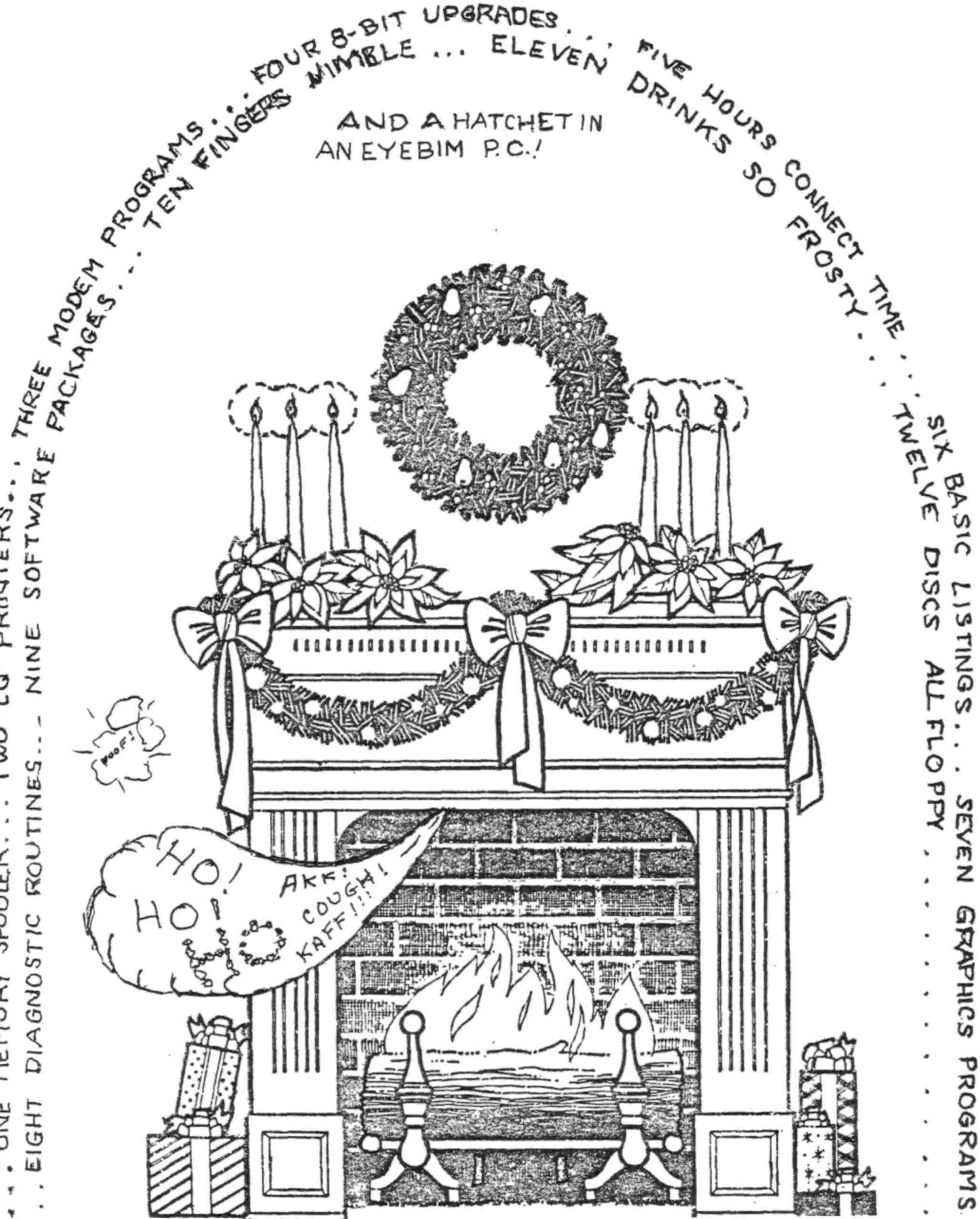
Saving Our HEATH Eight-Bit Machines!

SEBHC JOURNAL

Volume 1, Number 5 $2.50 a copy, $12.50 a year December, 1986

FOR THE TWELVE DAYS OF CHRISTMAS MY TRUE-LOVE GAVE TO ME:



LETTERS, etc.

This month we have several separate letters from one writer. Since they're all very interesting and informative, we're printing them in one bunch. Enjoy!

[Dear] SEBHC JOURNAL:

#1--

I am glad to see a journal devoted to the HEATH H8 and H89.

I would like to make a suggestion for your format. I think it would be helpful if your page layout would permit enough of a margin on the INSIDE (i.e. the left side of odd-numbered pages and the right side of even-numbered pages) to permit 3-hole punching the journal for storage. [See anything different about this issue? -ed]

#2--

I would like to offer some praise for a small firm selling disc drives and other accxessories: I.F.E., 1119 S. Washington Avenue, Piscataway, NJ 08854. Most of their business is mail order, but the owner, Mr. Bob Darling, will go out of his way to fill orders person-to-person, and to carry out repairs; this can be very useful to people living in the NYC-to-Princeton area.

My experience with Mr. Darling is that his prices are the best, that he is honest as well as knowledgeable, and that he will go out of his way to accomodate.

#3--

Your article on p.3 of Number 4 mentions ALGOL 60 and ALGOL 68.

Does anyone know whether there are any ALGOL compilers available for MICROs, especially under CP/M?

#4--

The review of programming languages in your November issue once again shows the prejudices of computer scientists. The needs of the users are ignored in the same tradition as General Motors' attitude.

Believe it or not, computers were invented to "compute" rather than be examples of the latest elegance in computer science. And for computing purposes FORTRAN (any FORTRAN!) has no equal. The reasons for this are many, but two are outstanding: Firstly, International Standard FORTRAN is transportable. Secondly, it is a compiled language based firmly on the idea that separate subroutines tackle separate tasks, and the library of available computing routines throughout the world is so vast and accessible, that translation into any allegedly More Elegant language is a completely unrealistic task.

And what is BASIC if not a "translated" version of FORTRAN, suitably simplified for beginners, and with a few graphics commands added that are useful on micros?

[Incidentally] the two authors of the file are computer scientists from the University of Waterloo where the most-widely-used FORTRAN compilers, WATFOR and WATFIVE were written.

Sincerely,
Harry Spencer

Thanks for the kind words and information, Harry! I'm sure all our members shall appreciate reading your letters as much as I!

MISCELLANEOUS NOTES FROM YOUR EDITOR --

New member/subscriber Bill Derby in Livermore, CA is working on a nifty CP/M patch package which allows one to enhance CP/M 2.2x with many of MS-DOS capabilities and make the system easier to use, especially in a small-capacity 8-bit machine with limited disc capacity. The condensed enhancement package is not expected to take up more than about 6k of CCP space. Bill expects to release it in Beta Test form Early Next Year. If you're interested, write him! Wm. S. Derby, UC/LLNL (L-300), P.O. Box 808, Livermore, CA 94550. [Please enclose a Stamped, Self-Addressed Envelope as a matter of courtesy.]

AW, SHUCKS!

Due to a number of unforseen incidents, we couldn't get the review of O'Neill Software's ELECTRA-FIND program done in time to put it in this issue. We'll see to it that it gets printed in the January JOURNAL! Hope we're forgiven. . .

MORE MISC. NOTES --

Finally got the new H8-37 soft-sector card I bought from Henry Fale installed and running properly in Hachiban-san (my H8's female Japanese name). Hachiban-san has been working nearly 10 hours a day for quite some time because my H89A is laid up with a missing D.G. Super89 CPU, and some terminal board ROMs which need re-burning. For a little while, I had Hachiban-san's H19 patched into the H89 (original Heath CPU card working in the S89 slot) because the H8 was opened up for that '37 transplant operation. The temporary lashup ran ok, but Heather (H89) sits about 5 feet away from the H19 and I had lots of disc swapping at more than arm's length. Got lots of involuntary excercise!

I'll be glad to have Heather back running again. Then Hachiban-san can get back on her regular schedule; she's been working too much overtime, getting this issue out!

Finally discovered how to make SkyCastle Computer's CP/M Calligraphy-II graphics text formatter put two columns on one page. It's a bit more complicated than the technique I developed for use with Lindley System's HDOS MXV44.DVD. But I did this issue's entire formatting under Calligraphy-II, and it looks nicer than the patched-together earlier issues! Next month I'll tell you how I do it with both HDOS and CP/M. Until then,

* H * A * P * P * Y *
* H * O * L * I * D * A * Y * S *

Graphics "Easy as PIE" On The H/Z19 Screen

by George M. Ewing

Like many article ideas, this one was sparked by an argument. A knowledgeable engineer, intimately familiar with the insides of Heath/Zenith H19 terminals, the H/Z89s and other machines using H19 graphics, told me flat out, "You can't use PIE as a Graphics editor!" A rather extreme opinion I thought, but then I remembered another collegue and friend--who also is a Famous Science Fiction Author--had postulated a law that says in part, "When an aging and eminent scientist or engineer says something is possible, he's probably right; when he says 'it can't be done,' he's almost certainly wrong."

What my friend implied was, "It's too inconvenient to use PIE as a graphics editor; there are much better and more efficient, albeit more expensive, tools for the job."

I considered that statement a rude challenge and so I experimented with my aged copy of Walt Bilofsky's Software Toolworks PIE V1.5. I proved to myself how it can sucessfully be used with H19-H89 graphics. During these experiments I discovered a few limitations, but generally results were so encouraging that I just have to share them with you!

A BRIEF OVERVIEW --

PIE (available in various releases for many H/Z machines) is an inexpensive screen editor, great for straight typing. When teamed together with a formatting program such as TEXT, it makes a pretty fair poor man's word processor.

