	
@EXP	EXPONENT	
@LON	NATURAL LOGARITHM	
@LOC	COMMON LOGARITHM	
@INV	INVERSE	
@RND	RANDOM NUMBER GENERATOR	
@INT	INTEGER	
@ABS	ABSOLUTE	
@FIX	TRUNCATE	
@CIN	ROUNDING	
@MAX	FIND MAXIMUM VALUE	
@MIN	FIND MINIMUM VALUE	
&SUM	SUMATION	

On Benchmarking

As mentioned earlier in this article, a benchmark is a must to make some comparisons with commercial programs and to realize how good Super-Cheapcalc is. The Sextant article by Kenneth A. Patrick, was published in issue 4, page 40 and compares 8 spreadsheets: SuperCalc/PeachCalc, ZenCalc, CalcStar, Multiplan, PlannerCalc, MasterPlanner, Microplan and PerfectCalc.

Regarding math functions, neither one has Inverse, Random, CIN, Secant, Cosecant or Input Constant functions as Super-CheapCalc have. And neither one has the possibility to include any additional function, either.

Two benchmarks were published. Bench1, the first one, consists of four columns of 100 cells. (Super-Cheapcalc allows 50 cells only. Thus, twice that number of columns were used.) Cell A1 was the number 1010, and subsequent column entries were merely incremented by 1 from that. The second column is the value of column 1 multiplied by a constant 2. Column 3 checks out the function @LON(x), using the number at column 1 as an argument to the function. The last column merely sums the first three columns using the built-in &SUM function. In all cases, all four columns were set to 15 characters wide, 5 digits of precision.

	SuperCalc/ PeachCalc	ZenCalc	CalcStar	Multiplan	PlannerCalc	Master Planner	MicroPlan	PerfectCalc	SUPER-CHEAPCALC
Program Load Time	0:06	0:06	0:13	0:29	0:11	0:13	0:17	0:08	0:08
Bench1 Load Time	1:16	2:41	1:20	0:19	0:06	0:24	0:10	2:56	0:09
Bench1 Calc Time	1:00	1:24	3:55	1:06	3:00	1:23	0:19	1:24	1:43
Bench1 File Size	16K	8K	18K	8K	4K	2K	4K	6K	10K
Bench2 SIN	0:42	1:29	N/A	0:13	N/A	N/A	N/A	1:29	0:62
+300	0:02	0:07	1:25	0:04	---	0:23	0:01	0:09	0:49
n4	1:47	0:27	2:31	0:25	N/A	N/A	N/A4	N/A	0:27
n$n*n$n	0:045	0:17	2:22	0:05	---	0:26	0:04	0:17	1:15
Price (1983)	$375	$99.95	$295	$275	$99	$495	$495	$495	Free to $20

Table 2

Bench2, the second benchmark, consisted of the numbers 1–200 in column one (4 columns were used), while next column contained the function to be tested, 200 times using column one as the argument value to the function. I tested only four functions, because I considered it was enough.

Table 2 shows the results as published against Super-CheapCalc performance. Only results at 2 MHz were tested, since my computer clock uses that frequency only.

Enough for comparisons, but if you ever have interest to run all tests, grab the Sextant article and convince yourself. However, don't think that Super-CheapCalc will replace Lotus 1-2-3 or Multiplan. If you do a lot of calculations either for finance or engineering, your choice must be the commercial version. If your Company or business cannot afford the expense of the commercial spreadsheet, then it means your needs are limited and Super-ChepCalc may well fill the bill.

To finish, let me tell you that I don't pretend that this program is be perfect. After all, that's why we all are HUG members, to help each other and to improve programming and procedures. I guess that one field of experimentation is the nesting of math functions. Perhaps the use of parentheses is the trick to solve the problem. I did include the math functions the dirty way, hence it is possible to modify the subroutine in order to allow nesting with better coding and a better logical approach. Another area of improvement, that I think may be done very easily, is the fast page move, both vertically and horizontally, with "true next page moving." Finally, I think that the structure of the program may be modified to get a more efficient coding. As I told before, I preserved the original structure of CheapCalc and certainly the listing does not reflect all of my recommendations to speed up a program.

One final advice. Don't type in the program with the REMarks and ignore the line formatting. Once loaded under MBASIC, I got 18512 bytes free. My system is an H-89, 64K, CP/M 2.2.03 and MBASIC 5.21. I compiled using BASCOM=SCC/N/E and L80 SCC, SCC/N/E. After compilation, L80 (ver. 3.4) reported 713 bytes free.

Listing

```
0 'ELECTRONIC WORKSHEET
1 'WRITTEN FOR THE APPLE II BY WILLIAM V R SMITH
2 'MODIFIED FOR H/Z-89/90 BY BOB MCFARLAND MAY 1983
3 'SUPER-CHEAPCALC VER 2.0 FOR HEATH-ZENITH H/Z-89/90
4 'UNDER MBASIC VER 5.21 AND CP/M VER 5.3.03 (APR 1986)
5 'BY LUIS E  SUAREZ
6 'PO BOX 66994, CARACAS 1061-A, VENEZUELA
7 '*****************************************************
8 ' INITIALIZATION
9 '
10 DEFINT B-Z:MR=50:MC=20.
   DIM A$(50,20),B$(50,20),CW(20),YX(20)
20 SY=1:XM=1:SX=1:WIDTH 255:W$=IV$+"DEGREES"+NV$:W=1:
   AU=.0174533
30 BP$=CHR$(7):E$=CHR$(27):EH$=E$+"H":CS$=E$+"E".
   COF$=E$+"x5":CON$=E$+"y5"
40 CL$=E$+"K":IV$=E$+"p":NV$=E$+"q":EN$=E$+"xl":DS$=E$+"yl"
50 DEF FN PC$(R,C)=E$+"Y"+CHR$(R+31)+CHR$(C+31)
60 PRINT E$+"w"COF$CS$EN$FN PC$(25,1)CL$
70 FOR X=1 TO XM:FOR Y=1 TO YX(X):A$(Y,X)="":B$(Y,X)="":
   NEXT:NEXT
80 PRINT EN$FN PC$(25,28)"Super-CheapCalc  Ver  2.0"EH$
90 FOR X=1 TO MC:YX(X)=1:CW(X)=9:NEXT:OF=1
100 T$="ABCDEFGHIJKLMNOPQRST"
110 ON ERROR GOTO 3130:GOTO 1380
120 L=LEN(A$(Y,X)):F=2:A1=0:A2=0:P=1:H$="":IF L=0 THEN 440
130 IF P>L THEN 440
140 GOSUB 620:IF C=64 THEN P=5:GOTO 2700
150 IF C>64 THEN GOSUB 480:IF P>L THEN RETURN
160 IF C=46 THEN 220
170 IF C>41 AND C<48 THEN GOSUB 350:F=C-41:GOTO 130
180 IF C=94 THEN GOSUB 350:F=7:GOTO 130
190 IF C=38 THEN 840
200 IF C>47 AND C<58 THEN 220
210 IF C=58 THEN 880 ELSE 440
220 H$=H$+CHR$(C):IF P>L THEN GOSUB 350:GOTO 640
230 GOSUB 620:GOTO 150
240 IV=1:I$=A$:PRINT A$;FN PC$(2,1),
250 PRINT CON$;:A$=INPUT$(1):A=ASC(A$)
    IF A<>8 AND A<>127 THEN 280
260 L=LEN(I$):I$=MID$(" "+I$,2,L-1):PRINT FN PC$(2,1)I$;:
    IF A=127 THEN PRINT CL$;
270 IF LEN(I$)=0 THEN A$="":RETURN ELSE 250
280 IF A=21 THEN 330
290 IF A=13 THEN 340
300 IF A<31 THEN 250
310 IF A=34 THEN 250
320 I$=I$+A$:PRINT COF$;FN PC$(2,1)I$CON$;.IV=LEN(I$).
    GOTO 250
330 A$=MID$(A$(Y,X),IV+1,1):GOTO 320
340 IF LEFT$(I$,1)="\"THEN A$=STRING$(30,MID$(I$,2,1)):
    RETURN ELSE A$=I$:RETURN
350 A2=VAL(H$):H$="":F1=F:F=2
360 ON F1 GOSUB 380,390,370,400,370,420,410
370 RETURN
380 A1=A1*A2:RETURN
390 A1=A1+A2:RETURN
400 A1=A1-A2:RETURN
410 A1=A1^A2:RETURN
420 IF A2<>0 THEN A1=A1/A2 ELSE ERROR 254
430 RETURN
440 H$=MID$(A$(Y,X),1,L)
450 IF LEFT$(A$(Y,X),1)="?" THEN 470
460 B$(Y,X)=LEFT$(H$,CW(X))
470 GOSUB 800:RETURN
480 X3=C-64:IF C=94 THEN RETURN
490 IF X3>MC OR (L=2 AND VAL(H$)>.00001) THEN GOSUB 440:
    RETURN
500 H$="":IF L=1 THEN 440
510 GOSUB 620:IF C<48 OR C>57 THEN 440
520 GOTO 540
530 GOSUB 620
540 IF C<48 OR C>57 THEN 570
550 H$=H$+CHR$(C):IF P>L THEN 570
560 GOTO 530
570 Y3=VAL(H$)
580 IF Y3>MR OR X3>MC THEN H$="ERROR":P=L+1:GOTO 660
590 H$=B$(Y3,X3)
600 GOSUB 350:IF P>L THEN GOSUB 640
610 RETURN
620 C=ASC(MID$(A$(Y,X),P,1)):P=P+1
630 RETURN
640 IF A$(Y,X)="" THEN B$(Y,X)="":RETURN
650 IF LEN(STR$(INT(A1)))>CW(X) THEN H$="ERROR"
660 ON OF GOSUB 700,750,760,770,830
670 IF OF=4 OR OF=1 OR OF=5 THEN 800
680 IF F2<>1 THEN B$(Y,X)=STR$(A1) ELSE H$=STR$(A1)
690 GOTO 800
700 H$=STR$(A1):IF INSTR(H$,"E") THEN 740
710 H$=H$+".00":LP=INSTR(H$,"."):H$=LEFT$(H$,LP+2)
720 H$=STR$(A1)+".00":LP=INSTR(H$,"."):H$=LEFT$(H$,LP+2)
730 IF RIGHT$(H$,1)="." THEN H$=LEFT$(H$,LEN(H$)-1)+"0"
740 B$(Y,X)=H$:RETURN
750 A1=INT(A1):RETURN
760 RETURN
770 A1=CINT(A1):IF A1>30 THEN A1=30
780 IF A1<1 THEN A1=1
790 B$(Y,X)=STRING$(A1,"*"):RETURN
800 IF X=>XM THEN XM=X
810 IF Y>YX(X) THEN  YX(X)=Y
820 IF F2=1 THEN 2730 ELSE RETURN
830 H$=STR$(A1):B$(Y,X)=H$+"%":RETURN
840 P=P+4:GOSUB 620:GOSUB 480:Y4=Y3:X4=X3:GOSUB 620:
    GOSUB 480
850 A1=0:A2=0:X5=X3:Y5=Y3:IF Y4=Y5 THEN 870
860 X3=X4:FOR Y3=Y4 TO Y5:P=1:GOSUB 590:NEXT:GOSUB 640:
```

```
    RETURN
870 Y3=Y4:FOR X3=X4 TO X5:P=1:GOSUB 590:NEXT:GOSUB 640
    RETURN
880 GOSUB 620
890 IF C=36 THEN OF=1
900 IF C=73 THEN OF=2
910 IF C=70 THEN OF=3
920 IF C=42 THEN OF=4
930 IF C=37 THEN OF=5
940 GOTO 130
950 X2=X:Y2=Y
960 FOR X1=1 TO XM
970 PRINT FN PC$(1,40)IV$"WORKING"NV$;:FOR Y1=1 TO YX(X1)
980 PRINT FN PC$(1,50)MID$(T$,X1,1)Y1;
990 IF LEFT$(A$(Y1,X1),1)<>"?"
    THEN 1020 ELSE PRINT FN PC$(2,1)CL$CON$A$(Y1,X1),.
    LINE INPUT" ";T3$:PRINT COF$
1000 IF T3$<>"" THEN B$(Y1,X1)=T3$:T3$=""
1010 SWAP A$(Y1,X1),B$(Y1,X1):T3$=B$(Y1,X1)
1020 IF A$(Y1,X1)="" THEN 1050
1030 X=X1:Y=Y1:GOSUB 120
1040 IF LEFT$(T3$,1)="?" THEN A$(Y,X)=T3$:T3$=""
1050 NEXT:NEXT:X=X2:Y=Y2:GOSUB 1170:PRINT FN PC$(1,25)CL$;
1060 PRINT FN PC$(1,25)CL$;
1070 GOTO 1390
1080 A=X:Y1=Y-SY:IF LEN(A$(Y,X))>0 THEN GOSUB 120
1090 C0=5:IF X=SX THEN 1110
1100 FOR X2=SX TO X-1:C0=C0+CW(X2):NEXT X2:
    IF C0>80 THEN C0=80
1110 IF CW(A)<LEN(B$(SY+Y1,A)) THEN ERROR 5
1120 IF LEFT$(B$(SY+Y1,A),1)="*"
    THEN PRINT FN PC$(Y1+5,C0)B$(SY+Y1,A)SPACE$(CW(A)-
    LEN(B$(SY+Y1,A)))NV$;:GOTO 1160
1130 IF LEFT$(B$(SY+Y1,A),1)="<"
    THEN PRINT FN PC$(Y1+5,C0)MID$(B$(SY+Y1,A),2)
    SPACE$(CW(A)-LEN(MID$(B$(SY+Y1,A),2)))NV$;:GOTO 1160
1140 IF LEFT$(B$(SY+Y1,A),1)="^"
    THEN PRINT FN PC$(Y1+5,C0)SPACE$((CW(A)/2)-
    (LEN(MID$(B$(SY+Y1,A),2))/2))MID$(B$(SY+Y1,A),2)
    SPACE$((CW(A)/2)-(LEN(MID$(B$(SY+Y1,A),2))/2))NV$;:
    GOTO 1160
1150 PRINT FN PC$(Y1+5,C0)SPACE$(CW(A)-
    LEN(B$(SY+Y1,A)))B$(SY+Y1,A)NV$;
1160 RETURN
1170 PRINT CS$ FN PC$(4,1)IV$SPACE$(5);:MT=MC
1180 PP=0:FOR FX=SX TO MC:PP=PP+CW(FX)
1190 IF PP>77 THEN MT=FX-1:FX=MC+1
1200 NEXT:FOR FX=SX TO MT
1210 H1=CW(FX)/2:H2=CW(FX)-H1-1
1220 PRINT SPACE$(H1)MID$(T$,FX,1)SPACE$(H2);
1230 NEXT:PRINT CL$FN PC$(5,1);
1240 YD=18:IF SY+YD>MR THEN YD=MR-SY
1250 FOR FX=SY TO YD+SY:PRINT IV$;FX;.
    IF FX<10 THEN PRINT " ";
1260 PRINT NV$;CL$:NEXT:A=SX:T=5
1270 PRINT FN PC$(5,1);:YD=YX(A)-SY+1.IF YD>18 THEN YD=18
1280 FOR Y1=0 TO YD
1290 IF LEN(B$(SY+Y1,A))=0 THEN 1350
1300 Z=T+CW(A)-LEN(B$(SY+Y1,A)):IF CW(A)<LEN(B$(SY+Y1,A))
    THEN ERROR 252
1310 IF LEFT$(B$(SY+Y1,A),1)="*"
    THEN PRINT FN PC$(Y1+5,T)B$(SY+Y1,A)SPACE$(CW(A)-
    LEN(B$(SY+Y1,A))),:GOTO 1350
1320 IF LEFT$(B$(SY+Y1,A),1)="<"
    THEN PRINT FN PC$(Y1+5,T)MID$(B$(SY+Y1,A),2)
    SPACE$(CW(A)-LEN(MID$(B$(SY+Y1,A),2))),:GOTO 1350
1330 IF LEFT$(B$(SY+Y1,A),1)="^"
    THEN PRINT FN PC$(Y1+5,T)SPACE$((CW(A)-
    LEN(MID$(B$(SY+Y1,A),2)))/2)MID$(B$(SY+Y1,A),2),
    GOTO 1350
1340 PRINT FN PC$(Y1+5,Z)B$(SY+Y1,A);
1350 NEXT
1360 T=T+CW(A):A=A+1:IF A=<MT THEN 1270
1370 RETURN
1380 DF=1:PRINT CS$:GOSUB 1170:X=1:Y=1
1390 PRINT FN PC$(1,1)MID$(T$,X,1)Y;
1400 PRINT FN PC$(2,1)CL$A$(Y,X)FN
    PC$(1,5)CL$SPACE$(CW(X)-4)"¦<-"CW(X)
1410 PRINT FN PC$(1,60)IV$W$CL$;:GOSUB 1080
1420 PRINT BP$CON$CHR$(8),:A$=INPUT$(1):A=ASC(A$):
    PRINT COF$;:IF A<>27 THEN 1490 ELSE A$=INPUT$(1)
    B=INSTR(" C,L,M,D,B,A,@,N",A$)/2
1430 ON B+1 GOTO 1480,1440,2570,2570,1450,1460,1470,3250,3260
1440 DF=-1:GOTO 1570
1450 DF=-1:GOTO 1660
1460 DF=1:GOTO 1570
1470 DF=1:GOTO 1660
1480 A$=CHR$(0)+A$:GOTO 1900
1490 PRINT FN PC$(3,1)CL$;:IF A=47 THEN 1880
1500 IF A=4 THEN 2560
1510 IF A=35 THEN A$(Y,X)="":GOTO 1790
1520 IF A=33 THEN 950
1530 IF A>43 THEN 1770
1540 IF A=38 THEN 1810
1550 IF A=34 THEN A$=CHR$(0):GOTO 1770
1560 GOTO 1420
1570 GOSUB 1080:ON DF+2 GOTO 1580,1390,1610
1580 X=X+1:IF X>MC THEN X=MC:GOTO 1600
1590 IF X>MT THEN SX=SX+1:GOSUB 1170:GOTO 1590
1600 PRINT IV$:GOSUB 1080:GOTO 1390
1610 Y=Y+1.IF Y>MR THEN Y=MR:GOTO 1650
1620 IF Y>18+SY THEN X3=-1:SY=SY+10:Y=SY+9
1630 IF Y>MR THEN Y=MR:SY=MR-18
1640 IF X3=-1 THEN GOSUB 1170:X3=0
1650 GOTO 1600
1660 ON DF+2 GOTO 1670,1760,1720
1670 GOSUB 1080
1680 X=X-1:IF X>=SX THEN 1600
1690 SX=SX-1:IF X=0 THEN X=1:SX=1:GOTO 1710
1700 GOSUB 1170
1710 GOSUB 1600
1720 GOSUB 1080
1730 Y=Y-1.IF Y=>SY THEN 1600
1740 SY=SY-10:IF SY<=0 THEN SY=1:IF Y<=0 THEN Y=1:SY=1
1750 GOSUB 1170:GOSUB 1600
1760 IF A$="&" THEN 1810
1770 PRINT FN PC$(2,1)A$(Y,X):PRINT FN PC$(2,1),
    GOSUB 240
1780 IF A$="" THEN 1800 ELSE A$(Y,X)=A$
1790 GOSUB 120
1800 GOTO 1390
1810 PRINT FN PC$(1,1)CL$;
1820 PRINT FN PC$(2,1)CL$CON$;:INPUT "&SUM(",A$
1830 IF RIGHT$(A$,1)<>")" THEN A$=A$+")"
1840 PRINT FN PC$(2,1)CL$COF$"&SUM(";A$;
1850 B$=MID$(A$,INSTR(A$,"-")+1):
    A$=LEFT$(A$,INSTR(A$,"-")-1)
1860 B$=LEFT$(B$,LEN(B$)-1):A$(Y,X)="&SUM("+A$+"-"+B$+")"
1870 GOSUB 120:GOTO 1390
1880 CLOSE #1:PRINT FN PC$(2,1);CL$;
1890 PRINT EN$FN PC$(25,1)CL$"      WIDTH  GOTO  HELP1
    HELP2   ANGL  ERASE   LOAD  SAVE  PRINT
    esc-DELETE >"CON$;.A$=INPUT$(2):PRINT EH$;COF$;:
    IF ASC(A$)<>27 THEN 1390
1900 IF MID$(A$,2)="J" THEN 3000
    ELSE IF ASC(MID$(A$,2))=27 THEN 1390
1910 PRINT EN$FN PC$(25,1)CL$EH$
1920 IF MID$(A$,2)=CHR$(127)
    THEN PRINT CON$"CLEAR SHEET? Y/N",:A$=INPUT$(1)
    IF A$="Y" OR A$="y" THEN 70 ELSE 1390
1930 ON ASC(MID$(A$,2))-79 GOTO 2150,2070,2370,2000,
    2230,1950,1980,2030
1940 PRINT BP$:GOTO 1890
1950 PRINT EN$FN PC$(25,1)CL$"@SINn  @COSn  @TANn  @ASNn
    @ACSn  @ATNn  @SQRn  @EXPn  @LONn  @LOCn
    <CR> >>>"CON$;:A$=INPUT$(1)
1960 PRINT EN$FN PC$(25,1)CL$"@INTn  @ABSn  @FIXn  @CINn
    @INVn  @RNDn  @MAXn @MINn  ?input prompt <CR>
    >>>"CON$;:A$=INPUT$(1)
1970 PRINT EN$FN PC$(25,1)CL$" I=Integer :$=Dollars/Cents
    .F=Floating :%=Percent :*=Graphics
    @=Math functions"EH$:GOTO 1390
1980 PRINT EN$FN PC$(25,1)CL$"#=ClearCell !=Recalc &Sum
    /=Help \=Fill cell <=Left justify ^=Center text
    >>>"CON$;:A$=INPUT$(1)
```

```
1990 PRINT EN$FN PC$(25,1)CL$"IL=Copy   DL=Replicate
     IC=Peek  DC=Poke  CTRL-U=move cursor/Edit
     CTRL-D=Quit"EH$:GOTO 1390
2000 PRINT FN PC$(2,1)CL$CON$;:LINE INPUT " WIDTH= ";A$:
     PRINT COF$;:A=VAL(A$):IF A>30 OR A<4 THEN PRINT BP$;
     :GOTO 1390
2010 IF RIGHT$(A$,1)="." THEN FOR X9=1 TO 20:CW(X9)=A.
     NEXT ELSE CW(X)=A
2020 YH=Y:XH=X:FOR Y=1 TO YX(X):GOSUB 440:NEXT:Y2=YH:
     X2=XH:GOTO 960
2030 ON W GOTO 2040,2050,2060
2040 W=2:W$="RADIANS":AU=1:GOTO 1390
2050 W=3:W$="GRADS":AU=.015708:GOTO 1390
2060 W=1:W$="DEGREES":AU=.0174533:GOTO 1390
2070 GOSUB 2220:PRINT FN PC$(1,1)CL$;
2080 PRINT CON$;:LINE INPUT"FILE NAME TO SAVE = ";A$:
     IF A$="" THEN RETURN
2090 PRINT FN PC$(1,1);CL$
2100 OPEN "O",#1,A$:PRINT #1,XM
2110 FOR X=1 TO XM:PRINT#1,CW(X),YX(X)
2120 FOR Y=1 TO YX(X):PRINT#1,CHR$(34)A$(Y,X)CHR$(34)
2130 PRINT#1,CHR$(34);B$(Y,X);CHR$(34)
2140 NEXT:PRINT #1,"<>":NEXT:PRINT #1,"<>":CLOSE #1:
     Y1=1:X1=1:GOTO 2550
2150 PRINT FN PC$(1,1);
2160 PRINT CON$"FILE NAME TO LOAD = "CL$;:LINE INPUT A$
2170 IF A$="" THEN GOSUB 2220:GOTO 1390 ELSE FOU$=A$
2180 OPEN "I",1,FOU$
2190 WHILE NOT EOF(1):INPUT #1,XM:FOR X=1 TO XM:
     INPUT #1,CW(X),YX(X)
2200 FOR Y=1 TO YX(X):INPUT #1,A$(Y,X),B$(Y,X):NEXT:
     INPUT#1, B$:NEXT:INPUT#1,B$
2210 WEND:CLOSE #1:GOSUB 2220:GOTO 2550
2220 FOR B=1 TO 3:PRINT FN PC$(B,1)CL$;:NEXT:RETURN
2230 GOSUB 2240:GOTO 1390
2240 GOSUB 2220
2250 PRINT FN PC$(2,1)CON$;:A$="":
     LINE INPUT "GOTO CELL, <CR> FAST MOVE: ";A$:
     IF A$="" THEN A$=CHR$(X+X+1-SX+64)+STR$(Y):
     A$=LEFT$(A$,2)+RIGHT$(A$,2)
2260 GOSUB 2320
2270 IF X1*Y1=0 THEN RETURN
2280 X=X1:SX=X1:Y=Y1:SY=Y1:PRINT CS$
2290 SY=SY-18:IF SY<1 THEN SY=1
2300 SX=SX-3:IF SX<1 THEN SX=1
2310 GOSUB 1170:GOTO 1390
2320 L=LEN(A$):IF L<2 THEN X1=0:Y1=1:RETURN
2330 X1=ASC(A$)-64:Y1=VAL(RIGHT$(A$,L-1))
2340 IF X1<1 OR X1>MC THEN X1=1
2350 IF Y1<1 OR Y1>MR THEN X1=0:Y1=0
2360 RETURN
2370 GOSUB 2220
2380 PRINT FN PC$(2,1)CL$CON$;:INPUT "PRINT 1)SHEET
     2)FORM ";A$:IF LEFT$(A$,1)="2" THEN 2950
2390 PRINT CON$FN PC$(2,1)CL$;.INPUT "UPPER/LEFT CORNER:";
     A$:GOSUB 2320
2400 X3=X1:Y3=Y1
2410 PRINT FN PC$(2,1)CL$CON$;:INPUT "LOWER/RIGHT CORNER: ";
     A$:GOSUB 2320
2420 X4=X1:Y4=Y1:GOSUB 2430:GOTO 2490
2430 PRINT FN PC$(2,1)CL$CON$;:INPUT "1)80 CPL  2)96 CPL
     3)132 CPL";A$:PRINT COF$:IF A$="" THEN PRINT BP$:
     GOTO 2430
2440 ON VAL(A$) GOTO 2450,2460,2470
2450 LPRINT E$+CHR$(33)+CHR$(0);:RETURN
2460 LPRINT E$+CHR$(33)+CHR$(1);:RETURN
2470 LPRINT E$+CHR$(33)+CHR$(4),
2480 RETURN
2490 FOR Y1=Y3 TO Y4:FOR X1=X3 TO X4
2500 IF LEFT$(B$(Y1,X1),1)="<"
     THEN LPRINT MID$(B$(Y1,X1),2)SPACE$(CW(X1)-
     LEN(B$(Y1,X1))),:GOTO 2540
2510 IF LEFT$(B$(Y1,X1),1)="^"
     THEN LPRINT SPACE$((CW(X1)-
     LEN(B$(Y1,X1)))/2)MID$(B$(Y1,X1),2)SPACE$((CW(X1)-
     LEN(B$(Y1,X1)))/2),:GOTO 2540
2520 IF LEFT$(B$(Y1,X1),1)="*" THEN LPRINT B$(Y1,X1);:
     GOTO 2540
2530 LPRINT SPACE$(CW(X1)-LEN(B$(Y1,X1)))B$(Y1,X1);
2540 NEXT:LPRINT:NEXT:LPRINT E$+"@"
2550 X1=1:Y1=1:GOTO 2280
2560 PRINT DS$CS$E$+"z":WIDTH 80:END
2570 PRINT FN PC$(3,1)CL$BP$:IF CR<1 THEN XH=X:YH=Y
     T2$=A$(Y,X):CR=CR+1:GOTO 1420
2580 LD=0:A$(Y,X)=A$(YH,XH):IF A$="L" THEN 2690
2590 DX=X-XH:DY=Y-YH:LA=LEN(A$(YH,XH)):IF LA<1 THEN 1420
2600 IS=1:IF ASC(A$(Y,X))=38 OR ASC(A$(Y,X))=64 THEN IS=5
2610 FOR I=IS TO LA:A$=MID$(A$(YH,XH),I,1).A=ASC(A$)
2620 IF A<65 OR A>65+MC THEN 2680
2630 IF I=1 THEN A$(Y,X)=CHR$(A+DX):GOTO 2650
2640 A$(Y,X)=MID$(A$(Y,X),1,I-1+LD)+CHR$(A+DX)
2650 N=VAL(MID$(A$(YH,XH),I+1))
2660 N$=STR$(N):LN=LEN(N$):N=N+DY:N$=STR$(N):LD=LEN(N$)-LN
2670 A$(Y,X)=A$(Y,X)+MID$(N$,2)+MID$(A$(YH,XH),I+LN)
2680 NEXT
2690 CR=0:GOSUB 120:GOTO 1390
2700 F2=1:T3$=MID$(A$(Y,X),2,3):H$=MID$(A$(Y,X),6)
2710 T4$=LEFT$(H$,LEN(H$)-1):IF T3$="MAX" OR T3$="MIN"
     THEN GOSUB 3030:GOTO 640
2720 L=LEN(T4$):SWAP A$(Y,X),T4$:GOTO 120
2730 SWAP A$(Y,X),T4$
2740 F2=0:A2=VAL(H$)
2750 B=1+INSTR(" @ATN@INV@COS@EXP@LOC@LON@RND@SIN@SQR
     @TAN@INT@ABS@FIX@CIN@ASN@ACS","@"+T3$)/4
2760 ON B GOTO 2770,2780,2790,2800,2810,2820,2830,2840,
     2850,2860,2870,2880,2890,2900,2910,2920,2930
2770 ERROR 255
2780 A1=ATN(A2)/AU:GOTO 2940
2790 IF A2<>0 THEN A1=1/A2:GOTO 2940 ELSE ERROR 5
2800 A1=COS(A2*AU):GOTO 2940
2810 A1=EXP(A2):GOTO 2940
2820 A1=LOG(A2)/LOG(10):GOTO 2940
2830 A1=LOG(A2):GOTO 2940
2840 A1=RND(A2):GOTO 2940
2850 A1=SIN(A2*AU):GOTO 2940
2860 A1=SQR(A2):GOTO 2940
2870 A1=TAN(A2*AU):GOTO 2940
2880 A1=INT(A2):GOTO 2940
2890 A1=ABS(A2):GOTO 2940
2900 A1=FIX(A2):GOTO 2940
2910 A1=CINT(A2):GOTO 2940
2920 IF ABS(A2)>.999999 THEN ERROR 5
     ELSE A1=ATN(A2/SQR(-A2*A2+1))/AU:GOTO 2940
2930 IF ABS(A2)>.999999 THEN ERROR 5
     ELSE A1=(-ATN(A2/SQR(-A2*A2+1))+1.5708)/AU:GOTO 2940
2940 P=P+1:GOTO 640
2950 GOSUB 2430:FOR X=1 TO XM:FOR Y=1 TO YX(X)
2960 IF A$(Y,X)="" THEN 2990
2970 LPRINT CHR$(X+64);-Y;TAB(7)A$(Y,X);
2980 IF LEFT$(A$(Y,X),1)<>LEFT$(B$(Y,X),1)
     THEN LPRINT TAB(45)B$(Y,X)ELSE LPRINT
2990 NEXT:LPRINT:NEXT:LPRINT:LPRINT:LPRINT CHR$(27)+"@"
     GOTO 2550
3000 FOR X=1 TO XM:FOR Y=1 TO YX(X)
3010 IF B$(Y,X)=A$(Y,X)
     OR ABS(VAL(A$(Y,X)))=ABS(VAL(B$(Y,X)))
     THEN 3020 ELSE B$(Y,X)=""
3020 NEXT:NEXT:GOTO 1380
3030 IF T3$="MAX" THEN A1=-1E+32 ELSE A1=1E+32
3040 B$=MID$(T4$,INSTR(T4$,"-")+1):
     A$=LEFT$(T4$,INSTR(T4$,"-")-1)
3050 R1=VAL(MID$(A$,2)):C1=ASC(A$)-64
3060 R2=VAL(MID$(B$,2)):C2=ASC(B$)-64
3070 FOR C0=C1 TO C2:FOR R0=R1 TO R2
3080 IF ASC(MID$(B$(R0,C0),2))>47
     AND ASC(MID$(B$(R0,C0),2))<58
     OR ASC(MID$(B$(R0,C0),2))=46
     THEN A2=VAL(B$(R0,C0)) ELSE 3120
3090 IF T3$="MAX" AND A2<A1 THEN 3120
3100 IF T3$="MIN" AND A2>A1 THEN 3120
3110 A1=A2
3120 NEXT:NEXT:F2=0:P=P+1:RETURN
3130 IF ERR=254 THEN PRINT FN PC$(3,1),
     "Division by zero?":IF ERL=420 THEN RESUME 430
     ELSE A1=0:RESUME 640
```

```
3140 IF ERL=620 AND ERR=5 THEN H$="ERROR":RESUME 630
3150 IF ERL>1100 AND ERL<1160
     THEN PRINT FN PC$(Y1+5,C0)STRING$(CW(A),"#"),:
     RESUME 1160
3160 IF ERR=252 OR ERL=1310
     THEN PRINT FN PC$(Y1+5,T)STRING$(CW(A),"#"),:
     RESUME 1350
3170 IF ERL>2570 AND ERL<2680 THEN PRINT FN PC$(3,1)
     "Wrong procedure":A$(Y,X)=T2$:CR=0:RESUME 1390
3180 IF ERL>2700 AND ERR=5 THEN B$(Y,X)="Error":RESUME 370
3190 IF ERR=255 THEN A$(Y,X)="Wrong function":RESUME 600
3200 IF ERL=1850 AND ERR=5 THEN A$(Y,X)="ERROR":RESUME 1390
3210 IF ERL=590 AND ERR=5 THEN A$(Y,X)="ERROR":RESUME 600
3220 IF ERR=53 THEN GOSUB 2220:PRINT FN PC$(1,30)
     "File not found":FOR A=1 TO 900:NEXT:CLOSE:RESUME 1390
3230 IF ERR=6 THEN A$(Y,X)="Overflow":RESUME 1390
3240 RESUME 1390
3250 WZ$=A$(Y,X):GOTO 1410
3260 A$(Y,X)=WZ$:GOTO 1400
3270 REM SUPER-CHEAPCALC VER 2.0 by Luis E. Suarez, Apr 86.
```

-- SHIPPING NOW -- SHIPPING NOW -- SHIPPING NOW --

WILDFIRE

An INCREDIBLE PERFORMANCE IMPROVEMENT from Software Wizardry

SPEED... WILDFIRE has spirit. It will spur your Z-151 or Z-161* to run faster than the 8mHz Z-158! And WILDFIRE isn't just a simple crystal change like some products; it gives you more speed and much greater compatibility.

COMPATIBILITY... WILDFIRE uses the NEC V-20 chip plus lots of other enhancements to make an unbeatable combination of performance and reliability. It outdoes the 80286 processor chip additions for compatibility with the IBM PC and costs up to $600.00 less!

FEATURES... the heart of WILDFIRE is a daughter board that mounts on your processor board, saddled comfortably between it and the video board. Several higher-speed chips are included to replace socketed chips on your processor board. A high speed/low speed switch that mounts on your machine's front panel keeps WILDFIRE on a tight rein, under your control to trot at normal speed, or break into a real gallop when you really want to ride!

