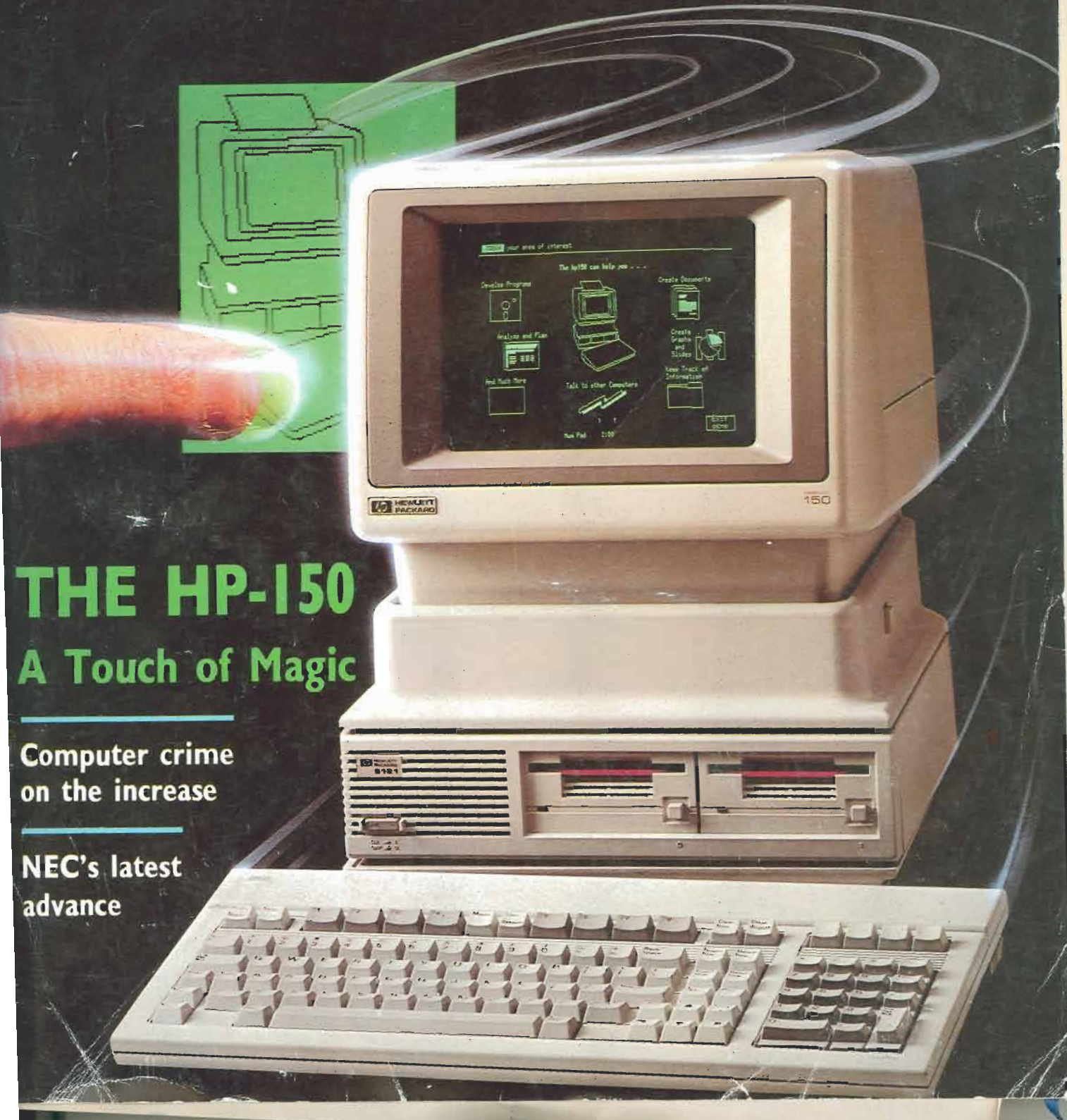
12490 ®

BYTE

the small systems journal

OCTOBER 1983 Vol. 8, No. 10
$3.50 in USA
$3.95 in Canada/£2.10 in U.K.
A McGraw-Hill Publication
0360-5280

UNIX ON MICROS



THE HP-150

A Touch of Magic

Computer crime on the increase

NEC's latest advance

The HP 150

Magic is the code name for Hewlett-Packard's personal-computer project in Sunnyvale—and it fits. Something magical happens when you use the HP 150. The optical touchscreen trademarked as HPTouch goes beyond other pointing devices; it makes you feel that you have remarkable powers in your fingertips. It's almost as if the touchscreen turns your finger into a conduit between your mind and the computer.

Hardware: Compact, Powerful, and Innovative

This compact machine packs the system-processor unit, memory, video-display unit and control circuitry, three I/O (input/output) ports, and touchscreen electronics into an elegant package that is 1-foot square. Two free-expansion slots permit a network-interface board and expansion to 640K bytes of RAM (random-access, read/write memory). An optional user-installable thermal printer fits in an enclosure at the top of the unit, with its own connecting cable. The dual-floppy-disk unit contains two single-sided Sony 3½-inch disks holding 270K bytes each and has a footprint not much larger than the main unit's. The HPIB bus used to connect the disk drives permits the controller to reside with the drives; no expansion slots are required to add additional floppy or hard disks.

The compactness of the HP 150 does have one drawback for personal-computer users who do intensive computation. There was no way to squeeze a socket in for an 8087 coprocessor. Early indications are that the HP 150 will be an open system with respect to both hardware and software, and perhaps someone will develop an 8087 board for one of the two expansion slots.

Photo 1 shows the back of the HP 150 with its various I/O connectors. In photo 2, part of the back of the system has been removed to expose the two expansion slots, one of which is occupied by a memory board. Photo 3 shows the system with the back removed. The motherboard sits one level above the two expansion slots and holds it own piggyback 256K-byte memory board. Above that, you can see the video-controller board. Photo 4 shows the system from the front with the bezel removed and turned around to face the camera.

The bezel contains the touchscreen electronics—the grid of light-emitting diodes and photo diodes. There are 24 holes in each side of the bezel and 40 holes in both the top and the bottom. This provides touch sensitivity for each row of the display and for each unit of two columns.

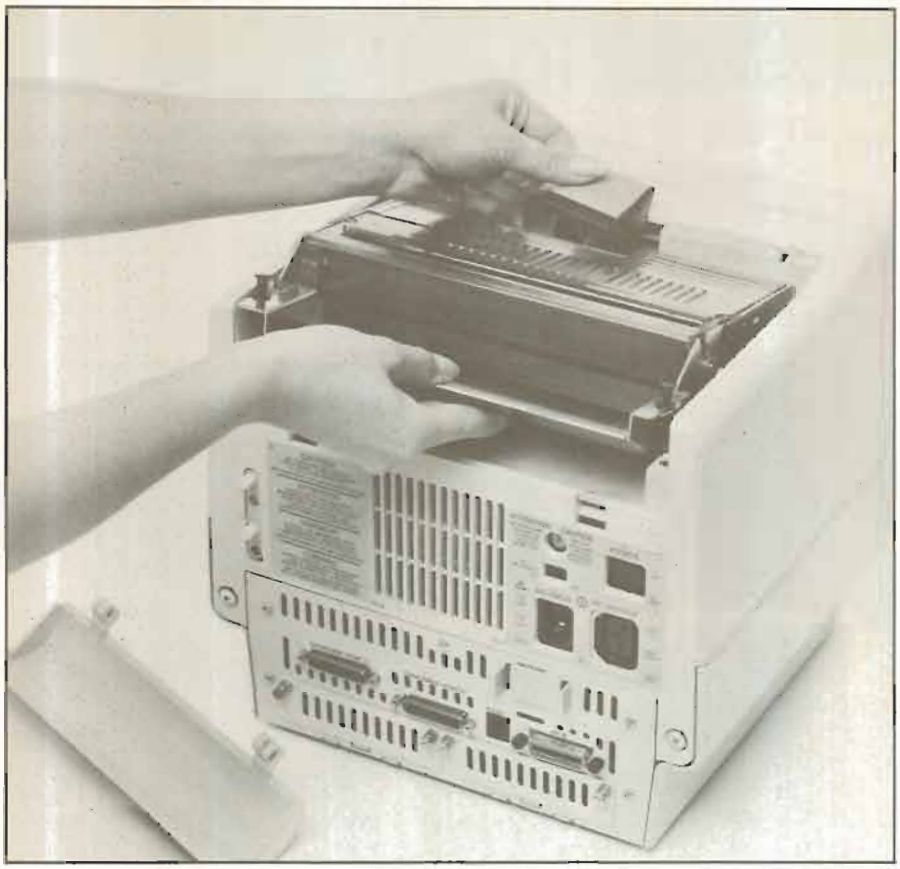
The touchscreen leaves no doubt that the HP 150 intends to deliver the power of personal computers to more people—nontechnical people. For HP, always known as an outstanding manufacturer of high-performance products for engineers, the 150 signals a bold entry into the broader market now dominated by IBM and Apple. Cyril Yansouni, general manager of HP's new personal computer division, confirms this interpretation and describes the 150 as the first of a new family of products. Yansouni also says that the HP 150 will be priced competitively, another departure for the company. A standard system with 256K bytes of RAM and two Sony 3½-inch disks providing 540K bytes of storage, MS-DOS, the Personal Applications Manager software, and Microsoft BASIC will retail for approximately the same price as the IBM PC with equivalent memory and mass storage. The HP 150's unique touchscreen and user interface provide the magic in an already powerful computer.