But PIE is limited to 80-column pages; if you're trying to design graphics for display (or printout) on anything but an 80-column H19, it may be unsuitable. Note that I said, "unsuitable," not "impossible!" It's possible to make up some simple graphics designs, such as a horizontal bar graph in compressed form at 80 column width, then use another editor, such as CP/Ms ED to splice in missing columns with a macro to send out to a dot-addressable printer. Rather tricky but less troublesome than trying to do the whole thing from scratch with ED--or worse yet --trying to do it at machine language level with DDT!

Another limitation is PIE's inability to handle files larger than memory space available, unlike ED. PIE will handle only twelve to thirteen 24-line screens running on a 48k CP/M system, about half the 24 to 26k free memory space. You might stuff in a few more by creative use of tabs and blank lines.

With a 64K system available space is slightly over 40K, plenty for simple graphs, charts, and puzzles. But for something such as an elaborate animation program you should create many separate picture files with PIE, and then later splice them together with PIP. (I'm assuming a CP/M operating system for the discussion here, as that's what I am familiar with. The HDOS version of PIE should work fine also.) [It does. -- ed]

A third major limitation: The H/Z Terminal Logic Unit (TLU) has its own separate Z80 processor, communicating with the CPU over an internal RS-232 serial link and a complex array of escape and control sequences.

For example, keyboard small letter A generates the 7-bit Octal number 141 (Hex 61, or Decimal 97). If the CAPS LOCK key is down you get Octal 101 (Hex 41, Dec 65). Any one of six different characters can be generated; capital or lower-case letter "A" in either regular or reverse video, or in graphics mode, two horizontal rows of dots across the center of the 8x10 character matrix, also in regular or reverse video.

In order to be able to use an editor like PIE and still "see what you're getting" in real time without stopping, saving the file, and using TYPE or PIP to check it out, you have to consider embedding escape sequences in the screen as you go along, and to take the terminal off line and feed it a separate escape sequence every so often. (This last part was what caused my engineer friend's grouchiness; THERE'S NO OBVIOUS WAY TO EMBED ESCAPE SEQUENCES WITH PIE!)

THE GREAT ALPHABET SOUP ESCAPE --

When a beginner first sees all those TLU escape sequences in the manual it's quite intimidating! In addition to the ASCII characters in Octal, Decimal, Hex and control Mnemonics, there are the ZDS sequences (from ZCUH to ZERV and ZXRV to ZXMP) plus ANSI codes from CRM to FETM --either SET, RESET, or N/A, of course!--and FEAM to PRCP. Then throw in tables for converting between ZDS and ANSI, pile on the extra 25th-line sequences plus regular and alternate keypad modes and things rapidly degenerate into alphanumeric chaos.

Graphics "Easy as PIE" Continued

While you may eventually find most of this information is useful or even essential, all a beginner really needs are the manual's two pages of graphic characters (asterisks in 8x10 matrices) and four simple commands to get started.

Remember this Mnemonic:

Foolish Girl! Mind Your p's And q's!

I suggest you disregard much of the appendix stuff, though you'll probably want to go back to some of it later. Forget Octal, unless you have to modify existing software that was written in it. Octal went out with the 6SN7GT dual-triode vacuum tube! Also skip hex for now.

To keep things simple, alphabet keys are shown as capital letters in quotes ("A", "Z", etc.) and specify whether upper or lower case. Thus, "lower-case A" means ASCII character 'a' (Dec 97 Hex 61) and so on.

Back to the "Foolish Girl" Mnemonic. There are four escape sequences that you really do have to memorize:

1. ESC-F - Escape key followed by upper-case "F"
   Enter Graphics Mode, switches TLU to graphics mode.

2. ESC-G - Escape key followed by upper-case "G"
   Exit Graphics Mode, switches TLU to regular mode.

3. ESC-p - Escape key followed by lower-case "P"
   Enter Reverse Video Mode, displays black characters on white background. Experiment to find mode best for appearance or legibility.

4. ESC q - Escape key followed by lower-case "Q"
   Exit Reverse Video Mode, cancels reverse video.

Escape sequences F, G, p, and q give us the "Foolish Girl" mnemonic. Remember which letters are upper and which are lower case!

Before setting out to explore the H-19, let's briefly review how the TLU display is set up. There are 2,000 characters in all (NOT 2K, or 2048 but 1920 in a 24x80 array and a separate row of 80 at the bottom). The 33 graphics characters're in an 8x10 matrix avoiding "cracks between the columns" problems. (See Fig. #1.) The TLU writes characters sequentially to the screen from left to right and top to bottom--the same way you read a printed magazine page. To embed a control or escape character in a screen it must be placed to the left of a group of characters on a line so as to be "upstream", and to the right of the group if "downstream".

LET'S DO IT! --

The first thing to do with PIE before getting involved with a complicated design, or practicing by doodling is to copy the distribution disc "Patches.DOC" file on your printer. It's a good idea to shut off PIE's way of using tabs to save space. Tabs can mess up a screen of columns such as vertical bar graphs. Read the documentation BEFORE proceeding with this patch! Then keep track of which version (patched, not patched) you're using with a project. You'll be glad later that you did!

Okay, you've booted up your system, loaded PIE, and started a test file, called "DOODLE.GFX," or some such. Now what?

Here's a brief first excercise: Press OFF LINE key, next ESC key and then lowercase "p" key. Now type an alternating upper and lower case line of letters across the screen's top. See the TLU "switching" properly between regular and graphics mode? Press RETURN then LINE FEED key and cursor goes back to left side of screen and down a line and type another test line. Wasn't that easy and fun? When you're finished press ESC, then lowercase "z" to blank screen, then press OFF LINE key to return on line.

PIE still loaded? Now embed an ESC-F sequence in the top line which tells the TLU to display all subsequent characters entered as graphics characters instead of text until an ESC-G sequence, then switch back to text mode.

"But wait a minute," you say, "There ISN'T any provision for embedding escape sequences in a PIE file! A few control codes yes, but PIE has already allocated the "ESC" key on the H-19 keyboard as a reverse tab; if I hit the key, all that happens is that the cursor jumps 8 spaces to the left."

True, the ESC key is already allocated. But PIE has the ability to embed control sequences. You can embed a "CTRL-L" form feed for an Epson or Okidata printer, for example. Just hold down the "CTRL" key while first typing letter "K" key, then the control character key. You'll see the control character in reverse video on the screen, but it will later be invisible when the file is TYPED. This works for all but a few characters.

In the CP/M version of PIE, control "I," "J," and "M" are illegal, because they interfere with the operating system or are already allocated other functions. Fortunately the TLU accepts a control key plus a lower-case

Graphics "Easy as PIE" Continued

square bracket key as an "ESC". To embed the "ESC-F" sequence, hold down the "CTRL" key, hit the "K" key, (either upper or lower case), and then the lower-case square bracket key, then upper-case "F" without "CTRL." The square bracket appears on the screen in reverse video, followed by the upper-case "F" in regular video. as graphics when the file is TYPED.

Try it by typing in some characters. They look like standard alphanumerics but now the computer sees they're graphics characters. Save the file with a "CTRL-E" then send it to the screen with TYPE or PIP and it is displayed as graphics.

A few paragraphs back we promised a real-time display, or a what-you-see-is-what-you-get mode. There is a problem caused by the TLU Z80 being separate from the CPU board's Z80. The two processors only talk to each other when they think it's necessary. To see graphics in real time we must take the TLU off line and manually force it into graphics mode. To do this, press the OFF LINE key, next "ESC", followed by upper-case "F" keys, and the OFF LINE key again. Now press the "f-3" key; PIE then clears the screen, returns to the top of the file, and writes characters entered as graphics.

Now embed an "ESC-G" sequence ("CTRL" key, then "K" followed by a lower case "[" key, and upper-case "G" without "CTRL") in a convenient location "downstream" of the graphics characters group. Note that you can mix regular text and graphics in the same file--even on the same line--as long as the graphics are deliniated by upstream and downstream embeds, and you're willing to take the terminal off line to manually force it from one mode to the next with "ESC-F" and "ESC-G".

Quite often upper-case-only text is used in a graphics display for graph labels, game screen instructions, etc. To do this just leave the terminal in graphics mode all the time you're working, avoiding tiresome switching between modes. Uppercase letters then show as text, the letter key or "QWERTY" graphics keys are lowercase. But this is NOT true for numeric and punctuation key characters.

Use this technique to embed "ESC-p" and "ESC-q" in a PIE file to switch reverse video mode as desired. And use the same escape sequences offline to switch the terminal. But remember: the Terminal Logic Unit sees "CTRL-K + CTRL-Square Bracket" in a PIE file as the ESC key. Also remember when using reverse video, that blank lines in a large block of graphics or text won't be highlighted, nor will any line be highlighted past the rightmost character. In order to get a pleasing and legible display you may insert a column of characters, such as periods, down the right margin of the screen or block within the screen to make an even and attractive pattern. (See Figure 2.)

Here's something else to keep in mind: Embedded escape and control characters, both reverse video and any following characters in a sequence such as "CTRL-K+CTRL-SQUARE BRACKET+"F", may be invisible to the display screen when a file is TYPED but they still take up space in the line and must be taken into consideration as you work.

While editing, you can manipulate and move characters around in a file. But when the file is called back for display, the additional spaces control groups occupy in a line can shift other characters away from their intended locations. This is particularly annoying in designs with a number of vertical columns of characters, such as vertical bar graphs.

Remember, PIE only allows 80 characters in a line, and this includes the "invisible" embedded ones. If you have 8 characters on a line taken up by embedded escape sequences, you only have 72 spaces left open, even though the line LOOKS empty when sent to the screen! If you didn't patch out PIE's compressed-tab feature before starting to edit a file, be especially careful to watch out for this "gotcha" because it's "easy as PIE" to accidentally "eat" an invisible tab when manipulating embedded escape characters!