PRICE... Software Wizardry brings you WILDFIRE, complete with attractive front panel, installation instructions, and reliable daughter-board design, for only $249!

●●● 15-DAY MONEY-BACK GUARANTEE! ●●●

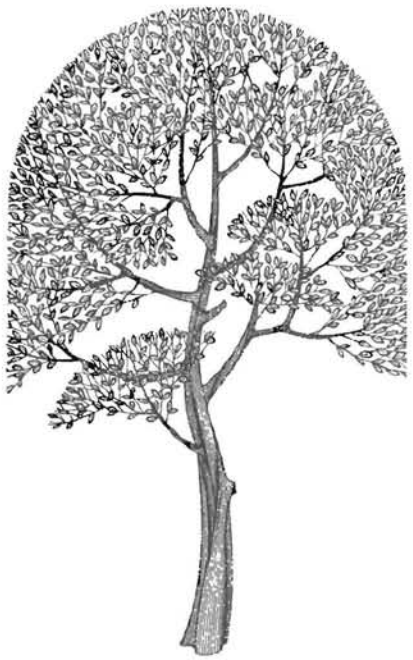
If you order now, you can try WILDFIRE with a 15-day money-back guarantee! If you don't like it, you can return it within 15 days and get a full refund!

WILDFIRE...the ultimate solution to your speed problems. And with our special low price, AND our 15-day money-back guarantee, you can't go wrong! Order now!

Contact your local dealer, or order direct from Software Wizardry by calling (314)724-1738. Dealer inquiries invited.

Software Wizardry

1106 First Capitol Drive
St. Charles, MO 63301
(314)724-1738



Roots II

A Dream Program For Genealogists

Part 2

Tom Huber
Senior Technical Writer
Zenith Data Systems

Roots II is the latest of a series of genealogical programs from CommSoft. It is designed for use on MSDOS-based computers, including some of the older computer designs that have been converted to use the PC (8088 processor) design. In short, it is the cadillac of all genealogy programs. Last month, I briefly described the program, provided an explanation about genealogy, and talked briefly about installing *Roots II* on a PC-compatible computer. This month, I will continue my discussion on installing the program and talk about some of its features. Next month, I will conclude the discussion on the features of *Roots II* and summarize my findings.

The Zenith Z-100 And Installing *Roots II*

On 5.25-inch, single-drive floppy-only systems, normal installation requires that Z-DOS or MSDOS be placed on the disk. To do this the manual tells you to use the SYS command, but there is something in the way the Z-DOS and MSDOS operating systems are implemented on the Zenith Z-100 that does not permit SYS to operate on PC-compatible software. Therefore, Z-DOS and MSDOS on the Z-100 will not install onto the distribution disks. Instead, an error message is generated.

If you reformat the disk, the key is destroyed. After some experimenting, I discovered a way to set up the operating system on the disks without destroying the key. However, it is different for Z-DOS and MSDOS.

Zenith's MSDOS Version 2 and 3 for the Z-100 are the easiest. When you call the FORMAT utility, use the /C/S switches in conjunction with each other. They work together. For those of you who have not explored MSDOS version 2 or 3, the /C switch in the FORMAT utility clears the directory, but does not reformat the disk. Therefore, the key is left alone. But, before you use the /C switch, make sure you have backed up your disks (with the COPY) command . . . at least twice. And verify the copy process while you do it (use the VERIFY ON command). Then enter FORMAT B:/C/S (assuming you have the *Roots II* distribution disk in drive B).

MSDOS version 2 occasionally caused problems when I tried this procedure, but the latest copies appear to work properly. Earlier copies of MSDOS version 2 format the disk, even with the /C and /S switches. While I was not able to repeatedly duplicate this problem, be aware that you may experience some problem with MSDOS version 2. If you can detect (hear) the heads stepping across the disk when you run FORMAT, then listen carefully. If you hear the heads stepping at regular intervals (click, click, click . . .) then **OPEN THE DISK DRIVE AT ONCE**; the disk is being formatted. If you catch it early enough, you will not destroy the key. Once a disk has started the format procedure, you will need to reset your computer. Then run SYS on the disk. If the transfer takes place without any error messages, then you probably can use the disk. If SYS causes an error message, then you may have other problems as well. Attempt to copy files back onto the disk with VERIFY set to ON. If you have no problems getting the system installed or the files copied to the disk, transfer the Z100 console (.COD) file as instructed and start Roots. If Roots starts with no problems, then you have successfully transferred the system (albeit, in an awkward, not-recommended fashion) without destroying the key. If Roots produces an error, then you have destroyed the key. At that point, you might as well contact CommSoft for instructions on returning the disk along with a copy of your operating system. They will make the transfer and install the operating system for you.

Under Z-DOS, you can partially format the disk so that it will accept the SYS command. However, this takes a bit of trickery with the operating system to do the job, and one I do not recommend unless you really know your computer and have relatively "noisy" disk drives. What you will be doing is to prepare the first few tracks of each distribution disk for the Z-100 version of MS-DOS. After copying and verifying the distribution disk files onto other disks, place the selected *Roots II* distribution disk in drive B and enter

FORMAT B:/S. After you have heard the disk step about five tracks into the disk (you must be able to hear the disk heads stepping across the tracks during normal formatting) stop the process (Open the disk drive doors and reset the computer). While this aborts the format process, the key is not destroyed and the SYS command can now be used successfully.

A better procedure uses Z-DOS's DEBUG to copy the first few tracks of a newly formatted system disk onto the distribution disk. Here is the procedure:

1. Back up all distribution disk files *at least twice*.
2. Use FORMAT to format a blank disk (*not one of the distribution disks!*):

   ```
   FORMAT B:/V/S <RETURN>
   ```

3. Start DEBUG:

   ```
   DEBUG <RETURN>
   ```

4. Enter the following:

   ```
   F 1000:0 3FFF 00 <RETURN>
   ```

5. Place the newly formatted disk in drive B and enter:

   ```
   L 1000:0 1 0 10 <RETURN>
   ```

6. Remove the newly formatted disk and place the distribution disk in drive B and enter:

   ```
   W 1000:0 1 0 10 <RETURN>
   ```

7. Exit DEBUG; enter:

   ```
   Q <RETURN>
   ```

8. With the Z-DOS system disk in drive A, enter:

   ```
   SYS B: <RETURN>
   ```

The screen should display:

```
Sys version 1.5 (or something similar)
System transferred
```

9. Use CHKDSK or DIR to verify the transfer. CHKDSK will display something similar to the following:

   ```
   322560 bytes total disk space
    18432 bytes in 2 hidden files
     6144 bytes in 2 user files
   297984 bytes available on disk
   ```

   DIR will display something similar to the following:

   ```
   COMMAND COM  5114 8-23-83 8:04a
   ALTCHAR SYS   431 6-22-83 8:49a
                 2 File(s)
   ```

10. Copy the appropriate files back onto the distribution disk from your backups and you are ready to go. Now you can follow the Roots manual.

This same procedure will work with MSDOS version 2 or later but you must use a higher address (2000:0) or you will lock up your computer.

Owners of any of the Zenith PC-compatible computers (Z-150, Z-158, Z-161, Z-138, Z-148, Z-170, or Z-200 series) will be able to use SYS without any problems to copy the operating system. Only the Z-110 and Z-120 non-PC-compatible computers exhibit the reluctance to transfer the operating system.

Portable computers, such as the IBM Convertible and the Zenith Z-180 use 3.5" disks. I do not know if CommSoft plans to market a version of *Roots II* in this format or not. If you are planning to use one of these computers with *Roots II*, you need to purchase the non-keyed version of the program and transfer the files to the 3.5" format. Zenith will be selling a 5.25-inch disk accessory for the Z-180 and I understand IBM is offering a 3.5-inch accessory for their PC compatible computers.

Initializing Roots

Before you start using *Roots II*, you should thoroughly read through the appendix covering installation. INSTALL will allow you to set up almost all the operating parameters of the system. The only thing that is left out is the filename of the database for your genealogical records. That is specified the first time you start the Roots program.

If you are a professional genealogist, then you may want to create a customized database for each client. That requires separate, customized ROOTS.EXE files for each of your clients. Since this file is keyed to the distribution disk, I highly recommend the unprotected system if your computer has only one disk drive. But with the prices of disk drives being relatively low, you may want to consider getting a second drive instead. Since the keyed distribution disk does not need to be in the same drive as your boot or database disks, the unkeyed version is not absolutely required.

Before you start *Roots II* for your own records, you need to make a number of decisions. But only a couple must be made before you start entering your own data. Since the program is supplied with a sample database, you can quickly familiarize yourself with its features. While this helps one "thumb" through the program quickly, it took me several passes at actually setting up my database before I was fully satisfied. Hopefully my experiences will help you make some decisions before you start.

First, there are four user-defined date/place fields provided in the individual (subject) and marriage records. There are three ways to set these up.

1. The Commsoft standard predefines the date/place fields as *buried, probate*, and *immigration* for individuals and *engagement* for marriages. For most non-LDS genealogists, these fields are acceptable, although I would think that adoption dates might be more useful to cover modern families where divorce and adoption of children from previous marriages sometimes takes place. The next option offers a solution to just that problem.
2. If you don't like any or all of the defined field names, then you can use the non-standard option to define your own date/place fields. The INSTALL program is used to set up your definitions of these fields.
3. LDS genealogists will find that the third option provides the necessary fields for LDS applications: *baptized, endowment*, and *sealed to parents* date/place fields for individuals and a *sealed to spouse* date/place field for marriage records. Although place fields are supplied, LDS generally do not record the place of LDS proxy baptism or endowment. The place where a sealing of parents or spouse is generally recorded as a two-letter abbreviation for a specific temple's location, such as SL for Salt Lake, LG for Logan, WA for Washington (DC), and so on. Some LDS may prefer to record the full city. For now, there has been no more than one temple built in any one city; however, that may change as more temples are constructed throughout the world and the ratio of LDS to general population continues to grow.

If you are not LDS, you may want to consider utilizing the LDS format if you use their facilities to do research. Temple ordinance

data is available to the public and I have observed a number of non-LDS making use of this data. Therefore, you may want to keep track of the LDS ordinance data, just as it is common to record christening dates from church records.

The other decision that should be made fairly early is how to link your Roots database with text files. There are pros and cons to setting up a reference field for every subject entry. Here is what I found.

The FAMILY routine allows you to view the text files as you use Roots. Because text files are disk-based, a search is made of the disk directory every time you move forward or backward through the family routines. This slows the performance of Roots considerably, especially if you are searching for a particular family group and must pass through several generations to reach it. Therefore, you may want to avoid this search by not setting up the reference fields until later, something I strongly recommend.

Another decision that you must make is how to utilize Roots to its best advantage, especially if you are researching a particular surname and gathering a lot of data on that name. Here I strongly recommend you set up a database on each unconnected surname. By keeping these databases small, Roots will run with surprising speed. Also, you keep your main database uncluttered.

By making some of these decisions in advance, you will save time and effort in setting up your genealogical records. Then Roots can become the best possible tool you could obtain for doing research.

Operation Of Roots

The program has ten functions (or applications, if you prefer) that appear in a menu once Roots has been initialized. Some of these functions act as an aid in managing the database created by *Roots II*, while others create reports from that database. The remainder of the issue's review contains a brief description of some of the functions and my thoughts toward each one's usefulness. Next month, I'll review the remaining functions.

But before I start explaining how I use each function, you need to know that Roots uses a 10-slot table called the fetch table or "f-table" for short. It is a scratch pad area that is used by the program to link various functions. Almost every function performed with Roots requires the f-table.

My first impression was that this was going to be very tedious. The thought of having to transfer every person into the f-table was a bother! Or so I thought.

After I used Roots for a while, I found the f-table to be worth its weight in gold. It is the key to linking families together, and without it, many errors could be introduced into a database that might be impossible to correct. This is probably the one thing that sets Roots apart from any other database in compiling family records.

Names located in many of the program's functions can easily be transferred into the f-table. Not all instances where this can be done are indicated in my review of each function.

LIST — The first function on the menu allows you to perform a fast search on your database on any single surname, given name, or partial name.

This function is extremely helpful if you are trying to connect one family to another and think you have discovered a common point through a marriage. As an example, let's say that I discovered a record (but no date or place) that says that Maria Hess and Jacob Huber were married. First, I would use the LIST function to search for Maria Hess using the given name "Maria." I have a very large number of Hess relatives in the database, but few Maria's. The LIST function shows me fifteen Maria's but only two have surnames of Hess; they both have birth and death dates that correspond to the approximated time period involved. So I would move both Maria's to the f-table and then go to the FAMILY function to check relationships and possible existing marriages that have already been recorded. I don't find marriage records for one Maria and the other was married to a Hertzler. Furthermore, the first Maria's father matches the father shown in the marriage record I had found. This establishes that one Maria Hess fits the category. Now I return to the LIST function and search for all the Jacob Hubers. Here I use Huber for the search because my family has a lot of Jacobs in it. As before, I will transfer the most likely candidates to the f-table and check the relationships with the FAMILY function. Again I find a Jacob Huber that fits the category. Now I would move to the EDIT function and join the two together with the MARRIAGE subfunction.

Actually, I would probably want a little better proof, such as exactly matching both parents' names with the nuptials. However, I used this as an example to show how Roots can be very useful in reviewing all the Hubers and Hess's for just the "right" two people. Before Roots, this task involved manually searching through a large number of family group sheets with the strong possibility that one or the other person might be missed.

There are several valuable features to the LIST function: they include the ability to show occupations or birth and death dates, to search on any part of the name, to search only surnames (identified when entering records) using a soundex code (either entered manually or generated by the computer), and to transfer any names found into one of the ten locations in the f-table.

The one drawback to having the soundex feature in the LIST function is that only one name can be used for the search. Hence, you cannot use two names, such as "Jacob Huber." Apparently, the author felt that being able to use a soundex code (which locates most valid variations of the name) was more important than trying to find a match against two or more names in a search. I tend to agree, but since the soundex search is optional, I would also like to be able to search on two or more names.

ANNIVERSARY — This function allows you to perform a search of the database by month and day. I often use this function to locate individuals when I know the date of an event for that individual. If this search does not locate that person, then I know that his or her record has to be edited.

In addition to the names of persons who have an event anniversary on that date, the event, number of years since the event, and year of the event are shown. As a side benefit, you can use this feature to search forward or backward from the current date for anniversaries (birthdays, in particular) of living relatives and thus avoid the embarrassment of forgetting some grandchild's or great-grandchild's birthday . . . or your own wedding anniversary!

PLACES — This function allows you to perform a search by placename. Like the LIST function, you may search by a complete or partial name. Unlike list, which allows just one name as the search key, you may enter more than one name (such as St. Joseph) as the search key. Because single names are not identified (such as Joseph, in St. Joseph), Soundex searching is not provided. You can use partial names, however, and that often allows you to find a variety of spellings.

I have used this function in three ways: first, to check the spelling of placenames; second, to make sure placenames are used consistently throughout the database; and third, to obtain a list of persons that lived in a particular place.

Recently, I used this function to prepare a list of people that lived in Indiana before I visited the South Bend (Indiana) public library. It saved me considerable time while looking for my relatives in the local histories from Indiana counties.

SEARCH — This last search function in Roots covers many possibilities. For instance, you can set flags on one or more of the nine fields in a subject record. For LDS, this includes one flag for each of the ordinance fields. In addition to selecting flags to search on, you can also specify ranges of years for the various date fields, whether the person is still living or not, the sex of the persons to be searched, birth codes, and parent codes. For instance, If you wanted to know if any of your ancestors were soldiers in the Revolutionary War, this would quickly locate all known possibilities. The SEARCH function allows you to search either subject or marriage records.

But even more important, you can search for names filed by source. And, searches can be cumulative. This could allow you to find all married men who died at the time of the Civil War who were recorded in the same source.

FAMILY — I use this function more than any other function except for EDIT, to continually check family lines for being complete and to determine what areas need work.

The display created by this function shows family groups. It helps verify that I have recorded all the known children in all the marriages of any given individual and the known birth and death dates of the people in that family group. Places are not shown, but are not needed either. If one or more text files exist, I can optionally view them.

To start the function, the desired name must be in the f-table. Once displayed, you can move forward in time to display the family group of a spouse or one of the children in the display, or backward in time to the family that forms the preceding generation.

TRACE — This function displays pedigree charts, which shows the ancestry of an individual. This function requires that the individual's name be in the f-table. Then, as you scan through the various branches of the pedigree chart, you can view either that person's vital statistics or a time line display which is useful in establishing where the person fits chronologically with his or her descendants. Symbols on the time line indicate birth, christening, marriage (and divorce), and death dates. Birth or death dates that are not recorded are approximated. Symbols representing other events are omitted if the date of the event is not recorded.

The pedigree chart is the basic outline to preparing and recording genealogical information. The vital statistics display will give you a quick glance of all recorded data (except for text files) on any individual in the pedigree.

The time line chart can be used as a troubleshooting tool to determine gaps (which might indicate a missing generation), dates out of logical sequence, and missing dates. This is a very powerful tool since ordinary pedigree charts do not provide this kind of graphic display.

Next month, I will finish this discussion of Roots' features and provide you with my overall feelings about this program.

Roots II is available from:

CommSoft
2452 Embarcadero Way
Palo Alto, CA 94303
(415) 493-2184
$195.00 (keyed version)
$295.00 (unkeyed version)

For those of you who are unfamiliar with Soundex indexing, the following might be helpful.

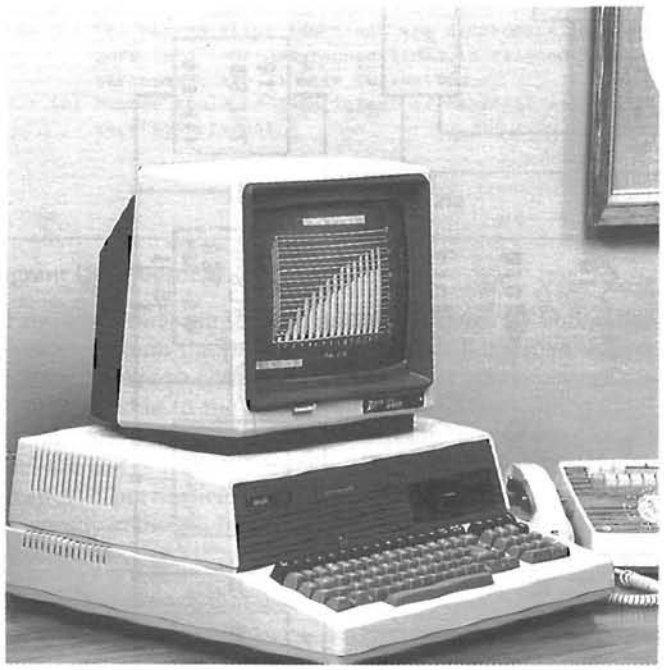
Soundex (an acronym for *sound index*) is an indexing system whereby words that sound alike, regardless of spelling are given the same values to produce a four-character code. The code consists of the word's first letter plus a three-digit numeric code based on the consonants in the word. The soundex code numbers are:

1 = b, f, p, and v
2 = c, g, j, k, q, s, x, and z
3 = d and t
4 = l
5 = m and n
6 = r

Notice that the vowels and a few consonants are not given code letters; they are: a, e, h, i, o, u, w, and y. If the word does not produce the three digits, then zeros are added to the end of the code. Thus, the name *Huber* is coded H160. If the word produces more than three digits, only the first three are used. So a word, such as *alphabetic*, would be coded A411.

When used in searching the database, names that sound similar are located. Thus a Soundex search for Smith will also display all Smythe, Smyth, Smitt, and Smithe names, all of which are valid variations of the name. In the case of Huber, Hoover is also located. This is correct, because Hoover is an anglicized version of Huber. Of the twenty-five original Hubers to settle in Lancaster County, Pennsylvania, one of them turned out to be the ancestor of President Herbert Hoover. Two publicly available federal census records are indexed by Soundex. *

ZPC Update #12



Pat Swayne
HUG Software Engineer

This is the twelfth in a series of articles in support of ZPC, a program that allows you to run IBM PC software in H/Z-100 (dual processor) computers. ZPC is available from HUG as part no. 885-3037-37. An upgrade disk for ZPC is now available as part no. 885-3042-37. See the New Hug Products page of this issue of REMark for more details on the upgrade disk.

In this installment, I will present an improved version of the ZPC Hardware Support circuit (ZHS) that was originally presented in the April 1986 issue of REMark. I will also give you a brief description of a commercial version of the ZHS circuit, the Scottie Board that you may have seen advertised in REMark, and mention some other commercial boards that are coming.

On the software side, I will tell you about two CAD (computer aided design) programs that can be used under ZPC, Prodesign II and Generic Cadd. I will also present patches for the non-copy protected version of Dbase III Plus.

An Improved ZHS Board

The schematic diagram accompanying this article is of an improved version of the ZHS circuit. This circuit not only corrects problems observed in the original design, but it includes an additional video port that makes it possible to run programs that change video modes or colors by writing to ports. Support for the new port is included in the new ZPC upgrade disk. The board also includes a wait state generator that alleviates the "slow 8259" problem encountered with previous boards, and the ports are decoded properly. The circuit has not actually been built as a complete circuit as of this writing, but the individual sections of it have been tested separately, and I feel confident that it will work. If you build it, be sure to refer to the original ZHS board instructions provided in your ZPC Version 2 manual, and install the required video board jumper wire, if you have not already done it. Note: Pin numbers are not shown on ICs that are part of a package containing multiple gates of the same type. Consult a TTL data book for pin numbering.

Commercial ZHS Boards

A couple of days before writing this, I received one of the first production models of the "Scottie Board", a commercial implementation of the ZHS circuit. This board incorporates an improved ZHS circuit similar to the one I have designed here, and it also includes a battery backed up clock/calendar chip and two IBM-style serial ports. The clock/calendar and serial ports are options. The purpose of the serial ports is to allow software that directly accesses ports, such as mouse drivers and modem programs, to run under ZPC. Software support for these ports has not been written as of this writing, and it is not known how successful we will be at making the ports work for PC programs.

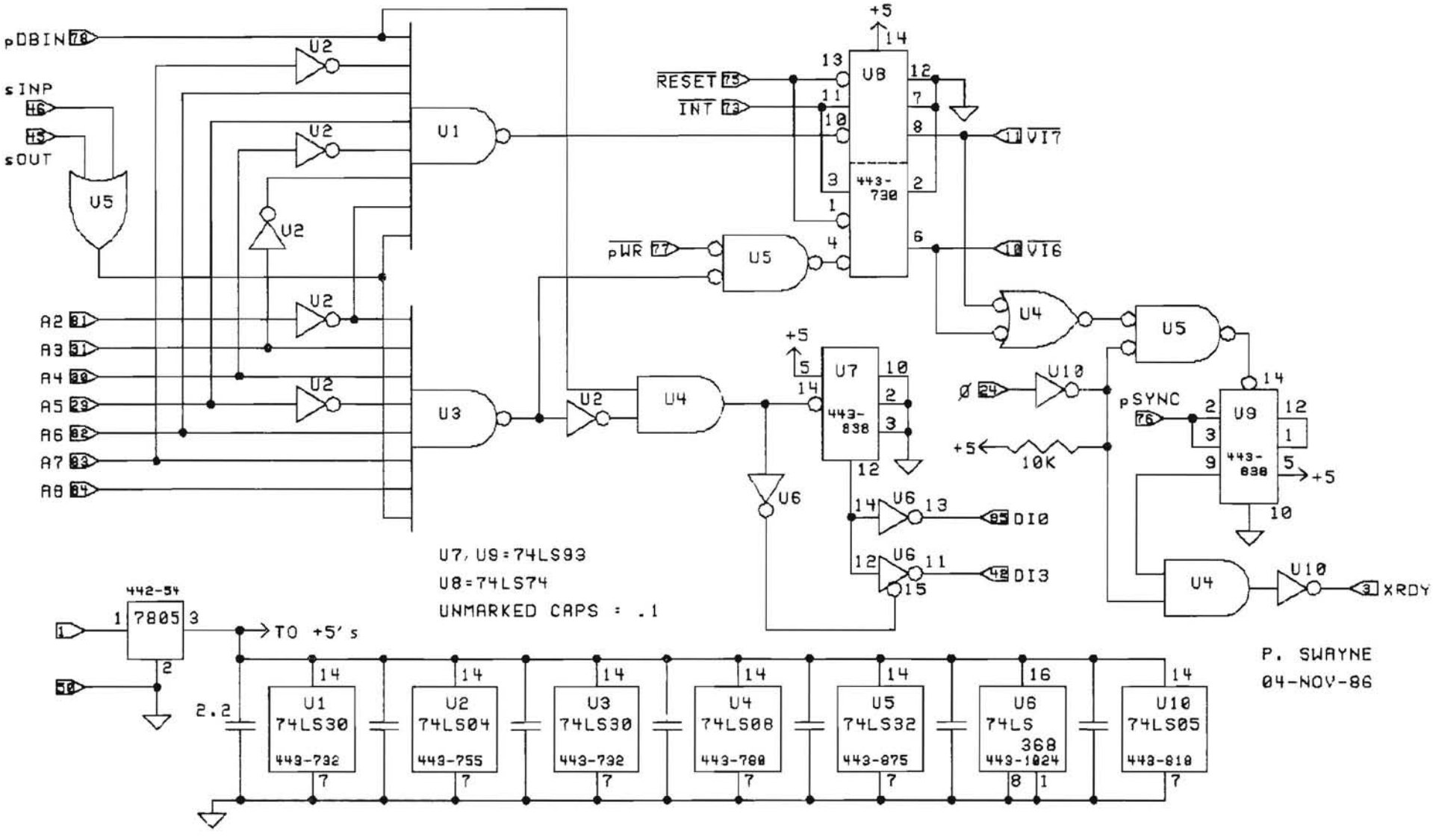
The Scottie Board is well built and works properly. Its designer, Robert Sicotte, said that he would release the design into the public domain. The wait state generator in my new do-it-yourself version is derived from his board.

Other commercial ZHS boards are under development as of this writing. One contains the basic ZHS circuit and one serial port, and the other is the UCI Easy I/O Plus board, which provides two IBM-style serial ports for Easy PC users, with the ZHS circuit added. UCI also plans to put ZPC in an EPROM on their board. I should be getting some of these new boards for review soon.

CAD Programs

I have found that the Prodesign II (version 2.0) CAD program can be used under ZPC without any patches or hardware support. Just

Improved ZHS Circuit



P. SWAYNE
04-NOV-86

"Have Your Computer Talk To Ours!"

"HUGPBBS is on line, 24 hours a day, with over 10 megabytes of free software available for downloading. There's software for every Heath/Zenith operating system, with the majority being for MSDOS, and specifically the Heath/Zenith PC compatible computer systems. Also included is software for HDOS, CP/M, and MSDOS for the H/Z-100 computer system. In addition to this software is a message base through which you can exchange information with other HUG members. Have your computer call (616) 982-3956, 'The Heath Users' Group Personal Bulletin Board System', and make connection at 300, 1200, or 2400 baud. Type a carriage return several times to get my attention. Registration requires that you supply your human's first name, last name, HUG ID number, and some sort of secret password (up to 16 characters). Alternatively, your human can call Jim Buszkiewicz at HUG, and register via voice connection at (616) 982-3837. Call today! All it takes is a computer, modem, and a phone call for your computer to talk to ours!"

configure it for an IBM color graphics card when you set it up. It requires 640k of memory, so be sure that you do not have any other memory resident programs besides ZPC loaded when you run it. The article "Cheap Speed" in this issue contains a drawing done using Prodesign II on a Z-100. It was printed with an Epson MX-80 (with Graftrax Plus) printer. The ZHS circuit in this article was also done with Prodesign II, but on an H-248. I had learned more about Prodesign by the time I did the ZHS drawing, and that is why it looks better (not because of the different machines used).

The Generic CADD (version 2.0) program can also be used under ZPC, but it must be patched first. The patch is needed because the program accesses the graphic character table, something that cannot be fixed with a hardware board. To patch Generic CADD, add the following lines to your PATCHER.DAT file:

```
Generic Cadd V. 2.0 (Logitech)
Insert the disk containing CADD.EXE
CADD.EXE
DABD,B0
DB27,B0
DB64,B0
DBCD,B0
DC1E,0,0
DC86,0,0
DD01,0,0
z
Generic Cadd Dot Plot V  2.0
Insert the disk containing DPLOT.EXE
DPLOT.EXE
5A13,B0
5A7D,B0
5ABA,B0
5B23,B0
5B74,0,0
5BDC,0,0
5C57,0,0
z
```

Then use the PATCHER program to patch CADD.EXE and DPLOT .EXE (the Dot Plot option). These patches were done on the version of Generic CADD provided by Logitech, for use with their mouse. It is not known if the version sold by itself is any different, but if it is, the patch may not work.

Dbase III Plus 1.1

The patch provided with ZPC for Dbase III Plus will not work on the new unprotected version. You can patch the new version by adding the following lines to your PATCHER.DAT file:

```
DBASE III + vers  1.1 (unprotected)
Insert the disk containing DBASE.EXE
DBASE.EXE
2E02,F9,90,90
2E0F,F8,90,90
3099,90,90,90,90,90
30A3,90
34C4,90,90,90,90,90
34CE,90
z
```

Use DISKCOPY to copy your System Disk no. 1 BEFORE you run the ID program, then run ID on the copy, and then apply the patch. ✱

Order Out Of Chaos, Cheaply

Stephen Marschall
4507 John Tyler Ct, #102
Annandale, VA 22003

"What I really need", I said to myself, "is a card catalog like the library. Then, I could find the right albums when I look for them."

"But that's dumb. You don't have the time to build anything like that", I argued back. "You'd never sit still to make out all those cards. Besides, a lot of the albums have several songs on them. How would you do that? More cards? Hah! Fat chance."

I was right, and I knew it. I was just too doggone lazy to set up a filing system that complicated. Still, I did need something. I had too many albums to be flipping through all of them trying to find the one single cut I wanted to hear.

My trusty H-89 gave me the answer. And it was simple. And cheap. And easy.

I just used the capabilities of Magic Wand to create a file containing the information I needed, store the file until I wanted it, and then retrieve that information in a useful way. The system didn't require any special skills, special software, special hardware, or anything other than Magic Wand and an elementary knowledge of how to use it.

Magic Wand (and just about every other text editor) has a search capability so that you can find words or character strings in a bunch of text. That capability can be used to operate a filing system not at all unlike the card catalog of a library. You can look up specific titles, authors, or subjects, in the case of books. Or you can find specific albums, artists, songs, composers, or even your favorite orchestra, if music is cataloged.

Here's how it works.

A card catalog is based on a set of data about a document with the data repeated on a series of cards, but organized differently on each card. At the top of one card is the author's name while others start with the document title or a subject reference. The rest of the data on the card is the same as on all other cards for the same document. By filing these cards in alphabetical order, a person looking for a title, author, or subject only has to look through one set of cards. However, the preparation and maintenance of all the cards needed for a catalog like this is extremely time consuming, not to mention boring.

A system that does almost the same thing, but is much simpler to create and operate can be built with Magic Wand. The system is, in essence, a collection of 'cards', each containing data about a document (i.e., in this case, an album of music), but each 'card' is present only once. The data on the 'card' can be retrieved in almost any imaginable way (not just by "author, subject, or title") by using the search capability of Magic Wand.

The process of setting up and using this system is easy. The steps described below will get you going in the right direction.

Step 1: First of all, decide on what it is you want to catalog. There isn't much point in going through this drill if the things to be cataloged don't really need cataloging. If you have a collection of something (music, in my case) that is too big to be memorized or sifted through easily when you want one specific item, then it deserves cataloging. I have several hundred records, tape reels, and cassettes and that's what I needed help with.

Step 2: Next, decide how you would like to get at the items in your collection. For example, my record albums all have stock numbers printed on them somewhere and usually an album title of some sort ("The Beatles Sing Beethoven"), but I don't often remember these things. On the other hand, I do know the names of the composers and a lot of the pieces they wrote, as well as the types of music in the collection. So what I needed was a way to find an "Oboe Concerto", or a piece by "Bacharach", or "The Stars And Stripes Forever" wherever they happened to be. I also

know the names of the artists who perform these works, such as Isaac STERN or the "Carpenters". Those were the keys I needed for my filing system. You figure out your own.

Step 3: Now comes the fun part. Set up a coding system that will let you mark words according to their function in the filing system. I used this scheme:

C: Denotes a composer's name (last name only).

T: Denotes the title of a specific piece of music.

A: Denotes the name of an artist performing a piece of music.

I: Denotes the instrument upon which a piece of music is played.

M: Denotes the kind of music.

These codes will be used as prefixes for the actual titles, names, or terms that will be in the system 'cards'. Pick codes that are meaningful to you so that you can remember them. I like single letters and colons myself, but if you prefer short phrases like 'AUTH:' or 'TUNE:', use them. The important thing is to use a symbol other than a letter or number as part of the code. The colon is my choice since it is a logical divider.

Why this scheme? Well, one of the features of a text search capability is that the character string you send the software out to find may be buried just about anywhere and that's what the search might turn up: just about anything, anywhere. What we want, though, is a specific thing, not just anything. For instance, if I want a march written by Frederick FENNELL, I don't want to find all the marches played by the band he conducts. Therefore, I would code FENNELL, the composer, like this: C:FENNELL and FENNELL, the artist, like this: A:FENNELL. Now, I can use the text search capability to find only his march. All I have to do is hit the blue key, enter C:FENNELL, hit RETURN and, bingo, I get what I want. Easy, isn't it?

Here's another example. My collection includes a lot of albums by the Carpenters, but not all of the songs were written by Richard Carpenter. Some of them were written by other composers who also sing or play their own music. (There's a lot of cross connection in the pop music field.) By coding all the songs in each album properly, I can find whatever I want without too much trouble. This is how I partially coded two such albums:

Album PV-018 The Singles - 1969-1973 by the Carpenters
A:CARPENTERS
T:CLOSE TO YOU C:BACHARACH C:DAVID
T:WE'VE ONLY JUST BEGUN C:WILLIAMS
T:TOP OF THE WORLD C:CARPENTER

Album PI-030 Burt Bacharach's Greatest Hits
A:BACHARACH
T:CLOSE TO YOU C:BACHARACH C:DAVID
T:ALFIE C:BACHARACH C:DAVID
T:THE LOOK OF LOVE C:BACHARACH C:DAVID

Now, by entering the character string "T:CLOSE TO YOU" as a search key, I can look through my record collection very rapidly and find just what I want. The search would turn up two hits (pun intended) for me to choose from. Then I could check to see what album the desired song is on and retrieve it without too much trouble.

Step 4: Enter the data for the 'cards' in the system. (This part of the system requires the most work, but, believe me, it's worth it.) This is not at all complicated and you can do it at your own pace. Having called up Magic Wand in the EDIT mode with a file name of your own invention (I called mine "ALBUMS01" in anticipation of growth), simply take each album you want to catalog and extract the data you want as you sit at your friendly H-89. Actually, this turned out to be a lot of fun as I got to see albums I had forgotten I even owned!

If you own a lot of albums, you may want to create several files, each one holding the 'cards' for a different kind of music. I eventually did this when my initial file got too big. Another factor that can affect file size is disk capacity. I have dual double-sided, double-density (96TPI) drives so I have some fairly large files on a single disk.