High Performance

Hewlett-Packard's engineers did not forsake their reputation for building high-performance products when they designed the 150. The HP 150's 8088 runs at 8 MHz compared to the usual 5 MHz or less, and the standard amount of dynamic RAM is 256K bytes. There are also 6K bytes of static RAM for the screen and 160K bytes of ROM (read-only memory), bringing the total memory for the standard machine to 422K bytes. One of the two standard RS-232C serial-communications ports also serves as a higher-speed RS-422 port, and several peripherals can be daisy-chained off the HPIB connector, permitting high-capacity mass storage.

Hewlett-Packard makes some magic

by Phil Lemmons and Barbara Robertson



Hewlett-Packard plans to introduce the HP 150 at the November Comdex show in Las Vegas and is already working with its dealers to provide all necessary support for that introduction, including major television and print advertising campaigns. According to Cyril Yansouni, general manager of HP's personal computer division, the company is establishing Personal Computer Centers for training dealers and end users. While these centers will not sell equipment or software, a professional training and marketing staff will be available to answer questions, conduct seminars, and refer prospective customers to dealers. Twenty of these centers are already open and 65 additional centers, located worldwide, are scheduled for completion by the end of 1983. For more information on the availability of the HP 150, or the location of a Personal Computer Center, call (800) FOR-HPPC.

Photo 1: *A back view of the HP 150. Note the two serial ports (DATACOM1 and DATACOM2), the port for HPIB bus (used to connect a series of disk drives and parallel printers), and the easily removed battery.*

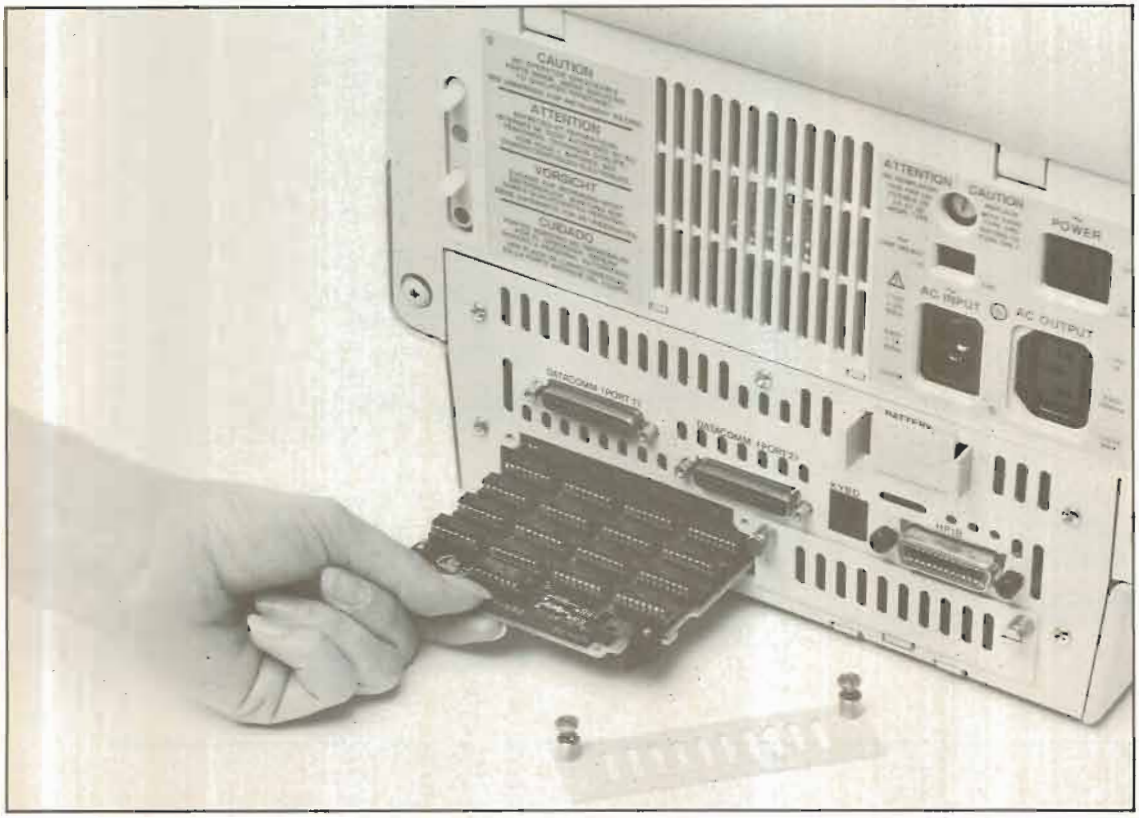


Photo 2: *The HP 150 comes with 256K bytes of RAM. This photo shows an optional 256K-byte memory board in one of the two expansion slots.*

Photo 5 shows the back of the 150 with its complement of I/O connectors and the optional thermal printer on top.

Available disk units include 5¼-inch and 8-inch floppy-disk drives as well as the 3½-inch disks, plus high-capacity Winchester disks. The Sony 3½-inch disks run at 600 rpm (revolutions per minute) rather than slowing down to the 300-rpm standard adopted by an ANSI committee.