To minimize that problem, carefully keep track of all embedded characters and locations. Sometimes problems can be avoided by moving the embedded characters to another column or row where they won't screw up the design, but will still perform their intended switching function. For example an ESC-F can be anywhere upstream of a graphics block rather than in the most immediate left space. Until you become experienced in this technique it's a good idea to save your edited file quite frequently, and check it with TYPE. Often a dislocated line in a screen can be repaired by deleting a few spaces in the right place with the keypad DC key.

QWERTY CADEF And The TVA BUS --

Once you start seriously playing around with H-19 character graphics, some of the manual's listing short-

Graphics "Easy as PIE" Continued

comings can be a nuisance. The alphabetical and numerical-order chart is useful in finding, for example, exactly what is displayed when the terminal receives Hex 61. But it's not, if you want to know something like, "How do I draw the top left corner of a box so that it will match up with the other characters, and not leave a gap?" Then it's back to the 8x10 matrix charts, and lots of tedious dot-counting.

Figure 2. shows all 33 graphics characters in modified QWERTY order; that is, with the three rows of alphabet keys listed in the order they are encountered, and the numeric and punctuation keys separately at the right. They are given in blocks of 8 to show how they match up when inserted with the repeat key.

Figure 3. shows the same characters when displayed in reverse video. I put a column of periods along the right side of the screen to even out the display.

Almost any functional grouping of characters can be learned easier than by plowing through the higgledy-piggledy manual listing. For example Figure 4. shows the four box-corner characters arranged in a rectangle.

Make up your own mnemonics--for example, "Clockwise Draw an Exploding Figure"--to help remember "C", "D", "E", and "F". (Actually entered as lower case, but are easier to remember as upper case.

Remember, you can type upper case text in a file and see the letters as you go, even when the terminal is manually forced into graphics mode.) Adding the "A" key and the grave accent or "backward apostrophe" (next to BACK SPACE key) allows you to draw rectangular boxes, game mazes, etc. with a two pixel-wide line. These six are perhaps the most useful characters.

Note that three characters, (grave accent ['], broken line [¦], and shifted curly-bracket [}]) also generate vertical lines two pixels wide, but only the first (') matches with the CDEF corner characters and the next group, the TVA-BUS.

THE TVA-BUS --

Not a Department of Energy vehicle full of bureaucrats shuttling between Oak Ridge and Knoxville but my mnemonic for the other rectangular-line H19 characters. (See Figure 5.) With these, plus the first six you can make up all kinds of mazes, street maps and so on, without gaps. The "exploded" figure shows the new characters as respectively, a left and right branching "T" connection, a cross and an up-and-down branching T.

If you think of them as all "T" connections beginning with the letter "T" in "TVA-BUS" and in "left-right, up-down order" with the horizontal dash following the "A" character, it will be easy to remember them. They all form contiguous figures with the "C-D-E-F" corners and the vertical bar (') key.

Figure 6. is a simple maze built up from CDEF. I've created the repetitive geometrical border with macros six to 8 keystrokes long, inserting them with the "REPEAT" key, although the "ENTER: nn...Blue Key" sequence works fine; "nn..." is number of desired repeats--to 999 max.

Another logical grouping are four special symbols generated by the "G," "H," "J," and "K" keys. They're not strictly graphics characters. These are: plus/minus sign, right-arrow, divsion sign, and down-arrow (standard H19 graphics ROM only). You might want to include the paragraph symbol, generated by the tilde (shifted grave accent) key in this grouping. Make up your own mnemonics for these keys. You might also group the diagonal and "sawtooth" characters together, the various vertical or horizontal bars together, and so on, in your own scheme.

COLORED KEYS AND MACROS --

The editing shortcuts which make PIE easy to use with text also work in graphics mode, allowing you to do all kinds of time saving things with macros, put-and-take, global search-and-replace, etc.

For example, if your design includes a complex and frequently-recurring pattern, use the "ENTER+BLUE" key function to record key-strokes. Now every time you press the BLUE key you can insert an entire block with one key stroke at any location. Combine with the global search-and-replace function and move pre-written designs to any spot you've marked with a special character.

You can manipulate embedded escape characters around in a file like any others, though they'll be ignored by the global search function. The RED and WHITE key put-and-take function allows you to record up to an entire 24-line page of graphics and text, including embedded escape characters, and re-insert multiple copies (up to 999, if there's room in the file!) anywhere you want.

When using macros it's VERY important to keeping close track of embedded characters, possible missing tabs, etc. One mistake may repeat 999 times and irrevocably trash a large file! Make frequent backup copies, and ALWAYS SAVE FILES BEFORE trying a complex macro! If you end up with a mess, exit without saving (ENTER-CTRL-E sequence) and restart without much lost time or wasted effort.

Graphics "Easy as PIE" Continued

GETTING IT OUT OF THE GREY BOX --

Now that you've created a graphics masterpiece, what do you do with it? If you are going to call up your design from other software, you'll have to add a few things to make it work properly. You may do this with ED which adds line numbers; it's easier to work with a file with absolute line numbers. And you can turn the line numbers on and off at will with the "v" and "-v" commands in ED. [I've never been happy with ED! - ed] Testing simple animation sequences is easy with ED. Just use the "nn...T" command to type any selected group of lines, and then use the "nn...Z" "sleep" command to establish timing delays. It's much more easy than with a machine-language timing loop.

If you're working with one of the BASICs as an editor, convert each file line into a PRINT statement with line numbers and graphics characters enclosed in quotation marks. A macro is probably the best way to do this. You also may have to have a complete set of embedded escape characters for each line; check the rules for your version of BASIC for the correct way to deal with control and escape characters.

In FORTH or RUNIC, you usually create a single "dictionary word" that prints as much of a file as you want. The exact format depends on what language variant you're using, available memory, etc. The RUNIC "SAY" command is analogous to BASIC's "PRINT" and requires that each block of text or graphics be set off with quote marks.

HARD COPY --

If you want hard copy of a graphics design, there are basically two ways to go: photograph the screen, or use a screen dump utility and a good dot-matrix printer with H19 graphics to put your design on paper. (I use photography; don't have a dot-matrix printer.)

Photography has many advantages. It's quick and simple, Using color filters or gels, tinkering with the H19/89 brightness control and making multiple exposures lets you create striking multiple-color slides you can show at meetings or to classroom groups. To do this you need a 35mm SLR camera capable of making multiple exposures without rewinding film, a sturdy tripod, closeup or macro lens and several 12-exposure rolls of color slide film. Have your slides duplicated or large prints (up to 11x14-inch) made by any "pro" photo lab quite inexpensively. For a good quick-and-easy record get hold of a Polaroid 'scope camera. Another neat trick is shooting animated Super-8 or 16mm movies off screen one frame at a time. Tedious, but can be very rewarding.

If you have a dot-matrix printer, check your local HUG, CP/M-UG, or other user group to see what software is available. I've seen ads for MPI and Okidata packages which specifically include the H19 graphics set. [Epson printers have built-in H19 graphics; I use an excellent screen dump program you can get from Skycastle; see ad in Nov. 86 issue. - ed] If your printer has user-character-font downloading capability, and supports 8 x 10 matrix characters, you can probably type in the set by hand from the manual charts, but you may have to include some special control and escape sequences for your particular machine. A real fanatic would probably fake up an 8x10 set of dot graphics with periods or something that would run on a proportional-print daisy wheel, but that's out of my league. Good Luck !

ADDENDA --

Several users have reported that while in graphics mode, PIE sometimes spontaneously switches the TLU back into regular text mode. This usually happens when the cursor nears the right-hand screen margin and "INSERT MODE" switch (keypad IC key) is toggled "on". I could duplicate the effect by repeatedly trying to insert a character into a line at the right margin. The LINE FULL error message comes on screen, and about one time in twenty, a spurious "ESC-G" somehow gets generated. That switches the TLU back into text mode. If it does happen, take terminal off line, enter "ESC-F", then return on line. Doesn't happen often enough to be more than a minor nuisance. -- GME [Don't happen with PIE 1.6 - ed]

Two new product announcements of special interest were released as this article was being prepared:

Ricoh makes a computerized XRP 35mm camera with three pre-programmed modes for CRT photography: American NTSC video, European PAL and SECAM color, and monochrome (including H/Z89) or RGB computer monitors.

ATG Systems, Inc. of Wellesley, Mass. has an inexpensive (about $20) screen dump CP/M utility program for the H19 character set called the ZSD-89. I haven't yet had a chance to evaluate it, also unsure of printers it supports although ATG claims it's an extensive list. -- GME

[See all Figures mentioned above on page 8. - ed.]

[EOF]

** SUBROUTINE **

Boss: "This computer seems to be going crazy! What's the matter with it?!"

Programmer: "I saw someone throw a rubber band into it and now all it does is make snap decisions."