While you enter the data into your file, keep a few things in mind. The search function is good, but it will only find character strings that exactly match the search argument, not any that are just close to what you want. For this reason, be consistent in entering your data. I like to use all caps to prevent any chance of winding up with one entry that says "STREISAND" and another that says "Streisand", making retrieval a little difficult. Also, try to use only the last names of people, such as "SHOSTAKOVICH" or "SINATRA", rather than "P. CLARK" or "F. FENNELL". Unless you are very careful, you could accidentally forget to put in the initial or even put in the wrong initial. In addition, always spell the same name the same way every time. This may seem trivial, but one misspelled name may put an entire album into never-never land as you search in vain for something done by "BERNSTEIN" which has been filed under "BERNSTIEN".

These are samples of the entries that I created for my catalog system.

Album CO-038 Symphonic Dances
A:SLATKIN A:HOLLYWOOD BOWL SYMPH. ORCHESTRA
C:TCHAIKOVSKY T:SLEEPING BEAUTY WALTZ
C:KABALEVSKY T:GALOP
C:GRIEG T:NORWEGIAN DANCE
C:RAVEL T:PAVANE FOR A DEAD PRINCESS
C:GLIERE T:SAILOR'S DANCE
C:BIZET T:FARANDOLE
C:MASSENET T:NAVARRAISE
C:DELIBES T:PIZZICATO POLKA
C:KHACHATURIAN T:SABRE DANCE
C:SAINT-SAENS T:BACCHANALE

Album PI-032 In-A-Gadda-Da-Vida by Iron Butterfly
(Only title song cataloged)
A:IRON BUTTERFLY M:ACID ROCK
T:IN-A-GADDA-DA-VIDA

Album CI-019 (Title same as contents)
A:VAN CLIBURN M:PIANO CONCERTO I:PIANO
C:RACHMANINOFF T:CONCERTO NO. 3

Album PI-010 Have a Ball with Arthur Fiedler - Boston Pops
A:FIEDLER A:BOSTON POPS
T:HEY JUDE C:LENNON C:MCCARTNEY
T:MISTY
T:SPANISH FLEA
T:I LEFT MY HEART IN SAN FRANCISCO
T:CHATTANOOGA CHOO CHOO
T:HIGH NOON
T:CABARET
T:DANCE OF THE SWANS T:SWAN LK C:TCHAIKOVSKY
T:SERENADE C:SCHUBERT
T:HUNGARIAN DANCE NO. 2 C:BRAHMS

A Sound Holiday Offer...

From Gemini Technology

50% OFF

Gemini Sound Boards*

That's right . . . if you've got a **Gemini Emulator Board,** or you know someone who does, we have a special holiday offer for you. Just write us or phone our toll-free number and give the name and full address of the person to whom we should send the new **Gemini Sound Board,** along with your cheque or Visa/MasterCard number and expiry date, and we'll deliver it in time for Christmas. You'll pay only **$14.95** — plus $3.00 for shipping — compared to the regular price of **$29.95.**

Washington State residents please add 8.1% sales tax.

This special Holiday offer
is our way of saying thanks to the nearly 15,000 H/Z-100 users
who have helped us become the recognized leader
in IBM emulation technology.
But don't delay, the offer expires on
December 12, 1986.

BEST WISHES FOR A HAPPY HOLIDAY
AND A PROSPEROUS NEW YEAR!

GEMINI technology

Gemini Technology Corp.
13227 N.E. 20th Street
Bellevue, Washington
USA 98005
1-800-663-8331

*The Gemini Sound Board delivers sound support to Gemini Emulator equipped H/Z-100's operating in IBM mode. H/Z -100 are trademarks of The Heath Company and Zenith Electronics Corporation respectively; IBM is a trademark of International Business Machines Corporation. Offer good while quantities last.

T:FLIGHT OF THE BUMBLEBEE C:RIMSKY-KORSAKOFF
T:ON A CLEAR DAY

Album SH-011 South Pacific Original Broadway Cast
M:SHOW
A:PINZA A:MARTIN
T:SOUTH PACIFIC C:RODGERS C:HAMMERSTEIN

Album B-003 The Drums and Pipes of the 1st Battalion of the Royal Irish Fusiliers
M:MARCH I:BAGPIPES I:DRUMS

Album SV-008 A Treasury of Gregorian Chants boxed 4 record set
M:CHURCH I:CHOIR M:CHANTS

You'll notice that each album has a letters-number code just in front of the title. That's my own filing system arrangement for the records themselves. It is not related to the computer system, except that the file entries refer to the album numbers so that I can find the albums I search for. Any filing system for the actual objects (e.g., records, tapes, books) will work including simple sequential numbers corresponding to the position of an object on a shelf or in a box or some other storage mechanism. However, for the Magic Wand system to be of any value, there must be some way to locate the real thing after you find the reference to it.

Something else you may have noticed is that the format of each entry is different. With the exception of having an album number and title at the top of the 'card', there is no set pattern for the rest of the data. You may include or omit anything you want. The idea is to record the data you will need to retrieve a reference. I purposely did not list all the tunes on the In-a-Gadda-Da-Vida album because I simply don't care about them — only the title song is important to me. Likewise, I didn't list the titles of all the Gregorian chants in the 4 record set because the titles are not critical. However, the kind of music is important, so I made several coded entries reflecting that aspect of the album. Remember, you don't have to list everything. You only have to list what you feel is important.

In the examples above, the artist (A:) code is shown only once for each album since all the pieces on the album are performed by the same artist. In the case of several artists, the artist for each song can be shown after the title. The location of the coded entries doesn't mean anything to the Magic Wand search software. It only has meaning to you as you look at what the search found as the result of your input. So arrange the coded entries any way you want. Two of the entries above, for example, could have been done like this:

Album PV-018 The Singles - 1969-1973 by the Carpenters
T:CLOSE TO YOU A:CARPENTERS C:BACHARACH C:DAVID
T:WE'VE ONLY JUST BEGUN A:CARPENTERS C:WILLIAMS
T:TOP OF THE WORLD A:CARPENTERS C:CARPENTER

Album PI-030 Burt Bacharach's Greatest Hits
T:CLOSE TO YOU A:BACHARACH C:BACHARACH C:DAVID
T:ALFIE A:BACHARACH C:BACHARACH C:DAVID
T:THE LOOK OF LOVE A:BACHARACH C:BACHARACH C:DAVID

With this arrangement, you can search for the title and artist at the same time. Just enter "T:CLOSE TO YOU A:CARP" or "T:CLOSE TO YOU A:BACH" to go straight to your goal. However, I prefer to browse a little even when I'm looking for a specific item. After all, this system is such a time saver, I can afford to slow down some and enjoy the ride.

The header information (i.e., album reference, title), while it can be used as a search target, is intended to tell you about the album that contains the piece you were looking for. You can include anything you want here, like the album stock number, the date you got it, the cost, or even a comment about the album ("GREAT!" or "A little slow for dancing but the violins sound good."). And don't forget to show whether the medium is a record, a cassette, or a reel so you won't waste your time looking in the wrong place. With the advent of video tapes, video disks, and compact disks, this distinction could become even more important.

Right now, my video tape collection is small, so I don't need an index for it. Someday, however, I might reach the point where looking through a stack of VHS cassettes to find my favorite movie ("Casablanca", of course) isn't that much fun anymore. Then I'll set up a file for my video albums just like I have for my music collection. It's easy.

Oh yes. There is one other advantage of the Magic Wand system which I discovered the hard way. I had a box of cards (real 3 × 5 cards) that I had been using to keep track of my record and tape collection. I dropped it one day, inadvertently creating a very undesirable random file. This is always a danger with real card files. However, a Magic Wand file system is never "out of order" no matter how you enter the data. That is, the search operation isn't affected at all by the order of the entries in the file. The only reason you may care to maintain any order at all is if you want to list your album data. Even then, the insertion of record data at any place in the file is easily accomplished with Magic Wand.

Well, there it is. My problem and my solution to it. If you have a similar problem and you also have an H/Z-89 (or something similar) and Magic Wand (or something similar), then you also have the solution right in front of you.

I'm a retired Air Force officer currently working as a systems analyst for Quality Systems Incorporated in Vienna, Virginia. I've been a computer maintenance instructor, a programmer, an intelligence officer, and a personnel officer. I enjoy getting something "extra" out of almost anything, as my article demonstrates. *

HUG Price List

The following HUG Price List contains a list of all products in the HUG Software Catalog. For a detailed abstract of these products, refer to the issue of REMark specified.

Part Number	Description of Product	Selling Price	Vol. Issue
HDOS HARDCOPY SOFTWARE			
885-1008	Volume I Documentation	9.00	
885-1013	Volume II Documentation	12.00	
885-1015	Volume III Documentation	9.00	
885-1037	Volume IV Documentation	12.00	8
885-1058	Volume V Documentation	12.00	
MISCELLANEOUS HDOS COLLECTIONS			
885-1032	Disk V H8/H9	18.00	8
885-1044-[37]	Disk VI H8/H89	18.00	
885-1064-[37]	Disk IX H8/H89 Disk	18.00	
885-1066-[37]	Disk X H8/H89	18.00	10
885-1069	Disk XIII Misc H8/H89	18.00	
885-1135-[37]	HDOS Variety Pkg.	20.00	76
GAMES			
HDOS			
885-1010	Adventure Disk H8/H89	10.00	4
885-1029-[37]	Disk II Games 1 H8/H89	18.00	8
885-1030-[37]	Disk III Games 2 H8/H89	18.00	8
885-1031	Disk IV MUSIC H8 Only	20.00	25
885-1067-[37]	Disk XI H8/H19/H89 Games	18.00	12
885-1068	Disk XII MBASIC Graphic Games	18.00	10
885-1088-[37]	Disk XVII MBASIC Graph. Games	20.00	14
885-1093-[37]	D&D H8/H89 Disk	20.00	16
885-1096-[37]	MBASIC Action Games H8/H89	20.00	18
885-1103	Sea Battle HDOS H19/H8/H89	20.00	20
885-1111-[37]	HDOS MBASIC Games H8/H89	20.00	23
885-1112-[37]	HDOS Graphic Games H8/H89	20.00	23
885-1113-[37]	HDOS Action Games H8/H89	20.00	23
885-1114	H8 Color Raiders & Goop	20.00	23
885-1124	HUGMAN & Movie Animation Pkg	20.00	41
885-1125	MAZEMADNESS	20.00	41
885-1130	Star Battle	20.00	45
885-1133-[37]	HDOS Games Collection I	20.00	59
885-8009-[37]	HDOS & CP/M Galactic Warrior	20.00	32
885-8022	HDOS SHAPES	16.00	45
885-8026	HDOS Space Drop	16.00	49
885-8032-[37]	HDOS Castle	20.00	59
CP/M			
885-1206-[37]	CP/M Games Disk	20.00	11
885-1209-[37]	CP/M MBASIC D&D	20.00	19
885-1211-[37]	CP/M Sea Battle	20.00	20
885-1220-[37]	CP/M Action Games	20.00	32
885-1222-[37]	CP/M Adventure	10.00	35
885-1227-[37]	CP/M Casino Games	20.00	38
885-1228-[37]	CP/M Fast Action Games	20.00	39
885-1236-[37]	CP/M Fun Disk I	20.00	55
885-1248-[37]	CP/M Fun Disk II	35.00	69
ZDOS			
885-3004-37	ZDOS ZBASIC Graphic Games	20.00	37
885-3009-37	ZDOS ZBASIC D&D	20.00	50
885-3011-37	ZDOS ZBASIC Games Disk	20.00	52
885-3017-37	ZDOS Contest Games Disk	25.00	58
885-8042-37	ZDOS/MSDOS Poker Party	20.00	77
UTILITIES			
HDOS			
885-1022-[37]	HUG Editor (ED) Disk H8/H89	20.00	20
885-1025	Runoff Disk H8/H89	35.00	
885-1060-[37]	Disk VII H8/H89	18.00	
885-1061	TMI Load H8 ONLY Disk	18.00	
885-1062-[37]	Disk VIII H8/H89 (2 Disks)	25.00	
885-1063	Floating Point Disk H8/H89	18.00	
885-1065	Fix Point Package H8/H89 Disk	18.00	10
885-1075	HDOS Support Package H8/H89	60.00	
885-1077	TXTCON/BASCON H8/H89	18.00	
885-1079-[37]	HDOS Page Editor	25.00	15
885-1080	EDITX H8/H19/H89 Disk	20.00	
885-1082	Programs for Printers H8/H89	20.00	
885-1083-[37]	Disk XVI Misc H8/H89	20.00	11
885-1089-[37]	Disk XVIII Misc H8/H89	20.00	20
885-1090-[37]	Disk XIX Utilities H8/H89	20.00	22
885-1092-[37]	Relocating Debug Tool H8/H89	30.00	14
885-1098	H8 Color Graphics ASM	20.00	19
885-1099	H8 Color Graphics Tiny PASCAL	20.00	19
885-1105	HDOS Device Drivers H8/H89	20.00	24
885-1116	HDOS Z80 Debugging Tool	20.00	27
885-1119-[37]	BHBASIC Support	20.00	29
885-1120-[37]	HDOS 'WHEW' Utilities	20.00	33
885-1121	HDOS Hard Sec Sup Pkg 2 Disks	30.00	37
885-1123	XMET Robot Cross Assembler	20.00	40
885-1126	HDOS Utilities by PS:	20.00	42
885-1127-[37]	HDOS Soft Sector Support Pkg	30.00	45
885-1128-[37]	HDOS DISKVIEW	16.00	46
885-1129-[37]	HDOS CVT Color Video Terminal	20.00	46
885-8001	SE (Screen Editor)	25.00	28
885-8003	BHTOMB	25.00	28
885-8004	UDUMP	35.00	28
885-8006	HDOS SUBMIT	20.00	31
885-8007	EZITRANS.	30.00	30
885-8015	HDOS TEXTSET Formatter	30.00	42
885-8017	HDOS Programmers Helper	16.00	42
885-8024	HDOS BHBASIC Utilities Disk	16.00	46
CP/M			
885-1210-[37]	CP/M ED (same as 885-1022)	20.00	20
885-1212-[37]	CP/M Utilities H8/H89	20.00	21
885-1213-[37]	CP/M Disk Utilities H8/H89	20.00	22
885-1217-[37]	HUG Disk Duplication Utilities	20.00	26
885-1223-[37]	HRUN HDOS Emulator 3 Disks	40.00	37
885-1225-[37]	CP/M Disk Dump & Edit Utility	30.00	40
885-1226-[37]	CP/M Utilities by PS:	20.00	40
885-1229-[37]	XMET Robot Cross Assembler	20.00	40
885-1230-[37]	CP/M Function Key Mapper	20.00	42
885-1231-[37]	Cross Ref Utilities for MBASIC	20.00	43
885-1232-[37]	CP/M Color Video Terminal	20.00	46
885-1235-37	CP/M COPYDOS	20.00	54
885-1237-[37]	CP/M Utilities	20.00	55
885-1245-37	CP/M-85 KEYMAP	20.00	63
885-1246-[37]	CP/M HUG File Manager & Utilities	20.00	64
885-1247-37	CP/M-85 HUG Bkgrd Print Spooler	20.00	67
885-5001-37	CP/M-86 KEYMAP	20.00	51
885-5002-37	CP/M-86 HUG Editor	20.00	52
885-5003-37	CP/M-86 Utilities by PS:	20.00	54
885-5006-37	CP/M 8080 To 8088 Trans. & HFM	20.00	64
885-5009-37	CP/M-86 HUG Bkgrd Print Spool	20.00	66
885-8018-[37]	CP/M Fast Eddy & Big Eddy	20.00	43
885-8019-[37]	DOCUMAT and DOCULIST	20.00	43
885-8025-37	CP/M-85/86 Fast Eddy	20.00	49
ZDOS/MSDOS			
885-3005-37	ZDOS Etchdump	20.00	39
885-3007-37	ZDOS CP/EMulator	20.00	47
885-3008-37	ZDOS Utilities	20.00	47
885-3010-37	ZDOS Keymap	20.00	51
885-3022-37	ZDOS/MSDOS Useful Programs I	30.00	63
885-3023-37	ZDOS/MSDOS EZPLOT	20.00	63
885-3026-37	MSDOS SMALL C Compiler	25.00	65
885-3031-37	ZDOS/MSDOS Graphics	20.00	69
885-3037-37	MSDOS Z-100 PC Emulator II	60.00	76
885-3039-37	ZDOS/MSDOS HelpScreen	20.00	82
885-8029-37	ZDOS Fast Eddy	20.00	53
885-8035-37	MSDOS DOCUMAT and DOCULIST	20.00	70
885-8041-37	ZDOS/MSDOS Orbits	25.00	75
H/Z100 ZDOS/MSDOS - H/Z150 PC MSDOS			
885-3012-37§§	ZDOS HUG Editor	20.00	52
885-3014-37§§	ZDOS/MSDOS Utilities II	20.00	54
885-3016-37§	ZDOS/MSDOS Adventure	10.00	57
885-3020-37§	MSDOS HUG Menu System	20.00	62
885-3021-37§§	ZDOS/MSDOS Cardcat	20.00	63
885-3024-37§	ZDOS/MSDOS 8080 To 8088 Trans.	20.00	64
885-3025-37§§	ZDOS/MSDOS Misc. Utilities	20.00	64
885-3029-37§§	ZDOS/MSDOS HUG Bg. Print Spool	20.00	66
885-3035-37§§	MSDOS SPELL5 & SPELL5F	20.00	72
885-3038-37§	ZDOS/MSDOS DEBUG Support Util	20.00	77
885-8039-37§§	MSDOS DPATH	20.00	74
885-8040-37§§	MSDOS HELP Programs	20.00	74
885-8045-37§§	MSDOS MATT	20.00	80
885-8046-37§	MSDOS ASM Language Utilities	20.00	82

§ All program files run on both

§§ Program files run partially on both

Part Number	Description of Product	Selling Price	Vol. Issue
PC/IBM COMPATIBLE			
885-6001-37	MSDOS Keymapper	20.00	59
885-6002-37	CP/EMulator II & ZEMulator	20.00	59
885-6003-37	MSDOS EZPLOT	20.00	65
885-6004-37	MSDOS CheapCalc	20.00	67
885-6005-37	MSDOS Skyviews	20.00	67
885-6006-37	MSDOS Cardcat	20.00	69
885-6007-37	MSDOS DND (Dung. & Dragons)	20.00	70
885-6009-37	MSDOS Screen Saver Plus	20.00	76
885-8033-37	MSDOS Fast Edit	20.00	62
885-8037-37	MSDOS Grade	20.00	70
885-8044-37	MSDOS TCSpell	20.00	79
PROGRAMMING LANGUAGES			
HDOS			
885-1038-[37]	Wise on Disk H8/H89	18.00	
885-1042-[37]	PILOT on Disk H8/H89	19.00	
885-1059	FOCAL-8 H8/H89 Disk	25.00	13
885-1078-[37]	HDOS Z80 Assembler	25.00	21
885-1085	PILOT Documentation	9.00	
885-1086-[37]	Tiny HDOS PASCAL H8/H89	20.00	13
885-1094	HDOS Fig-Forth H8/H89	40.00	18
885-1132-[37]	HDOS Tiny BASIC Compiler	25.00	59
885-1134	HDOS SMALL-C Compiler	30.00	63
CP/M			
885-1208-[37]	CP/M Fig-Forth H8/H89 2 Disks	40.00	18
885-1215-[37]	CP/M BASIC-E	20.00	26
BUSINESS, FINANCE AND EDUCATION			
HDOS			
885-1047	Stocks H8/H89 Disk	18.00	
885-1048	Personal Account H8/H89 Disk	18.00	
885-1049	Income Tax Records H8/H89 Disk	18.00	
885-1055-[37]	MBASIC Inventory Disk H8/H89	30.00	
885-1056	MBASIC Mail List	30.00	
885-1070	Disk XIV Home Fin H8/H89	18.00	
885-1071-[37]	MBASIC SmBusPk H8/H19/H89	75.00	17
885-1091-[37]	Grade/Score Keeping H8/H89	30.00	14
885-1097-[37]	MBASIC Quiz Disk H8/H89	20.00	18
885-1118-[37]	MBASIC Payroll	60.00	30
885-1131-[37]	HDOS CheapCalc	20.00	47
885-8010	HDOS Checkoff	25.00	32
885-8021	HDOS Student's Statistics Pkg	20.00	44
885-8027	HDOS SciCalc	20.00	50
CP/M			
885-1218-[37]	CP/M MBASIC Payroll	60.00	31
885-1233-[37]	CP/M CheapCalc	20.00	47
885-1239-[37]	Spread Sht. Contest Disk I	20.00	
885-1240-[37]	Spread Sht. Contest Disk II	20.00	
885-1241-[37]	Spread Sht. Contest Disk III	20.00	
885-1242-[37]	Spread Sht. Contest Disk IV	20.00	
885-1243-[37]	Spread Sht. Contest Disk V	20.00	
885-1244-[37]	Spread Sht. Contest Disk VI	20.00	
885-8011-[37]	CP/M Checkoff	25.00	32
885-8036-[37]	CP/M Grade	20.00	70
ZDOS/MSDOS H/Z100 ONLY			
885-3006-37	ZDOS CheapCalc	20.00	47
885-3013-37	ZDOS Checkbook Manager	20.00	54
885-3018-37	ZDOS Contest Spreadsheet Disk	25.00	58
885-8028-37	ZDOS SciCalc	20.00	50
885-8030-37	ZDOS Mathflash	20.00	55
885-8043-37	MSDOS Calc	20.00	80

Continued on Page 83

HUG NEW PRODUCTS

HUG P/N 885-3040-37

HADES $40.00

Introduction: HADES is an acronym for Hug's Absolute Disk Editing System. In short, HADES is a screen oriented byte (or disk) editor with file recovery and attribute modifying capabilities.

Requirements: HADES requires MS-DOS version 2.0 or greater on any Heath/Zenith 16 bit computer with at least 128k of RAM. These systems include: H/Z-100, 110, 120, 130, 140, 150, 160, 170, 180, and 200 series of computers. HADES will also work with the IBM PC, XT, and AT.

The following files are included on the HUG 885-3040-37 HADES disk:

README	.DOC	- Preliminary information and disclaimer.
DHADES	.EXE	- Demo / Non-Destructive version of HADES.
HADES	.EXE	- Full implementation version.

Also included with this software, is an extensive users manual.

Author: Jim Buszkiewicz (Heath Users' Group)

Program Content: HADES is a program that gives the MS-DOS user, almost total control over the data residing on his disk. In addition, file attributes can be displayed and modified (within the capabilities of MS-DOS). Finally, the software has the ability to recover any file or sub-directory that's been accidentally erased (providing no other disk writes have occurred since the erasure).

When HADES first signs on, the main menu is displayed, and looks like the following:

```
F1 - Help
F2 - Directory Mode
F3 - Sector Mode
F4 - File Mode
F5 - Un-Erase Mode
F6 - Change Drives
F7 - Color/Monochrome
F9 - Exit To MS-DOS
```

Online help is always available, and can be called up by pressing the F1 function key.

The F2 enters the Directory Mode. From this mode, the user can view all the files (one page at a time), on his currently logged directory, along with each files' attributes. These include: <R>ead Only, <H>idden, <S>ystem, <A>rchive, and <D>irectory. Each file can have its attributes changed or removed including hidden sub-directories! The user can also change directories from this mode.

The F3 key enters the Sector Mode. This mode gives the user a 128 byte window into each sector on his disk. This window can be

TABLE C

Product Rating

10 - Very Good
9 - Good
8 - Average

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

7 - Has hardware limitations (memory, disk storage, etc.)
6 - Requires special programming technique
5 - Requires additional or special hardware
4 - Requires a printer
3 - Uses the Special Function Keys (f1,f2,f3,etc.)
2 - Program runs in *Real Time
1 - Single-keystroke input
0 - Uses the H19 (H/Z89) escape codes (graphics, reverse video)

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

ORDERING INFORMATION

For Visa and MasterCard phone orders; telephone Heath Company Parts Department at (616) 982-3571. Have the part number(s), descriptions, and quantity ready for quick processing. By mail; send order, plus 10% postage and handling ($1.00 minimum charge, up to a maximum of $5.00. UPS is $1.75 minimum -- no maximum on UPS. UPS Blue Label is $4.00 minimum.), to Heath Company Parts Department, Hilltop Road, St. Joseph, MI 49085. Visa and MasterCard require minimum $10.00 order.

Any questions or problems regarding HUG software or REMark magazine should be directed to HUG at (616) 982-3463. REMEMBER-Heath Company Parts Department is NOT capable of answering questions regarding software or REMark.

NOTE

The [-37] means the product is available in hard-sector or soft-sector. Remember, when ordering the soft-sectored format, you must include the "-37" after the part number; e.g. 885-1223-37.

Note: All special update offers announced in REMark (i.e. ZPC II update) must be paid by check or money order, payable to the Heath Users' Group. **NO CREDIT CARDS ACCEPTED.** ZPC II contains only one disk. It is a combination of ZPC I and the ZPC Support disk plus added improvements. Thank you.

moved anywhere within the sector, as well as to any sector or cluster. Any byte in any DOS accessible sector can be viewed, modified, and then permanently recorded on disk! Data can be entered in hex or ascii. The entire disk can be searched for strings of hex or ascii data. Once found, the same string can be searched for again from the current disk position. Finally, the currently viewed sector can be displayed on any listing device. In this sector mode, the keypad keys act as your 'steering wheel' to all the data on your disk. On the H/Z-100 (Not PC), not only are the cursor and HOME keys active, but the rest of the keypad acts in the same manner as the keypad on the PC series of computers, allowing both systems to be operated in an identical fashion.

The F4 key enters the file mode of operation. The name of an existing file must be entered before HADES allows this mode to continue. Once found (in the currently logged directory), each cluster of the file is displayed, in the same manner individual sectors are. Any byte, in any of the file's clusters, can be viewed, modified, and then permanently recorded back on disk. HADES always displays the current cluster, along with the next eight clusters (if the file is that long) of the file. As the user moves through each cluster, the cluster list is updated in ticker-tape fashion. No other sectors or clusters can be viewed or modified except those contained in the original requested file. Like in the Sector Mode, you can search for strings of either hex or ascii data. Only each cluster in the file itself will be searched for this data.

The F5 key allows the user to recover a file or sub-directory that's been accidentally erased. The file recovery is done automatically, and two important conditions must be met before a successful recovery can be made.

1. The file being recovered must be the LAST file to have been erased on the disk.

2. After the actual file erasure, no disk writes should have occurred.

The file recovery method used, is called a 'blind search'. HADES uses the first cluster pointer in the directory entry into the FAT. From there, free clusters are allocated sequentially to the lost file until the file size is satisfied.

The F6 key allows the user to log into a different drive. This function allows HADES to record all the necessary parameters of the drive being used.

The F7 key toggles between a white on black monochrome display and a white on blue with gray border color display.

The F9 key returns you safely to the MS-DOS operating system. This is the only SAFE way to exit other that a cold boot.

Comments: The abilities of HADES have been tested on virtually every Zenith 16 bit computer and drive combination including the H/Z-100 with 8" single density single-sided drives, as well as the new Z-181 with new 3" drives. HADES has yet to fail!! In the hands of an experienced MS-DOS user, HADES can be a very powerful tool. To the beginner, HADES is a window to the secrets of disk formats and file structures. Even if you never use HADES for anything other than to recover a single file, the day you see that file magically return to the world of the living, you'll say "..thank Heaven for HADES"!

TABLE C Rating: 10

HUG P/N 885-3041-37
SCREENDUMP $20.00

Introduction: SCDMP is a utility that allows reproduction of a complete video screen on a dot matrix printer, including both text and graphics, without having to exit the current program. The SCDMP program may be loaded manually (by entering SCDMP <cr>) or automatically, via 'autoexec.bat', into memory at the beginning of a session where it remains resident until needed. This version of SCDMP is a update to and supersedes the version released on HUG P/N 885-3022-37, ZDOS/MSDOS Useful Programs I. Versions for both the Z-100 and Z-150 (IBM PC and most compatibles) are included. Additional features have been added to the programs which provide greater screen dump flexibility. A summary of the new features are as follows:

For Z-100

- Dumps All VRAM banks (everything on screen regardless of color), as well as choice of red, green, or blue.
- Dumps TEXT only (similar to PSC.COM).

For Z-150

- Dumps GRAPHICS (must be in graphics mode) or TEXT (even if in graphics mode).
- Version for ZPC on the Z-100 (all Z-150 features supported) or both.
- Dump either a positive or negative image of screen.
- Higher density dump supported on all printers.
- Dump either a 24 or 25 line screen.
- Change default settings by the command line on initial entry or via SHIFT-F12/SHIFT-PrtSc.
- Print out current default settings any time desired.
- More printers supported (C. Itoh, Epson, Gemini, IBM, MPI, NEC, Anadex, and IDS) (Versions in preparation for the Epson, Toshiba, and other 24 pin printers).
- Sample routines to call from within your own program. See the documentation file (SCDMP.DOC) for complete description of features and instructions for use.

Requirements: This program requires MSDOS operating system (Version 1.1 or higher) on an H/Z-100 or H/Z-150 computer. The printers currently supported are the C. Itoh 8510/1550 Series with 2K buffer, NEC 8023A, Epson MX/RX/FX Series with Graftrax, Star Micronics Gemini Series, Okidata Microline Series, Zenith/MPI 99/150 Series, Anadex Silent Scribe Series, IBM Proprinter, and IDS Paper Tiger printers.

The following files are included on the HUG P/N 3041-37 SCREENDUMP disk:

README	.DOC	Initial startup info and disclaimer
SCDMP	.DOC	Screen Dump information
SCDMPCIT	.100	Z-100 Version for C. Itoh 8510/1550
SCDMPCIT	.150	Z-150 Version for C. Itoh 8510/1550
SCDMPCIT	.ZPC	Z-100 ZPC Version for C. Itoh 8510/1550

SCDMPNEC	.100	Z-100 Version for NEC 8023A
SCDMPNEC	.150	Z-150 Version for NEC 8023A
SCDMPNEC	.ZPC	Z-100 ZPC Version for NEC 8023A
SCDMPCIT	.ASM	Source for C. Itoh 8510/1550 and NEC 8023A
SCDMPEPS	.100	Z-100 Version for Epson MX/RX/FX
SCDMPEPS	.150	Z-150 Version for Epson MX/RX/FX
SCDMPEPS	.ZPC	Z-100 ZPC Version for Epson MX/RX/FX
SCDMPIBM	.100	Z-100 Version for IBM Proprinter
SCDMPIBM	.150	Z-150 Version for IBM Proprinter
SCDMPIBM	.ZPC	Z-100 ZPC Version for IBM Proprinter
SCDMPEPS	.ASM	Source for Epson MX/RX/FX and IBM Proprinter
SCDMPGEM	.100	Z-100 Version for Star Micronics Gemini
SCDMPGEM	.150	Z-150 Version for Star Micronics Gemini
SCDMPGEM	.ZPC	Z-100 ZPC Version for Star Micronics Gemini
SCDMPGMX	.100	Z-100 Version for Star Micronics Gemini 10X
SCDMPGMX	.150	Z-150 Version for Star Micronics Gemini 10X
SCDMPGMX	.ZPC	Z-100 ZPC Version for Star Micronics Gemini 10X
SCDMPGEM	.ASM	Source for Star Micronics Gemini Series
SCDMPOKI	.100	Z-100 Version for Okidata Microline
SCDMPOKI	.150	Z-150 Version for Okidata Microline
SCDMPOKI	.ZPC	Z-100 ZPC Version for Okidata Microline
SCDMPOKI	.ASM	Source for Okidata Microline
SCDMPMPI	.100	Z-100 Version for Zenith/MPI 99/150
SCDMPMPI	.150	Z-150 Version for Zenith/MPI 99/150
SCDMPMPI	.ZPC	Z-100 ZPC Version for Zenith/MPI 99/150
SCDMPMPI	.ASM	Source for Zenith/MPI 99/150
SCDMPANA	.100	Z-100 Version for Anadex Silent Scribe
SCDMPANA	.150	Z-150 Version for Anadex Silent Scribe
SCDMPANA	.ZPC	Z-100 ZPC Version for Anadex Silent Scribe
SCDMPANA	.ASM	Source for Anadex Silent Scribe
SCDMPIDS	.100	Z-100 Version for IDS Paper Tiger
SCDMPIDS	.150	Z-150 Version for IDS Paper Tiger
SCDMPIDS	.ZPC	Z-100 ZPC Version for IDS Paper Tiger
SCDMPIDS	.ASM	Source for IDS Paper Tiger
SCDMP-RL	.SKL	Skeleton versions for use in building a program
SCDMP-LR	.SKL	for your unique printer or application
SCDMP	.BAT	Batch file for auto load of SCDMP .COM (rename the appropriate program to SCDMP.COM)
INT5	.BAS	Sample BASIC program to call int-5
INT5	.ASM	Assembly subroutine for above
INT5	.BIN	Assembled subroutine for call from BASIC
CINT5	.BAS	Sample BASIC program for Compiler to call int-5
CINT5	.ASM	Assembly subroutine for above
CINT5	.OBJ	Object file for linking with Compiled BASIC

Program Author: Leslie L. Bordelon

Operation: For the Z-100, the program allows a choice of which color bank of video RAM is dumped (if the user has all banks of color RAM in his Z-100). The number of VRAM banks installed in your computer is determined by SCDMP upon initial load. For the color version, entering a <B> for blue, <R> for red, <G> for green, or <A> for all banks immediately after the <SHIFT-F12> will select the VRAM bank to be printed. If no character is entered, <A>ll banks is the default. If only one bank of color RAM is installed, only the green bank can be dumped. Entering a <T> will cause only the text on the screen to be dumped unformatted to the printer.

For the Z-150, you can option to print only text, even if in the graphics mode, by entering a <T> for text immediately after the <SHIFT-PrtSc>. However, if in the Z-150 text mode (modes 0, 1, 2, or 3) then only screen text can be dumped (using the ROM text dump), even if SCDMP is set for graphics dump.

The program allows the printing of the screen image as either a positive or negative. Entering a <P> for positive or <N> for negative after the <SHIFT-F12> or <SHIFT-PrtSc> will change the image printed. When set to negative image, text on a black background is printed as black on white. This is the default setting for SCDMP. A positive image would appear as white text on a black background. This provides some additional flexibility when printing complicated artwork using various shades of colors.

The program also allows multiple density printing. This is accomplished by using a higher dot density in graphics mode if the printer supports such. For those printers that do not, high density is achieved by overprinting each line a second time. Entering an <H> immediately after the <SHIFT-F12> for Z-100 or <SHIFT-PrtSc> for Z-150 or color selection would cause the printer to use a higher density mode for printing. The default density is normal or standard density.

The program also allows a choice of printing the twenty-fifth line (only if displayed for the Z-100). Entering a <D> immediately after the <SHIFT-F12> for Z-100 or <SHIFT-PrtSc> for Z-150 or color or density selection will cause SCDMP to ignore the 25th line even if it is displayed.

Two skeleton versions of the program are also provided as an aid in building a program for a non-supported printer. The skeleton versions present an outline for both options of scanning the Z-100/Z-150 screen depending on your specific type of printer. With this data and a little trial and error on the user's part, a successful product for a unique printer can be developed.

Comments: none

TABLE C Rating: (10)

HUG P/N 885-3042-37

ZPC Upgrade Disk $20.00

Introduction: This disk upgrades ZPC (885-3037-37) version 2.0.x to version 2.1.x. ZPC is a program that allows H/Z-100 (dual

processor) computers to run many IBM-PC compatible programs. This upgrade allows you to run more programs, and provides support for hardware enhancements (such as the "Scottie Board") to support even more PC programs.