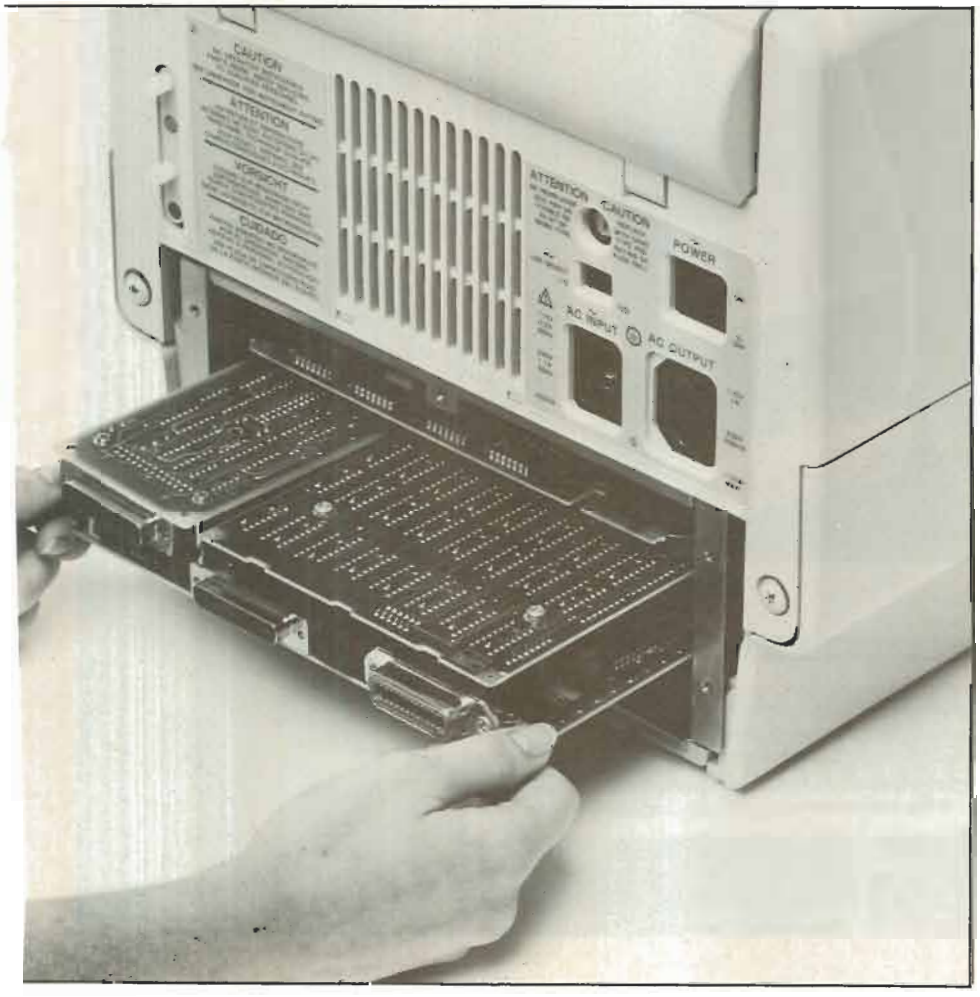
The performance of the video display is also outstanding. The 9-inch screen looks too small until you turn it on. The "At a Glance" box on page 41 shows the HP 150 screen displaying essential facts about the system in a format similar to that of a card in the Personal Card File, an electronic Rolodex-like program available for the 150; a lot of information is displayed quite clearly on the 9-inch screen. With a resolution of 720 by 378 as an alphanumeric display and 512 by 390 as a bit-mapped graphics display, the green-phosphor screen actually displays more pixels (picture elements) than the IBM PC 12-inch monochrome monitor, which has a 720 by 350 display. As an alphanumeric display, the HP 150's little screen provides 27 lines by 80 characters instead of the usual 24 by 80. Each character is formed by a 7 by 10 dot matrix in a 9 by 14 dot cell. The dots shift by half a dot to form clear characters, as the screen photos demonstrate.

The bottom line of the screen is reserved for system-status messages, and the two lines above that label screen areas programmed to perform specific commands when touched (softkeys). That leaves a full 24 by 80 screen plus the extra lines at the bottom for system enhancements.

Two controllers handle the video display. A custom gate array controls the graphics display, while the Standard Microsystems Corporation 9007 VPAC (Video Processor and Controller) takes charge of alphanumerics. As explained in "The CRT 9007 Video Processor and Controller" (April 1983 BYTE, page 96), the 9007 has powerful memory-addressing capabilities and flexible video-timing control. Its 30 registers include 12 to keep screen parameters and others for cursor control and light-pen operation. The 9007's memory-addressing power provides row-table-oriented memory addressing that relieves the system's central processor of the task of moving data on the screen.

The 9007 is partially responsible for the high performance of the HP 150's display in alphanumeric mode and for the gate array for the high-speed graphics, but systems software also plays an important role. Just as separate controllers control the two modes of the display, two separate software modules control screen I/O. The AIOS (alphanumeric input/output system) optimizes character I/O and the GIOS (graphical input/output system) optimizes bit-mapped graphical I/O. The GIOS includes routines for powerful graphics functions such as filling areas.

HP designed the keyboard on the assumption that people should be able to use the computer to its fullest without the touchscreen. The keyboard (see photo 6) has 107 sculpted keys, including cursor controls, editing keys, a numeric pad that can be shifted into a graphics-control pad, and eight programmable func-



Benchmark	HP 150	IBM
Empty Do Loops	6.13	6.43
Division	16.75	23.80
Subroutine Jump	11.80	12.40
MID$ (substring)	19.33	23.00
Prime Number	151.60	190.00

Table 1: *Benchmark results for the HP 150 against the IBM PC. The HP machine was running under MS-DOS 2.0, BASIC86, prerelease version 5.28. The IBM PC was running under PC-DOS 1.0, IBM BASIC. The benchmark programs are from "A Closer Look at the IBM Personal Computer" by Gregg Williams (January 1982 BYTE, page 36).*

Photo 3: *The motherboard holds a 256K-byte memory board on the left. The video-controller board is directly above the motherboard.*

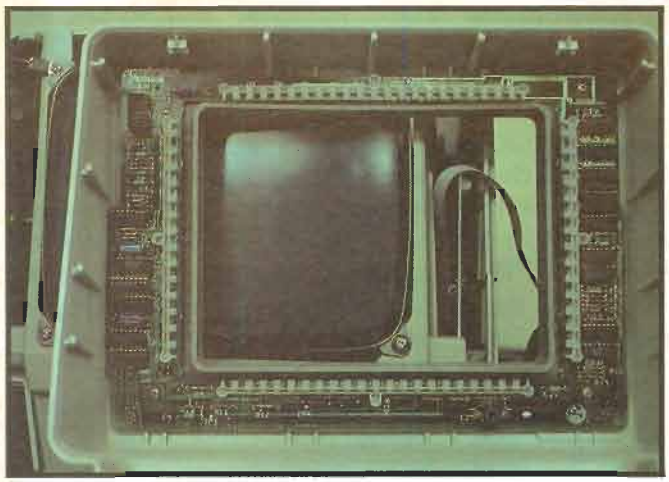


Photo 4: *The inside of the touchscreen bezel with its light-emitting and photo diodes. The system recognizes a touch when an object breaks the light beams crossing the screen.*

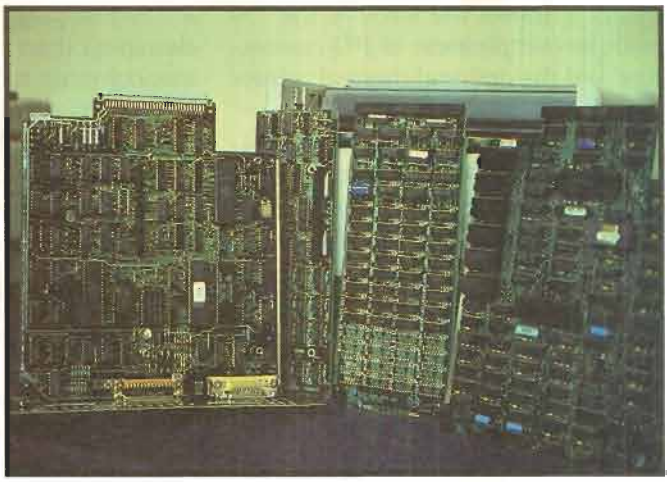
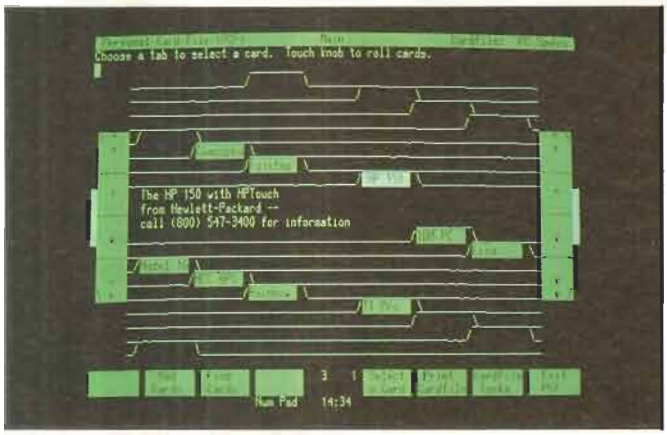