Graphics "Easy as PIE" Continued

Figure 1. H/Z19 Graphics 8x10 matrix --
a. ASCII character "b" (98 Decimal, Hex 62):

```
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
(*) (*) (*) (*) (*) (*) (*) (*)
(*) (*) (*) (*) (*) (*) (*) (*)
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
( ) ( ) ( ) (*) (*) ( ) ( ) ( )
```

b. ASCII "b" in reverse-video mode:

```
(*) (*) (*) ( ) ( ) (*) (*) (*)
(*) (*) (*) ( ) ( ) (*) (*) (*)
(*) (*) (*) ( ) ( ) (*) (*) (*)
(*) (*) (*) ( ) ( ) (*) (*) (*)
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
(*) (*) (*) ( ) ( ) (*) (*) (*)
(*) (*) (*) ( ) ( ) (*) (*) (*)
(*) (*) (*) ( ) ( ) (*) (*) (*)
(*) (*) (*) ( ) ( ) (*) (*) (*)
```

Figure 2. H19 Graphics Characters, QWERTY Order:

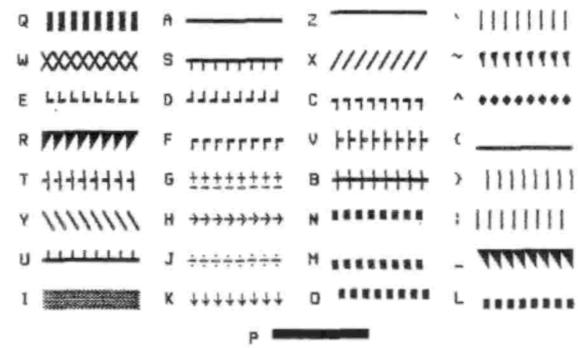


Figure 3. H19 Graphics Chars, Reverse Video:

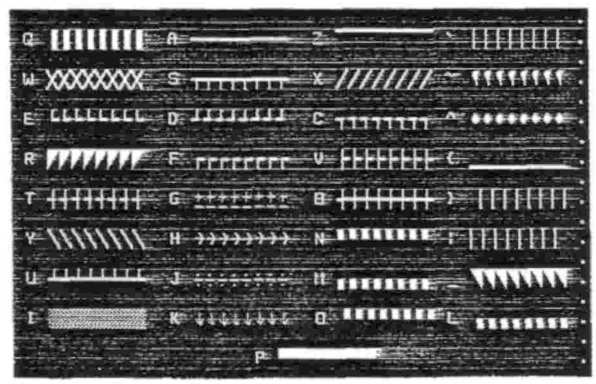


Figure 4. H19 Clockwise Draw an Exploding Figure Mnemonic:

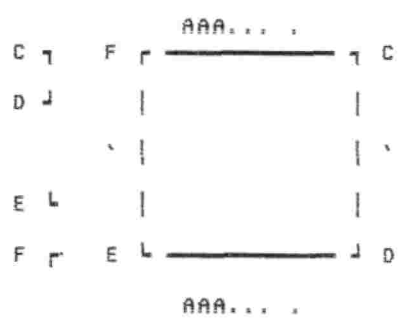


Figure 5. H19 TVA-BUS Characters Mnemonic:

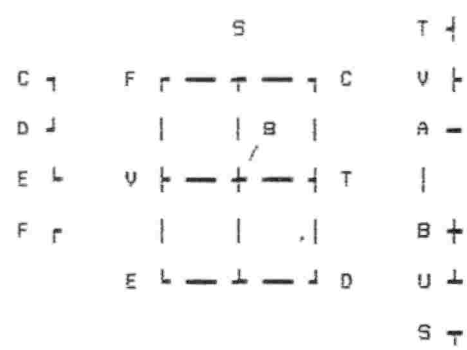
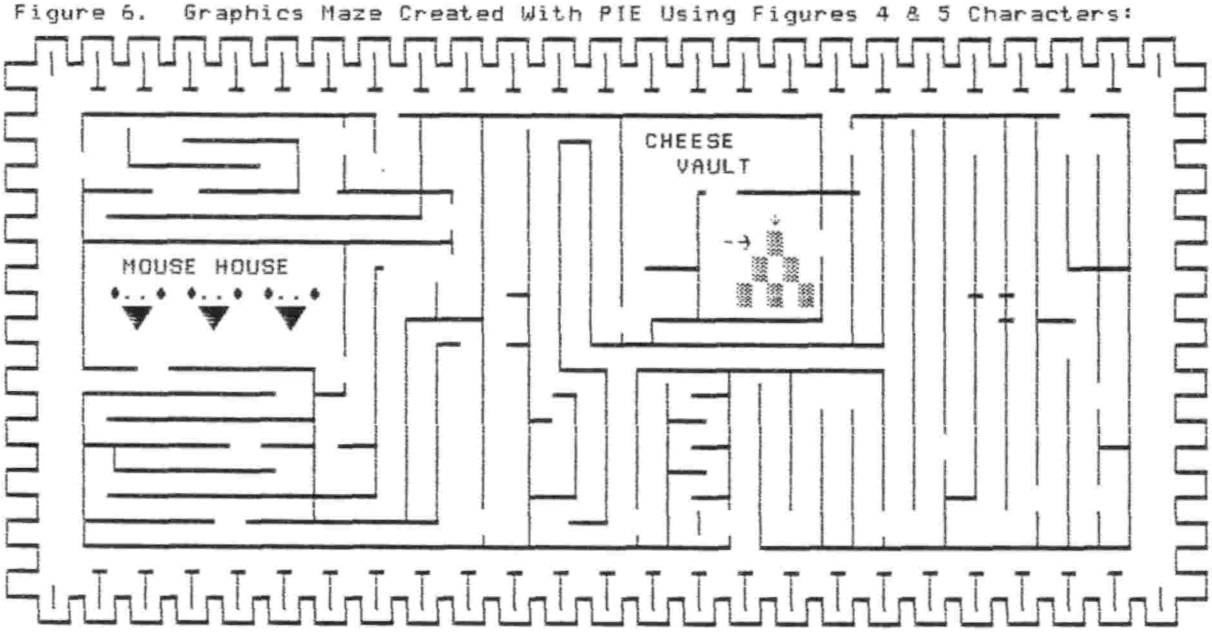


Figure 6. Graphics Maze Created With PIE Using Figures 4 & 5 Characters:



I remember when I got my first H89. In addition to a real keyboard and an 80-column screen (with lower case, too), it had a DISK DRIVE. This was no game machine, with a chintzy cassette recorder. It was a serious computer for professional accounting and word processing.

The H89 came with an H88-1 hard-sector controller, and one internal drive storing a whopping 100K bytes. Since this was twice the RAM size, I couldn't imagine it being any kind of limitation.

Funny how a computer's memory seems to shrink even faster than the waistline on my old suits. It wasn't long before 100K seemed like nothing at all. A bootable disk with a text editor and all the necessary utilities left precious little space for the project at hand. Heath's answer was of course, more disk drives.

A pair of external 40-track drives were soon added. Now I could put my programs on the A> disk, and my data on another in the B> drive. I could comfortably edit files up to 1/2 the disk size, and still have room for a file and its backup.

It didn't take long until I was getting "Disk Full" messages again. By now, Heath was touting the 80-track Tandon drives, which could store up to 800K on a single disk. But they were for the $300 soft-sector controller board. Would they work with the hard-sector as well? Heath said "no".

But Ray Livingston said "yes". He wrote a program called BIOS-80 that let you connect double-sided and/or 80-track drives to a hard-sector controller, and thereby double or quadruple your disk storage. Best of all, it cost only $50 and was fully compatible with my existing disks. I replaced my internal drive with a Tandon TM100-4 and bought a copy of BIOS-80 from Keyboard Studios (now Studio Computers). Total cost was just $150, 1/2 that of the soft-sector controller alone.

It worked great. I could store 390K on disks in the Tandon drive, yet it could read and write all my present disks and software. I had the usual learning problems (ya canna read a 2-sided disk inna 1-sided drive, etc.), but I learned to identify the disk format on the label.

I finally got an H37 soft-sector controller courtesy of my employer, TMSI (who was footing the bills). Wow! 800K on a single disk. Disks were less expensive, and easier to find. It was noticeably faster, too (see fig.1). But soft-sector controllers are a scarce commodity around here. Just when I'd start getting used to it, somebody would sell it and ship it off to a customer. For a time I literally padlocked the cover to keep them from scavenging my machine for parts!

Time (seconds)	H17 hard-sector	H37 soft-sector
warm boot (^C)	4	3
load 8K (PIP)	5	4
PIP a 1K file	5	4
PIP an 8K file	10	6
PIP a 90K file	97	57
FORMAT a disk	78	19

Figure 1 - Disk Read/Write Performance

While I'd never give up my soft-sector controller, the H17 still has advantages. The hard-sector disks are so tolerant of abuse that I use them regularly for archival and backup storage. They have survived being fingerprinted, dropped on the floor and run over by my office chair, drowned by spilled drinks, and even bent, folded, and mutilated by the Post Awfuls.

So I got to thinking. Three years ago BIOS-80 could put 4 times more data on an H17 disk. Technology has moved a lot in 3 years. What can be done <u>today</u> with the H17 controller?

The H17 Hardware

The hard-sector board is about as simple as a disk controller can get. It consists of an AMI S2350 USRT to serialize the data, and an 8-bit parallel port to control the disk drive mechanics. It depends almost totally on software to do the work. But since software is easily changed, there is a LOT more that can be done!

Figure 2 shows the storage capacity of the hard-sector controller with various drives. These are the formatted capacities; CP/M will report this value minus the space reserved by the directory and boot tracks (typically 10K).

DISK DRIVE: tracks:	40-track		80-track	
sides:	1	2	1	2
1. standard Heath BIOS and 10-sector disk	100K			
2. #1 with BIOS-80 software	100K	200K	200K	400K
3. H17 with 8" drive and 15-sector disks			288K	577K
4. high-capacity 5.25" 16-sector disk			640K	640K
5. soft-sector disk in H17 controller	100K	200K	200K	400K
6. #5, extended density (1 sector/track)	120K	240K	240K	480K
7. #6, high-capacity 5.25" drive			400K	800K

Figure 2 - Disk Capacities with H17 Controller

As you see, BIOS-80 is pretty good (#2 in fig. 2). It adds single/double side and 40/80 track support with no hardware changes. The H17 controller and an 80-track DS drive puts 400K on a disk; more than an IBM PC. But can we do better?

The traditional route is to go to 8" disks (option #3). Being larger, they have more tracks, and each track is longer. 8" disks spin faster than 5-1/4" (360 vs 300 rpm). This plus the larger diameter permits the read/write speed to be doubled. Hard-sectored 8" disks are available; they have 15 256-byte sectors and 77 tracks per side. Storage is 288K single-sided, or 577K double-sided.