Requirements: To use this disk, you need ZPC Version 2 (885-3037-37), an H/Z-100 computer with 768k of memory, and MS-DOS version 2 or 3.

The following files are included on the disk:

README	.DOC	ZPCUP	.DOC
ZPCUP	.COM	INT14	.COM
MASK	.COM	PATCHER	.COM
PATCHER	.DAT	SETZPC	.COM
FIXLTS	.COM	FIXSYM	.COM
BLINK	.COM	DOS	.ACM
KEY	.ACM	ZPCASM	.TXT
INT14	.ASM	MASK	.ASM
PATCHER	.ASM	SETZPC	.ASM
FIXLTS	.ASM	BLINK	.ASM

Here is an explanation of some of the files on the disk:

ZPCUP.COM — This program automatically patches and upgrades your ZPC.COM file. The improvements made include the following:

Improved Keyboard Handling. The codes returned by ZPC as you type keys more closely resemble PC codes, so that difficult programs such as QuickBASIC version 2 can be made to work, that would not work previously.

Support For More Hardware. The modified ZPC supports new hardware that allows programs which change video modes or colors by writing to ports to work properly.

Improved Control-C and Control-Break Handling. Handling of these codes more closely approximates the real PC world.

INT14.COM — This program provides support for a real IBM-style serial port, such as is provided as an option on the "Scottie Board".

MASK.COM — ZPC unmasks some interrupt lines on the S-100 bus when it is loaded. This program allows you to mask off those lines, in case you experience any difficulty with software that otherwise runs OK with ZPC not loaded.

PATCHER.COM, PATCHER.DAT — A new PATCHER.COM is provided because some early copies of ZPC Version 2 contained a flawed PATCHER.COM. The new data file, PATCHER.DAT, contains patches for more programs.

SETZPC.COM — A new SETZPC is included because some early copies contained bugs.

FIXLTS.COM, FIXSYM.COM — These programs patch LOTUS 1-2-3 release 2 and Symphony for use under ZPC.

BLINK.COM — This program allows you to have a non-blinking cursor while ZPC is in the PC mode.

DOS.ACM, KEY.ACM — These modules are part of the ZPC assembly source code that have been upgraded.

ZPCASM.TXT — This file lists changes that must be made to your ZPC.ASM file in case you want to reassemble the upgraded version of ZPC.

*

X-10® POWERHOUSE™

NUMBER ONE IN HOME CONTROL

Use the latest electronic technology to control lights and appliances throughout your home. It's now possible with our radio home control system.

Transmitter
Works from up to 150ft. away, even from outside your home!

Experience remote control of your home via radio control

Imagine turning on the lights in any room of your house from your bed or from your car as you enter your driveway. With the BC-5000 Radio Control System it's easy to put all your lights and electric appliances at your beck and call.

This dynamic system includes a Radio transmitter (BC-504) for transmitting radio frequency signals, and a Radio Transceiver/Appliance Module for receiving the signals to turn electrical devices on or off. Because it is a "Transceiver" it re-transmits, to standard X-10 Modules, over your house wiring, the signals it receives from the Radio Transmitter. **This lets you control up to 8 X-10 modules (sold separately) from the Radio Transmitter.**

Plus you can have control from any other X-10 controller, such as our Timer, Maxi Controller, or Home Control Interface.

BC-5000 Radio Remote Control system consisting of one Radio Transmitter and one Radio Transceiver/Appliance Module

Just $49.99 Complete

Enjoy Computer-Programmed Home Control

Now you can use the power of your computer to control all the lights and appliances in your home! **All without tying up your computer. The BC-290 Home Control Interface** is an RS-232 Controller which you program from your computer and *then disconnect,* freeing up your computer for other uses. Your interface sends signals over your house wiring to control lights and appliances plugged into the X-10 modules.

• Interface with menu-driven software for the IBM PC and compatibles. **BC-290P *Was* $59.99 *Now Only* $49.99** ***• Interface with icon-based software for the Apple Macintosh*** **BC-290M *Was* $69.99 *Now Only* $59.99** ***• Interface with color graphics software for the Apple IIe and IIc.*** **BC-290A *Was* $59.99 *Now Only* $49.99**

SPECIAL BUYS!

- **Purchase one BC-5000 RF Radio Remote Control System plus any 2 modules* and receive one Lamp Module FREE...A $14.99 Value! *You Pay Only $79.97!***
- **Purchase one computer interface, your choice, plus any 2 modules* and receive one Lamp Module FREE...A $14.99 Value! *You Pay Only $89.97* (Apple® IIe/IIc and IBM) or $99.97 (Macintosh™).**
- **Purchase one Maxi Controller plus any 2 modules* and receive one Lamp Module FREE...A $14.99 Value! *You Pay Only $54.97!***

***Choice of either Lamp Module, Wall Switch, or 3-PIN Appliance Module.**

Prices good through December 31, 1986

Available at your nearest Heath/Zenith Computers & Electronics Store

See pages 28 & 29 of our current Heath/Zenith catalog for more details on our exciting line of X-10 POWERHOUSE products.

MegaBASIC:

Everything You've Always Needed In A BASIC . . .

Richard A. Tilden
Software Toolsmith
10 Thurston Street
Somerville, MA 02145

About The Author

Richard A. Tilden, Software Toolsmith, is an independent software consultant who learned to program on the IBM 704 serial #1. The MegaBASIC/Z-100 utilities mentioned in the article are available free of charge with a formatted disk in a prepaid mailer, or for $5 without the disk, to 10 Thurston Street, Somerville, MA 02145.

A fast, capable interpreter for a rich BASIC dialect providing a comprehensive development environment, not cheaply.

After many years' struggle with the approachable, but limited, environment of ZBASIC (latterly GW-BASIC) and Bascom for program development on my Z-100, I have discovered an alternative. With a language providing most capabilities whose absence I had been lamenting, and a development environment including tools it had never before been my pleasure to discover imperative, MegaBASIC provides access to the potential of the Z-100 in ways previously thinkable only in C.

Published by American Planning Corporation (APC), MegaBASIC has apparently been around awhile (it is currently at version 5.22), but I had not been aware it would run on a Z-100 — I get brainwashed by "runs on IBM PC and close compatibles", and silence when I send postcards asking "Does It Run On A Zenith Z-100?" Indeed, MegaBASIC will apparently run on damn near everything: IBM, Compatibles, North Star, Seiko, HP, Victor, Televideo, 16-bit S100, MS-DOS generic and many more, running MS-DOS 2.0+, MPM-86, CPM-86, CCPM-86, TurboDOS; even over several networks. Programs are transportable, as is, across the range.

Addressed here, primarily, is the Z-100, running MS-DOS version 2.21. The first of the pleasant surprises is that MegaBASIC's universality is not accomplished by compromise, nor has age resulted in liver spots. Many generic products are flawed by the restrictions imposed by compatibility and history: not so here. More on this later.

This article will present some of the features of MegaBASIC of which I have become aware — there are many unplumbed depths — and from the point of view of a Z-100 user specifically: most of the material will be applicable across the Zenith line and beyond. The discussion will frequently contrast MegaBASIC with GW-BASIC; they'll be known as Mega and GW for short. GW features absent from Bascom, such as windows, will be ignored; thus references to GW may be taken to include Bascom, where context implies.

A Simple Benchmark

To begin with the bottom line, Mega is fast compared to GW interpreted, 2.5 — 5 times faster. It is, of course, not a compiler, and GW compiled is something like 5 times faster than it. And GW compiled is not the maximal use of the machine: probably good C code is again an order of magnitude faster. (Indeed, Bascom isn't even GW compiled, apparently we are not to be vouchsafed the GW-BASIC compiler.)

The comparison was deliberately made with the programs as identical as possible, and hence is unfair to Mega. As will be seen, the Mega language is far richer than GW's, and many things requiring circumlocution there can be expressed concisely in Mega. Thus for some applications, parts which must run interpreted in GW will essentially run compiled in Mega: file-handling and character manipulation programs may be expected to be even closer to the speed of Bascom.

Table 1 shows the results of running the benchmark listed in Listings 1 and 2. Listing 1, the Mega version, was generated by Mega, including the cross reference, which is built-in. Since the GW LLIST command fails for other than simple statements, Listing 2, the GW version, was produced with my program Lx. As in EPA mileage figures, the numbers should be examined only with respect to one another.

The benchmark is a simple sort program which reads a file of random 15 digit records into a Heap and writes a second file dequeuing, and hence, sorting the records. The two subroutines are copied from Jon Bentley's excellent Programming Pearls column, which runs in Communications of the ACM. This one appeared in March 1985. Written originally in GW, the program was altered for Mega only where incompatibility required; since a

direct comparison was desired, no improvements made possible by the Mega language were exploited. (Not that, in this case, any significant ones occur to me.) The Z-100 is an 8 Mhz, 768K one with an SDC external hard disk.

Two files, of 1000 and 2000 records were generated by the Mega program shown in Listing 3 and used in all runs. Five variants are shown, three run with DEMO.BAS, the GW version, and two with DEMO.PGM, the Mega version. The latter two require explanations: Mega is sold in two versions.

One's initial $380 buys what APC calls the Development version. This provides the full capabilities described here for program development, debugging, and execution. Programs so developed are usable by anyone else with a Mega Development version and may be distributed to such people. You promise in the Licensing Agreement, of course, that you won't distribute Mega itself. (Which, incidentally, is not copy protected. Can you copy protect programs on the Z-100?)

Additionally available is the Runtime version. This is not a compiler, it is merely the Development version stripped of its development features, which buys an increase in interpreter speed. For an additional $250, one gets this version, which allows independent execution of Mega code. These standalone Mega programs may be distributed to anyone, and do not require that the second party own Mega.

So your second parcel of dollars buys you some improvement in speed of debugged code and the ability to sell Mega programs to non-Mega owners if you wish. Benchmark times for the Runtime version are identified as Mega(R).

Listing 1
'DEMO.PGM', MegaBasic Version of Benchmark

```
100 Rem Benchmark program for ZBasic et al   Sort a file
110 Def integer "A-Z"; Dim LIN$(2500,20)
120 Def real func SEC(T$)=val(T$(7:2))+60*(val(T$(4:2))
    +60*val(T$(1:2)))
130 Print 'Mega  Avail:',free(1),
140 Data 'data-1k','data-2k',''; Read F$:
    If F$='' then [Print, Stop]
150 Rem                  ------Begin timed area-----
160 T1$=time$; Open input #3,F$; Openc output#4,F$+".out";
    N=1; Input#3,LIN$(1); Rem          Get first record
170 While input(3),
      N=N+1, Input#3,LIN$(N),
      Gosub 250; Next; Rem    Enqueue remaining records.
180 CT=N; For J=1 to CT;
      Print#4,LIN$(1); Swap LIN$(N),LIN$(1); N=N-1;
      Gosub 290, Next, Rem  Pull from stack in sequence
190 Close; T2$=time$
200 Rem                  -----End of timed area-----
210 Open input #3,F$+'.out'; LIN$(0)=""; N=1;
    While input(3), Input#3,LIN$(1), N=N+1,
      If LIN$(1)<LIN$(0) then Goto 230
      else LIN$(0)=LIN$(1);
      Next
220 Print ", ",F$,":",CT," recs,",SEC(T2$)-SEC(T1$)," secs",;
    Close, Goto 140
230 Print "Error sorting ",F$,", record",N;
    Close; Goto 140
240 Rem
     Procedure SiftUp from Bentley's Pearls,
     CACM March 1985. New element at LIN$(N),
     first element at LIN$(1)
250 I=N; While I>1
260   P=I div 2; If LIN$(P)<=LIN$(I) then Return
270   Swap LIN$(I),LIN$(P), I=P;
      Next; Return
280 Rem
      Procedure SiftDown from ibid  New element at LIN$(1);
      N elements
290 I=1; C=2, While C<=N,
      D=C+1; If D<=N then If LIN$(D)<LIN$(C) then C=D
300   If LIN$(I)<=LIN$(C) then Return
310   Swap LIN$(I),LIN$(C); I=C, C=C+C;
      Next; Return

***** Real Functions in DEMO.pgm *****

SEC             120*   220

***** Integer Variables in DEMO.pgm *****

C               290    300    310
CT              180    220
D               290
I               250    260    270    290    300    310

J               180
N               160    170    180    210    230    250
                290
P               260    270

***** String Variables in DEMO.pgm *****

F$              140    160    210    220    230

LIN$            110*   160    170    180    210    260
                270    290    300    310

T$              120*
T1$             160    220
T2$             190    220

***** Line Number References in DEMO.pgm *****

140             220    230
230             210
250             170
290             180
```

Object Time Capabilities

GW users may look on Table 1 and weep at some entries in the Free Memory column. Yes, we have here a universal-type program that doesn't make MS-DOS look like CPM. 65k is a thing of the past.

Mega has full access to whatever memory is available on the machine, full access to MS-DOS features, including pathnames, directory hierarchy, environment, command line tail, etc. DOS is accessible from within Mega as a subshell. Proper filters may be written using I/O redirection.

Memory is accessed through a particularly nice extension to Peek/Poke: memory may be read and written as a file, with or without conversions. Further access to hardware is provided with assembly language calls, interrupt calls, I/O to ports, and — get this — accessibility of Mega programs via interrupts.

What Mega lacks, and understandably, is built-in support for Z-100 graphics. Should enough Z-100 use of Mega develop, and should the Z-100 live so long, someone will no doubt provide: the hooks are there. A generic graphics interface, GSX, is supported by APC, but not by Zenith. That's what that someone should look at.

Character-oriented graphics specific to the Z-100 are also not provided, but are easily added. I have developed a package of routines implementing all character-oriented facilities from GW not present in Mega (Color, Locate, Csrlin, etc.) along with some other utilities, and will make it available free of charge to anyone interested. See the end of the article for instructions.

Mega programs operate with 32-bit integers and BCD arithmetic. BCD precision is variable from 8 to 18 digits. A selection of precision is made in advance of execution and stuck to, though precision in files can vary dynamically. APC distributes two precision levels, 8 and 14, and will supply, I understand, others that might be required. The Mega environment initially loaded selects the precision for the duration of the session. Understandably, the 8087 is not supported.

Strings and string arrays are allocated statically (a maximum size must be declared), eliminating GW's time-consuming and uncontrollable garbage collection. Should the pre-declaration seem burdensome, note that arrays may be re-declared dynamically at run time.

Limitations on sizes of things are those one might expect in a full-memory environment: few. Length of character strings, size of arrays, etc., are limits imposed by hardware and available memory, not by Mega. Except, sigh, for good ol' 255 character statements.

No, I got most everything I wanted in the language except removal of this artificial limit, frustrating only when one attempts to write structured code using Ifs, or long string constants.

Listing 2
'DEMO.BSA', GW-Basic Version of Benchmark

```
List/Xref version 1.0                File. demo.bsa Page  1
Program listing                      04-30-1986 at 08:08:39

100 ' Benchmark program for ZBasic et al   Sort a file
110 DEFINT A-Z:DIM LIN$(2500)
    DEF FNSEC!(T$)=VAL(RIGHT$(T$,2))+60*(VAL(MID$(T$,4,2))
    +60*VAL(LEFT$(T$,2)))
120 PRINT "Basic  Avail:";FRE("");
130 DATA "data-1k","data-2k","":READ F$
    IF F$="" THEN PRINT:STOP
140 '                  -----Begin timed area------
150 T1$=TIME$:OPEN "I",#3,F$:OPEN "O",#4,F$+".out":
    LINE INPUT#3,LIN$(1):N=1'          Get first record
160 WHILE NOT EOF(3)
      N=N+1.LINE INPUT#3,LIN$(N).
      GOSUB 250:WEND'          Enqueue remaining records
170 CT=N:FOR J=1 TO CT:
      PRINT#4,LIN$(1):SWAP LIN$(N),LIN$(1):N=N-1
      GOSUB 290:NEXT'        Pull from stack in sequence.
180 CLOSE:T2$=TIME$
190 '
                        -----End of timed area.-----
200 OPEN "i",#3,F$+" out":LIN$(0)="":N=1
    WHILE NOT EOF(3):LINE INPUT#3,LIN$(1):N=N+1:
      IF LIN$(1)<LIN$(0) THEN GOTO 230:
      ELSE LIN$(0)=LIN$(1)
210   WEND
220 PRINT ", ";F$;":";CT;"recs,";FNSEC!(T2$)-FNSEC!(T1$),
    "secs";:CLOSE:GOTO 130
230 PRINT "Error sorting ";F$;", record";N:BEEP:
    CLOSE:GOTO 130
240 '   Procedure SiftUp from Bentley's Pearls,
     CACM March 1985. New element at LIN$(n),
     first element at LIN$(1)
250 I=N:WHILE I>1
260   P=I\2:IF LIN$(P)<=LIN$(I) THEN RETURN
270   SWAP LIN$(I),LIN$(P):I=P.
      WEND:RETURN
280 '  Procedure SiftDown from ibid  New element at
      LIN$(1): N elements
290 I=1:C=2:WHILE C<=N
      D=C+1:IF D<=N THEN IF LIN$(D)<LIN$(C) THEN C=D
300   IF LIN$(I)<=LIN$(C) THEN RETURN
310   SWAP LIN$(I),LIN$(C):I=C:C=C+C.
      WEND:RETURN
```

```
List/Xref version 1.0                File  demo.bsa Page: 2
References to Statements             04-30-1986 at 08:08:39

130  220,230              250  160
230  200                  290  170

                        References to Variables

A     110                LIN$ 110,150,160,170,200,260,270,290,
                               300,310
C     290,300,310
CT    170,220            N    150,160,170,200,230,250,290

D     290                                   P     260,270

F$    150,200,220,230                       T$    110
FNSEC! 110,220                              T1$   150,220
                                            T2$   180,220
I     250,260,270,290,300,310               TIME$ 150,180

J     170                                   Z     110
```

The Language

As a professional programmer familiar with many languages, I have long placed BASIC above only Cobol in my hierarchy of preferences. GW (I know BASIC this way) makes it impossible to write code of any significant size through absence of facility for program structuring and to a lesser extent through implicit declaration of variables. In addition, the ad hoc syntax of the language is to me inherently unpleasant for its inconsistencies.

Mega addresses the first problem, the important one, fully and comprehensively — to the extent that its program structuring capabilities are on the level of ADA or Modula-2.

It shares the second with other BASICs except that it provides two palliatives. One can follow the good programming practice of declaring variables, which can't be done at all in GW. And for variables not declared there is at least a means of discovering what has been implicitly done using commands to list variables in many different ways.

Let us say it alleviates the third objection: Mega has in many ways a more unified and planned flavor to its syntax, but it's still BASIC. It still has functions requiring funny characters to change the meaning of parameters and exceptional locutions unique to particular contexts.

The two languages are syntactically, as well as functionally incompatible. A 95% translator is available from APC, but I have found that the syntactic incompatibilities are superficial, and the functional ones are either easily overcome-able by writing a few routines (e.g. a CLS routine), or major enough to make me consider alternatives where they are important (graphics).

```
10 Def integer "a-z", N=rnd -1
20 N=3, Openc output #N,"data-1k"
30 For I=1 to 1000
40   Print #N,str$(rnd,"15f14")(2)
50   Next
60 Close, Openc output #N,"data-2k"
70 For I=1 to 2000
80   Print #N,str$(rnd,"15f14")(2)
90   Next
100 Close
```

Listing 3
MegaBasic program to generate test files

Statements are separated by semicolons, not colons. I/O lists contain only commas, not a mixture of commas and semicolons. Things are spelled differently (but there is — wait for it — a global string substitution command in their editor). The statement following an IF within the same line is not in the scope of the IF (is executed unconditionally). Non-decimal constants are identified by a single character suffix rather than a two character prefix. True is one, not minus one. And so on.

Moving from GW to Mega (graphics excepted) is easy, and I am inclined to do it manually because it affords the opportunity to replace entire code sequences with more concise Mega equivalents rather than blindly to transliterate. Moving from Mega to GW would be exceedingly difficult, as GW is not an extensible language, while Mega is one. Thus an absent statement in Mega wanted for a GW translation may be simply provided with a procedure of that name; an absent statement in GW needed for a Mega translation would need a Gosub and ancillary code to overcome the absence of parameter.

I snuck in the buzzword `extensible'. Mega is an extensible language in a very real sense. It means that user code may be added in such a way as to appear part of the language itself. Want to implement turtle graphics? In GW you'd end up with

```
A=5· GOSUB 2419· GOSUB 9506· A=40  GOSUB 49302
```

In Mega you can easily have

```
FORWARD 5; PENUP; LEFT 40
```

Mega provides single or multiline user functions, together with parameterizable procedures with local variables. Parameters may be passed in any of several ways. Lines may have labels.

Gosubs and line numbers remain available. (Indeed line numbers remain obligatory.)

The foregoing was all I felt necessary in terms of ability to package code in a reasonable way, but Mega lent me a great deal better. It goes beyond providing a subroutine library capability to provide Modula- style packages, complete with shared and local data and procedures, dynamically loadable and unloadable at runtime.

Mega is a very rich language. The constructs are for the most part familiar, but again and again one finds that GW restrictions have been removed, or that a capability has been generalized, removing the necessity for ad hoc syntax. An example of the former is that DATA statements may contain arbitrary expressions as data items, not just constants; of the latter, that file numbers are associated with printer and screen, simplifying things like PRINT/LPRINT/PRINT#.

There is not space here to cover this wealth in detail — the best I can hope to do is to give something of the flavor of the Mega language. This is the intent of Table 2, which contains some comparative examples of what can be accomplished with character operations. Another example of simplification by generalization, Mega contains the idea of character subscripts. Such a subscript, which may occur following any character entity (including a subscripted character string!), contains either a 'first,last' character specification or a 'first:count' character specification. At once concise and general, this locution eliminates LEFT$, MID$, RIGHT$ and probably some others. It has the additional important characteristic that such operations often take place in situ, not requiring working storage.

I'll mention just a few more language characteristics briefly, then move on to discuss the development environment.

Mega I/O is considerably simpler than GW's, containing far fewer statements to accomplish similar results. Operations within I/O buffers are gone, along with LSET, RSET, CVS, MKI$, GET, PUT, and that whole complex horde. It is arguable that a high level language user should know about buffers, although FIELD as a template has its uses. Mega does formatted or file I/O, period.

Formatted I/O, Input/Print, looks something like Fortran or C as opposed to GW's Cobol-style pictures. Given my bias, I prefer it.

File I/O, Open/Close/Read/Write, provides straightforward access to sequential or random files, including shared access for networks in a small number of statements and functions. Data objects down to the byte level may be transputted. Non-sequential file positioning is done within the I/O list. File formats are specified, in lieu of a FIELD statement, by use of (ugh) prefixes (%, &, etc) in I/O lists.

Mega provides complete access to the MS-DOS directory system including use of pathnames and subdirectory support. It goes beyond this in providing functions to examine such things as the date/time of a file, not available in GW. The first program I wrote in Mega did a treewalk of my hard disk, accumulating usage data I previously had no access to. (Given that the allocation unit is a minimum of 8k per file, which subdirectory is least efficient in space usage? If the tree at a certain level contains X amount of data, which subtree contains what percentage of that data?) Impossible in GW. Unfortunately, the program is prohibitively slow, since Mega insists I open a file before it will tell me anything about it.

User error handling in Mega is controllable in a hierarchical way, eliminating GW's inflexible single level error handlers. A subroutine which needs to field an error can do so without impact on superior control levels: upon return, the outer level's handler is reactivated. This is the only way that makes sense.

GW contains 15 arithmetic functions; Mega contains 26, including Min and Max, which every GW program I ever wrote had to have as FNMIN and FNMAX. The Mega versions, of course, permit a variable number of arguments, while my GW programs must talk about FNMIN2, FNMIN3, ... Better yet, a supplementary function, Index provides information about the results of selected previously executed functions. For Min/Max, the function value is the value of the selected datum, and Index indicates which one was selected; often invaluable.

The true math functions add ASIN, ACOS, LN, POLY, PI. There is a function to round to a specified precision and a CEIL function.

Functions are supplemented by an enriched set of operators, allowing more natural looking code. Math operators include Sgn, Round, Ceil, Trunc, and Int. Character operators include Match, Max, Min.

Strings may also be treated as bit strings, and a full complement of shifting, logical operations, and search functions is provided.

The functional richness barely touched on in all this makes the point that for benchmarks such as I gave in the beginning, and in discussions of interpreters versus compilers, there is great difficulty measuring the speed advantage (not to mention the maintainability factor) of language expressiveness. Mega has, for example, a built-in binary search function, so it is no longer necessary to spend half a day figuring out boundary conditions for this seemingly trivial bug generator. And though the result is running in an interpreter, the binary search isn't.

BENCHMARK DATA

	Secs for 1000 recs	Secs for 2000 recs	Relative percentages (2000 recs)				Free Memory	Disk Space
Bascom	12	23	100%	20%	15%	6%	51866	23964
Mega(R)	53	114	496%	100%	75%	27%	290415 b	50688
Mega	71	153	665%	134%	100%	37%	272447 b	1718+64000
GW-Basic	155	417 a	1813%	366%	273%	100%	53127	1680+65200
ZBasic	227	575 a	2500%	504%	376%	138%	53162	1680+45312

Notes: (a) The reason these two times are disproportionately more than the 1000-record times is related to string storage garbage collection
(b) Number would be 300k larger if resident utilities were not present

Table 1
Benchmark Data

Program Development

First the bad news, the single greatest shortcoming I find in this marvelous system: the program editor is not full screen. It isn't even comparable to GW's editor, buggy as it may be. Mega's is more comparable to the MS-DOS command line editing functions (you know, F1 copies one character from the template, F3 copies the rest of it, etc). There are more than a dozen functions rather than 8, but the heritage is the same. They are ctrl-something keys, mostly in the left hand. Once the fingers get used to them, they work, but that's about all that can be said for them.

But wait! Don't touch that dial!

Available when the left hand gets tired are equivalents of all the GW editor commands, of course, plus:

- A character string substitution command which can change any string to any other string, either globally or in a range of line numbers, with or without asking for permission for each occurrence. A limited form of wildcard specification may be used in the search string.
- A command which will either present an alphabetized list of defined names of a specified type (Functions, say, or Integer Functions) or will rename one name to another without interfering with comments and constants.
- Commands to rearrange program text: any block of text can be copied or moved anywhere else in the program that makes sense (interleaving at the destination doesn't), with or without changing the numbering increment on the lines.
- A Save command which remembers what you're editing. Warnings if you quit without Saving. No nonsense about quote marks and comma-As.
- A Load command which knows about the MS-DOS path.
- An Edit command which will accept a string and present for editing all and only lines containing it.
- A List command which will list the program, line ranges of it, or lines selected by a string, either to the screen, the printer, or a file.
- A built-in Xref command, operating like List, but providing flexibly defined cross reference listings. How many times have you wanted an immediate answer to the question, is this statement referenced elsewhere? (I have one problem with Xref: it takes it upon itself to issue formfeeds, usually in the wrong places, when going to the printer. If List doesn't, which it doesn't, Xref shouldn't.)

The developement environment has more. Remember I touched on packages earlier? During development different packages or programs can be loaded into separate named workspaces and can co-exist. One can easily and instantly switch from one to another. Several programs may be simultaneously resident and independent, or the workspace can share specified portions of code and data.

Getting lost? Type BASIC and get a whole new environment with the state of the current one retained. This is a reentry to Mega, not a DOS execute; it is instantaneous. Thus the execution of a program may be suspended, a fresh BASIC entered, that or another program run, and the original program then resumed.

MS-DOS execute is also available. Given memory, and the fact that it can service interrupts, I suppose that with a little work, one could create a Mega environment that served as a sort of super-Genie. The possibilities are endless.

It is in running the program that the fun begins. I am not sure of the exact rules, but it has been my experience that once I interrupt execution of a GW program almost anything I do but look at the values of variables loses me the ability to resume execution and loses the values. The opposite is the case in Mega. Almost anything that doesn't affect currently active statements may be done without harm to the possibility of continuing execution. A change to the statement that called the one currently executing, indeed loses it, with a warning (after the fact for some reason: You Can No Longer Continue, not If You Do That, You . . .). However, even in this case current data values stick around; they are available almost always rather than being evanescent.

The independence of program executability and data preservation has the important consequence that during editing and immediate statement execution the function library is active and available; not merely during the limited period when program resumption is viable.

Finally, there is running and debugging. Move over TRON and TROFF; there is a debugger here. Execution profiling in TRON's uncontrolled way is not available, but a fairly fancy single stepper is. TRACE may be asserted directly or from up to eight breakpoints within the program or on the basis of a condition (Trace starting whenever X changes, for example).

Once Trace is activated, 12 special single-step commands become available, bound to single keys. Space bar to execute one statement and break. '^' to execute until current loop or function completes, then break. 'r' to execute until control reaches here again. A line of code may be predefined and called out with one of these commands.

And if you don't see what you want here, one of the commands makes the program behave as if you had inserted a STOP, and you now have all of MegaBASIC available, including a way of inspecting the subroutine call stack to find out how you got in this mess.

Documentation, Packaging, And Support

Mega comes as a disk and an IBM-sized 450 page slip-cased manual. The disk, as well as containing the two (8 and 14 digit precision) Megas, contains manual updates, a configuration pro-

gram (no configuration is really necessary, all options being dynamically specifiable when Mega is loaded), a couple of sample programs, and a very useful library of Mega procedures and functions written by Chris Cochran. This is not your usual hodgepodge of user contributed goodies: Chris Cochran also wrote MegaBASIC.

The routine version also contains two Run programs (again, 8 and 14), a Crunch program which compresses Mega programs and can scramble them for security, and a Link program which will combine a Mega program with a Run program and make an .EXE file.

No installation of any program is necessary: they are runnable out of the box.

For one of my bent, the manual is excellent. The 450 pages are in fine print, and are written by programmers for programmers. This is not a tutorial manual, and will be relatively opaque to a novice. It is organized functionally rather than alphabetically and is well indexed. Each topic is covered carefully and in detail, often with examples, but with little redundancy and no condecension. I don't mean to imply that it is arcane: it merely is not for beginners, and knows its audience. It occurs to me that it is similar in flavor to the current article. I am attempting to describe MegaBASIC to Z-100 MS-DOS programmers who have not seen it. The audience is no broader.

The manual is a refreshing change from Zenith's, which feels it necessary to explain why I might want to use a variable, but often feels I have no need to know technical details of commands.

My major criticism of Mega's manual is that it is hard to use initially, and even when you know your way around, there is a lot of page turning. I'd like a cribsheet. There are summaries at the beginning of various chapters and elsewhere, but it takes a lot of paging through and references to the index. A unified key, command, operator, statement and function summary at the front or the back or on a card, giving page references, would help immeasureably.

Support at APC seems adequate. I have reached a knowledgeable and friendly technical person with questions a couple of times; if he is not there he returns calls. The line is not busy. I wrote them a letter (with comments, not complaints) and they didn't answer, but then no one does. I take that back: Zenith does, but only to stonewall.

Conclusions

MegaBASIC is a very expensive alternative to GW-BASIC, but offers dramatic advantages in performance and functionality. It is at once simpler and richer than GW, with fewer statements (63 vs 75) and more functions (78 vs 56). It provides a much improved base for the development of large programs.

Aside from its cost, MegaBASIC's major drawbacks are its line-oriented editor and its lack of graphics support.

Its major advantages are its extensibility, its language features for program structuring, its access to memory and system functions, and the avoidance of interpreter vs compiler problems. I will not change back.

MegaBASIC version 5.22. Development version, $380; Runtime version, an additional $250. American Planning Corporation, 4600 Duke Street, Suite 425, Alexandria, VA 22304. Orders: 1-800-368-2248; technical: 1-703-751-2574. I learned of MegaBASIC through their sending me a 28 page glossy brochure containing much solid data and not much hype. Presumably the brochure is still available on request.

*

Table 2
Some character operations

To	In Mega use	In GW use
Set the n-th character of Y$ to the n-th character of X$.	`Y$(n:) = X$(n ' n)`	`mid$(Y$,n,1) = mid$(X$,n,1)`
Set R$ to the concatenation of X$ and Y$	`R$ = X$ + Y$`	`R$ = X$ + Y$`
Set R$ to X$ less any instances of Y$	`R$ = X$ - Y$`	`R$=X$:j=len(Y$):i=instr(R$,Y$):while i: R$=left$(R$,i-1)+mid$(R$,i+j): i=instr(R$,Y$):wend`
Set R$ to n occurrences of Y$	`R$ = Y$ * n`	`R$="":for i=1 to n: R$=R$+Y$:next i`
Match corresponding characters of X$ and Y$, let R$ contain the smaller at each position	`R$ = X$ min Y$`	`10 R$=X$:for i=1 to len(X$): if mid$(R$,i,1)>=mid$(Y$,i,1) then mid$(R$,i,1)=mid$(Y$,i,1) 20 next i`
Insert Y$ to the right of the n-th character of X$.	`X$(n:) := Y$ (uses no stack storage)`	`X$ = left$(X$,n) + Y$ + mid$(X$,n+1)`
Print I without surrounding spaces	`print trim$(str$(I))`	`print mid$(str$(I),2)`
Set R$ to the capital letters	`R$ = chr$("A","Z")`	`R$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"`
Set R$ to X$ reading right to left	`R$ = rev$(X$)`	`R$="":for i=len(X$) to 1 step -1: R$=R$+mid$(X$,i,1):next i`
Blank out occurences of digits in X$	`X$ = tran$ (X$ '0123456789', ' '*10)`	`10 for i=1 to len(X$) if instr("0123456789",mid$(X$, i,1) then mid$(X$,i,1)=" " 20 next i`
Determine whether Y$ exists in X$, a concatenation of possible values of Y$, using binary search.	`if find(ord, X$=Y$,len(Y$))`	look up binary search in Knuth and copy the code.

Cheap Speed

A Review Of The Micronics Technology Speed Mod For The H/Z-151 And H/Z-161 With Some Suggested Improvements

Pat Swayne
HUG Software Engineer

Since I was not assigned to do the review of the Wildfire speed modification for REMark, and since I could not afford anything close to 249 bucks, I decided to order the speed modification advertised in the passed few months of REMark by Micronics Technology for only $34.95. For 1 dollar more, they included a disk containing a program to fix the real time clock. At the time, I did not know what that was about. I was thinking that the Micronics modification might be similar to the one described in the June 1986 issue of REMark ("H-150 Speed-up Modification"). One of the problems discussed in that article was how to keep the real time clock constant while changing the processor speed. The author of that article solved the problem in his hardware design.