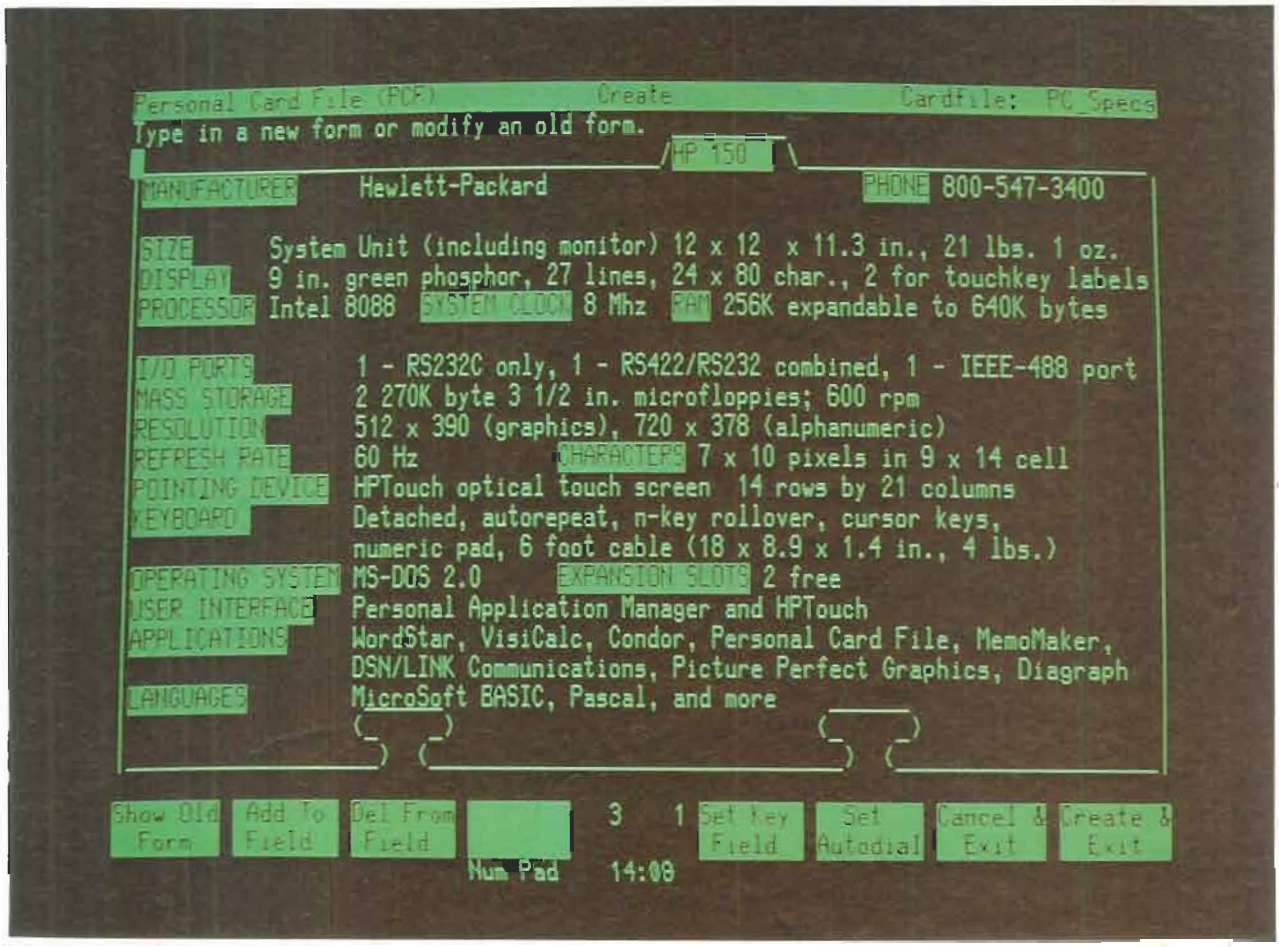
Photo 5: *These boards—the motherboard, piggyback and expansion memory boards, and the video-controller board—are all packed into the tiny monitor. In addition, two boards, the CRT sweep and power-supply boards, stand on end on each side of the video screen.*

Photo 6: *The HP 150 keyboard.*



At a Glance

Touch a tab in HP's Personal Card File program to look at a card at the card file.



tion keys. The layout is excellent. This keyboard will be the standard keyboard for all HP machines and terminals for years to come. An 8041 processor located in the system unit controls both the keyboard and the touchscreen.

An NEC 7201 controls the serial ports with a Texas Instruments controller chip handling the HPIB bus.

Both serial ports will operate at up to 19,200 bits per second (bps) as RS-232C ports and one will also operate as a higher-speed RS-422 port.

BASIC Benchmarks

Although it isn't possible to quantify the display's performance in this product preview, we did try five of BYTE's interpretive BASIC computational benchmarks on the 150, running a prereleased version of Microsoft's BASIC86 version 5.28. Table 1 shows the results. Not surprisingly, the HP 150 did well. In the prime-number benchmark, the HP 150 outdistanced the IBM PC by 38

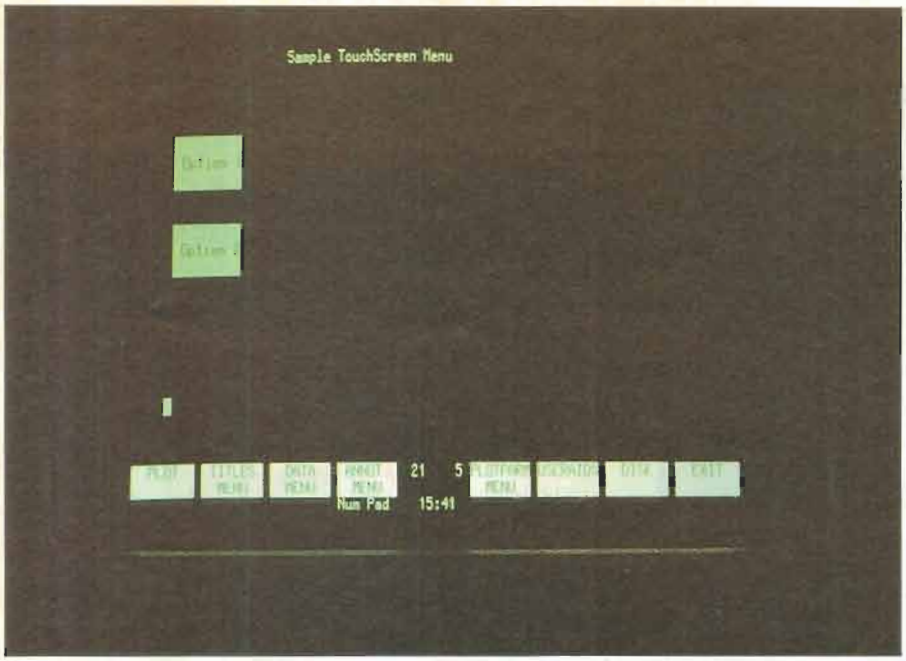


Photo 7: *Options 1 and 2 can be selected by touch. A simple BASIC program created this menu (see listing 1).*

seconds, even though the PC was running IBM PC BASIC, Microsoft's more advanced GW (Gee Whiz) BASIC, rather than BASIC86. The prereleased HP 150 was also significantly faster in the tests of division and string operations. The HP 150's 3½-inch disks performed well in simple disk I/O benchmarks— faster than all but a couple of the machines tested so far—but the results are not published here because HP plans to further improve the drives' performance.

Future products will expand the 150 family to include a compatible portable and transportable unit as well as a version with color graphics. Plans call for enhancement of the 150 family with faster clocks and a more powerful processor. The 150 family will also be able to communicate with non-HP computers through an Ethernet-compatible networking scheme.