8" drives come in full and 1/2-height versions, single and double-sided. They need +5 and +24 volt power, and sometimes 120 VAC for the motor. They have the same data signals as a 5-1/4", but use a 50-pin ribbon cable instead of 34-pin. To use an 8" drive with the H17 board, make a 34-to-50 pin ribbon cable adapter as in fig. 3. The higher data rate is achieved by changing the clock signal on the I/O bus (pin 13 of P506) from 2MHz to 4MHz.

5-1/4" cable	signal name	8" disk drive
pin 6	drive select 0	pin 26
8	index	20
10	drive select 1	28
12	drive select 2	30
14	drive select 3	32
16	motor/head load	18
18	direction	34
20	step	36
22	write data	38
24	write gate	40
26	track 0	42
28	write protect	44
30	read data	46
32	side select	14
all odd	ground	all odd

Figure 3 - 5-1/4" to 8" drive cable adapter

8" drives are rather passe' nowdays, but there is a better alternative. The Mitsubishi M4854, TEAC FD55GFV, and Chinon #506 ($100 to $150) are all part of a new breed of 1/2-height drives. They have the same connectors, mounting holes and use the same disks as 5-1/4" drives. But they perform like an 8" drive; spin faster, hold more data, and read/write twice as fast. You get the capacity and speed of an 8" drive without the cabling and mechanical disadvantages. IBM uses them in the -AT to store up to 1.2 megabytes on a 5-1/4" disk.

Installation is simplicity itself. Just bolt the high capacity drive in place of one of your 5-1/4" drives; the cables plug right in. To get the higher capacity, you have to use 16-hard sector 5-1/4" disks (standard for North-Star computers). You get 16 x 256 bytes = 4K per track; 4K x 80 tracks x 2 sides = 640K on a 5-1/4" disk with the H17 controller!

Like the real 8", these drives run at 360 rpm. You have to change the I/O bus clock (pin 13 of P506) to 4MHz. At this speed you can't read your old disks. But suppose you don't want to lose compatibility with regular 10-hard sector disks? Well, the drives have a speed select line on pin 2 of the 34-pin drive cable. Pull it low for normal 5-1/4" disks (300 rpm), and high for high-capacity disks (360 rpm). The "select" LED on the front is red in 5-1/4" mode, and green in 8" mode.

If you have a software-selectable 4MHz conversion, the same I/O bit that switches between 2/4 MHz should set the disk drive speed as well. Route the 2/4 MHz clock signal to P506-13 so it switches automatically, too. You can then read normal disks at 2MHz, and high-capacity disks at 4MHz. This makes the software easier because the timing loops change automatically.

Soft-Sector Disks in a Hard-Sector Drive

People complain that hard-sector disks are more expensive and harder to find. Is it possible for an unmodified hard-sector board to read soft-sector disks?

Amazingly enough, the answer is a partial "yes". The H17 board can be programmed to read soft sector disks, but only SINGLE density. Single density uses two bits on the disk to represent one bit of data. The bits alternate between clock-data-clock-data, etc. Since the clock is right there with the data, it takes only simple circuitry to recover it.

Double density has exactly the same number of bits, but every bit represents data (data-data-data-data). The clock is implied by the data bits, but it needs special circuitry to recover it. This circuitry is missing on the H17 board.

The Heath soft-sector single-density format has 10 256-byte sectors per track; exactly the same as a hard-sector disk. The data bytes are the same for both. But each sector on the disk has a prefix and a suffix that you never see. The soft-sector version is more complicated. (This is the "IBM 3740" format, and as you might expect, IBM never does anything simple). Fig.4 compares the two:

soft-sector bytes	description	hard-sector bytes
-------	-------------------	-------
6	bytes of 00h	30
1	ID address or sync	1
1	track number	1
1	side number	1
1	sector number	1
1	sector length	
2	checksum	1
11	bytes of FFh	
6	bytes of 00h	12
1	data/deleted or sync	1
256	bytes of user data	256
2	checksum	1
27	bytes of FFh	
---		---
316	total bytes per sector	305

Figure 4 - H37 vs H17 sector formats

The main difficulty is the method of checksumming. The H17 uses a simple 1-byte checksum; the IBM format uses an elaborate 16-bit cyclic redundancy check. The soft-sector board has hardware to do this automatically; the H17 board does not. So the CRC must be calculated in software, an awkward process.

But it IS possible; as proof, I have an S-100 bus computer that reads and writes normal 5-1/4" and 8" soft-sector disks with a "controller board" that is nothing but a USRT and a parallel port; just like the H17. I even have full source code for it so I know how it works.

The main drawback: it's S-L-O-W. The CPU is doing so much busywork that it can only read one sector per revolution. Then it wastes another revolution to find the next sector (without sector holes to count, you have to read everything until you find the sector you want). This makes it noticeably slower than even the hard-sector format.

The main advantages are the ability to use cheap, available disks, and as an exchange medium. Purchased software is often on single-density soft-sector disks, so this program would let you read them without a soft-sector controller. You could also read/write disks from a friend who doesn't have an H17 controller.

Extended Density on the H17

Heath CP/M 2.203 has what they call "extended density" for the soft-sector board. This is really double-density with five 1K sectors per track. By using bigger sectors, the gaps between sectors take a smaller percentage, and the total storage is increased. Can we implement a form of extended density on the H17?

The greatest gain is with the largest sector size. So let's go all the way and aim for one sector per track. Now where can we get hard-sector disks with one hole? Aha! A soft-sector disk is just a hard-sector disk with one sector per track!

The H17 clocks its data with the 2.048 MHz I/O bus clock. This is divided by 16, so the data rate is 128K bits/sec. At 300 rpm one revolution takes 0.2 sec, so a track will hold up to 25,600 bits = 3200 bytes. Round this off to 3K (=3072 bytes) to allow an inter-sector gap. A 40-track SS disk now holds 120K (fig.2 line #6); a DS 40-track disk is 240K; and a DS 80-track drive 480K. Our DS 80T high-capacity drive holds 5K bytes per track, or 800K!

Performance-wise, there is a substantial improvement. You still get only 1 sector per revolution, but that's 10 times more data than before. In fact, this "track buffering" is exactly what high performance systems do.

The Software

You need the source code for the disk drivers to make such fundamental changes. Heath obligingly supplies it for both CP/M and HDOS.

HDOS allows you to write installable device drivers. A device driver is a separate program that gets run after boot-up. For this reason, it should be easier to write the enhancements first for HDOS, even though fewer people will use it. The development and debug cycle should be faster. Unfortunately, I know zilch about HDOS so this will probably remain idle speculation.

CP/M puts all its disk handling routines in the BIOS (the program BIOS.SYS on a bootable CP/M disk). The Heath BIOS is well written, and nicely modularized; you can add or delete drivers for hard and soft-sector controllers, winchesters, etc. So our new H17 driver can be built as a modification of the H17 routines in the existing BIOS.

Once the changes are made, you have to go through the miserable MAKEBIOS nonsense. A shortcut is to build a test BIOS with a big hole in it, and patch your test programs into the hole via DDT. Then you've got room to poke, prod, and patch.

The high-density formats move data past the head twice as fast, so all the time delays should be halved. The fastest way is to run the Z80 at 4MHz. That way the timing constants in the existing BIOS need little or no change.

Changing the number of sectors per track is fairly straightforward. It just changes a few constants in the existing programs. CP/M needs a new deblocking algorithm, but a perfect example is supplied by the soft-sector source; it uses 16 256-byte sectors per track.

The hardest job is a new FORMAT program. It should actually be the first thing written, but that's rough without an example source code to look at.

For our extended density H17 format, we set up a 3K "track buffer" in RAM. Like the 1K sector buffer used for the soft-sector board, the track buffer allows CP/M to read or write in 128-byte logical sectors, while the actual sector size is 3K. The first read of a particular sector reads the whole track into the buffer. All subsequent reads come instantaneously from the buffer; all writes go into the buffer, and then the buffer is written to disk. This algorithm is exactly what a high-performance BIOS does anyway, so disk read/write speeds will increase dramatically.

Disk System Performance

While we're on the subject of disk read/write speed, it should be noted that neither the hard- nor soft-sector controllers are anywhere near the actual data transfer rate of the disk drives themselves. Even in single density, the data is coming off the disk at a peak rate of 128K bits per second. That's 16K bytes per second!

Of course, we have to allow up to 1 second for the motor to start if it wasn't running already. And up to 30 milliseconds for each time the head has to step to another track. But we still aren't even close to the actual times in figure 1. Where did all our performance go?

Most of the time is spent waiting for the desired sector to come around on the disk. What REALLY sets read/write speed is how many bytes per revolution can be read from the disk. The soft-sector board is faster because its SOFTWARE is better written, not because of the hardware.

Most programs read consecutive sectors from the disk. Hard-sector disks put the sectors in numerical order; 1-2-3-4 etc. CP/M pauses after reading sector "n" for internal housekeeping functions. So sector "n+1" is already past the head before CP/M is ready for it. CP/M therefore gets only one sector per revolution (256 bytes).

For soft-sector disks, the BIOS uses a method called "interleaving". The FORMAT program puts adjacent sector numbers about 1/3 of the way around the disk from each other (for example; 1-4-7-2-5-8-3-6-9). So CP/M can read sector "n", fool around a while, and still catch sector "n+1" on the same revolution.

Getting just the right interleaving can increase read/write performance dramatically. It would be interesting to see how much better the H17 would do with interleaving.

Conclusion

By now you're probably wondering if this software is going to be made available. The answer is; I don't know.