Installing The Micronics Modification

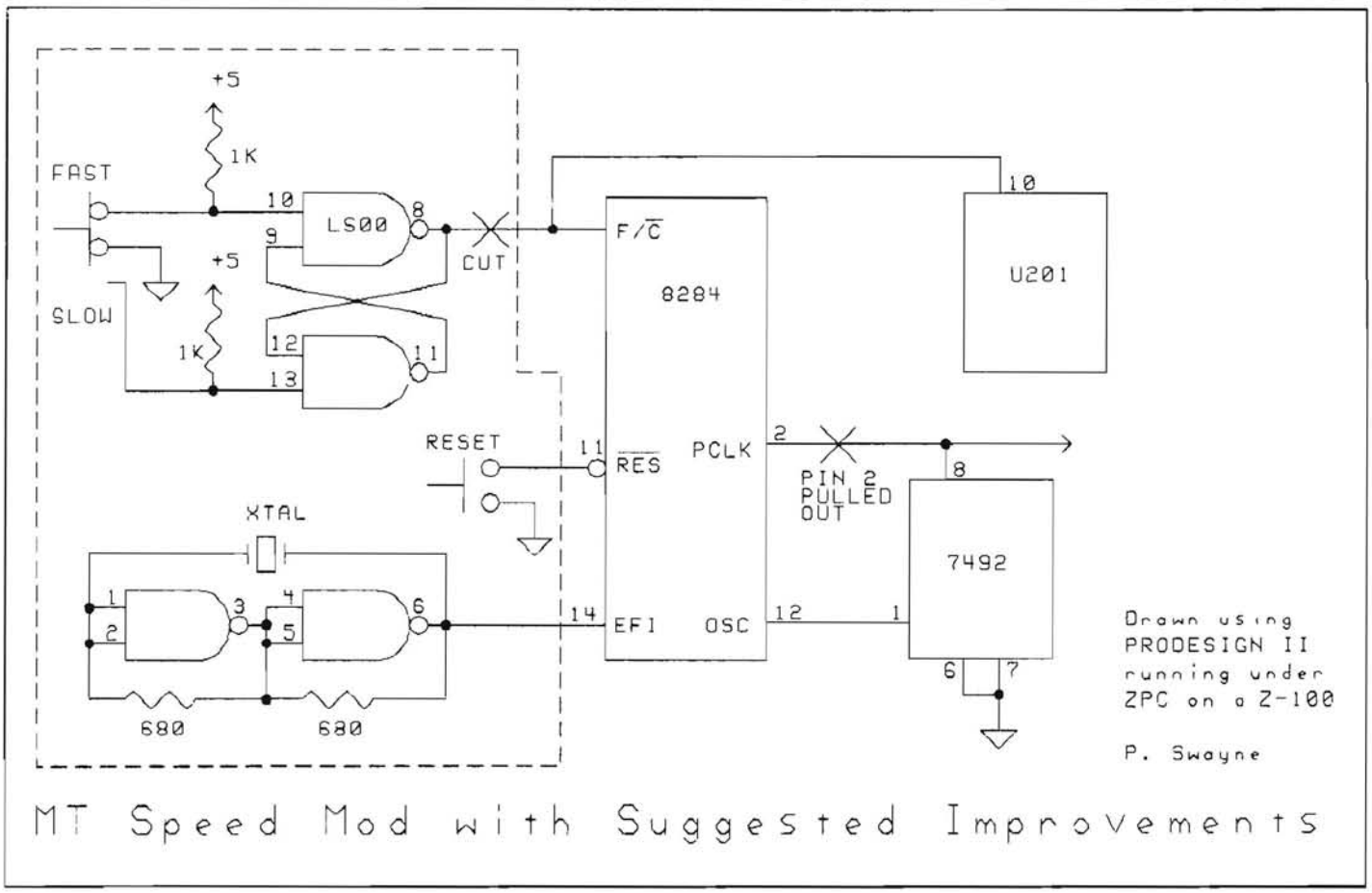
The Micronics kit arrived a few days after I placed the order by first class mail. It consists of a small circuit board with one IC, a socketed crystal, and an empty socket with pins extending through the board, and another socket (for extension) plugged onto the pins. The board is wired to two switches mounted on a mounting plate of the type that are used to fill blank board slots in the computer. The small board installs on the CPU board in your H/Z-151. You remove the 8284 clock driver chip (U211), plug the board's extension socket into its place, and plug the 8284 in the empty socket on the Micronics board. You can mount the plate with the switches in any unused board slot. One of the two switches in the plate is a slide switch that is used to change processor speed between the normal 4.77 MHz and the faster 6.67 MHz provided by the modification. The other switch is a pushbutton that can be used to reset the computer if it hangs up, and Ctrl-Alt-Del does not work.

The documentation provided with the Micronics kit explains installation fully, but the only suggestion they provide if the unit does not work is to try a slower crystal. That is unfortunate, because you can almost certainly get it to work at 6.67 MHz or even faster by changing a chip or two on your CPU board. The June 1986 REMark article suggests that you check the CPU (U212), the DMA controller (U209), and the bus controller (U243), and the author describes the brands of chips he had good results with. Another chip that can cause trouble, and did in my case, is the clock driver (U211). At first, my computer would not reset with the switch in the fast position, although it would run fast if first reset with the switch in the slow position. Replacing the clock driver fixed that problem. The original was an AMD plastic cased part, and I replaced it with an NEC 8284 in a ceramic case.

How It Works

In Figure 1, the area in dotted lines represents the circuitry on the Micronics speed mod board. In operation, it is similar to the CDR speed modification for the Z-100 (see "Speed Mods for the Z-100" in the April 1985 REMark). It takes advantage of the ability of the 8284 to be driven by two different frequency sources — internal or external oscillators. Normally, the internal oscillator is used, and it is controlled by the 14.31818 MHz crystal on the CPU board. This frequency is divided by 3 to provide the 4.77 MHz clock for the computer. There are two other outputs on the 8284. One is the undivided output of the internal oscillator (OSC, pin 12), and the other is a signal called PCLK (pin 2), which is the input frequency divided by 6 (normally 2.38 MHz).

The 8284 chip has a pin (EFI, pin 14) that determines whether it will use its internal or an external oscillator. This pin is normally grounded, but the Micronics board connects it to a debounced switch. When you move the switch to the FAST position, the select pin is brought high, which causes the 8284 to be driven by the external oscillator on the Micronics board. As a result, the clock supplied to the CPU and the PCLK clock change frequency, but the OSC output does not, since it always comes from the internal oscillator. (It's a good thing that OSC does not change, because the



MT Speed Mod with Suggested Improvements

video card uses that signal, and will not produce a proper screen display if the frequency changes.) The Micronics oscillator runs at 20 MHz, so the CPU clock changes to 6.67 MHz, and PCLK changes to 3.33 MHz. The H/Z-151 can operate under these conditions, but unfortunately, the PCLK signal is used to drive the timer chip that provides the real time clock for MS-DOS and is also used to generate sound. So with the Micronics board switched to FAST, the MS-DOS TIME command no longer provides accurate time, and any sounds produced by games or other programs are pitched higher than they should be.

That is where the $1 disk comes in. A program called FIXCLK is provided on it that is supposed to fix the timer interrupts so that real time is kept accurately. A number of other public domain programs are thrown in on the disk, so it's not a bad deal. However, the FIXCLK program takes a rather klutzy approach to fixing the problem. It captures the timer interrupt vector, and then allows 10 of every 16 interrupts to be passed on to the rest of the system. This means that any program "watching" the timer interrupts will see a burst of 10 interrupts faster than normal followed by a period of no interrupts for a while, etc. While this process makes the real time clock more accurate than it is without FIXCLK, it is still off one minute per hour.

The Software Improvement

The first thing I set out to do after installing my Micronics speed mod was to write an improved version of FIXCLK. The timer interrupt is produced by an 8253 timer chip by dividing down the PCLK signal. You can easily re-program the 8253 to change the divisor, but unfortunately, it is already programmed with the largest divisor possible, so you can only program the timer interrupts to occur at a faster rate, not a slower one. When you speed things up by setting the Micronics mod to FAST, you need to slow the timer interrupts down somehow.

My solution to the problem was to write a FIXCLK program that caused the 8253 to produce timer interrupts at exactly twice the normal rate when the Micronics mod was set to fast, and then pass every other interrupt on to the system. The result is a steady interrupt at the original rate. I also wrote a program called PRGCLK that allows me to re-program the 8253 back to the original rate, for running games, etc. at the original processor speed. My FIXCLK program detects when the 8253 has been altered, and does not skip interrupts unless the 8253 is programmed to the right divisor for the 6.67 MHz clock speed. I did that because some games take full control of the 8253 and the timer interrupts. The source code for FIXCLK and PRGCLK are at the end of this article.

The Hardware Improvements

Almost as soon as I wrote the new FIXCLK, I decided that it wasn't good enough, and the only thing that would satisfy me would be to modify the hardware so that the PCLK signal remained at its original value when the CPU clock changed speeds. In the mod presented in the June 1986 article mentioned previously, two 8284's are used to accomplish this. One produces the CPU clock, while the other is used to make OSC and PCLK. On the Micronics board, we already have an OSC signal that remains constant, so all that is needed is to divide it by 6 to get a constant PCLK. I used a 7492 (Heath p/n 443-34) IC to accomplish it, and you can see how I connected it in Figure 1. I attached the 7492 to the Micronics board by bending out pin 10, which is the power supply ground pin, and soldering it to pin 9 of the 8284 socket on the board. The other connections are connected with fine wire to the appropriate 8284 socket pins. Pin 2 of the 8284 was bent out to disconnect the

original PCLK signal. Please note that the power supply +5 and ground connections to the 7492 (pins 5 and 10) are not shown in Figure 1.

The 7492 modification fixes the PCLK problem, but it introduces a new problem. When the computer is first turned on or reset, it tests the timer IC by comparing the timer interrupt speed to a software loop. Therefore, if the CPU clock speed has been changed but the PCLK speed has not at reset time, the timer IC test will fail, and the reset operation will halt with an error message. In the June 1986 modification, the solution was to provide an RC network on the board that caused it to start out at the slow CPU clock speed at reset, and then switch to the high speed after a brief period. Since I did not want to hang any more parts on my little Micronics board, I looked for another solution, and found one as I watched the little status LEDs light up and go out each time the computer was reset.

When you first reset the computer, all of the LEDs turn on, and then one by one they go out as various tests are passed. I figured that if I used the signal driving the appropriate LED to also control the speed of the Micronics board, I could be assured that the computer would start out slow, and then switch to the fast speed after the timer test was done. I decided to use the signal that controls the INT LED, because it is the next to the last one to go out, and because the DSK led, which is the last one to go out, comes back on when you boot MS-DOS. It then occurred to me that since the LEDs are controlled by a port, I could control the speed of my computer by writing to that port myself. So I decided to eliminate the hardware speed switch altogether, and just ran a wire from pin 10 of U201, which controls the INT LED, to pin 13 of the 8284 socket on the Micronics board, and cut the trace that connects that pin to the switch debouncer. After all, who wants to reach around to the back of your computer to change speeds when you can do it from the keyboard.

Now all I have to do to change speeds is to call up DEBUG or the built-in MFM-150 debugger, and enter

```
-OC0,17
```

to switch to the 4.77 MHz clock speed, or

```
-OC0,1F
```

to switch to the 6.67 MHz speed.

To make things even easier, I wrote a little program (CS.COM) that allows me to switch to the slow speed by typing

```
CS S
```

at the system prompt, and to the fast speed by typing

```
CS
```

If you type in and run the following BASIC program, it will generate CS.COM. Or you can use the assembly source for CS, listed at the end of this article.

```
10 REM THIS PROGRAM CREATES CS.COM
20 DEFINT A-I:OPEN "0",1,"CS.COM
30 S=0:S1 = 1915  FOR I=1 TO 17
40 READ B:S=S+B:PRINT #1,CHR$(B),
50 NEXT I:IF S<>S1 THEN PRINT "TYPING ERROR!":END
60 CLOSE #1:LOCATE 23,1:PRINT "DONE!":SYSTEM
70 DATA 138,38,93,0,176,23,128,252,83,116
80 DATA 2,176,31,230,192,205,32
```

As stated in the Micronics documentation, the FORMAT and DISKCOPY programs do not work when you are in the fast mode of operation. This is true whether you make the changes presented here or not. But with a software speed switch, you can write a batch file that will switch to the slow speed, run FORMAT or DISKCOPY, and switch back to the fast speed. For example, a file called FORM .BAT could contain these lines:

```
CS S
FORMAT %1
CS
```

To format a disk in drive B:, you would type FORM B: instead of FORMAT B:, and you would not have to worry about the processor speed.

How Fast Is Fast

If you are thinking that 6.67 MHz is not very fast when compared to, say, an 8 MHz H/Z-158, consider that the H/Z-158 inserts a wait state on each access to memory, while the speeded up H/Z-151 does not. I wrote a simple benchmark in the form of the following BASIC program:

```
10 FOR I=1 TO 5000:NEXT I
20 BEEP
```

Then I ran the program on my speeded up '151 and on a '158 and timed the two. The same version of GW-BASIC and the same operating system were used on each system. The '151 actually came out a few hundredths of a second faster than the '158.

Final Comments

While doing the experiments that resulted in this article, I wore out the socket that comes attached to the Micronics board, which is used to plug it into the CPU board and extend it above other parts. It seemed to be a poor quality socket, so you might consider replacing it with a Heath p/n 434-310 socket if you plan to do some experimenting. Be sure to pull out pins 13 and 14 of the new socket. I experienced strange failures, accompanied by beepings and clickings, as the socket wore out.

While I had my computer apart, I took the power supply apart and turned the fan around, so that it now blows out of the computer instead of into it. If you study the path that cooling air has to take in the computer, you can see that it makes sense to turn the fan around. With the fan blowing in, air must pass over the top of the boards, down between them, and out the slots at the bottom of the other side. Since heated air normally rises, the fan is fighting the natural air flow. With the fan turned around, air comes in at the slots in the bottom, rises up between the boards, and is exhausted out the other side. If the components in the computer generate more heat because of the increased clock speed, turning the fan around should compensate for it. Some people have relocated the boards in their computers so that the CPU board is closest to the power supply, for greater cooling. If you reverse your fan, the CPU board should be put in its original position, because more air will flow between boards in this configuration than across the surface of the board closest to the power supply.

Another reason for turning the fan around is that dust collects in the power supply with the fan blowing in, and could eventually destroy it. With the fan blowing out, dust collection anywhere in the computer is greatly reduced. My 6-year-old H-89 is proof of that, because it has far less dust on its power supply components than the two-year-old H-151 had when I opened it. And the H-89 is still used a lot, because it is my wife's word processing station.

Assembly Source Code For FIXCLK.COM.

```
        PAGE    ,132
        TITLE   FIXCLK - ACCURATE VERSION
;       THIS PROGRAM FIXES THE SYSTEM CLOCK SO THAT ACCURATE
;       TIME IS KEPT WITH THE MICRONICS SPEED MODULE INSTALLED.
;
```

```
;   THIS PROGRAM CAUSES THE 8253 TO PRODUCE CLOCK
,   INTERRUPTS 36.412962 TIMES A SECOND (TWICE THE NORMAL
;   RATE), AND SENDS HALF OF THEM THROUGH TO THE SYSTEM
,   IT ALSO CHECKS THE COUNTER, AND IF IT HAS BEEN SET TO
;   A NEW VALUE, CLOCK SKIPPING IS DISABLED
;
.   BY P  SWAYNE, HUG  07-OCT-86
;
;   THE FOLLOWING VARIABLE SHOULD BE SET TO 45771 FOR A
;   20 MHZ CRYSTAL, 42178 FOR AN 18.43 MHZ CRYSTAL, 41194
.   FOR AN 18 MHZ CRYSTAL, AND 36616 FOR A 16 MHZ CRYSTAL

COUNT   EQU     45771           ;COUNT FOR 8253 CHIP

JMPF    MACRO
        DB      0EAH            ,DEFINE FAR JUMP
        ENDM

DUMMY   SEGMENT STACK           ;AVOID LINK ERR  MSG
DUMMY   ENDS

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

START:  JMP     SETUP           ,SET UP PROGRAM

CLKFLG  DB      0
CLKCNT  DB      0

        DB      'FX'            ;IDENTIFIER
MYCLK:  TEST    CS:CLKFLG,80H   ;TEST FOR SKIP OFF
        JNZ     NOSKP           ;IT'S OFF
        XOR     CS:CLKFLG,1     ;TOGGLE CLOCK FLAG
        JNZ     SKCLK           ;TIME TO SKIP CLOCK
NOSKP:  DEC     CS:CLKCNT       ;DECREMENT COUNT
        JNZ     NOTZRO          ;NOT ZERO
        MOV     CS:CLKCNT,18    ,ELSE, RESET COUNT
        PUSH    AX
        MOV     AL,0
        OUT     43H,AL          ;READ COUNT FROM 8253
        IN      AL,40H          ;READ LSB
        IN      AL,40H          ,READ MSB
        CMP     AL,COUNT SHR 8  ,PROGRAMMED TO COUNT?
        MOV     AL,0            ;ASSUME IT IS
        JZ      ATCNT
        CMP     AL,(COUNT SHR 8)+1 ;CHECK COUNT + 1
        JZ      ATCNT           ,CONSIDER IT COUNT
        MOV     AL,80H          ;ELSE, FLAG NOT COUNT
ATCNT:  MOV     CS:CLKFLG,AL    ;SET SKIP CONDITION
        POP     AX
NOTZRO  JMPF                    ;ELSE, GO TO BIOS
OLDCLK  DW      0,0
SKCLK:  PUSH    AX
        MOV     AL,20H
        OUT     20H,AL          ;RESET INTERRUPT CONTROLLER
        POP     AX
        IRET

,       SET UP CLOCK VECTOR AND 8253

SETUP:  PUSH    DS              ;SAVE DS
        XOR     AX,AX
        MOV     DS,AX           ,POINT TO INT  PAGE
        MOV     SI,8*4          ;POINT TO CLOCK VECTOR
        LES     DI,DWORD PTR [SI]       ;GET VECTOR
        CMP     ES:WORD PTR -2[DI],'XF' ;INSTALLED?
        JNZ     NOTIN           ;IF NOT, DO IT
        POP     DS              ;FIX DS
        MOV     DX,OFFSET INMSG
        MOV     AH,9
        INT     21H             ;ELSE, SAY "ALREADY IN"
NOTIN   CLI                     ,KILL INTERRUPTS
        MOV     WORD PTR [SI],OFFSET MYCLK ;SET NEW VECTOR
        MOV     2[SI],CS
        POP     DS
        MOV     OLDCLK,DI       ;SAVE OLD VECTOR
        MOV     OLDCLK+2,ES
        STI
        MOV     AL,36H
        OUT     43H,AL          ;SELECT 8253 CHANNEL 0
        MOV     AX,COUNT        ;GET COUNT VALUE
        OUT     40H,AL          ;SEND LSB TO LATCH REGISTER
        MOV     AL,AH
        OUT     40H,AL          ,SEND MSB
        MOV     DX,OFFSET SIGNON
        MOV     AH,9
        INT     21H             ;PRINT SIGN-ON MSG.
        MOV     DX,OFFSET SETUP
        INT     27H             ,EXIT, CLOCK CODE RESIDENT

INMSG   DB      13,10,7,'FIXCLK is already installed.$'
SIGNON  DB      13,10,'FIXCLK is now installed.$'

CODE    ENDS
        END     START
```

Assembly Source Code For PRGCLK.COM.

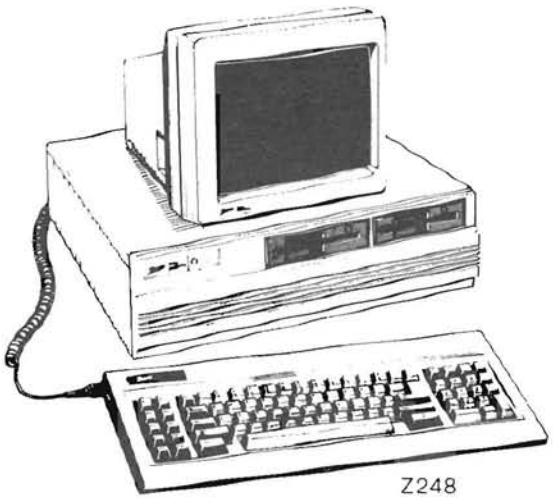
```
        PAGE    ,132
        TITLE   PRGCLK - CLOCK PROGRAMMER
,   THIS PROGRAM ALLOWS YOU TO PROGRAM THE SYSTEM CLOCK FOR
,   ACCURATE TIME KEEPING WHETHER YOU ARE IN THE FAST OR
;   SLOW MODE OF OPERATION OF THE MICRONICS SPEED MODULE.
;
,   TO USE THIS PROGRAM, ENTER
.
,   PRGCLK F                TO PROGRAM CLOCK IN FAST MODE
;
;   PRGCLK S                TO PROGRAM CLOCK IN SLOW MODE
.
.   BY P  SWAYNE, HUG  07-OCT-86
;
;   THE FOLLOWING VARIABLE SHOULD BE SET TO 45771 FOR A
;   20 MHZ CRYSTAL, 42178 FOR AN 18.43 MHZ CRYSTAL, 41194
,   FOR AN 18 MHZ CRYSTAL, AND 36616 FOR A 16 MHZ CRYSTAL

COUNT   EQU     45771           ;COUNT FOR 8253 CHIP

DUMMY   SEGMENT STACK           ,AVOID LINK ERR  MSG
DUMMY   ENDS

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     5CH
FCB     LABEL   BYTE
        ORG     100H

START   MOV     AL,FCB+1        ;GET ARGUMENT
        MOV     BX,COUNT        ,ASSUME FAST
        CMP     AL,'F'          ,FAST?
        JZ      PRGCLK          ;YES, PROGRAM CLOCK
        MOV     BX,0            ;ASSUME SLOW
        CMP     AL,'S'          ;SLOW?
        JZ      PRGCLK          ;YES
        INT     20H             ;ELSE, EXIT

PRGCLK  MOV     AL,36H
        OUT     43H,AL          ;SELECT 8253 CHANNEL 0
        MOV     AL,BL           ;GET COUNT VALUE LSB
        OUT     40H,AL          ;SEND LSB TO LATCH REGISTER
        MOV     AL,BH
        OUT     40H,AL          ;SEND MSB
        INT     20H             ,AND EXIT

CODE    ENDS
        END     START
```

Assembly Source Code For CS.COM.

```
        PAGE    ,132
;   SPEED CHANGE PROGRAM FOR MODIFIED H/Z-151
;   TO USE THIS PROGRAM, ENTER
;
,   CS S            TO SELECT SLOW SPEED
,
```

```
,   CS              TO SELECT FAST SPEED

,   BY P  SWAYNE, HUG  14-OCT-86

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     5CH
FCB     LABEL   BYTE            ,DEFINE FCB
        ORG     100H

START:  MOV     AH,FCB+1        ;GET ARGUMENT
        MOV     AL,17H          ,ASSUME SLOW
        CMP     AH,'S'          ;IS IT SLOW?
        JZ      SETSPD          ;IF SO, SET IT
        MOV     AL,1FH          ;ELSE  IT'S FAST
SETSPD  OUT     0C0H,AL         ,SET SPEED
        INT     20H             ,AND EXIT

CODE    ENDS
        END     START
```

First Capitol Delivers Zenith. Your Way.

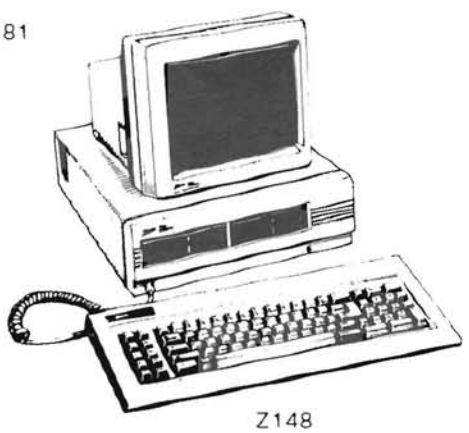


Z248

Z181

Specialists in

CAD Systems
Local Area Networking



Z148

ZENITH | data systems

AUTHORIZED SALES AND SERVICE

Have it your way.

If you're looking for a PC, you couldn't do better than Zenith. And you couldn't buy your Zenith at a better place than First Capitol. We'll put together a system the way YOU want it. And we'll do it for a price that competes with the guy who just sells boxes that fall on his loading dock.

Great Prices. And A Great Machine.

We're committed to Zenith. And to you. the standard Zenith models that come from the factory are a good buy. And we'll be glad to sell you one at a price that will match anyone. But if you want the full potential, we'll fix it up for you. Larger disk drives. More memory. Internal tape back up. Removable winchesters. All at a budget pleasing price.

We make Zenith Scream.

When a new model comes out, we're among the first to get it. Our Zenith fanatics take it apart, prod it and poke it until it screams for mercy. Then we put it back together (usually with some new parts) so that it REALLY screams. With power.

You'll be back.

Check with us. Check elsewhere. We'll put together a custom quotation for you we GUARANTEE you'll find interesting. We think you'll come back to us. There's a reason for the growing list of companies, government agencies, and end users that know us as the source for Zenith. Your Way.

First Capitol Computer

1106 First Capitol Dr.
St. Charles, MO 63301

1-800-TO-BUY-IT (800-862-8948) Orders and quotes

1-314-724-2336 Technical support and order status

The StockSystem Challenge.

We will meet or beat any dealer's price on new stock Zenith systems. And for enhanced systems, no one can touch our combination of value, performance, and custom service.

Before you buy . . . . . .

Call Us First. And Last.

Paper Bits
A Review Of Softstrips On Zenith Computers

Joseph Katz
103 South Edisto Avenue
Columbia, SC 29205

he Softstrip System from Cauzin Systems is a potentially significant new way to store and transfer files. It lets you use ordinary sheets of paper almost the way you now use floppy diskettes, writing to and reading from paper instead of magnetic media. But don't run away with my analogy. You can't tell your word processing program that the Softstrip is a drive and write directly to it. Softstrips are intermediate containers for files: they archive existing files that were created the usual ways, allowing you to reconstitute those archives into the original files at a later time.

It's really an extremely powerful capability because Cauzin's Softstrips set files free. Once files are transferred to a Softstrip, they are completely independent of the system on which they were created. At least where text and data are concerned, system incompatibilities become mostly irrelevant. Cauzin's Softstrips, therefore, can become the lingua franca of the computer world.

When The Softstrip System writes files to a Softstrip, it actually prints graphic representations of all their component bits onto a "data strip." Cauzin suggests "data strip" as the generic term and reserves "Softstrip" as its trademark for data strips produced with its technology. That technology includes a convention for representing bits graphically in a form based on what Cauzin calls a "DiBit." A DiBit is a pair of rectangles assigned the following code: if the pair is "black/white," the bit is a zero; if the pair is "white/black," the bit is a one; if the pair is "black/black" or "white/white," there is an error.

At first glance the printed product might remind you of bar codes, but Softstrips are much more sophisticated than bar codes. They have greater density, packing information into a complex structure that includes a header, data section, file section, and boundary lines. All but the file section — the files stored on the Softstrip — is internal information used by The Softstrip System to insure data integrity. Such details are hidden from the user, who need know nothing at all about them.

When The Softstrip System reads a Softstrip, it pulls all that graphic information out of the data strips, translates it back into electronic information, and uses it to reconstitute disk files with exact duplicates of the original contents. A file created on a Macintosh, for example, becomes the same file on a Z-158, but conforms to DOS file conventions.

Think of the transfer in both directions as depending on the paper record of a modem file transfer and you'll be right on the mark. But keep in mind everything implied by that way of thinking. One implication is the ability to store in a Softstrip anything you can store in a disk file. Look at the actual Softstrip in Figure 1, for example. In it are an executable MS-DOS program, instructions for using it, and the source code to it. The Softstrip System will read that Softstrip from the printed page and reconstruct its information as three disk files. Then, if you had a PC compatible, you could run BEEP.COM (the program), perhaps after reading BEEP.DOC (the instructions), and prior to studying BEEP.ASM (the source code) so you might improve the program. But what if you owned a Macintosh? Although you could still read the three files to your system, once they were there you could do things only with the text files, BEEP.DOC and BEEP.ASM. BEEP.COM would not run on your system, and therefore, its instructions and source code would have little real value. Just as with file transfers by modem, the transfer itself does not necessarily create anything useful: it's a vehicle only. The Softstrip System is just such a vehicle.

Of course there are tradeoffs in The Softstrip System, the least consequential of which is the purchase price. It is reasonable, especially for the owner of a Heath or Zenith PC compatible. All you need to read Softstrips is The Softstrip Reader System at $199.95. It requires a serial port, which comes with all Heath and Zenith PC compatibles. There are a few different ways to go if you want to write Softstrips. The cheapest way assumes an Epson FX or RX printer. If so, all you need on a Heath or Zenith PC compatible is

Cauzin's Stripper package, which costs $19.95. That's all you need because these computers come with a parallel printer interface. In fact, Heath's and Zenith's PC compatible computers seem made for The Softstrip System, and vice versa.

The consequential expenditure is time. Although all three files held by the Softstrip in Figure 1 total under 1K, I spent about three minutes waiting for my Epson FX-185 to print it. Most of that time seems to be overhead, because an entire page of seven Softstrips takes about seven and one-half minutes. Plan to spend about an hour waiting for an Epson like mine to print a full diskette into Softstrips. Reading a Softstrip takes time too, but much less: about a half minute for each Softstrip, a minimum of about half an hour to read the Softstrips for a full diskette.

What one gets, though, is portability, durability, and reliability far above what magnetic media provide.

The Softstrip System makes possible the porting of all files among the three kinds of computers it currently supports: PC compatibles, such as the Z-100 PC series; AT compatibles, such as the Z-200 PC series; Apple II compatibles; and Apple Macintosh. You can transfer files among any of those computers. Since you can't run a program like MacPaint on a Z-158, you probably won't want to bother transferring executable programs between different computers. But since you can use some data and most text files, there is real point in being able to transfer them: it is possible, for example, to massage MacPaint's graphic files into something usable on a Z-158, and no trouble at all to do the same thing with files from an Apple version of WordStar. In fact, if there are different kinds of computers in distant locations, The Softstrip System looks like the most economical way to make such transfers: a Softstrip produced on any computer can be read with The Softstrip Reader System on any supported computer.

Softstrips are as durable as printed paper — which is much more durable than files on magnetic media. For one thing, Softstrips are impervious to stray magnetism; for another thing, they have no magnetic imprint to bleed or decay over time. File them, box them, or punch them for inserting in binders. Give a Softstrip reasonable care and it should last as long as the paper on which it was printed.

As for Softstrip reliability, it seems absolute. If you print them, use a ribbon in half-way decent condition and check the output with DIBITEST. When you read them, CAUZCOMM checks for errors, corrects up to two, and passes none.

Uses

One application for which Cauzin is pushing The Softstrip System is as a cheap way to publish software. Its ads with short sample programs have been appearing in several computer magazines since about the beginning of January 1986, and at least one magazine (Byte) has started publishing Softstrips of source code from some of its articles. Cauzin itself publishes a series of StripWare (another trademark) containing programs from books and magazines at prices from $2.98 to $12.98. It appears a promising application.

Fulfillment of that promise seems to depend on a minimum of two factors. The obvious one is whether Softstrips catch on. If enough publishers produce enough worthwhile Softstrips, enough consumers may buy The Softstrip Reader System to increase Softstrip demand and feed the cycle. But until a great many people demand Softstrips, there's no motive for publishers to produce them, and therefore, little motive for consumers to buy Readers. There we have a high-tech version of the chicken-and-egg dilemma.

The less obvious factor concerns authors' and publishers' attitudes towards having material available in machine-readable form. In areas like texts on C programming, there is a tendency for authors or publishers to sell diskettes containing the book's source code. Those diskettes presumably contribute revenue that would disappear if the publishers included Softstrips in the books. Maybe, though, book sales would increase commensurately? Who knows. Only time can tell how these two factors, and perhaps others, will resolve.

Of course, your own situation might not depend on those resolutions. You might, for example, be interested in providing information instead of receiving it. In that case, Softstrips might be a reasonable way to disseminate data, as well as programs in machine-readable form. All you would need to establish this kind of cottage-publishing business is your computer, an Epson printer, The Softstrip System, and an office copier or the services of a cheap offset printer. Or perhaps you have to archive tax or other records economically but in machine-readable form, without worrying about what to do should you change computer systems. Softstrips seem a reasonable solution.

Before you leap into publishing Softstrips, though, keep in mind that you will need a license from Cauzin for any, but the most strictly non-commercial use of the system. The Stripper manual says, "A special license is available, free of charge, to educational institutions, user groups, and computer clubs to permit them to publish original data strips to distribute public domain software, or to include programs and data in newsletters." The idea, I think, is that all others pay cash.

My firm conclusion is that right now The Softstrip System might well be one of the following: the future direction of microcomputer publishing; the best solution for storage and transmission problems; a novelty now and a collectible in the future; none of the preceding. It's just too early to know which or whether. Cauzin's advertising blitz started around January 1986, and the first units started shipping around March. It's July 4, 1986, as I write. These are early days for a commitment. No matter: The Softstrip System is at least worth knowing about and, if you have the money and inclination, experimenting with.

System Overview

The Softstrip System comes in two cartons for $199.95: one holds The Softstrip System Reader, the other holds The Softstrip System Accessory Kit. It's the same Reader no matter which computer you own. What defines the System configuration for a specific kind of computer is the Accessory Kit. For Heath/Zenith's compatibles, you want The Softstrip System Accessory Kit marked on the box "For IBM PC and Compatibles." Cauzin sells additional Accessory Kits separately at $19.95 each. Since you can swap a Reader from one kind of computer to another easily, you therefore can make The Softstrip Reader System available on all the computers it can support for a dime less than $40 additional.

The Reader (mine is marked "Model RDR 100") is an optical scanner in a grayish-brown molded plastic housing, 16 × 3 × 2-1/2 inches. At the top right is aN LED power indicator; at the right end is an on/off slide switch for the power. Beneath it are two sockets: a three-prong socket for the cable from the 10 volt transformer that powers the unit, and a modular telephone jack for the Reader's end of the cable that goes to the computer's serial port. The transformer comes with the Reader. The serial cable, which will differ among kinds of computers, comes with the Accessory Kit. For compatibles it has a standard DB-25 connector at one end and a standard telephone plug — RJ-14, not RJ-11 — at the other.

(If you own an AT-compatible, such as the Z-241 or Z-248, you'll need to get an adaptor from the DB-25 female on the cable to the DB-9 male on the computer.) With the Accessory Kit also comes software to control the reader. Cauzin makes no H-DOS, CP/M, or generic MS-DOS version of the software, but the sidebar to this article might interest you if yours is that kind of Heath or Zenith computer.

Setting Up The Reader

Setting up the hardware — the Reader and its Accessory Kit — on a compatible is absolutely foolproof: all you do is plug things into their obvious mates. The plug on the transformer goes into a standard 110-volt electrical outlet and the plug on its power cord goes into the socket on the Reader; the DB-25 on one end of the serial cable goes into the RS-232 jack on the back panel of the computer and the telephone plug on the cable's other end goes into the modular jack on the reader. That's it.

Software setup requires a little more judgment, but not much. There are only two necessary files on the floppy diskette included with the Accessory Kit: CAUZCOMM.COM and CAUZCOMM .DEF, which together total little more than 13KB. Make a working copy of those files (on either a floppy diskette or a hard disk) and run CAUZCOMM, the program that links the reader to the computer. There are two ways to do that: the command "CAUZCOMM" alone executes the program in a TSR ("Terminate-and-Stay-Resident") mode, as a RAM-resident program which can be invoked by pressing ALT R at any time; "CAUZCOMM/N," however, executes the program in a normal, non-resident mode, so it occupies no RAM between uses. Since CAUZCOMM prohibits the operation of other programs while it is running, I see no advantage to the TSR mode and, therefore, suggest using it in the non-resident mode.