The Touchscreen and Compactness

You don't actually have to touch the screen to make the touchscreen work because the beams of light pass slightly above the surface of the screen. One reason for choosing the optical touchscreen was to avoid coating the display screen with a material that would impair the sharpness of the display. Using a screen coat allows greater precision, but to take advantage of this precision you must point with a device much smaller than a fingertip. The smaller pointing devices seem to sacrifice the intuitive correctness of pointing with your finger.

While the 9-inch screen contributed to the compactness of the HP 150, it also reduced the size of the touch cells. You never have difficulty pointing to the defined touch areas at the bottom of the screen, or at the name of a file or program that you want to run, but a single character is difficult to select precisely. The cursor keys provide an easy alternative for fine movements, and pressing the select key selects the desired object.

The touchscreen also minimizes this problem in another way. The system recognizes a touch when your finger breaks the vertical and horizontal beams of light that cross above the object, and shows its recognition by displaying the object in inverse video. But a touch is not equivalent to a selection. The system only recognizes a selection when you withdraw your finger from the area and the interrupted light beams again cross the screen to the photo receivers under the bezel. You can move your finger around the screen for as long as you want, and the

A Potential User Looks at the Software

Phil Lemmons and I sat at his kitchen table late one night with the preview machine and prereleased copies of some of the software that Hewlett-Packard will offer for the HP 150. The touchscreen concept sounded interesting, but the only way to determine how it worked out was to use it.

Before trying any serious applications, we booted up the demonstration disk because I wanted to play with the graphics game. To create a drawing, you must first touch the screen in a least three places to mark the periphery. As I slid my finger around the screen, a small dot of light followed it. When I lifted my finger, the dot changed to a highlighted bar about the size of a typical cursor. This was obviously the first end point for the drawing. I selected several points this way, touched the label Draw Graphic in one of the eight function blocks (softkeys) at the bottom of the screen, and was fascinated as the drawing started from each point and filled in toward the center (photo 8).

I hadn't touched the keyboard once, and I had learned everything I needed to know about using the HP 150 touchscreen: to select a point on the screen, I lift my finger, and to start an operation, I touch one of the highlighted blocks in the row at the bottom of the screen.

Then I moved on to more serious work.

The HP 150 will be packaged with

system continues to highlight every object as your fingertip passes or touches it. When the desired object is displayed in inverse video, you simply withdraw your fingertip and the system acts on your selection. This visual feedback compensates for the limited precision of the array of touch areas. Whether the HP touchscreen will meet your needs depends on the precision you require. If you need to select individual pixels, HPTouch won't do. But the touchscreen takes care of much of the interaction during applications programs, making many system-level operations effortless and natural. Touch is the easiest input device to learn and the hardest to give up.

But can you use the touchscreen as an input device in your BASIC programs? Yes. Miles Kehoe of HP provided a quick example of a touch-sensitive menu. (See photo 7 and listing 1.)

If you wonder what it's like to use the touchscreen in more sophisticated programming, read the programming sidebar, "Adapting Existing Programs to Use HPTouch: Picture Perfect, Diagraph, and Wordstar" on page 48. It describes Micropro's experiences in adapting Wordstar (written in assembly language) to use the touchscreen and Computer Support Corporation's experiences in adapting the graphics programs Picture Perfect (written in BASIC) and Diagraph (written in Pascal).

—Phil Lemmons

MS-DOS, Microsoft BASIC, and a program called Personal Application Manager. I wanted to follow the process a new buyer would—booting the operating system, formatting some disks, and copying the master. From there I would look at some of the application-software packages that will be available for the machine, including some old friends that have been modified for the touchscreen and some new programs developed for the HP 150.

Listing 1: *This simple BASIC program will create the menu shown in photo 7.*

```
1000 '
1010 'Sample Menu Selection Sub-Program us
1020 'Provided by TLA. All Rights Reserved
1030 'use of these routines is permissible
1040 '
1050 WIDTH 255                          'Set 'infinite' screen width
1060 CLS$=CHR$(27)+"h"+CHR$(27)+"J"     'Home up, clear display
1070 '
1080 'FNLOCATE is similar to the LOCATE command in some BASICs
1090 '
1100 DEF FNLOCATE$(ROW,COL)=CHR$(27)+"&a"+
          STR$(ROW)+"r"+STR$(COL)+"C"
1110 '
1120 'FNTOUCH will define a touch field on the screen. The field
1130 '        will be three rows long and eight columns wide
1140 '        starting at the specified row and column. The
1150 '        character string to be returned is specified in
1160 '        the last parameter, and may be from 0 to 80 bytes
1170 '        in length.
1180 '
1190 '        The function as defined uses just one of many modes
1200 '        of touch field definition. The general form of the
1210 '        escape sequence is:
1220 '
1230 '     ESC  - z g <row1>,<row2> r <col1>,<col2> c
1240 '
1250 '            <0/1> b <onenh> e <offenh> f
1260 '
1270 '            <rptmode> m <attr> a  <len> L  <response>
1280 '
1290 '   where:
1300 '
1310 '       The 'b' field determines whether a touch will beep;
1320 '
1330 '       The 'e' field specifies the video enhancement of the
1340 '       field when NOT being touched
1350 '
1360 '       The 'f' field specifies the video enhancement of the
1370 '       field when it IS being touched
1380 '
1390 '       The 'm' field specifies the type of field (ie, row/col
1400 '       reporting, ASCII (as shown here), etc.)
1410 '
1420 '       The 'a' field specifies when to report (on touch, on
1430 '       release, or both). ASCII mode is considered a 'keyboard
1440 '       replacement', hence reports on touch only regardless of
1450 '       this field (just like a keyboard key)
1460 '
1470 '       The 'L' specifies the length of the desired response string
1480 '
1490 '       <response> indicates the string to return when touched and
1500 '       may include a carriage return if desired
1510 '
1520 DEF FNTOUCH$(ROW,COL,CHARS$)=CHR$(27)+"-zg"+
     STR$(ROW)+","+STR$(ROW+2)+"r"+
     STR$(COL)+","+STR$(COL+8)+"c"+
     "1b10e2f2m1a"+STR$(LEN(CHARS$))+"L"+CHARS$
1530 '
1540 REPT.MODE.ON$=CHR$(27)+"-z2n1A"    'Enable touch reporting
1550 REPT.MODE.OFF$=CHR$(27)+"-z0N"     'Disable touch reporting
1560 TOUCH.DELETE$=CHR$(27)+"-zD"       'Delete ALL touch fields
1570 '
1580 '  Display Main Menu
1590 '
1600 PRINT CLS$;                        'Clear screen
1610 PRINT FNLOCATE$(0,20);"Sample TouchScreen Menu"
1620 PRINT FNTOUCH$(5,5,"1");           'Define touch field in
1630 '                                   row 5-7, column 5-13 to
1640 '                                   return ASCII '1'
1650 '
1660 PRINT FNLOCATE$(6,6);"Option 1";   'Put label in field 1
1670 '
1680 '
1690 PRINT FNTOUCH$(10,5,"2");          'Define touch field in
1700 '                                   row 10-12, column 5-13 to
1710 '                                   return ASCII '2'
1720 '
1730 PRINT FNLOCATE$(11,6);"Option 2";  'Put label in field 2
1740 '
1750 'Now enable reporting and wait for a character
1760 PRINT REPT.MODE.ON$;               'Turn on reporting mode
1770 'Input single character from keyboard OR touchscreen
1780 A$=INPUT$(1)
1790 PRINT REPT.MODE.OFF$;              'Turn off reporting
1800 IF INSTR("12",A$)=0 THEN PRINT CHR$(7);:
                              GOTO 1760         'Did not type a 1 or 2
1810 IF A$="1" THEN CHAIN"PROG1"                'Selection 1
1820 CHAIN"PROG2"                               'Must be Selection 2
1830 END
```