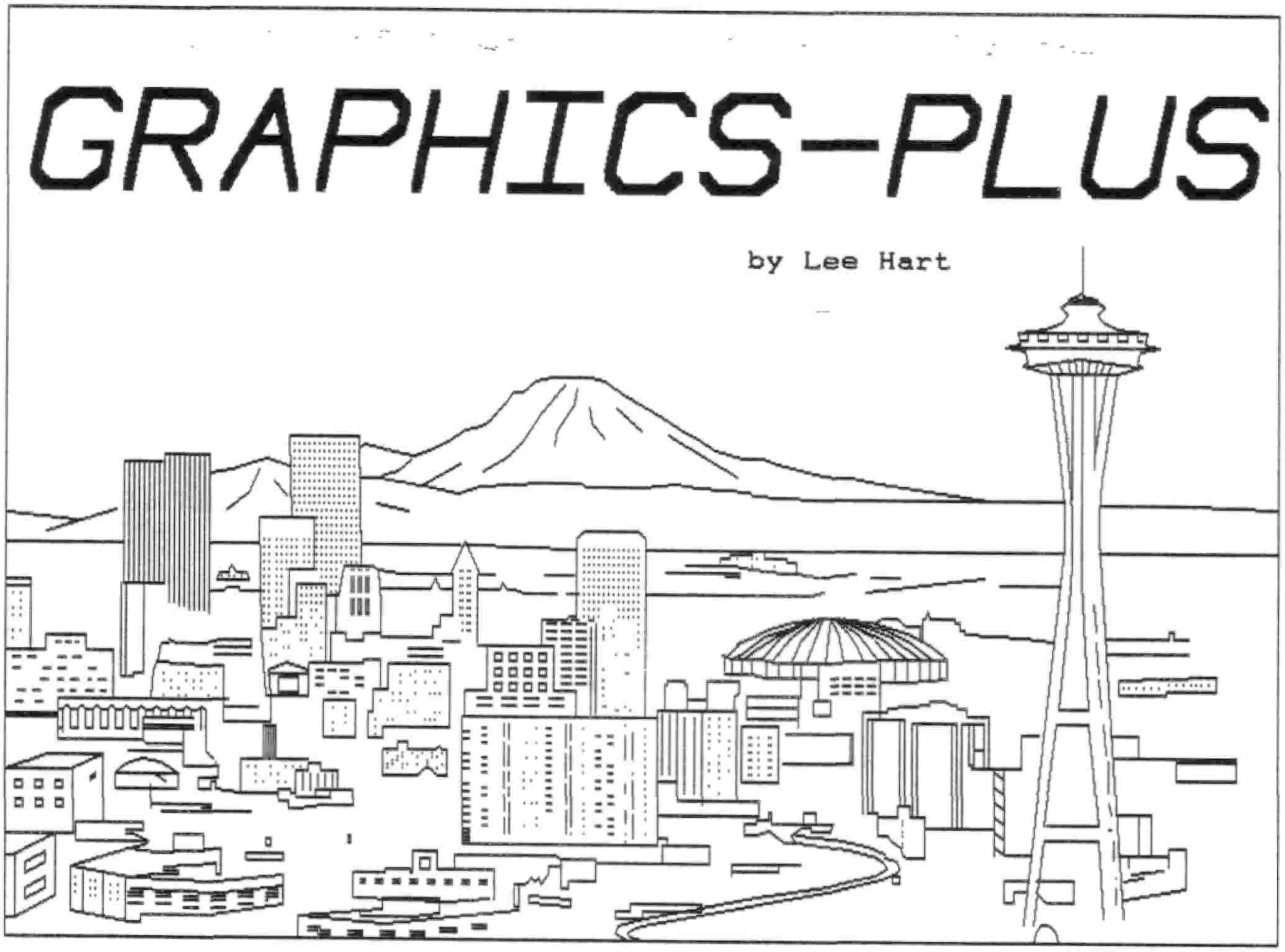
The obvious approach is to write a new version of BIOS-80 with the new formats. I talked to Ray Livingston on his experience. He found that the financial rewards were negligible compared to the amount of work. Even though offered at a very reasonable price, BIOS-80 may be the most widely pirated program in the Heath world - there seem to be more pirates than purchasers. Frequently a customer bought one, and passed copies out to all his friends. Ray is almost hostile toward HUGgies as a result. His advice was "don't do it; you'll get screwed."

So the HUGgies may have fouled their own nest. You thus see a direct consequence of software piracy. Ray is an excellent programmer, and made significant and lasting contributions to the H/Z world. Yet the attitude of those who think "a copy isn't hurting anybody" have lost us a friend. The H/Z community is poorer as a result.

So I am publishing this information as a kind of marketing study. Writing programs as tricky as disk drivers is tedious and exacting enough as it is; testing, debugging, and documenting them is HARD WORK. I need your feedback to tell me if I'm just wasting my time, or if someone really cares.

Lee Hart
366 Cloverdale
Ann Arbor, MI 48105
(313) 994-0784

"Wouldn't it be nice, Alice, if somehow we could freeze this moment in time!"



There are several graphics boards available for the H/Z89 (and H/Z19 terminal). The most common is probably the "Imaginator" by Cleveland Codonics. The one most heavily advertised is undoubtedly the Sigmasoft board. But there's another one that you don't hear too much about. It's the Northwest Digital "GRAPHICS PLUS" model GP-19.

Northwest Digital is a small, high-tech company in Seattle Washington. They specialize in high performance graphics terminals for mini- and mainframe computers. No "toy" graphics here! A typical model might have 1/2 megabyte of display memory, 1024x780 dot resolution, 16 frames for animation, color, grey-scale, and more. But at over $1000, not many hobbiests are interested. So Northwest Digital concentrates its attention (and advertising) on mini- and mainframe computers; the big users of such terminals.

The GP-19 was their first graphics board, and is still going strong today. It adds 512 x 250 dot graphics capability, with both Tektronix 4010 and easy-entry command modes.

Text performance is also considerably enhanced. The GP-19 has nearly all the features of the Super-19 and Ultra-ROM as standard equipment. In addition it offers up to 50 lines of 132 characters, programmable function keys, 16K off-screen buffer, true VT-100 emulation, "help" screens, 256 displayable characters, and more. The graphics and text in this article were prepared and printed with the GP-19.

The GP-19 Installation

The GP-19 (fig.1) looks very much like an H19's Terminal Logic Board. On-board is a Z80 microprocessor, 16K of dynamic RAM, and 4 EPROMs totalling 16K. It is very professional in appearance, and good quality parts are used throughout (except for the same cheap tin-plated IC sockets as Heath, a pet peeve of mine).

Installation in an H19 terminal is literally a ten minute job. The Z80 and 6845 CRT controller ICs are removed from the TLB and transferred to the GP-19 board. A 40-pin ribbon cable plugs into the now-vacant Z80 socket. Nine other ICs (that once generated the video display) are also removed from the TLB to save power.

The GP-19 installs in the card slot just in front of the TLB. The 40-pin ribbon cable plugs into a matching socket. The original power cable plugs onto the GP-19, and a short jumper cable connects the GP-19 to the TLB's original power connector. That's it; no power supply mods, extra boards, or cables like the Imaginator and Sigmasoft products!

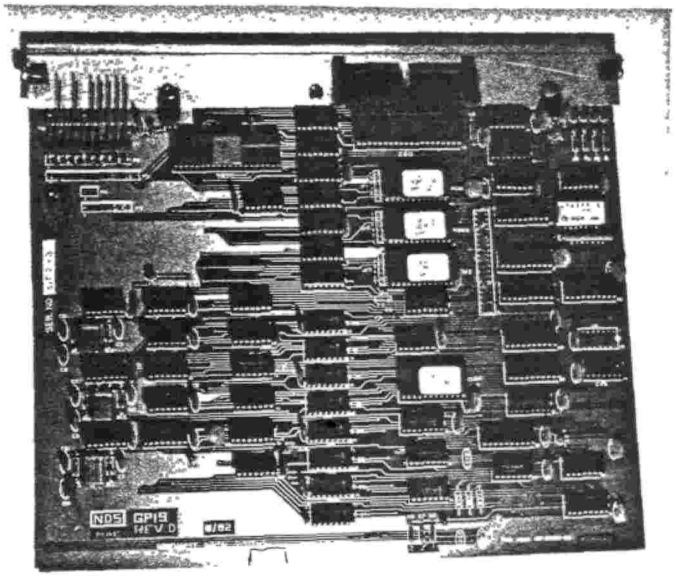


Figure 1 - the Graphics Plus GP-19

In an H89, the GP-19's slot is already occupied by the CPU board. So TMSI offers a mounting kit ($35) to add a third set of card guides. The GP-19 then fits between the TLB and CPU boards. It also works for the Imaginator which otherwise mounts horizontally on top of all the other boards. This blocks ventilation and gets in the way during servicing.

The GP-19 added 629 mA to my +8v power supply; less than the Imaginator, and much less than the Sigmasoft (but the latter comes with an extra power supply to handle the added load). My TLB alone was 1135 mA; the TLB plus GP-19 was 1764 mA. There was a trivial change in the other supplies.

A First Run

When you first turn on the computer, the familiar "beep" is gone. But four little red LEDs shine on the GP-19 board, visible through the top slots. These indicate that self-test is running. After 2-3 seconds, the lights go out and the cursor appears. Since the H89 CPU board was sending its "H:" at the same time the GP-19 was busy testing itself, the H: does not appear. However, you can "Boot" the system normally, and the DELETE key will display the familiar "H:" prompt.

Other than this, the GP-19 behaved exactly like a normal H19. Every program I tried, including games and graphics performed normally. If there are any compatibility problems, I have yet to find them.

Let's try the new features. Control-ESC displays the SETUP menu (fig.3). It's a full-screen display of the entire terminal state; baud rate, keyboard mode, graphics status, etc. To change a function, just move the cursor to it with the keypad arrow keys, and hit the space bar to toggle it to the desired setting. A second Control-ESC removes the setup menu and restores the previous screen display. No longer do I have to remember that "ESC x 2" disables the key click; I can do it in a flash any time, from within any program. The setup menu is so delightful to use that I wish someone had written it as a CP/M program long ago.