There are only a couple of things you might need to change in CAUZCOMM. Of course, the Reader needs somewhere to store files created when it reads a Softstrip: Drive A is the default for such files. If you own a stock Z-100 PC or Z-200 PC and you're happy with the default storage drive, you should be able to start reading Softstrips immediately. The only possible other installation decision pops up — literally, on the CAUZCOMM "Options" menu — if you have more than one serial port on your computer. There's only one COM device on my Z-152 so the menu was mute about port assignments; but when I installed CAUZCOMM on my H-158, in which I have an internal modem, the program's options gave me the choice of configuring the serial port for the Reader as either COM1 or COM2. (Since stock Heath/Zenith compatibles come with only one serial port, that option won't appear when CAUZCOMM is run on them.) The default is COM1: anticipating serial peripherals, I had installed my ZOOM/Modem PC as COM2 and, therefore, was happy with the default serial port assignment. So much for setting up the software on a Heath/Zenith compatible.

Reading Softstrips

Reading a Softstrip is fairly easy. Each Softstrip is associated with two alignment marks: a bullet (a big, bold, dot) at its lower left and a bar at its lower right. The front of the plastic housing on the Reader has a lip with a circular peepsight on its lower left. Energize the reader with the slide switch so the power LED glows, place the Softstrip on a flat surface so the bullet and bar face you, put the Reader atop the Softstrip so the bullet shows through the peepsight and the lip is aligned with the bar. Assuming you have run CAUZCOMM and are looking at its main menu, press "R" (for "Read").

Immediately, that menu disappears and is replaced with another offering you the sole option of "Q" (for "Quit"). What happens next, unless you do quit, is progressive. You will see a message that CAUZCOMM is "trying to establish communications" — which means it is trying to communicate with the Reader.

If the message remains on the screen more than a second or so, the communication failed and you will have to take appropriate action. Assuming the Reader has power, good cable connections, and access to the right COM device, you probably need to run CONFIGUR.COM from your MS-DOS distribution disk. Make sure the COM device matches that to which you have installed the Reader, then configure that device as a "WH-23/WH-33/WH-43 modem." It's option "F" in the configuration section of CONFIGUR.

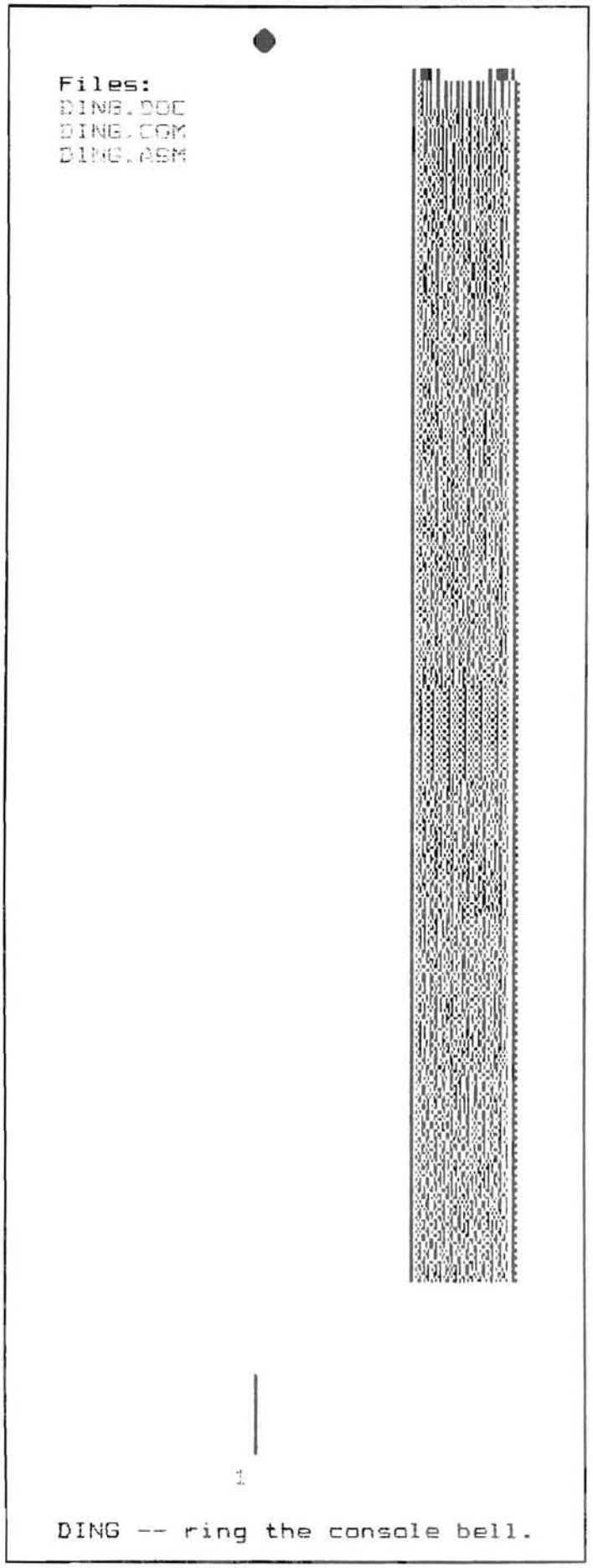
Once communications have been established, two new messages appear: "Reading strip #1" and "Reader countdown." A Softstrip page may have one or more Softstrips, and they may contain one or more files that comprise a "batch." "Batch" is my term for a group of files printed turned into Softstrips in a single run of the program that produces them. An analogy is the file groups that can be produced with archival programs such as ARC and LU. CAUZCOMM takes care of electronic details when separating out the files in a Softstrip batch: the program displays a list of all files and begins counting down from the total number of bits in that Softstrip. If there is only one Softstrip on the page and in the batch, you are finished and can do whatever it was you wanted to do with the resulting file. If there is more than one Softstrip, CAUZCOMM will prompt you to move the Reader to the next Softstrip: each is numbered at the alignment bar. You, therefore, take care of the mechanical work in moving the Reader as needed when there is more than a single Softstrip in a batch. It's neither hard nor tricky, but it is a nuisance and does take time. If The Softstrip System takes off, I would not be surprised to see Cauzin introduce a "production" Reader, completely automated to scan at least an entire page of Softstrips.

Printing Softstrips

Printing your own Softstrips is optional. There are three ways to do that right now.

If you own an Epson FX or RX series dot matrix printer, you can buy a Stripper package for $19.95: one is available for each kind of supported computer. You'll want the "IBM PC Series" version for a Z-100 PC or Z-200 PC. The slim package holds only an equally-slim 28-page booklet, which itself is evidence of the economies to be realized from The Softstrip System: the two programs needed to print Softstrips on an Epson, STRIPPER.EXE (the program that translates files into Softstrips) and DITBIT.EXE (a quality-checker for the Softstrips you print), are both printed as Softstrips, on pp. 16-18. Although you don't need a Reader to print Softstrips, you do need one if you want to check their quality — and to get these two programs into executable form so that there is quality to check. In the last analysis, you'll be silly to try printing Softstrips without having a Reader.

Stripper and an Epson — remember you need an FX or RX — allow a choice of two densities in printing Softstrips: normal density allows up to 844 bytes on each Softstrip; high density allows up to 1065 bytes. Each Softstrip has a header, so you can't simply divide the size of your files by those numbers to calculate how many Softstrips they will make. Stripper will calculate that and report as you give it a list of files to print. Cauzin could do customers a favor by providing a program that would allow really quick estimates outside Stripper.



There are two other ways to produce Softstrips. One is with a laser printer: my Stripper booklet says to call or write Cauzin for full details, which presumably will include information on just which laser printers are supported. That's not in the booklet. The booklet does mention an "on-site licensing arrangement" in connection with this option, which suggests to me a price other than the flat $19.95 charged Epson users. Interestingly, the booklet also says that so far all one can expect from a laser printer are "low to medium density" strips. I read that as meaning you are better off with an Epson than with a laser printer.

The second way to produce Softstrips is through Cauzin's own service bureau, which can produce negatives for offset printing — presumably from files you send on a floppy disk. I was quoted a charge of $20 per negative, which I am told will hold five or six Softstrips. It's a reasonable charge for production negatives, but check with Cauzin about security before sending any sensitive data. As a pretty good test of how much more data could be packed into Softstrips printed from a negative than into those printed on an Epson, I ran Stripper through itself. It took seven Softstrips on my Epson to reproduce the same 8KB program that took four Softstrips in the Stripper booklet.

Reliability And Error Checking

I wondered about how strict the Reader would be about alignment because the sight-and-lip arrangement on its plastic housing do not strike me as comprising a precision tool. Not to worry: the System is remarkably forgiving. When faced with situations it could not forgive — such as really bad misalignments and Softstrips out of sequence — it delivered a clear error message and encouraged a retry. After a while I tried reading Softstrips from magazines held in my lap. Even so there were no failures until I encountered a Softstrip printed too close to the magazine's gutter margin: when CAUZCOMM told me to try again, I flattened out the magazine until I got a successful read. No problem.

Nor were there any problems reading the Softstrips I made with my Epson FX-185, even though the ribbon is of middling age. So of course I took one of those pages to a heavily-used office copier, a Ricoh xerographic copier, and made some copies. The first two generations caused no problems at all: they were read the very first time. The third generation had CAUZCOMM complaining about — oddly enough — alignment problems. Maybe I could have fiddled around to get that copy read too; but since Cauzin says third generation copies won't work, I didn't bother to try.

What I did try next was copy protection: I used a red felt-tip marker to slash through the original a couple of times, then made a copy. The Reader had no problem with the rubricated original but simply could not read the copy — in which the red marks were reproduced black. Slick. I don't think this kind of copy protection can be defeated easily.

Next I crumpled the original into a ball and straightened it roughly by drawing it sidewise and lengthwise along the edge of a counter a few times. The Reader balked several times, but eventually I was able to straighten the paper enough to get it read. I think some judicious ironing would have sped the job. All I wanted, though, was to see if the Reader needed its Softstrips handled with kid gloves. It doesn't.

Nor is CAUZCOMM the least bit antic: I had feared that I might have to start over from the beginning if there was an error five strips into a six-strip batch. In fact the program is notably sensible: where errors are concerned, it handles each Softstrip as a discrete unit and simply has you reread that unit until it encounters no errors.

Softstrips have checksums in the headers, and CAUZCOMM uses them to ensure error-free reconstruction of the original files.

I did encounter one problem, though, and I think CAUZCOMM could have handled it better. Where a batch contains multiple files that span several Softstrips over several pages, CAUZCOMM's insistence on having all or nothing seems too uncompromising for my sloppy world. If the last Softstrip can't be read, the only option is to abort CAUZCOMM — which then erases everything in the files it had been building for that batch.

Since files are stored sequentially in a batch on Softstrips, I don't understand the need to lose good files at the beginning because of some insurmountable problem in one at the end. That penalty seems especially severe in the case of ASCII files, where — despite F. Scott Fitzgerald's lament — a few missing words might not mean so much if I could recover all the rest. I was bothered enough by this behavior to do some snooping in The Softstrip Reader System, which brought me to a solution that circumvents CAUZCOMM and permits using the Reader more flexibly with text files on both supported and unsupported computers.

Other Ways To Use The Softstrip Reader

Cauzin offers The Softstrip Reader System for IBM PC compatibles, the Apple II, and the Apple Macintosh, but you can use the Reader to read Softstrips on other computers, too. You'll be limited to reading ASCII files only, though, and you'll have to solve any problems yourself. Don't expect support from Cauzin or, for that matter, from me. Remember that Cauzin neither documented nor supplied the information that follows, and that I do not warrant it in any way. I developed it through experimentation and can't be held responsible if it does not work for you.

The Reader is DCE ("Data Communications Equipment") operating at a fixed speed of 4800 Baud. Connect it to the computer's RS-232 port as you would a modem on that computer. Since IBM PC compatibles use a male connector at the computer, the cable in The Softstrip System Accessory Kit terminates in a female connector. That's backwards for some computers, so on them you'll need a male-to-male RS-232 adaptor. You can buy one or make your own: the Reader uses only pins 2-8 and 20.

You will need a modem program that has the ability to capture data in a buffer and write it to a disk file. Most programs can. Set your program to the following parameters: 4800 Baud, no parity, 8 bit words, and 1 stop bit.

With the Reader connected and its power turned on, open the capture buffer and go to the modem program's "terminal" mode — the mode in which you can type characters from the keyboard to a modem. Align the Reader over the first Softstrip you want to read. Type "T" or "t" and press the "RETURN" key on the computer.

In about a second, the Reader should start sending characters. When the flow stops, align the Reader over the second Softstrip and send out another "T" or "t." Go on that way until all Softstrips in the batch have been read. Then close the capture buffer, save it to a file, and exit the program.

If the Reader has sent any error messages, they'll appear in English on the computer's screen and will be captured in the file. Error messages start this way: "**ERR - ."

You'll have to edit the capture file to remove them, the header at the beginning of the first Softstrip, and maybe the SUB characters (1AH) used for padding and to mark the ends of files — which is why you must not use this method to read binary files such as programs. Use it only for text files, only if you are absolutely confident about what you are doing, and only if your confidence is justified by your knowledge.

The Reader itself looks fragile but feels sturdy. Cauzin recommends treating it "with the same care you would a camera," because of the optics. I think the general idea is to use reasonable care: don't use the Reader as a hammer, don't poke things into the unprotected slot in which the lens assembly resides, and store the Reader in the plastic base provided with it.

Documentation

The Softstrip Reader System, The Softstrip System Accessory Kit, and Stripper each come with a slender book of instructions pitched at the most unsophisticated kind of user for each component — the kind who wants an appliance. That kind of user should be admirably well served by the clear, logical, and well-illustrated instructions, none of it the least bit threatening or intimidating. The list of error messages in The Softstrip System Reader Instruction Manual, for example, is headed "Reader Alert," lists each message, briefly explains the cause of the problem, and tells how to correct it.

Of course Murphy's First Law struck and the very first problem I encountered was an error condition not in the books: CAUZCOMM hung the computer while it displayed its "trying to establish communications" message. There was no explanation of what might be wrong. What was wrong was that I had configured the serial port for testing an odd peripheral, so resetting COM1 with CONFIGUR.COM straightened everything out. Maybe my problem is, by definition, one that would confront only people able to solve it without help. But I doubt that. Nevertheless, the documentation I've seen is among the best of its kind I've ever seen for the non-technical computer user. I want a technical manual

Should You?

The Softstrip System does absolutely everything it is supposed to do, and does it well. It's reliable, economical, and so easy to use that it seems more like an appliance than like a microcomputer peripheral. But it's too slow, at least in its present incarnation, for production work. For occasional use, though, the lack of speed may not be any inconvenience at all. It's for that kind of use that

Cauzin evidently is marketing the product. If you need what it does, there is nothing else like it. Even if you think you possibly could use what it does, the price of Cauzin's The Softstrip System might be low enough to justify the experiment. Besides, it's a Nifty Thing to have.

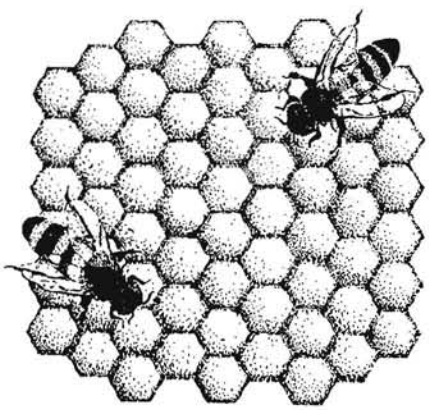
For Further Information

The Softstrip Reader System (Includes one Accessory Kit)	$199.95
The Softstrip System Accessory Kit (For IBM PC Compatibles, Apple II, or Macintosh)	$19.95
Stripper (For IBM PC Compatibles, Apple II, or Macintosh.)	$19.95

Cauzin Systems, Inc.
835 South Main Street
Waterbury, CT 06706
203/572-0150

*

Beyond IBM PC Compatibility:



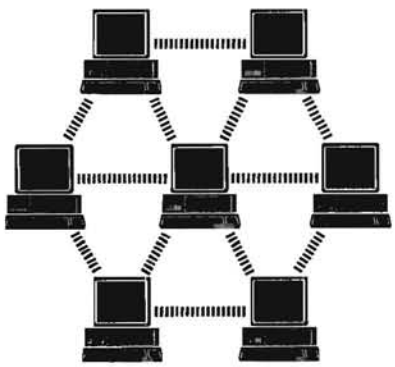
Z-100 users who are ready for PC compatibility, should keep one thing in mind. The future. New, state-of-the-art PC software is creating ever greater performance demands on computer hardware. Large scale, memory intensive programs for networking, graphics, number crunching and communications are quickly becoming the rule, not the exception. But software is only as good as the system that runs it.

EasyPC will run it all.

Even the newest copy-protected best sellers. That's because you actually boot up PC hardware, not software translation routines.

EasyPC runs faster, too.

Faster than an IBM PC, faster than a Z-150, and faster than any other emulation scheme you can buy.



EasyPC will support a fully integrated Local Area Network System (LANS) of Z-100's, PC's and AT's communicating at 115Kbaud.

Networking is now a reality for Z-100 users.

EasyPC will fully support a network of up to *sixteen* Z-100's, PC's and AT's. EasyPC is the core of an integrated system of components that includes RAM expansion, I/O, math co-processing, hard disk subsystem and more. For serious users, it's the only choice.

Ask to see an EasyPC in action.

Visit your Heath/Zenith Electronics Center or independent UCI dealer and see how easy it is to turn your Z-100 into a PC compatible system. For the name of your nearest dealer, just call 800-UCI-COMPUTER.

UCI CORPORATION

948 Cherry Street, Kent, Ohio 44240
(216) 673-5155 / 800-UCI-COMPUTER

WildFire: A Review

Kevin Lerch
Engineering Technician II
Zenith Data Systems
St. Joseph, MI 49085

On this rainy September evening I retire to the basement, put on one of my favorite albums, and fire up my NEW Z-161. A few days earlier my '161 was as slow and cumbersome as a dinosaur, and soon to become extinct. You see, I'm a technician at Zenith and I am forced to work with our Zenith products every day. Don't get me wrong, I love Zenith computers and use them exclusively at home and at work. The problem here is at work I use Z-158s and Z-248s, and after driving a Porsche, the old VW bug just doesn't cut it. I tried to sell the '161, but have you ever tried to sell a dinosaur? I don't enjoy being laughed at. So it came down to reinvest and lose my shirt or upgrade my '161. But then look at it this way, what happens when the new computer you buy gets to be 2 years old? The same thing.

Well, the other day I received a call from my wife. It just so happens that my wife is the Production Coordinator of REMark. She was wondering if I would be interested in giving my old dinosaur a kick in the pants, and all I had to do was write an article for the magazine. Great, one of my weakest subjects happens to be the written English language. The "kick in the pants" happened to be WildFire (a speed-up modification for the Z-151/152/161) from Software Wizardry. Alas, I agreed to the task. Just my luck the next day Lori (my wife) calls and says Joe Katz wrote an article on WildFire. Now what!! Well, I got a copy of Katz' article and read it (and I must say Katz did a great job as usual), but I have a slightly different angle on things than he does, being a technician. Besides, I called the oracle himself, Jeff Solliday (Engineer and Developer of WildFire), and got the inside scoop on the hardware.

I have a Z-161 unlike Katz' Z-152, the installation is pretty much the same as for a Z-151/152, and although the installation of WildFire is well engineered, Z-161 owners have a slight problem. You can't mount the switch panel on the front of the computer. When the panel is mounted on the front of the computer it gets in the way of the keyboard when it is in the mobile position. There is one solution to this small problem. Where the keyboard cable is stored, it is possible to mount the switch panel vertically. If you don't mind the reset switch behind closed doors.

I got WildFire installed and booted up my computer and got the message "Memory Parity Failure". The problem seems to vary intermittently like a hardware failure. After talking to Tom Jorgenson (Software Wizardry) and Jeff Solliday they assured me that they would help me solve my problem. Well, we think we got the problem nailed down.

Jeff told me that the system was designed to work 100% with 200ns memory chips and the failures could be caused by a speed sensitive memory chip. Jeff also said that the problem could be caused by a marginal 8087 co-processor. I have stock 200ns memory chips half of which I got from a salvage store, so I can't say the memory in my system is of flawless quality. They suggested I install RAM-PAL and 256k RAM chips that are much faster. Tom Jorgenson sent me RAM-PAL and 256k chips. Lo-and-behold that solved my problem. Now that I have 640k of faster RAM, I have not had a single RAM failure. I wrote this article and ran a lot of software on my '161 without any failures. It seems to be fixed. This is not to say that you can't run WildFire without RAM-PAL and 256k chips installed. Software Wizardry assured me I was one of a small minority of WildFire users to experience this problem.

When you install WildFire and it happens to give you "memory parity failure" messages, first of all suspect memory. This should be easy to find with Zenith Data Systems' Diagnostics. That will iso-

late the problem down to a bank of memory. If possible, either swap the whole bank of memory or rotate a good chip into the bank, one slot at a time. A memory board that has been working fine could show a speed sensitive memory chip when subjected to higher-speed signals.

Second, it could be a marginal 8087 co-processor. If memory is not your problem and you have a co-processor, take it out of its socket and test the computer again. But remember to move the switches on the CPU board to the correct position for no co-processor.

Now I'm going to talk a little bit about the hardware end of the WildFire system. When I was on the phone to Jeff he went through how the hardware works in the greatest of detail without giving away any proprietary information. I'll try and go through what Jeff told me. Here goes!

Jeff says he is doing 4 major things. Number One is, of course, speed up the clock. The WildFire system does not just replace the crystal with a faster one and hope that the rest of your system can handle the speed. You replace other surrounding ICs to cope with the new speed.

Number Two is he widens the clock pulses or the duty cycle of the processor clock. If you could watch the processor on an unmodified system, you would see the processor "miss" or skip clock beats at the rate of up to 1 in 10. Bingo, my math says you could lose up to 10% of your processor speed. By increasing the duty cycle of the clock, the processor would be less likely to miss clock beats. Also on the subject of the processor, the new V20 replacement processor is no ordinary V20. Jeff said that the V20 given to you in the WildFire upgrade kit is a special version. The new brain has a more enhanced internal clock gate. This will also discourage "missing" clock pulses.

Number Three is a bit touchy. Jeff said that he could not be real specific on how this is done. The principle here is to get everything synchronized, so that there is no wasted clock time. Basically, that's what he said. I know that's not real clear, but that's all I can say.

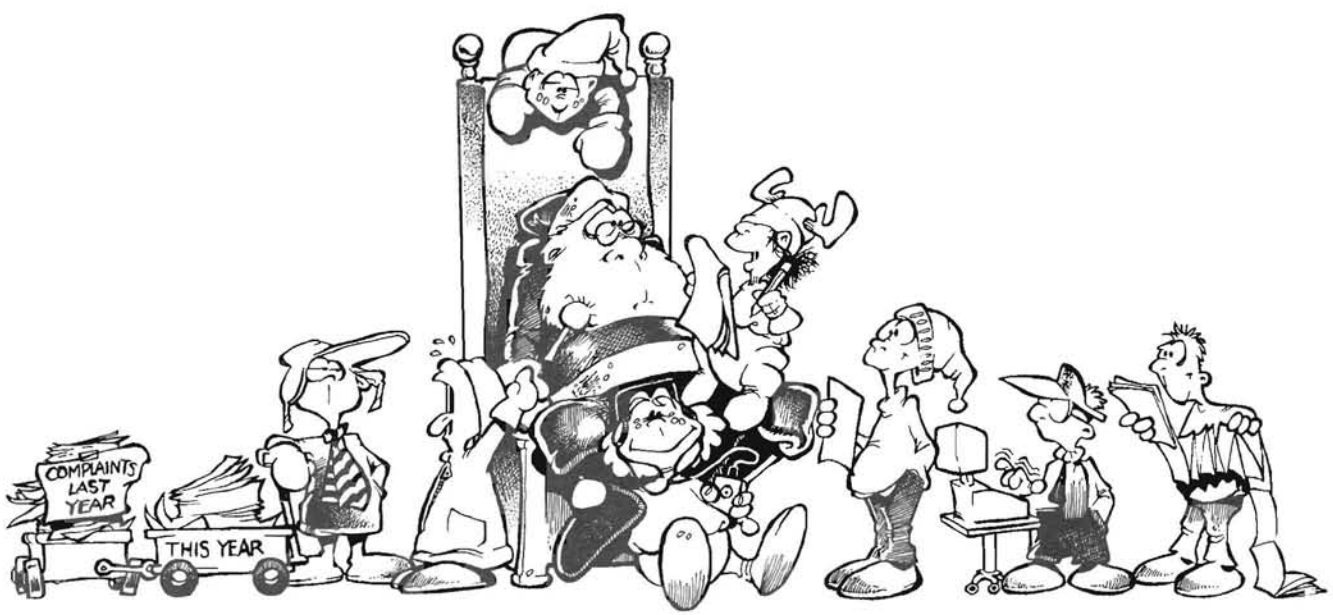
And last, but not least, Number Four. This one is somewhat subtle. Most people wouldn't notice, but most of the components that you remove and replace are TTL in nature. The ones you put back in are CMOS. There are two reasons for the CMOS parts. One is power consumption, and the other happens to be heat. Most folks don't want to heat their house with the exhaust from their computer. Zenith computers don't run hot, in general, but for instance, a TTL 8237 has 100,000 N gates inside that package, where a CMOS 8237 has 80,000 C gates. The C gates run much cooler and there are less of those little critters. There will also be less current drawn from your power supply, therefore, it will run cooler.

In general, I think that my computer runs much smoother and cooler. The video seems to flow instead of jump. To quote Jeff "all I'm basically doing is making your computer more efficient."

If you were thinking about selling your Z-150/161 because its too darn slow, don't bother, just buy yourself a WildFire upgrade kit. I wouldn't be without it now. My '161 rivals even my office machines and it saves me a lot of time and money. I ran a PRIME NUMBER benchmark and got a 33% increase in speed. If you desire more information on software, run times, and benchmarks, read Joe Katz' article in the November 1986 issure of REMark.

For more information on WildFire contact:

Software Wizardry
1106 First Capitol
St. Charles, MO 63301
(314) 946-1968
List Price: $249



The Home Security System Anyone Can Install

Dave Rye
P.O. Box 248
Northvale, NJ 07647

Crime prevention experts agree that the best deterrent against break-ins is to make your home look and sound lived-in. Most people agree that the best way to make your home look lived-in is to have lights and appliances going on and off when you're away. Okay, so you buy some of those clunky little timers and plug your living room lights into them, then you realize that to "really make your home look lived-in" you need a few more of those little critters, before you know it you have half a dozen of them sprinkled around the house. You feel secure in the knowledge that your home and family are protected while you're away, but what happens when the seasons change and it starts getting dark earlier — you have to go around the house resetting every one of those little things. Then you notice after you've had them for a while that they start to get noisy. There's nothing worse than sitting quietly trying to read a book with a constant hum in the background. So you think about replacing all your timers with those new "electronic" types, but the expense puts you off — well now there's a better way.

The ***X-10 POWERHOUSE*** Programmable Timer from X-10 (USA) Inc. controls up to eight lights and appliances by sending digital signals over your existing house wiring. You plug the lights and appliances you wish to control into "Modules". You set each Module to a different code which corresponds to the signals the Timer sends. You can purchase the Timer for as little as $30 and the Modules cost as little as $10 each. Setting the timer is like setting a clock radio and if you can plug in a light, you can install the system. Another problem with those "mechanical timers" is that you really have to dedicate a particular lamp to "making your home look lived-in" and forget about using the lamp for everyday uses. If you forget this and turn the lamp off manually — you've lost your security system! The advantage of the ***X-10 POWERHOUSE*** System is that there is a complete range of different types of controllers available allowing you (once you've plugged the lamp into a module) to control it automatically — from the Timer, manually — from push button Maxi or Mini Controllers located anywhere in your home, over the telephone — from X-10's Telephone Responder or even from your car or your back yard with the new Radio Controlled System — and you can still turn the lamp on from it's own power switch!

Once you've agreed that it's a good idea to have your lights go on when you're not at home, you think to yourself "I'd really like to have the outside flood lights, porch lights and driveway lights going on automatically". But they're controlled by wall switches so you can't plug them into a "timer". Simple — you just replace your existing wall switches with the Wall Switch Module from X-10, it's controllable from any of the X-10 Timers and Controllers and you can even dim it — in fact it installs just like a regular dimmer.

Not Just For Security

Security is just part of the story. You may initially purchase the ***X-10 POWERHOUSE*** System to make your home look lived-in, but you will soon come to appreciate how convenient the system is. Imagine going to bed at night without having to think twice about whether you left a light on downstairs, you fall asleep in a nice warm bedroom knowing that the central heating will automatically lower itself by about 15 degrees after you've fallen asleep (to save energy). Sure you could turn the thermostat down manually before you go to sleep but who remembers, and besides, you then wake up in a cold room in the morning. The ***X-10 POWERHOUSE*** System not only turns the heating down for you automatically, but can also turn it up for you just before you wake up.

If you are a light sleeper you may wake up when the bedside light goes on, but if you're a heavy sleeper you may not wake up until the TV turns on to Good Morning America or the stereo turns on to your favorite station. No more "tinny" sounding clock radios. For those who find it "too convenient" to be able to reach over and turn off the alarm before falling back to sleep, you can make sure that there is no controller within arms reach. On the other hand, for

those who find it convenient to be able to control the TV and/or stereo from the bedside there is the reassurance that comes from having programmed a "second ON time". In fact, you can have as many as eight different lights and appliances going on and off twice every morning. But don't use them all to get you up, you'll want to save one of the channels on the Timer to program the Appliance Module which controls your coffee pot so you can wake up to fresh brewed coffee.

Not everyone uses heated curlers! But if you do, I'm sure you'll agree, that you have to wait a long time for them to heat up in the morning. Now the ***X-10 POWERHOUSE*** Timer can turn them on before you wake up. When you leave the house for work you don't have to run around the house making sure that you have turned everything off. The Timer will do it for you. When you come home at night, the outside lights will be on so you can see to get your key in the door (but you didn't have to leave the lights on all day). The crock pot will have been on for the past hour for that special slow cooking stew, the house will be back at your comfort level even though the heating/air conditioning has been programmed to be off all day to save energy and your home will welcome you with a warm and comforting lived-in appearance.

The above scenario may sound a bit too "regimented" for some people, but it's surprising how quickly you get used to certain things happening at the same time every day. Just try taking the system out after a week and see the complaints you get from the rest of the family. Also, remember that there are Maxi Controllers and Mini Controllers available which let you override any programmed event at any time. In fact, you'd be surprised how convenient it is just to be able to control your lights and appliances remotely even if you don't purchase the Timer.

Manual Controllers For Convenience

Even without owning the Timer, you can turn the bedroom light on before you go upstairs — to light your way. You can turn on the electric blanket from your living room half an hour before you go to bed without having to run upstairs to do it. When you're tucked up in bed watching the Johnny Carson Show, you can press a button to turn the TV off when you feel like going to sleep. If you want to read in bed, but hear complaints like "turn that light off, I'm trying to sleep", you can easily turn off one bedside lamp and dim the other without even having to reach over the bed. When you are finally ready to go to sleep you can press one button to turn off everything in your home including any lights you may have inadvertently left on downstairs. If you wake up in the middle of the night feeling a bit too warm, you can reach over and turn on your window air conditioner without having to get out of bed.

Most Acts Of Violence Associated With Burglaries Occur As A Result Of The Intruder Being Disturbed

If you hear a frightening noise in the middle of the night, the last thing you should do is go downstairs to see what caused it. Most acts of violence associated with burglaries occur as a result of the intruder being disturbed. Instead of going downstairs to see what caused the noise, you can simply press one button to flood your home with light. You could also blast the stereo downstairs or even press "All Lights On" and "All Off" repetitively to flash all your lights. There aren't too many burglars who will hang around while all that's going on!

When you wake up in the morning, you can turn on the coffee pot from your bedroom so your coffee is ready by the time you have taken your shower. You can press one button to turn everything off before you leave for work or turn off selected items and leave your stereo on to make your home sound lived-in. All of these things can be done with the Maxi Controller which lets you control up to 16 lights and appliances at the touch of a button from anywhere in your home. The Maxi Controller has a Suggested Retail Price (S.R.P.) of $24.99. The Mini Controller does everything the Maxi Controller does but controls eight modules rather than sixteen, and has an S.R.P. of $11.99 so you can afford to have one in every room!

Control Your Home From Any Phone

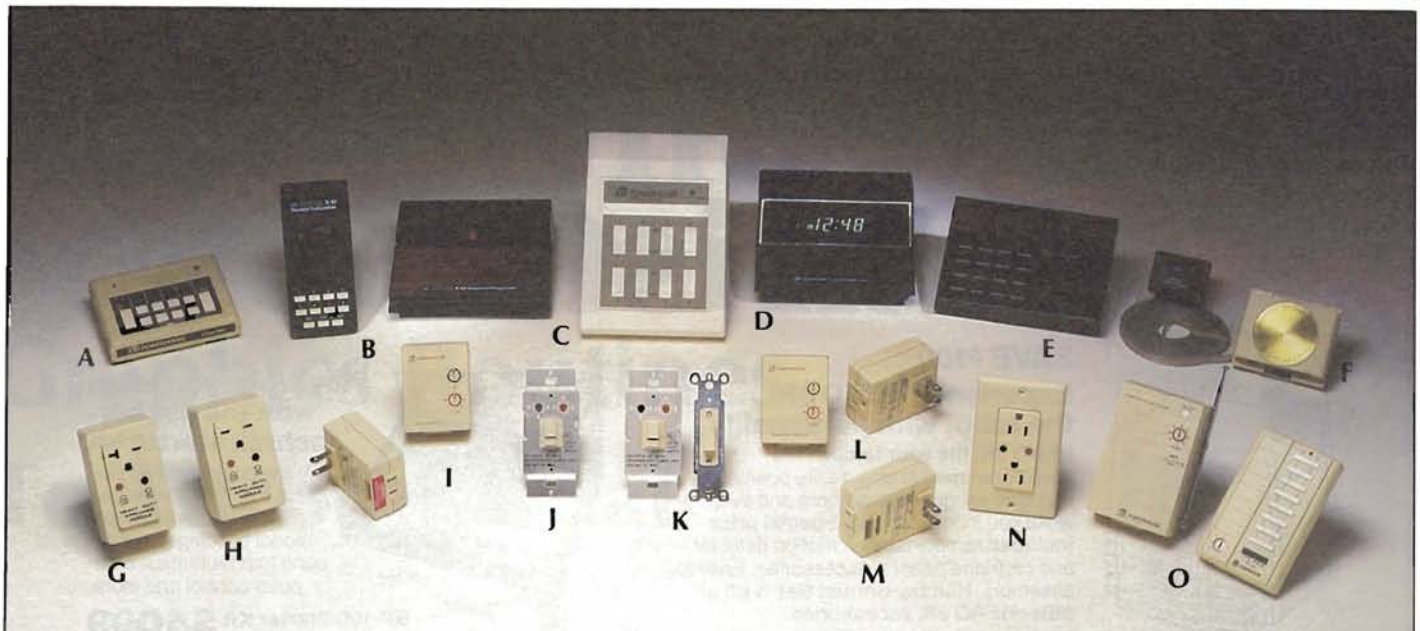
Imagine being able to control your lights and appliances from any phone in the world! You could, for example, call your second home and turn the central heating or air conditioning on so your home is comfortable for your arrival. Anyone who owns two homes *must* own the Telephone Responder! Even if you don't own two homes, have you ever left for your vacation and when you got to the airport thought to yourself "did I turn the coffee pot off before I left?" Instead of having your vacation spoiled by the worry that you may not have, you can call home and turn everything off just to be sure. You could also plug a window air conditioner into an Appliance Module or Wall Receptable Module (there are even 220V Appliance Modules available) and call home from the airport when you return from your vacation to turn on the air conditioner, so your home is nice and cool by the time you get there. The Telephone Responder can control up to eight modules over the phone, or manually from it's keyboard while you are at home. It has an S.R.P. of $49.99.