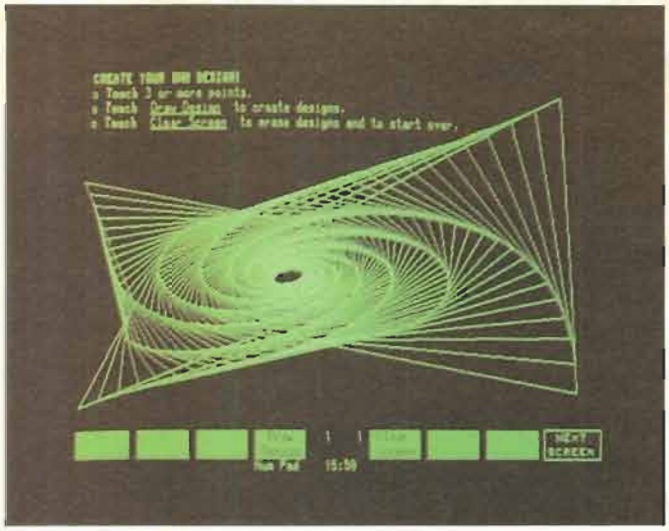


Photo 8: *A graphics-demonstration program by Hewlett-Packard for the HP 150.*

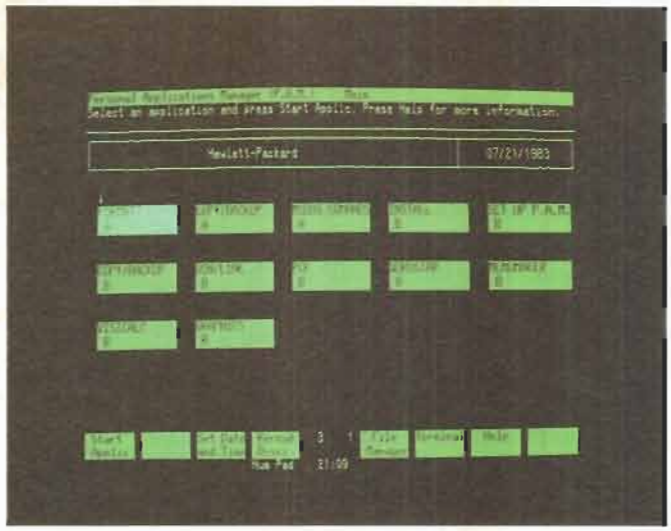


Photo 9: *HP's operating system shell. Touch an application name to select it, then touch "Start Applic." to run it.*

P.A.M.

When you boot the system, instead of the familiar A> from MS-DOS and a blank screen, you see Hewlett-Packard's Personal Application Manager, P.A.M., on the screen. P.A.M. automatically displays in alphabetical order the names of all the installed programs on the disk. Photo 9 shows the first screen with several installed applications displayed.

A small arrow pointed to the application named Format A, and it was highlighted, so I knew this was the default selection. I simply touched the Start Applic. softkey and the format screen appeared, with several bars indicating disk drives (photo 10). I touched the second bar for drive B, typed a label for the disk when prompted, then touched HP Format. A small asterisk appeared next to the label. When the formatting was completed, the Exit Format softkey returned the P.A.M. main menu.

Three touches to format a disk, and I didn't have to learn or remember any operating-system commands! The only time I used the keyboard was to type the optional disk-drive label.

Hewlett-Packard decided early in the project to use the standard MS-DOS operating system (although the company added some features—see the interview with Jim Sutton and John Lee on page 51) so that many popular software packages could run on the HP 150. To facilitate learning and use of the system, they created P.A.M. as a shell for the operating system.

The first P.A.M. screen (photo 8) demonstrates the convenient and simple user interface. This interface is common to all the programs Hewlett-Packard is offering for the HP 150: at the top, a program and menu name followed by a line for application prompts and messages, and at the bottom, a row of softkeys followed by a line for system messages and the clock. (Hewlett-Packard wasn't sure if the clock would be in the final version. Personally, I hope it is. I often lose track of time when I'm working on a computer.)

The softkeys replace traditional function menus whose items are usually selected by typing in code letters or numbers. In addition, each softkey in a row can lead to an entire tree structure of more softkey functions accessed by touching the screen.

The software discourages accidental selections. You must move your finger directly into the softkey area. Sliding to a softkey doesn't work. However, because you don't actually have to touch the screen, a finger hovering in the softkey area sometimes produces unexpected results. Selections are always highlighted for visual feedback, and in addition, when a softkey is selected, it clicks.

The function keys at the top of the keyboard mirror the softkeys on the screen. You can carry out any operation named on a softkey using the corresponding function key.

Now to copy the master disk. After I touched the Copy/Backup softkey, screen messages prompted me to type the drive letters to copy from and to. When I entered A: (copy from) the names of all the directories and files on the disk in drive A appeared on the screen. I chose the Copy File softkey when prompted; I suspected I could select any files I wanted to copy by touching the filenames, and in fact, I could. As I moved my finger from one to another, highlighting each in turn, the small arrow followed. Only names from which I lifted my finger remained highlighted (photo 11). The Start Over or Unselect by Name softkeys undo selections.

I wanted to copy the entire master disk, so when I finished playing with the touchscreen, I touched the Select All and Start Copy softkeys. The number of bytes available on the disk in drive B (in the counter at the top right of the screen) decreased until they matched the bytes available for drive A. The computer politely beeped when it was finished.

Four touches (and two disk-drive letters typed) from the File Manager menu, and I had copied the master disk.

In addition to the Copy Files func-

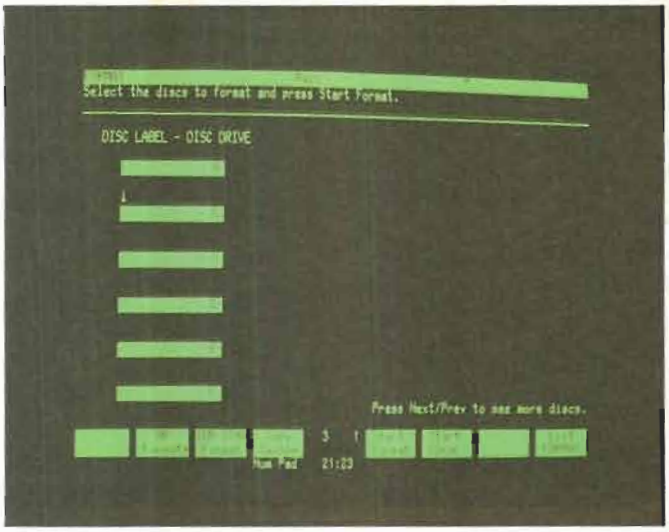


Photo 10: *You can format disks in several drives simultaneously.*

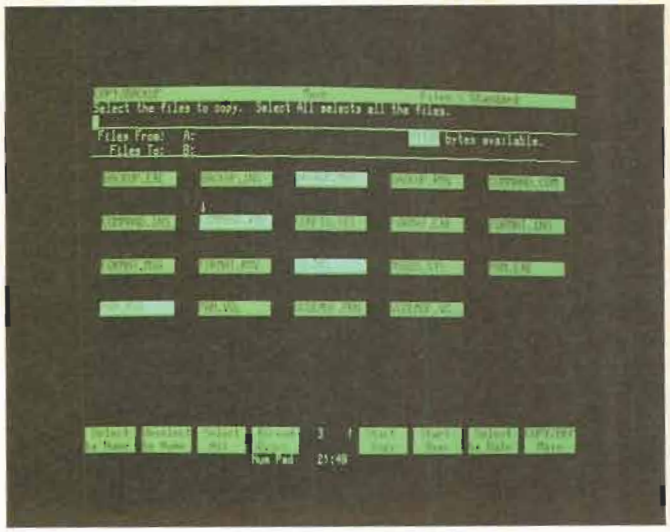


Photo 11: *Select files to copy by touching the filename on the screen.*

tion, P.A.M. has a Backup function that stores files in a compressed format. With this function, you can select files by name or date. The program has enough intelligence to know when there is not enough space left on a disk for the next file and will fill the remaining space with smaller files, then prompt you to change disks. Users with hard disks will find the Backup function especially helpful for archiving data on floppy disks.