The standard H19 function keys generate the same code regardless of the shift and control keys; i.e. f1 is ESC S no matter what. The GP-19 allows four combinations: f1, shift-f1, control-f1, and control-shift-f1 are all distinctly different. The f1-f5, ERASE, BLUE, RED, and WHITE keys are all initially the same as a normal H19 for compatibility. But the normal and shifted functions are user-programmable. Page 2 of the Setup menu lets you program them to any desired character string. The keys can be progammed directly from the keyboard, or downloaded from a disk file.

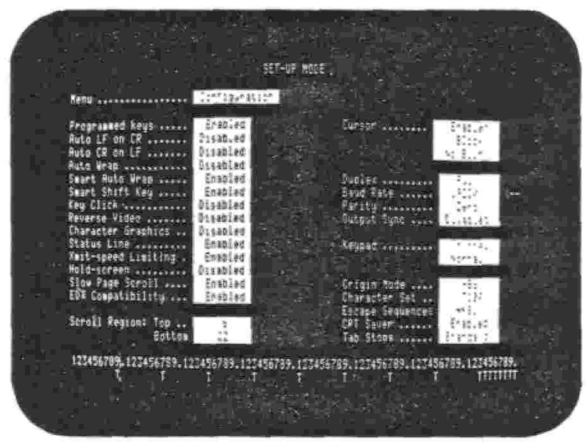


Figure 3 - the GP-19 Main Menu (1 of 3)

For example, create a file called MBASIC.SET which programs f1=PRINT, f2=LIST, f3=GOTO, etc. Now create the file MBASIC.SUB with these 2 lines:

```
type MBASIC.SET      ; program the function keys
MBASIC               ; run MBASIC
```

Now type SUBMIT MBASIC to program the function keys and run MBASIC. The same technique works for putting the GP-19 in any of its many modes to suit a particular program.

Figure 2 - the GP-19 Keyboard Decal

Yes, it Does Windows

The H19 has 2K of screen RAM. This is just enough for 25 lines of 80 characters. You can't have more lines or more characters per line because there is no memory for them. Likewise, when text scrolls off the top or bottom of the screen, it is lost.

The GP-19 has 16K of screen RAM. The screen is a "window" into this memory. It works like a text editor; you can scroll forward and backward to see any part of the last 16K that was sent to the terminal.

Suppose we type a long text file at the A> prompt. At the end, only the last 24 lines are visible. But the whole file (or at least the last 16K of it) is still in the terminal's memory. Control-f1 and control-f2 scroll up and down 1 line at a time; control-shift-f1 and control-shift-f2 scroll a page, and control-f3 and control-shift-f3 jump to the beginning and end respectively.

This effectively adds windowing to your software. For instance, type DIR to display a directory of your disk. Then type <RETURN>s to scroll it off the top of the screen. Now run your text editor. Most editors don't allow text to scroll off the top of the screen, so you can "peek" at the directory by just typing control-f2 to roll the screen down; then control-f3 to roll it back.

We can even insert this directory into the text we are editing. Position the directory on the top line of the display. Hit control-ESC for the Setup menu. Set Xmit Speed Limiting = Enabled (to slow the sending speed to 60 keys/sec so the editor can keep up). Another control-ESC removes the Setup menu. Then go offline and type ESC # (transmit screen). Our disk directory will be included by the text editor as if we typed it ourself!

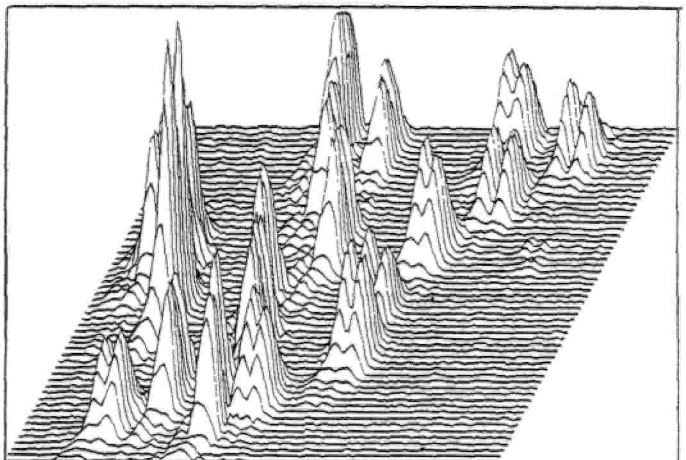
50 lines of 132 Characters

The extra screen memory can be put to work another way. Control-RED sets a 132 character by 24-line display. Characters are narrow, but fully formed and quite readable (like printing in condensed mode). Control-BLUE returns to an 80 by 24 screen.

Control shift-BLUE sets an 80 character by 50-line screen; control-shift RED is 132 by 50. Characters are sharp and clear, but there is a noticeable "flicker", like an old silent movie. That's because the 50-line displays run the H19 terminal in interlaced mode. This mode updates the screen only 30 times a second instead of 60. The flicker is aggravated by white CRTs, high brightness levels, and fluorescent room lights. A green CRT at low brightness makes it barely noticeable.

Programs run normally with these extended screens, but don't usually go past the 80th column or 25th line (after all, they don't know it's there). Therefore, the GP-19 Setup menu lets you define the top and bottom lines of a scrolling window; If you set the top to line 10 and the bottom to line 35, then your programs will "live" in that 25-line window, leaving the rest of the lines free for help messages, extra status lines, etc.

One delightful feature; Northwest Digital didn't invent new ESC sequences to control all these new attributes; they used the existing ones defined by the DEC VT-100. This is a very popular terminal, so many software programs know how to deal with it. I had no trouble configuring Wordstar, Vedit, Supercalc, and dBase-II to make full use of the extra lines and columns by just configuring them for a VT-100. Voila; a 132-column spreadsheet and a 50-line word processor!

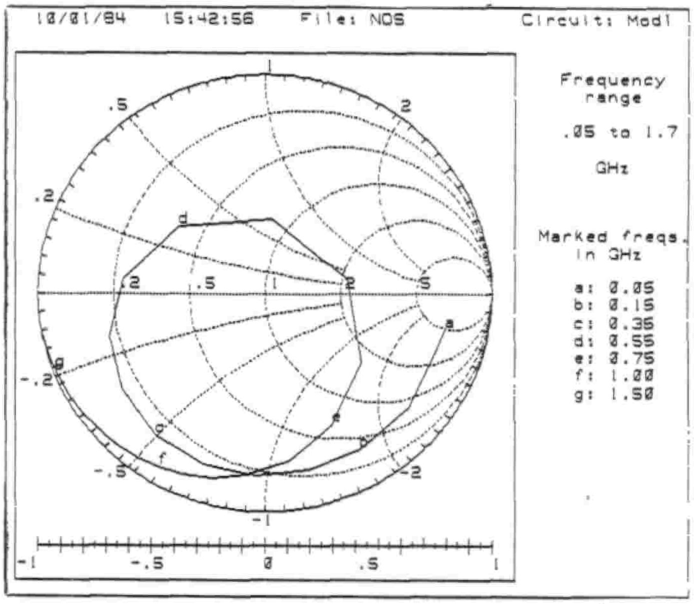


2D PROTON NMR SPECTRUM OF DNA
- Print Sample - Northwest Digital

Graphics!

Now it's time we talked about graphics. The GP-19 emulates a graphics terminal, the Tektronix 4010. The 4010 is a landmark product; it offered 4096 x 3120 resolution and drew perfect curves and lines with no "stair-step" effects. Its ingenious command language set the standard for every major graphics terminal today. Now consider that it was introduced 20 years ago, and you'll see why it influences high-end computer graphics to this day.

Control-WHITE sets the Tektronix mode. The GP-19 now behaves like a pen plotter. You position the "pen" with an x-y coordinate. Then give it a pen-down command, and another x-y coordinate. The pen draws a straight line connecting the coordinates.

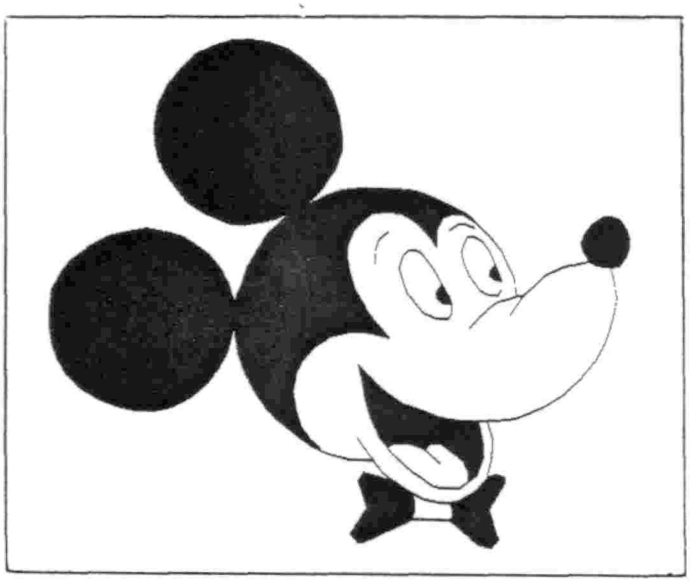


As an alternative to specifying coordinates with numbers, the GIN mode (hic) displays a crosshair cursor. The cursor is positioned with the keypad arrow keys, and then you hit ENTER to say "here!".

Northwest Digital has added some useful extensions to the 4010 as well. They include vector and block erase, block fills, and block moves.

The 4010 mode handles ordinary text, too. In this case, it acts like a 72-character by 32-line terminal. Characters are "drawn" on the screen as if by a pen. For instance, you can backspace over text and underline it, XXX it out, or strike it again to make it **bolder**, like a printer!