Save Energy (Save Money)

You can easily save enough money to pay for your complete ***X-10 POWERHOUSE*** System by just setting back you central heating or air conditioning by about 10 degrees when you go to bed. To do this you will need the Thermostat Setback Controller (TH2867). This neat little device mounts on the wall directly under your existing Thermostat and plugs into a regular Appliance Module or a Split Receptacle Module. When you turn the module on it supplies a small amount of local heat under the thermostat which "fools" it into thinking that the room is hotter than it really is. Your thermostat, therefore, turns your heating off (or your air conditioning on). The problem with most setback thermostats is that they replace your existing thermostat and are difficult to install. The instructions might say "connect the red and the green wires to the black and yellow wires", for example, but when you remove your old thermostat you find that you have two pink wires! The really neat thing about the TH2867 is the ease with which you can install it. You don't need to touch your existing thermostat wiring and it works with ANY kind of thermostat. Another problem with setback thermostats is that they can be difficult to program. The TH2867 is controlled by any ***X-10*** controller so the only programming necessary is to set the ***X-10 POWERHOUSE*** Timer. The TH2867 includes an Appliance Module and has an S.R.P. of $39.99.

Computer Control

Home Computers are for the home (hence the name) so naturally they should be able to *control* your home. Even before their popularity as a consumer product, hobbyists used computers to control their homes, but this used to be done the "expensive way" by hard wiring the lights and appliances to be controlled, to relays which were directly controlled by the computer. This approach did not appeal to Mr. Average. Since the introduction of X-10 technology, boards have been available which were installed inside the computer and transmitted signals to X-10 Modules.



A - Mini Controller
B - Telephone Responder
C - Home Control Interface
D - Programmable Timer
E - Maxi Controller
F - Thermostat Setback Controller
G - 220V 20Amp Appliance Module
H - 220V 15Amp Appliance Module
I - Lamp Module
J - Wall Switch Module
K - 3 Way Wall Switch Module
L - 3 Pin Appliance Module
M - 2 Pin Appliance Module
N - Split Receptacle Module
O - Radio Controlled System

These products, however, were severely limited by requiring that the computer be left on permanently. Most of them also dedicated the computer to the task of controlling your home so you couldn't use it for anything else!

Now you can use your Home Computer to control anything in your home without tying up the computer or having to leave it turned on and you don't need to know anything about programming. If you can plug in a light you can control your home! The Home Control Interface (S.R.P. $69.99) is programmed from an Apple, IBM, Commodore or Macintosh computer. You program the Interface from your computer using the easy to use software (included) and then disconnect the interface from the computer. It doesn't tie the computer up, you can turn the computer off or use it for something else. The Interface will continue to control up to 256 modules (software dependent, varies with computer brand) whether or not it is connected to the computer. It has it's own real time clock and battery back-up to protect the time and program during the event of a power outage of up to 100 hours. The Interface can be programmed to turn modules on or off at different times on different days so your stereo won't wake you up on weekends. You can program it to turn lights on at a specific brightness level at a specific time, so your living room lights can automatically dim to 50% brightness at the same time your tape deck turns on to romantic music (remote controlled seduction). X-10's dimmers can even be dimmed automatically (or manually) in groups! So the *whole room* can gradually dim to 70% brightness at a pre-programmed time.

Radio Control

For the ultimate in convenience and security you can now control lights and appliances from inside or outside your home, from your car as you pull into your driveway or from your back yard while you're sitting by the pool. You can do this with the Radio Controlled Transmitter/Transceiver set (RC5000). The Transmitter works like a garage door opener and will control whatever you plug into the Transceiver — from inside or outside your home. It has eight on/off buttons and a bright/dim button and controls lights and appliances by sending radio frequency (r.f.) signals to the Transceiver which then retransmits them over your house wiring to control up to eight standard X-10 modules. The system works without additional X-10 Modules, so if you don't own any X-10 Modules the TC5000 lets you control one light or appliance from inside or outside your home. If you already own standard X-10 Modules (or purchase them later) you can control up to eight lights and appliances. You can also add another RC5000 to your system to control up to sixteen lights and appliances. The Transmitter/Transceiver set (RC5000) has an S.R.P. of $49.99.

The ***X-10 POWERHOUSE*** System is the most versatile security system available, combining security, comfort and energy savings with ease of installation. There is a complete range of Controllers and Modules available for virtually any application. The system is available under many private brand names, all of which are manufactured by X-10 and all of which are totally compatible. Visit your nearest Heath/Zenith store for more details. *

Unique Gifts for Christmas

RTR-1-6

SAVE $100

Experience HERO JR., the robot with personality

HERO JR. fits your family. He's preprogrammed to sing, recite poems, play games, guard your home and even wake you in the morning. **Special price includes** remote control, motion detector and cartridge adapter accessories. Easy to assemble. **Holiday Bonus! Get ½ off all other HERO JR. accessories.**

KIT ONLY $199.95

BP-100

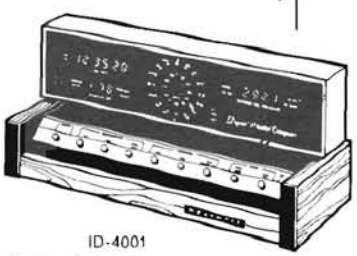
Play while you learn with the fischer-technik starter kit

A knowledge-building play system that grows with your child. Starts with simple model building – can expand to pneumatics, computer control and more.

BP-100 Starter Kit $49.99

Predict the weather with our Digital Weather Computer

It measures wind chill factor, indoor/outdoor temperature, wind speed and direction plus barometric pressure with microprocessor-controlled accuracy. Full line of weather instruments also available.



ID-4001

KIT ONLY $399.95

A gift for the whole family! Cometron Reflector Telescope

View the rings of Saturn and the moons of Mars. This super 4.5" reflector telescope features precision German optics in 40 and 111 power.

KIT ONLY $299

TL-400

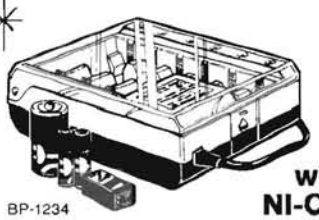
HS-148 with HVM-1220-A Monitor

*HUG members may take 10% off this price

IBM PC Compatible Heathkit Compact Personal Computer with monitor

Runs on MS-DOS operating system (included) and includes 256K RAM and 5.25" disk drive with 360K storage. Has both monochrome and color capability.

KIT ONLY $899*



BP-1234

Keep your batteries at their peak with this NI-CAD Charger

A convenient and quick way to recharge your batteries. This easy-to-use charger handles up to 13 batteries at once, including ten "AA," "C," or "D" batteries in any combination and three 9-volt transistor batteries.

ONLY $24.99

Heathkit Starter Kits for the beginner

Enter the exciting world of kitbuilding with easy-to-build Heathkit Starter Kits from as low as $4.95. We have a wide variety of kits for the beginner, from stereo synthesizers and audio amplifiers to speaker phones and fish callers. This new experience in product ownership is waiting for you today at your nearest Heath/Zenith Computers & Electronics Store.

LEASING AVAILABLE FOR YOUR BUSINESS NEEDS

Phone orders accepted.

Prices good through January 15, 1987.

Call 616-982-3614 for store locations or consult your Heathkit Catalog.

Casio Black & White Pocket TV
ONLY
$59.95
Quantities Limited

ZENITH data systems
AUTHORIZED DEALER

HEWLETT PACKARD

EXPERTS: CAD NETWORKING COMMUNICATIONS

HZC-264

Your TOTAL SERVICE computer center

• Service • Support • Software • Accessories • User Training • Competitive Prices

Heath® ZENITH®
Computers & Electronics

Units of Veritechnology Electronics Corporation

Transferring "MAILPRO" Files To Other Computer Systems

Kirk L. Thompson
#6 West Branch Mobile Home Village
West Branch, IA 52358

In the first installment of this series (REMark, November 1986), I discussed a program for changing the field layout of data files created by Clark Systems' "MAILPRO" mailing list system. As I noted there, I've been using this inexpensive package for some five years as a data file manager.

The program, MPFIELD, was an outgrowth of a project I completed in the fall of '84 for my employer, using "MAILPRO" to create, edit, and print a large database. The purpose was to compile a list of tools used in the Molding Department and organized by inventory part number. When completed, the project totaled 459 records of 61 bytes and 12 fields each, spread over three files. As this series progresses, you will get some feel for the problems I encountered using Clark Systems' package on my 64K H-89 in an industrial environment. An environment it wasn't really designed for!

Moreover, the order of publication here is the chronological reverse of the order of development (with some minor exceptions). MPFIELD was written after the conclusion of the project, but based on knowledge I acquired during it. The program I will discuss now, MPCONVRT in Listing 1, was the last program developed for the project.

The Final Problem. The last task in my employer's database project was to transfer the final data files from my H-89 to a Hewlett-Packard Model 9816. This is a 16-bit machine running one of the early, integrated, spreadsheet/database/word processing/telecommunications packages, "CONTEXT MBA". As is usually the case with older micros, the disk formats were completely incompatible, particularly since the H-P uses Sony's 3-1/2 inch microdrives. Another hardware problem was, in fact, a limitation in the destination software. The communications section of "CONTEXT" expects a modem at the RS-232 port, so high-speed serial-to-serial transfer was out of the question. But the biggest headache was the basic incompatibility between "MAILPRO's" random access files and "CONTEXT's" format. I perceived early on that, when converting from one type of file to another, you should know where you are coming from and where you are going to!

If you TYPE a short "MAILPRO" .DAT file to the screen, or better, examine one with a screen dump utility, you will notice several interesting things. Figure 1 is such a dump, using Mike Cogswell's "UDUMP". The first thing to observe is the total absence of new-line characters (10hex under HDOS) anywhere in the file. The second is that the data fields and records are simply run together like beads on a string. These two are typical of HDOS's and MBASIC's random file structure. A third, however, is characteristic of "MAILPRO".

This peculiarity is that the ORDER of the fields in the file are the REVERSE of the order in which you entered them. Figure 2 illustrates the normal order of entry. The reason for this is the way in which the program fields each record. Line 1000 in the listing is typical:

```
1000 FIELD#1,T%*I% AS B$,B%(12) AS B$(12),B%(11) AS B$(11),@
     B%(10) AS B$(10),B%(9) AS B$(9),B%(8) AS B$(8),B%(7) AS @
     B$(7),B%(6) AS B$(6),B%(5) AS B$(5),B%(4) AS B$(4),B%(3) @
     AS B$(3),B%(2) AS B$(2),B%(1) AS B$(1)
```

Notice that the subscripts for the numeric and string arrays are in descending order as the line is scanned from left to right. This order is the way the fields are stored in the file, even though you entered them in ascending order.

So, any conversion program I might write would have to reverse this order since it was of some significance for data entry and use.

```
                        Editing SY2:DRWHO.DAT
   Track:     13                                          File Sector:    Ø
   Sector:    12                                          File Length:   66

        Ø  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  Ø123456789ABCDEF
       -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  ----------------
ØØØØ:  34 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 6F 7Ø 65 6E 69 6E  4         openin
ØØ1Ø:  67 2Ø 73 65 72 69 61 6C 3B 61 6B 61 2Ø 54 68 65  g serial;aka The
ØØ2Ø:  2Ø 54 72 69 62 65 2Ø 6F 66 2Ø 47 75 6D 2Ø 2Ø 2Ø   Tribe of Gum
ØØ3Ø:  2Ø 2Ø 44 72 2Ø 57 68 6F 2Ø 61 6E 64 2Ø 61 6E 2Ø    Dr Who and an
ØØ4Ø:  55 6E 65 61 72 74 68 6C 79 2Ø 43 68 69 6C 64 2Ø  Unearthly Child
ØØ5Ø:  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 41 6E 2Ø 55 6E 65            An Une
ØØ6Ø:  61 72 74 68 6C 79 2Ø 43 68 69 6C 64 2Ø 2Ø 2Ø 2Ø  arthly Child
ØØ7Ø:  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 41 2Ø 2Ø 2Ø 37 2Ø 2Ø 2Ø          A   7
ØØ8Ø:  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 66 69 72 73 74 2Ø 44 61 6C 65        first Dale
ØØ9Ø:  6B 2Ø 73 65 72 69 61 6C 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø  k serial
ØØAØ:  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 44 72                  Dr
ØØBØ:  2Ø 57 68 6F 2Ø 61 6E 64 2Ø 74 68 65 2Ø 44 61 6C   Who and the Dal
ØØCØ:  65 6B 73 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø  eks
ØØDØ:  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 54 68 65 2Ø 44 61 6C 65 6B 73        The Daleks
ØØEØ:  2Ø 28 61 6B 61 2Ø 44 65 61 64 2Ø 5Ø 6C 61 6E 65   (aka Dead Plane
ØØFØ:  74 29 2Ø 2Ø 42 2Ø 2Ø 2Ø ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ  t)  B   ........
```

Figure 1: A screen dump of a "MAILPRO" random access file. This particular sector contains two records, with the break between them (controlled by the software) just before the "7" at address 07C hex. Also note that HDOS pads out the sector with ASCII nul characters (00 hex) at addresses 0F8 hex through 0FF hex.

```
TOP
 1 (Serial)  -- A
 2 (Title)   -- An Unearthly Child
 3 (Bk Title)-- Dr Who and an Unearthly Child
 4 (Comment) -- opening serial; aka The Tribe of Gum
 5 (VCR loc.)--
 6 (# of episodes)-- 4
This is record # 1 .

 Database `syl:drwho' currently has 131 record(s).
 Key field is # 1 --`Serial'.
 Ø addition(s) already made.  5 more allowed

Command?
```

Figure 2: A display screen from "MAILPRO's" UPDATE subprogram for the same file and first record shown in Figure 1, addresses 000 hex through 07B hex. The new "TOP" command and some of the other enhancements to "MAILPRO" shown here will be discussed later in this series.

```
SeriTitle                                Comment
A    An Unearthly Child                  opening serial;aka The Tribe of Gum
B    The Daleks (aka Dead Planet)        first Dalek serial
C    The Edge of Destruction             smtms "Beyond the Sun"
D    Marco Polo
E    The Keys of Marinus                 6-part
F    The Aztecs
G    The Sensorites
H    The Reign of Terror
J    Planet of Giants                    1st of 2nd season
K    The Dalek Invasion of Earth         Susan leaves
L    The Rescue                          Vicki arrives
```

Figure 3: The first dozen lines of the same file in Figures 1 and 2 after conversion by MPCONVRT, but with three fields excluded. By comparing this with Figure 2, you can determine which fields were.

But, more importantly, the program would also have to create an output file compatible with the software at the destination end.

Having files in immediately-useable form on the destination system is, I think, preferrable to doing the entire conversion there. In fact, doing the conversion on the H-P would have been next to impossible because of the absence of reasonable software development tools. But in the case of "CONTEXT", conversion proved to be easy. "CONTEXT" can break a sequential text file into columnar blocks for its standard, spreadsheet-like display. So my conversion program would simply have to concatenate the fields in each record and output this long string to a sequential file.

Some of the other higher-powered database and spreadsheet packages in the 16-bit world also have this ASCII conversion capability. In those cases, MPCONVRT may work fine as written. However, if you are moving your "MAILPRO" files to ANY other system, you should first determine the format of files used by the destination software. The simple concatenation algorithm I settled on might not work for you, so you may have to design and write your own.

The Final Solution. With the major questions answered, I was able to write the conversion program. This is, in fact, the easiest and simplest of the programs being presented in this series. For that reason, I see no point in offering source code and documentation as I did with MPFIELD. Like its predecessor, it assumes an H/Z-19/89 terminal and was written specifically for "MAILPRO" version 127.1.

As with the refielding utility, when the program signs on, it asks for source and output file names, checking for the presence of each and prompting you if there is a problem. After the .CHA and .PAR source files are read, you are presented with a menu of the fields in each record and asked if you wish to exclude any from the sequential file.

This was an option I deliberately included. I was a bit extravagant when I created the original files because I included more fields than were eventually needed for the database. By including this option, I could exclude these extraneous fields from the data transferred to the Hewlett-Packard. If you should choose this option, you are prompted for how many fields to exclude and for which ones. You respond to the latter by keying in a number from the field menu. These can be in any order. In fact, if you decide to exclude fewer fields than you originally input, you can select the same field number more than once. You cannot, however, "unexclude" one you have already selected. If you find yourself in this situation, you will have to abort the program with Ctrl-C and start from scratch with "RUN".

When you have completed the field exclusion process (or completely skipped it), data processing begins. However, the very first string sent to the sequential file isn't file data at all. To aid in column alignment and field identification at the destination system, the field names are truncated, or expanded with spaces, to fit their field lengths and concatenated. Then the records in the source file are read and processed.

Throughout this latter operation, you are informed of what the program is doing and where it is in the source file. After the data conversion is complete and the files closed, the program terminates. Figure 3 illustrates part of a file so converted.

The Transfer. With the program written and debugged, there were only two more steps to perform in the file transfer problem. The first of these, and the easiest, was to process the three data files. Then came the actual transfer, and for that I lugged my Neanderthal '89 (since it dates back to 1979), its outboard soft-sectored drives, and modem into work.

After setting up the hardware, I discovered (to my chagrin) that my 300-baud UDS modem wouldn't work — it derives its power from the telephone lines and the Hayes Smartmodem 1200 connected to the H-P simply doesn't supply that kind of thing! A few phone calls later, however, I was able to borrow a 300-baud Hayes from a local computer dealer.

To control the transfer at my end, I settled on the software package, ZLYNK, because of the full set of configuring options it has built in. But configuring the H-P end of the operation took half an hour of experimenting because of its complexity and the poor quality of the documentation. However, once the protocol was determined, transferring the 35 kilobytes of files took less than half an hour. The only glitch was that "CONTEXT" dropped the first one or two characters in each file, but since these, fortunately, were parts of the column header, they weren't of major consequence.

In any case, the data in the files was put to immediate use. The printouts (the problems I had generating these will be discussed in a later installment) were photocopied and are still used daily in two of the plant's departments. The data transferred to the H-P was used by an industrial engineer for cost and productivity analysis. In summary, my employer benefitted because the data was rationally organized for the very first time and I benefitted (and you through me) from the knowledge of "MAILPRO" gained and the useful utilities written.

Coming Attraction. I noted above that MPCONVRT is the easiest of the "MAILPRO" utilities I'm presenting. In the next installment, I'll discuss the most difficult, and what happens when you run out of memory and you still have one-third of the information yet to key into the datafile!

Acknowledgment: My thanks to Ansley Standard Products, EC Division, Thomas & Betts Corp., Iowa City, IA, for funding the work. Their patience was invaluable as I struggled through some interesting problems.

Software Sources

UDUMP — Studio Computers, 999 S. Adams, Birmingham, MI 48011

ZLYNK — Software Wizardry, 1106 First Capitol Dr, St Charles, MO 63301

Listing 1

```
10 REM                    MPCONVRT.MBS
20 REM                    Version 1.1
30 REM                         by
40 REM                    Kirk L Thompson
50 REM                    West Branch, IA
60 REM                    December, 1985
70 REM
80 REM Copyright (c) 1985 by Kirk L Thompson
90 REM This program is placed in the public domain for
       non-commercial use only
100 REM The author retains copyright to the source code
110 REM
120 REM VARIABLE TABLE:
130 REM ================================================
140 REM         A$      General string prompt answer, usually
                 yes or no
150 REM B$       Source file dummy string
160 REM B$()     Source file FIELD string array
170 REM B%()     Source file field lengths
```

```
180 REM C       Source file FOR/NEXT loop record number
190 REM CD$     Cursor down (ESC B)
200 REM CH$     Column header string for sequential file
210 REM CS$     Clear screen (ESC E)
220 REM CU$     Cursor up (ESC A)
230 REM EEL$    Erase to end-of-line (ESC K)
240 REM ERV$    Enter reverse video (ESC p)
250 REM F       Number of fields to exclude from sequential
                file
260 REM FLD     Field number to exclude
270 REM FLD()   Boolean field-to-exclude array
280 REM J       General FOR/NEXT loop index
290 REM L1%     Source file longest field name length
300 REM N$()    Source file field name array
310 REM N%      Number of fields/record in source file
320 REM O$      Output string written to sequential file
330 REM OF$     Output file name
340 REM PA$()   Source  PAR file field string array
350 REM Q       Number of records in source/output files
                (less one)
360 REM RCP$    Return to saved cursor position (ESC k)
370 REM S%      Number of records/sector in source file
380 REM SCP$    Save cursor position (ESC j)
390 REM SF$     Source file name
400 REM T%      Record length in source file
410 REM TB      Tab position
420 REM XRV$    Exit reverse video (ESC q)
430 REM
440 CLEAR 1000:DEFINT A-Z:CS$=CHR$(27)+CHR$(69):
    ERV$=CHR$(27)+CHR$(112):@ XRV$=CHR$(27)+CHR$(113):
    SCP$=CHR$(27)+CHR$(106):RCP$=CHR$(27)+CHR$(107):@
    EEL$=CHR$(27)+CHR$(75):CD$=CHR$(27)+CHR$(66):
    CU$=CHR$(27)+CHR$(65):@ FALSE=0:TRUE=-1
450 PRINT CS$:PRINT:PRINT STRING$(33,32);ERV$,
    "M A I L P R O";XRV$:PRINT:? PRINT STRING$(17,32);
    "F I L E   C O N V E R S I O N   U T I L I T Y"
460 PRINT:PRINT:
    DIM N$(12),B%(12),B$(12),O%(12),O$(12),FLD(12)
470 REM
480 REM***get & check source file name***
490 REM
500 PRINT:INPUT
    "Enter source filename (dvd:fname)[.DAT assumed]--";SF$
510 IF INSTR(SF$,".")>0 THEN PRINT ERV$;
    "Extension not allowed!";XRV$:GOTO 500
520 PRINT:PRINT"Is ";ERV$;SF$;XRV$;:INPUT" OK (Y/N)";A$
530 IF LEFT$(A$,1)="N" OR LEFT$(A$,1)="n" THEN 500
540 PRINT:ON ERROR GOTO 1230
550 OPEN"I",#1,SF$+".DAT":CLOSE
560 REM
570 REM***get & check output file***
580 REM
590 PRINT;INPUT
    "Enter output file (dvd:fname)[.DAT assumed]--";OF$
600 IF INSTR(OF$,".")>0 THEN PRINT ERV$;
    "Extension not allowed!":XRV$:GOTO 590
610 PRINT:PRINT"Is ";ERV$;OF$;XRV$;:INPUT" OK (Y/N)";A$
620 IF LEFT$(A$,1)="N" OR LEFT$(A$,1)="n" THEN 590
630 PRINT:OPEN"I",#2,OF$+".DAT":CLOSE:PRINT:PRINT ERV$;?
    "File already exists!";XRV$:GOTO 590
640 REM
650 REM***read source file .CHA &  PAR files***
660 REM
670 PRINT"Reading source  CHA and  PAR files."
680 OPEN"R",#1,SF$+" PAR"
690 FIELD#1,1 AS PA$(0),2 AS PA$(1),2 AS PA$(2),
    2 AS PA$(3),2 AS PA$(4),@ 2 AS PA$(5),16 AS PA$(6),
    16 AS PA$(7),16 AS PA$(8)
700 GET#1,1:S%=CVI(PA$(1)):T%=CVI(PA$(2)):N%=CVI(PA$(3)):
    L1%=CVI(PA$(4)):@ Q=CVI(PA$(5)):CLOSE
710 OPEN"I",#1,SF$+".CHA"
720 FOR J=1 TO N%:LINE INPUT#1,N$(J):INPUT#1,B%(J):
    NEXT J:CLOSE
730 REM
740 REM***field exclusion selection***
750 REM
760 PRINT CS$;TAB(27)"FIELDS (LENGTHS) IN RECORD"
770 PRINT TAB(15) STRING$(50,61):TB=14
780 FOR J=1 TO N%
790 IF J>(N%+1)\2 THEN PRINT RCP$;CD$;SCP$;:TB=0
800 PRINT TAB(TB) J;"  .";N$(J);
    " (";MID$(STR$(B%(J)),2);")";
810 IF J=1 THEN PRINT TAB(40) CU$;SCP$
820 PRINT:NEXT J
830 FOR J=1 TO 12:FLD(J)=TRUE:NEXT J.
    REM initialize boolean array
840 PRINT:INPUT
    "Do you want any of the above fields EXCLUDED (Y/N)",
    A$
850 IF LEFT$(A$,1)="Y" OR LEFT$(A$,1)="y" THEN GOSUB 1090
860 REM
870 REM***data transfer***
880 REM
890 PRINT CS$;"Transferring data.  "
900 OPEN"R",#1,SF$+".DAT":OPEN"O",#2,OF$+".DAT"
910 PRINT:PRINT "Creating output file column header   "
920 CH$="":FOR J=1 TO N%
930 IF FLD(J) AND (LEN(N$(J))<B%(J)) @
    THEN N$(J)=N$(J)+STRING$(B%(J)-LEN(N$(J)),32) @
    ELSE N$(J)=LEFT$(N$(J),B%(J))
940 IF FLD(J) THEN CH$=CH$+N$(J)
950 NEXT J:PRINT#2,CH$
960 PRINT:PRINT "Transferring record: ";SCP$
970 FOR C=0 TO Q:PRINT RCP$;EEL$;C+1;"of";Q+1
980 REM set up, field, & read source record
990 RS=C\S%+1.I%=C-S%*(RS-1)
1000 FIELD#1,T%*I% AS B$,B%(12) AS B$(12),B%(11) AS B$(11),
     B%(10) AS B$(10),@ B%(9) AS B$(9),B%(8) AS B$(8),
     B%(7) AS B$(7),B%(6) AS B$(6),@ B%(5) AS B$(5),
     B%(4) AS B$(4),B%(3) AS B$(3),B%(2) AS B$(2),@
     B%(1) AS B$(1)
1010 GET#1,RS
1020 O$=""
1030 FOR J=1 TO N%:IF FLD(J) THEN O$=O$+B$(J)
1040 NEXT J:PRINT#2,O$
1050 NEXT C
1060 CLOSE:PRINT:PRINT
     "End of RANDOM to SEQUENTIAL file conversion "
1070 END
1080 REM
1090 REM***exclude fields subroutine***
1100 REM
1110 PRINT:INPUT "Exclude fields (Y=<RETURN>/N)";A$:PRINT
1120 IF LEFT$(A$,1)="N" OR LEFT$(A$,1)="n" THEN 1210
1130 PRINT "How many fields (1 to";N%;:INPUT ")";F
1140 IF F<1 OR F>N% THEN PRINT:PRINT ERV$;
     "ERROR--number out of range!";@ XRV$:GOTO 1130
1150 PRINT TAB(10)"SELECT FIELDS TO EXCLUDE:"
     FOR J=1 TO F
1160 PRINT:INPUT "Which field no  from menu";FLD
1170 IF FLD<1 OR FLD>N% THEN PRINT:PRINT ERV$;
     "ERROR--number out of range!";@ XRV$:GOTO 1160
1180 PRINT "Exclude ";ERV$;N$(FLD);XRV$;
     INPUT " (Y/N)";A$
1190 IF LEFT$(A$,1)="Y" OR LEFT$(A$,1)="y"
     THEN FLD(FLD)=FALSE
1200 NEXT J
1210 RETURN
1220 REM
1230 REM***disk file error trap & handler***
1240 REM
1250 IF ERR=53 AND ERL=550 THEN PRINT ERV$,
     "File does not exist!";XRV$:? RESUME 500.
     REM source file not found error
1260 IF ERR=53 AND ERL=630 THEN RESUME 650:
     REM output file not found error
1270 ON ERROR GOTO 0
```

On The Leading Edge

Christmas Gifts, Ability, DSBACKUP, SmartNotes, Mace+ Utilities, HADES

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

Christmas is just around the corner, and I thought it might be a good idea to talk about some gifts for computer owners that cost under $100. This article will discuss a number of suggestions for gifts (or maybe you will want to buy one for yourself). If you need to drop a subtle hint to your Santa Claus, you can do so by circling the appropriate item on the list at the end of the article. And I think that you'll agree that some of these suggestions will need to be added to your "gotta have" list whether you get them for Christmas or not.

Because there are so many neat things to look at, it's difficult to know where to begin; so I guess we'll just jump in and see what comes up.

A Visit To The HUG Underworld

This visit has been provided courtesy of the "King of the Computerized Raspberries" — also known as HUG's own Jim Buszkiewicz. Jim has agreed to be our guide to the netherworld, and as such, has written the HUG Absolute Disk Editing System, otherwise known as HADES. If you have any interest in exploring the DOS disk formats or need to "fix" a disk, this is THE program for you. It allows you to visit the underworld of just about any standard disk format that you're likely to encounter in the world of MS-DOS or PC-DOS. That includes floppy disks of all varieties — 3.50", 5.25", and 8" — as well as hard disks. The distribution disk includes versions for both the Z-100 and the PC series. I have personally tried HADES on a PC/XT, and had no problem.

This program allows you to edit anything on a disk. It is an extremely powerful program that provides sector-level editing for any sector on a disk. You can "look" at the contents of each sector by simply paging through the disk. And you can change the disk contents using either ASCII or hex input — a nice feature. That also means that, since you can access the File Allocation Table (FAT) and the disk directory, you can easily destroy them which may irrevocably cause the loss of a file or a complete disk of data.

You can also edit a specified file on a sector-by-sector basis. That is a useful feature if a file has been damaged somehow, perhaps as a result of a bad sector. HADES can even change file attributes if you are interested in such things.

HADES is the most powerful and useful disk editor that I've ever seen. As such, a word of caution is in order. I already mentioned that you can easily destroy a disk through lack of knowledge of the DOS format; however, you can still use the program to view the contents. You don't have to change anything. Jim tells me that he will have a limited version (included on the distribution disk) that does not allow writing to a disk.

HADES has some rather subtle goodies that deserve "honorable mention". As I was working with HADES, I noticed that one of its particular features is on-screen hellp. You don't really need the documentation (I haven't seen it yet) since you can always go to hellp as part of the program — a highly useful feature. That should not be any big surprise since any underworld program should include some kind of hellp — think about it. The program has all kinds of "sound effects" including a "raspberry" when you make a mistake. That's why I mentioned that Jim is the obvious choice for "King of the Computerized Raspberries". Pat Swayne is a close runner-up for the title since Jim tells me that Pat was responsible for a similar sound effect on the Z-100 version.

I understand from a highly reliable HUG source (Nancy Strunk in this case) that the documentation for HADES has just been completed as I write this. Based on what she tells me, I claim no originality for some of the following comments. HADES is one of those programs that you may want to get, although you may have no immediate use for it. When you do need it though; you may thank heavens for HADES.

I think that HADES is an outstanding program that is extremely useful for disk-level editing. Although it's true that the program may not be for everyone, it certainly is one of the nicest HUG

programs available for its intended purpose. If you need a program like this, HADES is highly recommended.

If You Have A Hard Disk . . .

As I've mentioned before, one of the benefits of owning a personal computer is increased productivity. One of the unfortunate side effects is that you can also make mistakes more quickly and completely erase all data — many megabytes of it — in a matter of a few seconds.

When I lost all of the data on my hard disk as I was attempting to do some testing for last June's article, my first thought was: "Oh, no". That was quickly followed by a few (well, maybe more than a few) unprintable words. The good news is that I had complete and current backups, so it was just a matter of time to restore the data.

Sometimes hard disks fail without any warning whatever. There are quite a few documented cases where early IBM ATs with hard disks were powered on and could not bring up the system — total, complete, and unexpected failure. Loss of power and general mistakes are the most common causes of hard disk problems involving a significant loss of data.

Although it's always nice to say that you SHOULD take a full hard disk backup occasionally, and KEEP it current, some will say that the DOS BACKUP and RESTORE commands simply take too much time. Few computer users have the time to recreate lost data even if they can remember the contents. It is even more important for business users to keep current backups because time is money. From a convenience and time standpoint, it is clear that some kind of tape backup system is probably the way to go. Unfortunately, tape backup systems are relatively expensive, and costs can range from a minimum of $700 up (mostly up).

When I was doing the research and testing for the October article on MS-DOS, the '241, and the hard disk, I thought I would try to find some kind of software that provided a better way to create backups for my '241. As it turned out, I found a super backup utility that is not copy protected and is reasonably priced.

Hard Disk Backup Software

Although there seem to be lots of utilities that can be used to back up a hard disk, I spent some time reviewing the available market. Like most products that I review, I take a look at the results that have already been documented by other sources. When I began looking for backup software, it happened that PC Magazine had just come out with an article in the August 1986 issue on "Software Safety Nets for Hard Disk Data". My selection of DSBACKUP for this article was based in large part on that review, and I was very pleasantly surprised.

But first, some comments on how I select software to discuss in this column. In some cases, I receive software from vendors who are interested in my review of their product. In others, I read a lot of material to look for items that I think would be of interest to you. Since I have neither the time nor resources to objectively evaluate EVERY product on the market, I spend considerable research time before I request a specific product. I suspect that most of you do the same thing.

I rejected the hardware solution to hard disk backup simply because I believe that it is too expensive to be of general interest. By the time you get everything you need for a tape backup system, you will have spent around $1000 or so. And you can also spend quite a bit more. That's a little high in my opinion to be of significant interest to most home and some business computer users.

PC Magazine reviewed a total of 12 products. Three of them were copy protected, and because of that, I rejected them out of hand. Even though copy protection is always burdensome, I think it is absolutely ridiculous to have a hard disk backup utility that is copy protected. That seems to defeat the purpose of a utility that purports to help you preserve data, especially megabytes of hard disk data.

DSBACKUP

As I mentioned earlier, DSBACKUP was a very pleasant surprise from several points of view. Perhaps the most important aspect is that it was actually much faster on my '241 than I had expected it to be. On the average, it took about 90 seconds to back up one megabyte to a 1.2 MB floppy drive. By way of comparison, the DOS BACKUP command took nearly five times that long for the same partition and backup floppy drive.

Restoring files also demonstrated a similar performance improvement over the DOS RESTORE command. DSBACKUP restored files at an average rate of about one megabyte in 110 seconds. Again, the DOS RESTORE command took about five times that long. By the way, these times were simply measured with a regular watch, and I did not attempt to do any sophisticated time measurements. I also did not include the actual time that it took to change floppy disks in the backup drive since that will obviously vary.

I saw a similar performance improvement when 360K floppies were used for the backup drive, but I spent most of the time changing floppies. When using DSBACKUP for the backup and restore, it took 20-30 seconds per disk — about what I expected based on the 1.2 MB disk performance. During that testing, it seemed like I spent most of the time changing floppies.

Like most utilities of its type, DSBACKUP is menu driven and is quite easy to learn. But it has the added feature that you can also completely run DSBACKUP from command line input so that it can be included in a batch file. This is a terrific feature that allows the maximum amount of flexibility.