The P.A.M. shell works. It simplifies standard MS-DOS functions such as formatting and copying disks. Enhancements such as automatically displayed directories are convenient.

All the programs offered by HP are automatically installed to run under P.A.M., and programs added later are easy to install. You can customize P.A.M.—changing application names, rearranging names on the screen, setting any application to start automatically—all by touching softkeys. Users who prefer standard commands can select the MS-DOS Commands application on the main P.A.M. screen.

When you have disks in more than one drive (the HP 150 can handle 12 disk drives), P.A.M. will display, alphabetically, the names of all the installed programs on all the disks along with the letter code for the disk drive. Disks in several drives can be formatted simultaneously.

The P.A.M. instructions are easy to understand, consistent, and predictable. I used the system without once consulting a manual. There's no need to learn how to use the pointing device to position the cursor: there is nothing abstract about touching a particular place on the screen with your finger. In contrast, most computer systems require users to learn, remember, and always type correctly cryptic codes such as "dir a:" and "copy a:filename.xxx b:filename.xxx" in a precise and initially mysterious format. Utility programs created specifically for a touchscreen are remarkably easy to use. I wanted to see how touch changed some old friends: Visicalc and Wordstar. They are among the applications that will be available when the HP 150 is introduced.

Visicalc

I touched my way back to P.A.M., selected Visicalc, touched Start Applic., and saw Visicalc on screen. Just as in P.A.M., the top line of the screen contains the program and menu names, the second line a message, and at the bottom is the familiar row of softkeys.

Rather than create a new worksheet, I chose one from the directory displayed in the File Manager. It's easy to move back and forth from any application to the File Manager. Touching the File Manager softkey puts the application on hold; the Back to Visicalc key sends you back where you were (see photos 12 and 13).

Of course, I immediately wanted to know if I could select a cell by touching it. I could, although it's a little tricky. It's easy to highlight the column you want, but positioning the highlighting on a particular row takes some practice. Still, it's often a lot easier than typing.

The softkeys contain most of the Visicalc commands, and I was able to use touch alone to move cells, replicate, open multiple windows, and select format and printing functions. Data can be transferred to a graphics program by touching a softkey (see photos 14 and 15).

Experienced Visicalc users may find the familiar slash commands faster than touching softkeys. But they will be pleased with enhancements like additional print functions (photo 16), cell formats and protection, and multiple windows.

Wordstar

My fingers did the walking back to the familiar Wordstar opening menu, touched a softkey to open a file, and typed a new file name. A screen, blank except for the softkeys, quickly appeared (photo 17). Having the Wordstar help menus on the screen would be redundant, so the help level is set to zero, allowing most of the screen to be used for text.

I couldn't wait to try my nemesis, a block move. I succeeded in mark-

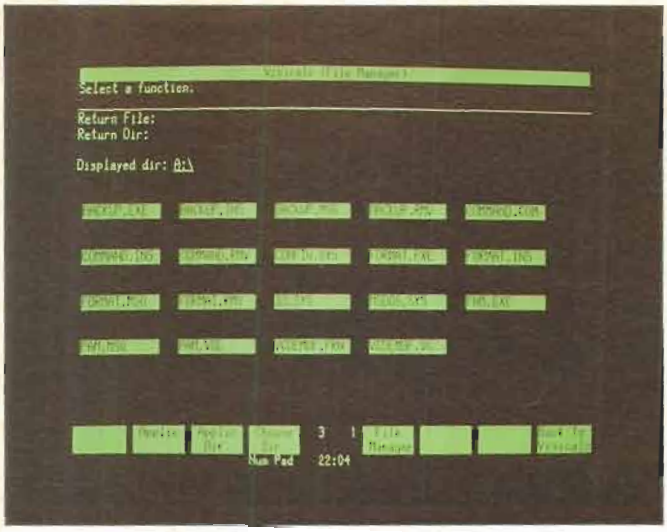


Photo 12: *You can move back and forth between an application and the File Manager by touching softkeys.*

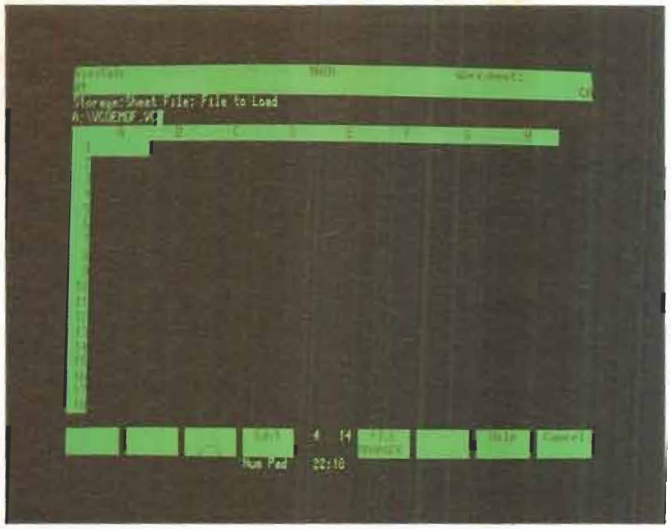


Photo 13: *A worksheet selected via the File Manager (see photo 12) being loaded into Visicalc.*

ing and moving a block of text with a few touches on the screen and softkeys (photos 18 and 19).

One of the most frequent objections to Wordstar is the difficulty new users have in learning and remembering the multitude of command codes. The IBM PC version 3.3 alleviates some of this problem by assigning 10 user-modifiable control codes to the 10 function keys and displaying labels across the bottom of the screen. The HP 150 version of Wordstar takes this idea a bit further: all the commands are on softkeys. Choosing a softkey label marked with lowercase letters produces more commands. Many of the softkeys lead to a whole tree structure of functions.

Typists who prefer not to take their fingers from the keyboard can use the inherent Wordstar commands or function keys. The keyboard has dedicated keys for common commands such as Insert Line, Delete Line, Clear Line, and Insert Character.

The ability to touch the screen to position the cursor makes many editing functions much easier. However, because the touchscreen is accurate only to a 2-character width (see the interview with Jim Sutton and John Lee on page 51), you may need to use the keyboard for exact positioning. Of course, you can't touch what you can't see, but the keyboard facilitates scrolling with Roll up/Roll down and Next page/Previous page keys.

In summary, the touchscreen improves Wordstar and Visicalc. Although these two programs are old friends to many of us, even the best of friends have a few nasty habits that we wish they could break. In the case of Wordstar and Visicalc, the nasty habit has been the refusal to understand anything but control codes and command strings. HPTouch has reformed these two old friends and made them much more agreeable.

Memomaker

You might well ask why, if Wordstar is available for the touchscreen, HP would offer a second word processor. HP's Memomaker is a simple word processor, compatible with Wordstar and designed for people