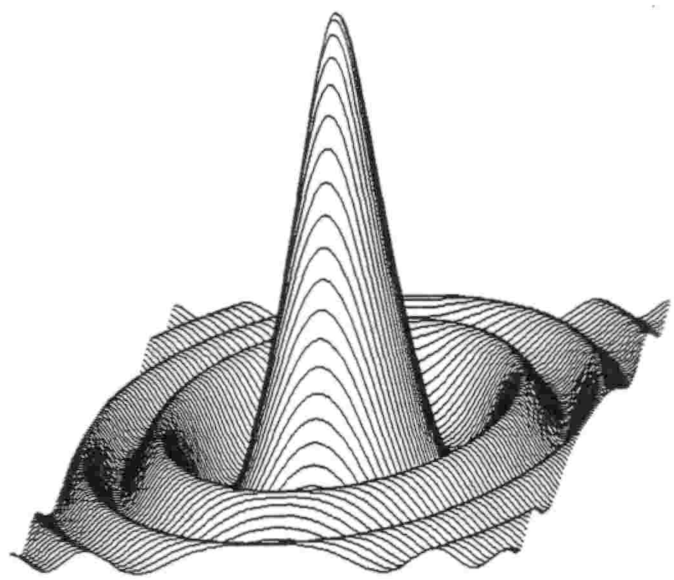
4010 commands are hardware independent; a 4010 picture looks the same regardless of what device made it. The x-y coordinates are based on an "ideal" 4096 x 3120 screen, not the actual 512x250 resolution of your screen. Like any 4010 device, the GP-19 scales these ideal coordinates to fit its actual resolution. Thus 4010 software is machine-independent; the program doesn't know (or care) what the actual resolution is.



Self-Portrait of the Author

4010 commands are not ASCII; they are 7-bit packed binary numbers. This is fast and efficient, but a 4010 picture file looks like machine code to a text editor. So the GP-19 also has an "easy entry" mode that uses normal ASCII characters.

Control-shift-WHITE sets Easy-entry mode. Commands consist of a letter identifying the command (P for point, L for line, M for move, etc.) followed by the x-y coordinate. For example, "L100,200" draws a line from the present position to coordinate x=100, y=200. It takes more bytes to define a picture, but they are easier to create and edit with normal text editors and BASIC programs. The "splash" graphic below was created with an MBASIC program in the easy-entry mode.



Comparing the Competition

As I've mentioned, the GP-19 has two main competitors; the Imaginator and the Sigmasoft board. While roughly equivalent, each product represents an entirely different design philosophy. How do they compare?

Northwest Digital's approach seems to be "keep it standard". They have tried (and succeeded) in emulating the industry's most popular terminals. This way, their features are immediately useful. The software already exists to support them, and future products will assuredly follow suit. When they added their own features, they were chosen to be usable without special programming. They thus avoided the need to write any custom software (except for the firmware already in ROM).

The Imaginator's hardware is very similar. But Cleveland Codonics is from the "we build hardware, not software" school. They wrote their own command language, but no software to use it. The commands are adequate to do the job, but no more. There are no text enhancements and you can't add a Super-19, Watzman ROM, or Ultra-ROM. The Imaginator would be lots more imaginative if they had software for it.

The Imaginator's graphic screen is independent of its text screen. This makes it makes it difficult to coordinate text and graphics, especially because the graphics screen is 20% narrower. The Tektronix 4010 is supported as a $75 option, but installing it sacrifices the easy-entry mode.

Sigmasoft takes an entirely different approach. It connects to the CPU board, and is I/O mapped. The hardware design is not very imaginative. There are easily twice the number of parts on 3 boards; an I/O card for the CPU, the graphics board itself, and a power supply board. Basically, it's a bank-selectable RAM card for the CPU, a portion of which can be made visible on the screen.

Their hardware has no high-level commands; every dot (or more precisely, every group of 8 dots) must be drawn individually. Since this is all done by your CPU board's Z80, it takes memory and execution time away from your running program.

Like Cleveland Codonics, Sigmasoft wrote their own command language. But they wrote the software to make it usable. It includes such unrelated functions as using the graphic memory as a RAM disk or print spooler. This makes an otherwise mediocre board truly useful. If you don't mind a unique language, the board has a lot to recommend it.

Sigmasoft says their graphics board is faster. Its 8-bit parallel port sends bytes to the screen at least 10 times faster than a serial port.

This is true if your application is plotting lots of individual dots, and the main CPU is doing nothing but running the graphics display. But using the terminal's Z80 is better if you will be plotting data from a running program.

For instance, an MBASIC program can calculate the data while the terminal is displaying it. In this case, both Z80s are running in parallel, and the job gets done faster. Similarly, the more high-level commands that are used (line drawing, area fills, etc.), the better the GP-19 performs with a separate Z80 doing the processing.

Complaints

Nothing is perfect, and the GP-19 is no exception. But the complaints are few and relatively minor. One problem is the interlace mode used for 50-line displays. The flicker is tolerable occasional use or in a dim room; for extended use you really need a long-persistence CRT to eliminate the problem (available from Langly St. Clair for about $100).

My TLB was modified to run at 3 MHz. At first the GP-19 didn't work right in interlaced mode. The culprit turned out to be the stock Hitachi 6845. Replacing it with a Motorola 6845 cured the bug.

The manual was complete and well written. It included parts lists, circuit descriptions, and schematics. But it was mainly a technical manual; it tells you WHAT it does, but not WHY you would want to do it. There are almost no examples. It took me months to gradually figure out what all the new modes and features are good for. In some ways it was like playing "Adventure" (Wave wand. Throw wand. Hit snake. Are you trying to deal with the snake? Yes!). A whole chapter could be written on configuration and learning to use the GP-19.

I found that handshaking was occassionally needed when erasing the screen or doing complex graphic operations. The GP-19 has hardware and software handshaking, but no instructions for adding it to your BIOS. I added hardware handshaking to my BIOS according to patches supplied by HUG's Pat Swayne.

Conclusions

I confess to being spoiled by the GP-19. The text enhancements have become a necessity of life. When I have to use an H89 without it, I'm lost without the programmable function keys, 132-column mode, screen dump capability, and off-screen memory. Other H19 enhancements such as the Super-19 and Ultra-ROM add some of these features, but not in as easy-to-use a form.

I rarely need the graphics, but when I do they are simple and straightforward to use. Lack of a good graphics editor is the main hindrance. I keep saying that someday I'll write one, but I probably never will. So til then I'll just enjoy the text modes, and dream of that pot 'o disks with 4010 software at the end of the rainbow.

Product: Graphics Plus, model GP-19

for H/Z-19 terminals, $495. Add $195 for printer port and screen dump. Order from:
Northwest Digital Systems
P.O. Box 15288
Seattle, WA 98115
(206) 524-0014

for H/Z89 computers, $545 with mounting kit and demo disk. $95 for screen dump software.
Technical Micro Systems
P.O. Box 7227
Ann Arbor, MI 48107
(313) 994-0784

⇨N O T I C E⇦

Single copy $2.50; 1-year subscription $15 as of 1 January, 1987.

Official Subscription Blank

Name____

Mailing Address____________________________________

City_______________ State__ ZIP_________ Country______

Phone number(s)____________________________________

Computer Type & Model #____________________________

Data-storage--H-17___H-37___H-47___H-67___Other__________

Accessories:_______________________________________

Operating System(s)_________________________________

Programming languages:_____________________________

Computer used mainly for:___________________________

Most liked vendor, publication________________________

Mail completed form to: SEBHC JOURNAL, 895 Starwick Drive, Ann Arbor, Michigan 48105, together with a cheque or money order made out in favor of L.E. GEISLER. Also indicate the back-issue reprints (if any) you're ordering at $2.50 each; indicate issue(s) of Volume 1 wanted. 1-[__] 2-[__] 3-[__] 4-[__]. Amount for reprint copies: $_____, plus Subscription $_____ = Total Amt $_____ enclosed.

----*----

Show your Journal to your H/Z Eight-Bit friends!

AND

* WHEN RESPONDING TO ADS, PLEASE SAY YOU SAW IT IN THE *JOURNAL* *

Thanks for your interest.

The SEBHC JOURNAL's Back Page

Society and Journal Policies

* The SEBHC JOURNAL is published twelve times a year and is mailed on or about the 22nd of each month. Editorial deadline is the 20th of each month.

* All advertising is printed free of charge. Vendors, please submit B&W "camera-ready" ad copy, seven inches wide by 9 inches high (one page/issue) no later than the 15th of a month in which it's scheduled to appear. Society Members get a free (new) 250-word want ad in each issue.

* Subscriptions are $15/year as of 1-Jan-87 in Canada, Mexico, the U.S.A. and possessions, and start the month following receipt of application. Make cheques or money orders payable to L.E. Geisler until further notice. Single back-issue copies now available on special order only--allow 6 weeks for processing.

* Subscribers automatically become members in the Society of Eight-Bit Heath Computerists. Member's ID number appears after their name on the JOURNAL mailing label. Any REGULAR member can vote and hold any Society office. * There are three classes of membership: REGULAR (H/Z 8-bit user), ADVERTI- SING (one vote only for each vendor), and ASSOCIATE (non-8-bit computerist, library, etc.). ASSOCIATE members do not hold office or vote in Society elections.

* The SEBHC JOURNAL is composed, edited and printed by Plain English Services (PES), and L.E. Geisler at 895 Starwick Drive, Ann Arbor, MI 48105. Phone 313-662-0750, 9am - 6pm EST M-F. Online "record-a-call" nightly, week-ends, holidays; message time about 50 seconds.

SEBHC Journal

895 Starwick Drive
Ann Arbor, MI 48105