DSBACKUP is configured by means of a setup file created by setting the various options in the menu and saving the setup under a file name. That means that you can have LOTS of different configurations depending on the type of backup or partition that you want. For example, you might have a setup file for a complete hard disk backup, a backup of files that have been updated since the last complete hard disk backup, and other special backups for your applications. You can also define setup files for the restore process too. The setup file concept, with the command line or menu driven processing, makes DSBACKUP a particularly effective and fast way to back up a hard disk.

And there certainly is no shortage of options in DSBACKUP. There are toggles that allow you to specify verify, prompt, hidden file backup, report generation, subdirectory backup, archive bit reset, and update bit reset. From the main menu, you can set the backup file specifications, source and destination directory, execute DSBACKUP backup and restore commands, run the DOS FORMAT program, and save and load the setup file. Online help is available and is quite good.

Since I can't resist exploring just about everything in software, I also came across another pleasant surprise in one of the documentation files on the distribution disk. DSBACKUP also is available in an ANSI version which means that it should run on the

Z-100 MS-DOS. I called the folks at Design Software, and they will send me the ANSI version since they did not know for certain about the Z-100. My assumption (and theirs) is that it will run just fine, but I'll let you know about that in a future column.

DSBACKUP comes in a unique plastic "folder" that is just slightly larger than a 5-1/4" floppy disk which includes the manual, software registration card, and distribution disk. The manual has about 100 pages which is a result of its small size. I found the manual to be comprehensive, and it has a number of examples showing the screen displays that you can expect to see.

The people at Design Software tell me that a number of other products are under development. Perhaps the most interesting is that I was told that they have improved the performance of DSBACKUP. I tested the PC version 2.4e on my '241 — they are currently shipping 2.4f. In addition, I understand that there will be a faster version available that cuts the current DSBACKUP time by about one third — a megabyte a minute or so depending on how fast you can change floppy disks.

All things considered, my guess is that you will probably be able to cut the DOS backup time to at least one third of the current value when disk changes are considered. The only thing that I found that was somewhat confusing was that the source and destination drives retain the same labels during the backup and restore. It's somewhat disconcerting to see the source drive as C when the DSBACKUP restore is processing. I recognize that the advantage of that scheme is to allow the same setup file to be used for a backup and restore, but I still think that maybe the menu labels could be changed depending on which function is invoked. That is perhaps being too nit-picky on a program that is as well thought out as DSBACKUP.

DSBACKUP is a program that no hard disk owner should be without. It is a time saving and cost effective way to do a hard disk backup. It is not copy protected so it can reside on your hard disk with no problem. Based on price versus performance, DSBACKUP represents one of the best software values on the market today. It is HIGHLY recommended.

SmartNotes

Occasionally I find something that is so spectacular that I believe it is a "required" acquisition for all computer owners. SmartNotes fits that category. Once you try it, you'll wonder how you ever got along without it. SmartNotes is the computerized equivalent of the Scotch brand Post-it notes — the ubiquitous yellow notes with the special glue on the back.

A SmartNote has a computerized "glue" that can be attached to just about any displayable output on your CRT. This concept is so unique that Personics Corporation has applied for a patent on it. After experimenting with it, I have become convinced that it is absolutely essential for recording information on just about anything. SmartNotes is an all-purpose note that can be attached to DOS file names, spreadsheet cells, programs (any language — even BASIC), data base files (e.g. dBase III), and any document generated by a word processor. It is one of the most versatile programs that I have ever seen.

SmartNotes is memory resident and takes about 90K of memory when activated. Although I'm not normally a fan of memory resident programs because of potential conflicts with application programs, I've had no problem with SmartNotes. And it's fast. Unlike some other memory resident programs, SmartNotes does NOT have to be loaded last, and it includes a command to remove SmartNotes from memory.

The program is menu driven and of course includes an editor that automatically word-wraps the text. You can also define two sizes of SmartNotes: 5 lines of text or 10 lines of text. Once SmartNotes is installed in memory, you press CTRL-F5 to enter SmartNotes and ESC to exit back to your program or DOS. Since that key sequence may conflict with some application programs, SmartNotes includes a configuration option that allows you to change that to CTRL-LEFT SHIFT-HOME. Although that sounds a little awkward, it really is easy to get used to. I had to change SmartNotes to use that activation key since WordPerfect uses the CTRL-F5 sequence.

SmartNotes' electronic glue normally "attaches" itself to the context of the text or display using the previous 47 characters. That means that you must be cautious when changing things since a SmartNote can be "lost". Not to worry. SmartNotes has a way you can see ALL notes just in case. One of the nice features of SmartNotes is that you can change the context for the electronic glue which can be a minimum of 5 characters. That is very useful for attaching notes to file names in a DIR display. Since SmartNotes are saved in a special file, there appears to be little chance that they will cause any problem with your data or other displays.

When using SmartNotes with subdirectory file names for DIR, I found it most useful to define the file name in the form of directory.SNS. My SYSTEM subdirectory, which contains all of the MS-DOS programs, has a SmartNotes file of SYSTEM.SNS. By the way, SNS is the recommended file extension since it makes the SmartNotes files easy to find.

If I had a choice of one single utility for my system, SmartNotes would be it. I have seen no other utility that provides the functionality and usefulness that SmartNotes has. It is definitely one of those "gotta have" programs. Unfortunately, it is only available for the PC compatibles and does not run on the Z-100. SmartNotes is HIGHLY recommended.

ace+ Utilities

As I mentioned in my October column, there are many ways to lose data on a hard disk. Aside from the fact that all prudent hard disk owners will make periodic backups of the files, there is another way to recover from most errors and catastrophes: the MACE+ Utilities. The front cover of the manual makes some rather spectacular claims. Lest I understate the capabilities of this software, I will quote from the cover page of the MACE+ Utilities manual.

"UNFORMAT will restore ALL your subdirectories and files automatically on any hard disk. UNDELETE gets erased files back with four keystrokes and without cross-linking. REMEDY automatically moves files to a safe place and locks out the bad spots. RECLAIM automatically extracts files from unreadable disks. CONDENSE unfragments up to 32 MB. SQUEEZE/SORT in a single operation: speeds up the PATH by squeezing deleted references from directories, and sorts your directories on any field."

When I received the software, I had tested everything but the UNFORMAT on my '241 hard disk. And everything worked as stated above. Then I got a call from my department chairman at UTA. He told me that someone had accidentally formatted the 20 MB hard disk on the department's PC XT. Could I help? That seemed like the ideal opportunity to really put the MACE+ Utilities to the acid test and perform a useful job at the same time.

To make a long story short, MACE+ Utilities performed exactly as advertised and recovered nearly all of the files on the hard disk.

That was IMPRESSIVE. For technical reasons, it is usually not possible to recover files in a root directory, so those files were lost. Fortunately, there is not much problem in recreating most root directory files like COMMAND.COM and AUTOEXEC.BAT.

But if you implement MACE+ Utilities BEFORE you format your hard disk, there is a way to recover ALL files on a formatted hard disk. A neat little utility, called RXBAK, is added to the AUTOEXEC file so that it is executed every time the system is booted. As far as I can tell, this program simply creates a backup of the disk directory and the File Allocation Table as a file that can be used to recover ALL files on a hard disk. That overcomes the technical problem that I mentioned earlier about not being able to recover files in the root directory since there IS a root directory in this file. It's amazing how simple ideas can be the best.

If you have a hard disk, you owe it to yourself to get a copy of the MACE+ Utilities. It's just a matter of time before you will need it. If you have spent the money and time to implement a hard disk, I think that MACE+ Utilities is a required item. If you are planning to buy a hard disk for your PC compatible system, include the cost of the MACE+ Utilities in your budget — it's just as essential as a hard disk controller. MACE+ Utilities is HIGHLY recommended for all hard disk owners.

There is an epilogue to this story. It turns out that the user who formatted the hard disk was using PC-DOS version 3.1. Like the old version 1 of Z-DOS, PC-DOS will cleverly format the default drive when you do not specify a drive letter in the command line. No prompts asking what drive is to be formatted at all — just press RETURN and you have formatted the hard disk or whatever the current drive is. That's how the hard disk was accidentally formatted. The moral of the story is to ALWAYS specify the drive letter in the command line when FORMAT is used. Then it doesn't matter which operating system you are using since you know that drive will be formatted without relying on defaults. If you have to operate a number of MS-DOS systems like I do, you will find that it really pays to use the "safe" way all of the time.

Ability

One of the very puzzling things about buying a microcomputer is the awesome array of software that is available. Which word processor and spreadsheet should I buy? That kind of software represents a substantial investment, and the wrong choice can lead to extensive frustration with computers in general.

Ability is one of the integrated software packages that provides just about everything that you will ever want. Word processing, spreadsheet, data base, communications, and graphics are included in the package. And it's still just under $100! For $20 a program, it's difficult to imagine a better deal.

These programs are not just for beginners either. They provide powerful capabilities that are not what you would expect from such an inexpensive program. While they probably can't equal a dedicated word processor or spreadsheet, they provide just about everything that a home computer user is likely to ever need. And some of the capabilities are not available until you begin talking about the $600-700 integrated programs.

For example, you can create and edit a document with Ability's word processor. Then, you can paste in a part of the spreadsheet to add some figures to the document. If you need graphics, you can also paste the graphics into your document. But let's say that you need to adjust the figures in the spreadsheet. No problem. Adjust the figures as usual, and they will also be adjusted in your document. That is a really powerful feature which is the forte of integrated software.

Even if you already have one or more of the programs that is part of Ability, you will find that the others provide capabilities that you don't currently have. And the integrated features are something that you will have to see.

On a price versus performance basis, it's tough to believe that you can find a better bargain. In the event you cannot find Ability at your local Heath/Zenith Electronics Center, you can order Ability from one of the Centers in my area at the address listed. Ability is a recommended addition to your software library.

WindowDOS

There is nothing more difficult than trying to manage a lot of disk files without appropriate software help. One nice program that will help you manage your disk files is WindowDOS. It is most useful with a hard disk because of the number of possible files, but I have found it to be quite useful with a floppy disk system too.

WindowDOS is a memory resident program that provides a sorted directory listing of ALL files in the specified directory. That means you can see things that are not displayed by the DIR command such as hidden files.

Did you ever try to remove a directory with the RMDIR command and get the "Invalid path, not directory, or directory not empty" error message. You checked the command line and it was OK. You checked the directory with the DIR command and no files were displayed. Yet, you still could not delete that directory.

The usual explanation for that problem is that there is still a hidden file in that directory. That file was probably created by some copy protected software since that is one of the less sophisticated of the copy protection tricks. And since the DIR command can't "see" the file, there is no way to ERASE it.

WindowDOS will allow you to see the file and change the attributes so that you can ERASE the file. In fact, WindowDOS displays all attributes of each file so that you know its current status. You can add, change or delete subdirectories, as well as erasing one or more files in a subdirectory. In addition, WindowDOS allows you to mark files (with a plus sign) for multiple file operations such as COPY or ERASE.

WindowDOS is quite useful for disk file management and is reasonably priced, as well. And by the way, if you combine WindowDOS with SmartNotes, that's a dynamite way to identify and manage files. It's a recommended addition to your software library.

"Fixing" The Z-100

As I was wrapping up this article, I received a letter from Dick Munroe that I found particularly interesting. He says: "Since installing the new motherboard (on the Z-100) I have not been able to run CP/M-85. Programs will run briefly and then I get an error message "Bus or memory parity error." I tried to run the programs on CP/M-86, but get the same result. I do not have a V20 in my Z-100, but I do have an 8087 installed."

While that is a specific problem with a specific computer, I thought it might be a good idea to look at some troubleshooting techniques that you can use on your computer in general. Just in case you haven't checked, getting a computer (any computer) "fixed" is not cheap. If you don't happen to live near a repair center, the problem is compounded since you may have to pack up your

system and ship it to be fixed. That is generally a pain, not to mention expensive. There are some ways to save money since Heath provides one of the best diagnostic tools available.

If you have any Heath or Zenith computer, the diagnostic software is one of the best investments you can make. In many cases, it was shipped with the computer, but it was not included with the Z-100 when I bought it about three and a half years ago. The diagnostic software was included with my '241. Since the diagnostic software is relatively inexpensive at $79.00, you may find that solving just one problem will pay for the software. Lest you think that is a frivolous expense, consider the idea that it may cost $100 just to have a technician LOOK at a computer and fix it by replacing a single IC. The best news is that you don't have to be any electronics expert to fix many of the common problems that microcomputers have.

Back to Dick's problem. My guess is that he probably has a bad memory chip on the motherboard. But I'm really getting ahead of the problem. The first trick in troubleshooting is to ISOLATE the problem based on the assumption that it is hardware related and has nothing to do with the software. That isn't a unique approach to computer problems, it's a general technique used in any mechanical problem — computers, cars or whatever.

The idea is to begin removing hardware until the problem disappears. In Dick's case, that means pulling the card that contains the 8087. Fire up the system and see if the problem continues. If it does, the next step might be to pull the additional memory boards. Since Dick has the new motherboard, he should not have that problem since all memory should be in 256K chips on the motherboard. Continue the process until you get down to the "bare" system hardware. Check the system EACH time you remove a board to see if the problem disappears. If it does, the LAST board you removed probably has a defective component. And I think most people would rather send a board for repair instead of their entire system. This works MOST of the time assuming that there is only one failure that is not caused by a combination problem with another piece of hardware.

One of the reasons that I HIGHLY recommend the Heath diagnostic software is that you may not have to resort to "stripping" the system to locate the problem. The diagnostics perform a sufficiently detailed check of the hardware so that a specific problem can usually be isolated with no difficulty. That is, the software may report that U-109 (memory chip) is bad or whatever. All you need to do is replace that chip and with a little luck, the problem is solved. And although the diagnostic software won't identify EVERY possible problem, it will help you locate most of the ones that will end up saving you some money and time. You may also find that one or more components are "marginal" and should be replaced anyway.

Let's assume that you are now down to the bare system with motherboard, disk drives and monitor. You still haven't been able to isolate the problem since you don't have the diagnostic software. Since Dick mentions that he runs the Z-100 AutoCad, I suspect that he has added two banks of 256K chips so that he has the maximum of 768K RAM in his system. The next step, assuming that you don't have the diagnostic software, is to start pulling memory chips. That is specific based on the type of message he saw. If I were doing it, I would pull both of the new banks of chips since the motherboard has been factory tested. My assumption is that the existing 256K on the motherboard is good based on those tests. That should make the problem disappear in his case. If it doesn't, I would then replace the first bank of chips with one of the other sets that I removed. If THAT doesn't fix the problem, you have gone just about as far as you can go with the hardware.

All of that is really tedious which is the specific reason that I like the diagnostic software. Although the ROM based diagnostics are good in the Z-100 and spectacular on the PC series, they won't test everything. The diagnostic disks test about everything on the system including video memory, disk drives, and RAM. There are special diagnostic disks for each system, and you must have one for each major system type. There are three basic diagnostics disks: one for the Z-100, one for the PCs (e.g. Z-151), and one for the Z-200 PCs. Each diagnostic disk is self contained and bootable, so you don't need anything else.

Preventative maintenance is always a good idea. I run the diagnostics disks occasionally (about every three or four months) just to check the system. That is especially true for the rotation speed of the disk drives. The rotation speed can be particularly critical since it may PREVENT your being able to exchange disks with someone else even though you may see no problem on your system. In addition, some forms of copy protection DEPEND on the rotation speed, and the install program does something that will NOT allow you to run the software on any drive that has a different rotation speed than the "installed" one. Moreover, it is a good idea to know what to expect from the diagnostics software BEFORE you have a problem. And if you use your computer, you will have a problem at some point. Mechanical problems (e.g. disk drives) and electronics failures (due to age and heat) are the most common.

I thought this would be a useful addition to this article since it provides another suggestion for computer owners for a nice piece of software under the $100 limit that I mentioned before. It is HIGHLY recommended to the point that I believe that it should be a personal requirement for your software library. And if you don't get the diagnostics disk for Christmas, I suggest that you consider getting it anyway.

Other Goodies

There are any number of other goodies that are quite useful for your system. S&K Technology has WatchWord and PC WatchWord if you need a good all-purpose editor and word processor. And if you want a good spelling checker, Resident Speller, PC Resident Speller, and Strike are highly recommended.

Barry Watzman has the neat Perks desktop utility for the Z-100 and has released PC Perks which I'll take a look at next month.

And if you're really out of ideas, there are always boxes of floppy disks and memory chips for expansion. Don't forget that each bank of RAM requires nine chips.

Closing REMarks

There seems to be a lot of confusion about computers and their model numbers in the Heath/Zenith line. I understand that HUG gets letters from readers wondering, for example, why the H-148 is not mentioned in articles. We'll take a brief look at the entire line of Heath/Zenith computers next month.

I expect to receive a copy of PC Perks in time for next month's column, so we'll also take a look at that.

I'll be glad to answer any questions about information in this article if you enclose a stamped, self-addressed envelope with your letter.

Products Discussed

HUG Software

HADES $ 40.00
Heath/Zenith Computer Centers
Heath/Zenith Users' Group
Attn: Nancy Strunk
Hilltop Road
St. Joseph, MI 49085
(616) 982-3571 (HUG Software only)

Software

MS-DOS Version 3
PC only (OS-63-31) $150.00
Z-100 only (OS-63-30) 150.00
Disk Diagnostics
Z-100 only (CB-463-13) $ 79.00
Z-150 only (CB-5063-28) 79.00
Z-200 only (CB-3163-31) 79.00

Hardware

Advanced Personal Computer (HS-241) $2499.00
Monochrome/Color Video Card (Z-409) 239.00
20MB Winchester (ZD-200) 1499.00
40MB Winchester (ZD-400) 2499.00
H-100 Desktop Computer (HS-1108-41) 999.00

Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-7057 (Heath Catalog orders only)

Ability $99.95
Mr. Don Murray
Heath/Zenith Computer Center
12022-C Garland Road
Dallas, TX 75218
(214) 327-4835 (VISA and Master Card accepted)

Mr. Dave Koslowsky
Heath/Zenith Computer Center
6825A Green Oaks Road
Fort Worth, TX 76116
(817) 737-8822

DSBACKUP $ 69.95
Design Software
2 N. 520 Prince Crossing Road
Suite #16
West Chicago, IL 60185-9990
(800) 231-3088

MACE+ Utilities $ 69.95
Paul Mace Software
206 Alicia Ave.
Ashland, OR 97520
(503) 488-0224

SmartNotes $ 79.95
Personics Corporation
2352 Main Street
Building Two
Concord, MA 01742
(800) 445-3311

WatchWord (Z-100 Only) $100.00
Resident Speller (Z-100 MS-DOS) 100.00
PC WatchWord (PC Series) 99.95
PC Resident Speller (Z-150/PC) 99.95
S & K Technology
4610 Spotted Oak Woods
San Antonio, TX 78249
(512) 492-3384

WindowDOS $ 49.95
WindowDOS Associates
Box 531650
Grand Prairie, TX 75053-1650

Variable Names

James Roy Davenport
P.O. Box 344 S.W.C.C.
AL-JUBAIL, SAUDI ARABIA 31951

Those of you who are methodical in your planning and record keeping need read no further but those of you who, like me, sit down at the keyboard and start banging away hoping something wonderful will happen please read on.

After purchasing my Heath Z-150 PC and Microsoft GW Basic, it wasn't long before I found that VNs (Variable Name) could be up to 40 characters long. The first character can be A-Z all others can be A-Z or 0-9 that would allow 26+(39^36) unique VNs. It would seem that there would be no end to the VNs that could be thought up — WRONG. I'm forever finding I have used the same VN in two or three places in the program, forgetting what VN I have used and where I used it. Some times a variable will change for no apparent reason and I can't find the line that's doing it.

To solve the problem I have written a little program to find all the VNs, the lines they are used in and the memory address where that line is stored.

The SEARCH (VNA.BAS) program will find all VNs and the line numbers where they are used, then print a hard copy with the VN followed by all line numbers where that VN is used.

Strange as it may seem the only thing that Microsoft Basic stores in memory in ASCII format is VNs, String Variables and two command words 'AS' and 'BASE'. Every thing else is stored as a one to eight byte code. PRINT is stored as a one byte code '145'. SIN(is stored as the two byte code '255 132'. Numbers 0 to 9 can be store as the codes 17 to 26. Numbers 0 to 255 can be stored as code '15' followed by one byte containing the number. The number 200 could be '015 200'. Numbers from -32768 to +32767 can be stored as the code '29' followed by a four byte code for the number. Numbers from 2^-127 to 2^+127 can be stored as the code '31' followed by an eight byte code for the number. I say "can" because Basic will try to store numbers so they use up the least memory but can be forced to save a number in another format just by adding a ! or # after the number. Some numbers are changed to negative by placing the code '234' before them; -200 could be '234 015 200'. Others change to negative by changing the first bit of the next to the last byte. There have been many good articles in REMARK and SEXTANT which can give you more information on how Microsoft Basic stores numbers.

When a line of program is entered, Microsoft Basic stores it in memory formatted as follows. The first two bytes are the MEMORY ADDRESS for the start of the next line, LSB (Least Significant Byte) first. The next two bytes are the LINE NUMBER, LSB first. Next is the line text followed by the line terminator which is always 0.

When a program is SAVED in Compressed Binary Format (Save" YOURFILE.BAS") the first byte of the file is always 255 followed by an exact copy of the program as it is stored in memory.

I was very careful to number the lines by 10's so you can use AUTO to enter all lines. To copy the program, just type AUTO followed by RETURN and start typing the program. Nothing is as aggravating as to think a program is numbered by 10's and one line is skipped or off by 5. If it is not seen, there can be a lot of lines to retype. I didn't number the remark lines so you can type them in leave them out as you wish.

I have tried to put in enough remarks so that the text could be short. All remark statements are in lower case as much as I could and marked with an * so they would stand out.

I know there must still be some combinations that I have not found so you may get an odd print out that is not a VN, but I don't think it will skip any true VN. Please let me know how you like the program and if you come up with any patches.

Running time to search itself: Interpreter 4:02 compiled 0:40

Search Program (VNA.BAS)

* NOTICE LINE 10 MUST BE THE FIRST LINE IN THE PROGRAM ENTERED JUST AS IT IS BELOW WITHOUT ANY CHANGES OR THE PROGRAM WILL NOT WORK. DO NOT PUT ANY REMARK OR OTHER LINES OF TEXT OR PROGRAM BEFORE IT.

```
*** Definition of major variables used

*** B          = Character code this BCOUNT
*** B$         = Buffer for Variable Name until completed
*** BCOUNT     = Byte COUNT - Pointer to Byte being tested
*** BUF100$    = String variable used to access data in file
*** FILENAME$  = NAME of FILE under test
*** NLSTART    = Next Line START memory address
*** RCOUNT     = Record COUNT
*** REC        = Points to one past last VNAME in array
*** PVN$       = Previous Variable Name in PRINT routine
*** PLN$       = Previous Line Number in PRINT routine
*** SF,SF1     = String Flag - SET when String Variable found
                 RESET after last character of String Variable
                 or end of line - SF1=NOT SF
*** BUF200$    = Group of 200 Bytes from file under test
***        This may give you trouble if you write  programs
           with  lines longer then 100  characters   If  so
           then  write me I know how to fix it but it  will
           make the program much longer
*** TLLEN      = This Line LENgth
*** TLNUM      = This Line NUMber
*** TLSTART    = This Line START memory address
*** VNAME$     = TLSTART+TLNUM+B$
*** VNAME1$(   = Array holding all Variable Names found
*** VNF        = Variable Name Flag - SET when Variable Name
                 found RESET at last character of Variable Name

10 A$="H" X=VARPTR(A$) TLSTART=(PEEK(X+1)+PEEK(X+2)*256)-8

*  Line 10 will automatically set TLSTART to the starting address
   of user program area  SEE NOTE ABOVE
*  Dim  array  to hold VN 500 should be  large  enough  for  most
   programs change it if you will have more then 500 VN

20 CLS:DIM VNAME1$(500)
30 BCOUNT=2:REC=0:BUF200$=""

*** routine to open file to be tested
40 LOCATE 10,1:PRINT STRING$(79," "):LOCATE 10,5
50 LINE INPUT "INPUT FILE NAME (YOURFILE.BAS) ";FILENAME$
60 LOCATE 12,1:PRINT STRING$(79," "):LOCATE 12,1
70 OPEN "R",#1,FILENAME$,100
*  does file exist
80 IF LOF(1)>0 THEN GOTO 130
```

```
90 PRINT "FILE ";FILENAME$;" DOES NOT EXIST"
*   close and kill file created by OPEN statement
100 CLOSE #1
110 KILL FILENAME$
120 GOTO 40

*** file found routine - FIELD it - GET first 200 bytes
130 RCOUNT=1
140 FIELD #1,100 AS BUF100$
150 GOSUB 870:GOSUB 870
*   first byte of file must be 255
160 IF MID$(BUF200$,1,1)=CHR$(255) THEN GOTO 210
170 COLOR 15:LOCATE 12,20
180 PRINT "THIS IS NOT A CONDENSED FORMAT BASIC FILE"
190 COLOR 7:CLOSE 1
200 GOTO 30

*** start of search routine
210 '
220 SF=0:SF1=1
*   gosub to compute TLNUM and NLSTART
230 CLS:GOSUB 610
*   compute this line length
240 TLLEN=NLSTART-1-TLSTART-4
250 PRINT USING "######";TLSTART,.PRINT USING "######";
    TLNUM;
260 PRINT "  ";
270 FOR LOOPA=BCOUNT TO BCOUNT+TLLEN
280 B=ASC(MID$(BUF200$,LOOPA,1))
*   if this is part of string then get next byte.
290 IF SF=1 AND B<>34 THEN GOTO 520
*   if start of string then set SF then get next byte
300 IF B=34 THEN SWAP SF,SF1:GOTO 520
*   if 2 byte number skip next 2 bytes.
310 IF B=13 OR B=14 OR B=28 THEN LOOPA=LOOPA+2:GOTO 520
*   if one byte number skip next byte.
320 IF B=15 THEN LOOPA=LOOPA+1:GOTO 520
*   if 4 byte number skip next 4 bytes.
330 IF B=29 THEN LOOPA=LOOPA+4:GOTO 520
*   if 8 byte number skip next 8 bytes
340 IF B=31 THEN LOOPA=LOOPA+8:GOTO 520
*   if byte is 143 'REM' or 132 'DATA' then skip line
350 IF B=143 OR B=132 THEN GOTO 530
*   if 2 byte code skip next byte.
360 IF B=253 OR B=254 OR B=255 THEN LOOPA=LOOPA+1:
    GOTO 440
*   was previous byte part of VN
370 IF VNF=0 THEN GOTO 410

*** previous byte was part of VN
*   then allow 0-9, .!.#.$ or % as part of VN
380 IF (B>=46 AND B<=57) OR (B=>33 AND B=<37)
    THEN GOTO 430
*   allow "(" as last byte of array VN
390 IF B=40 THEN GOTO 450
*   if not A-Z then VN complete close it out
400 IF B<65 OR B>90 THEN GOTO 460

*** if this byte not A-Z then not start of new VN
410 IF B<65 OR B>90 THEN GOTO 510
*   start of new VN found set VNF
420 VNF=1
*   change byte to character and add to B$
430 B$=B$+CHR$(B)
*   get next byte
440 GOTO 520

*   end of ARRAY VN found - add to B$
450 B$=B$+CHR$(B)

*   end of VN found.
*   'AS','BASE' are two commands which are not stored
    as code
460 IF B$="AS" OR LEFT$(B$,4)="BASE" OR B$=""
    THEN GOTO 510
470 PRINT B$;" ";
*   convert TLSTART and TLNUM to strings add VN, place
    in VNAME$
480 VNAME$=MKS$(TLSTART)+MKS$(TLNUM)+B$
*   store VNAMES$ in array
490 GOSUB 900
*   number of VNs in array  +1
500 REC=REC+1
510 B$="":VNF=0
520 NEXT LOOPA

*** set up for next line
*   Make sure BCOUNT points to start of nest line
530 BCOUNT=BCOUNT+TLLEN+1
*   clear string flag
540 SF=0:SF1=1
*   start new line on screen
550 PRINT
*   gosub compute TLSTART,NLSTART,TLNUM
560 GOSUB 600
*   if end of program then NLSTART = 0
570 IF NLSTART=0 THEN GOTO 660
580 GOTO 240
590 STOP

*** compute TLSTART,NLSTART,TLNUM
600 TLSTART=NLSTART
610 NLSTART=CVI(MID$(BUF200$,BCOUNT,2)):BCOUNT=BCOUNT+2
    GOSUB 810
620 TLNUM=CVI(MID$(BUF200$,BCOUNT,2)):BCOUNT=BCOUNT+2
    GOSUB810
630 RETURN
640 STOP
650 '

*** print hard copy of VN and associated line numbers
660 FOR LOOPC=0 TO REC-1
*   if this is same VN then just print line number.
670 IF MID$(VNAME1$(LOOPC),9,10)<>PVN$ THEN GOTO 730
*   if same line number for same VN then don't print
680 IF MID$(VNAME1$(LOOPC),5,4)=PLN$ THEN GOTO 780
*   print LF, CR if more then 72 characters.
690 IF LPOS(0)>72 THEN LPRINT :LPRINT TAB(11);
*   print line number
700 LPRINT USING "######";CVS(MID$(VNAME1$(LOOPC),5,4)),
*   make PLN$=line number
710 PLN$=MID$(VNAME1$(LOOPC),5,4)
720 GOTO 780
730 LPRINT
*   if new VN then print VN and line number
740 LPRINT USING "\        \";MID$(VNAME1$(LOOPC),9,10);
750 LPRINT USING "######";CVS(MID$(VNAME1$(LOOPC),5,4));
*   make PVN$=VN
760 PVN$=MID$(VNAME1$(LOOPC),9,10)
*   make sure PLN$ is cleared if new VN
770 PLN$=""
780 NEXT LOOPC
790 GOTO 970
800 '
*** the next 9 lines are to make sure that there is always
    at least 100 bytes past BCOUNT in BUF200$
810 IF BCOUNT<=101 THEN RETURN
820 BUF200$=RIGHT$(BUF200$,LEN(BUF200$)-100)
830 BCOUNT=BCOUNT-100
840 '
850 '
860 '
870 GET 1,RCOUNT:RCOUNT=RCOUNT+1
880 BUF200$=BUF200$+BUF100$
890 RETURN

*** the next 8 lines; store VN in array 'sorted'
900 IF REC=0 THEN VNAME1$(0)=VNAME$:RETURN
910 FOR LOOPD=REC TO 1 STEP -1
920 IF B$>=MID$(VNAME1$(LOOPD-1),9,10) THEN GOTO 950
930 VNAME1$(LOOPD)=VNAME1$(LOOPD-1)
940 NEXT LOOPD
950 VNAME1$(LOOPD)=VNAME$
960 RETURN
970 END
```

Continued from Page 41

Part Number	Description of Product	Selling Price	Vol. Issue
DATA BASE MANAGEMENT SYSTEMS			
HDOS			
885-1107-[37]	HDOS Data Base System H8/H89	30.00	23
885-1108-[37]	HDOS MBASIC Data Base Sys.	30.00	23
885-1109-[37]	HDOS Retriever ASM (3 Disks)	40.00	23
885-1110	HDOS Autofile (2 Disks)	30.00	23
885-1115-[37]	HDOS Navigational Program	20.00	25
885-8008	Farm Accounting System	45.00	30
CP/M			
885-1219-[37]	CP/M Navigational Program	20.00	31
MSDOS			
885-6008-37	MSDOS NAVPROG	20.00	73
885-8034-37	DBZ-A Database For The Z100	25.00	69
AMATEUR RADIO			
HDOS			
885-8016	Morse Code Transceiver Ver 2.0	20.00	42
CP/M			
885-1214-[37]	CP/M MBASIC Log Book (64k)	30.00	23
885-1234-[37]	CP/M Ham Help	20.00	49
885-1238-[37]	CP/M Ascirity	20.00	57
885-8020-[37]	CP/M RF Comp. Aided Design	30.00	44
885-8031-[37]	CP/M Morse Code Transceiver	20.00	57
MSDOS			
885-8038-37	MSDOS RFCAD Ver. 3.50	30.00	73
COMMUNICATION			
HDOS			
885-1122-[37]	HDOS MicroNET Connection	16.00	37
885-8005	MAPLE (Modem Appl. Effector)	35.00	29
CP/M			
885-1207-[37]	CP/M TERM & HTOC	20.00	26
885-1224-[37]	CP/M MicroNET Connection	16.00	37
885-3003-[37]	CP/M ZTERM (Z100 Modem Pkg)	20.00	34
885-5004-37	CP/M-86 TERM86 and DSKED	20.00	56
885-5005-37	CP/M-86 16 Bit MicroNET Conn.	16.00	61
885-5006-37	CP/M-86 HUGPBBS	40.00	62
885-5007-37	CP/M-86 HUGPBBS Source List.	60.00	62
885-8012-[37]	CP/M MAPLE (Modem Program)	35.00	34
885-8023-37	CP/M-85 MAPLE	35.00	45
MSDOS H/Z100 - H/Z150 PC			
885-3019-37	ZDOS 16 Bit MicroNET Connect.	16.00	61
885-3027-37	MSDOS HUG PBBS	40.00	66
885-3028-37	MSDOS HUG PBBS Source Listing	60.00	66
885-3033-37	MSDOS HUG MCP	40.00	71
MISCELLANEOUS			
885-0004	HUG Binder	5.75	
885-1221-[37]	Watzman ROM Source Code/Doc	30.00	33
885-4001	REMark Vol. I Issues 1-13	20.00	
885-4002	REMark Vol. II Issues 14-23	20.00	
885-4003	REMark Vol. III Issues 24-35	20.00	
885-4004	REMark Vol. IV Issues 36-47	20.00	
885-4005	REMark Vol. V Issues 48-59	25.00	
885-4006	REMark Vol. VI Issues 60-71	25.00	
885-4500	HUG Software Catalog	9.75	
885-4501	HUG Software Catalog Update #1	9.75	
885-4600	Watzman/HUG ROM	45.00	41
885-4700	HUG Bulletin Board Handbook	5.00	50
885-3015-37	ZDOS Skyviews	20.00	55
885-3036-37	MSDOS TREE-ID	20.00	77

NOTE: The [-37] means the product is available in hard sector or soft sector. Remember, when ordering the soft sectored format, you must include the "-37" after the part number; e.g. 885-1223-37.

Changing your address? Be sure and let us know since the software catalog and REMark are mailed bulk rate and it is not forwarded or returned.

CUT ALONG THIS LINE

HUG MEMBERSHIP RENEWAL FORM

HUG ID Number: _____

Check your ID card for your expiration date.

IS THE INFORMATION ON THE REVERSE SIDE CORRECT? IF NOT, FILL IN BELOW.

Name _____

Address _____

City-State _____

Zip _____

REMEMBER - ENCLOSE CHECK OR MONEY ORDER

CHECK THE APPROPRIATE BOX AND RETURN TO HUG

	NEW MEMBERSHIP RATES	RENEWAL RATES
U.S. DOMESTIC	$20 ☐	$17 ☐
FPO/APO & ALL OTHERS*	$35 ☐	$30 ☐ U.S. FUNDS

* Membership in France and Belgium is acquired through the local distributor at the prevailing rate.

Heath/Zenith Users' Group

Hilltop Road
Saint Joseph, Michigan 49085

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

POSTMASTER: If undeliverable, please do not return.

P/N 885-2083