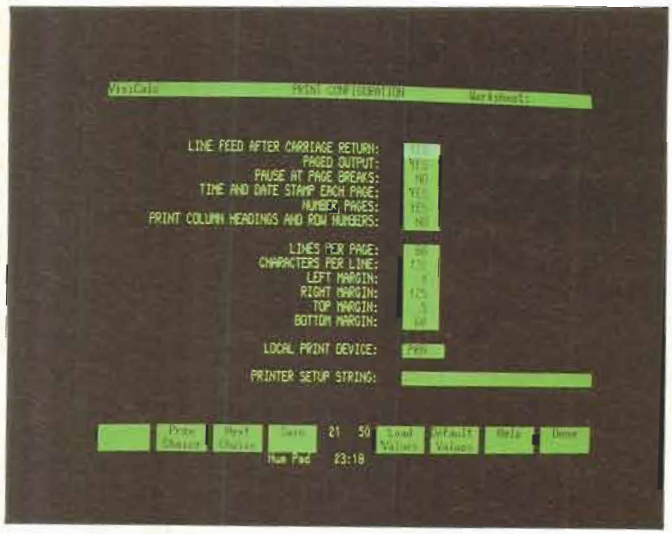


Photo 16: *Visicalc print options can be selected and changed with touch.*

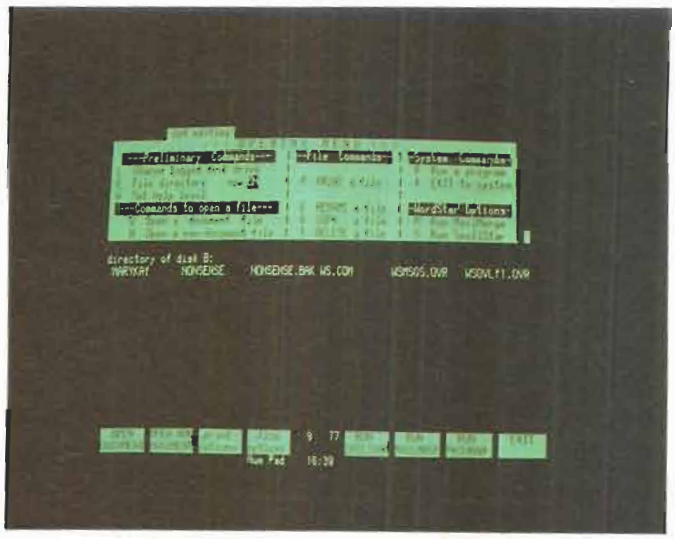


Photo 17: *Wordstar on the HP 150. Softkeys with lowercase letters lead to another layer of functions.*

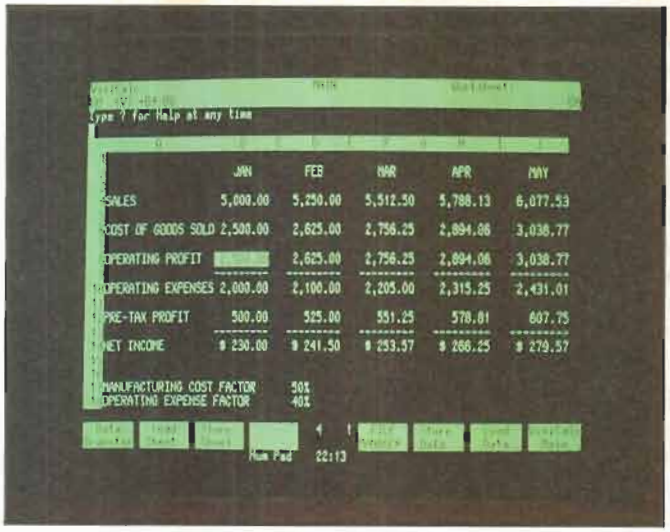


Photo 14: *To select a cell, touch it.*

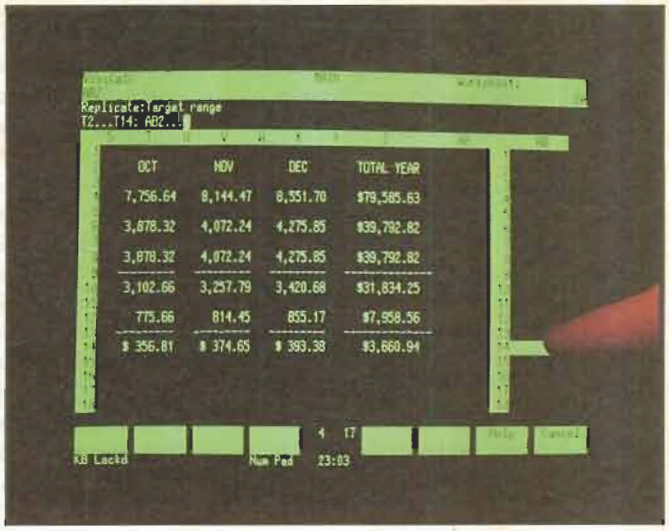


Photo 15: *HP's enhanced Visicalc allows multiple windows.*

who want a quick and easy way to enter text.

Even with the advantages the HP 150 gives to Wordstar, many people don't need or want to learn how to use a full-featured word processor. That's why HP offers Memomaker. With touch to position the cursor and select softkeys, and dedicated keyboard keys for functions such as inserting a line and deleting a character, you don't need to learn or remember any commands. Most people could use Memomaker fully—to write letters, create memos, and do rough drafts—without ever reading the documentation.

As in Wordstar, I used the touch feature to position the cursor and do block moves. I also tried changing the right margin and realigning the text. Touch makes these functions as simple as they should be. (See photos 20 and 21.)

I could go to File Manager, select files, and read them into Memomaker without ever using the keyboard or learning any commands (photo 22).

Memomaker shows text enhancements on screen—highlighting emphasizes boldface, and italic characters indicate underlining. Margins, tab settings, and standard memo formats can be stored in format files and read into a document.

Because Memomaker and Wordstar are fully compatible, documents created in one can be edited in the other. An experienced Wordstar user can add advanced features to Memomaker documents, while someone unfamiliar with Wordstar can use Memomaker to edit a Wordstar document. Memomaker is a great little word processor for people who don't want to do serious word processing.

Personal Card File

The demonstration program for the HP 150 includes a sample of a name and address program called Personal Card File (PCF). The screen for this program has a drawing that looks like a typical rotary card index found on many desks. To access a card in the file, you touch a tab as if the card file were made of paper and plastic; to rotate the card file, touch the handles—again, as you would with a

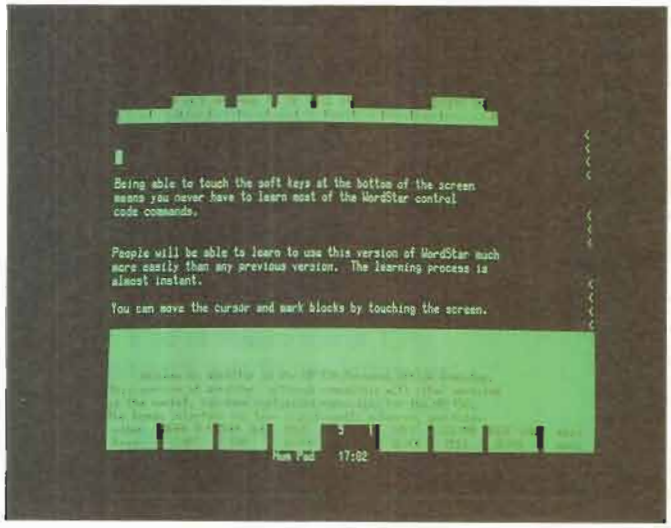


Photo 18: *The cursor shows where I touched the screen.*

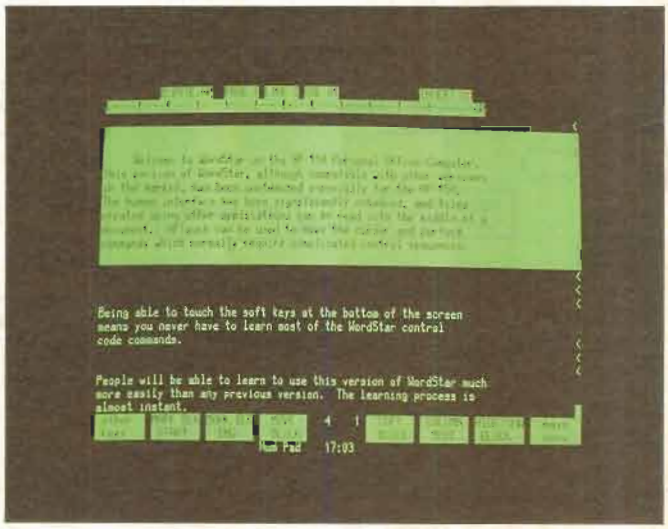


Photo 19: *The result of a block move accomplished entirely by touch.*

Adapting Existing Programs to Use HPTouch: Picture Perfect, Diagraph, and Wordstar

Two software houses experienced different levels of difficulty in adapting existing software for the HPTouch system of the HP 150 Personal Computer. Computer Support Corporation found it easy to adapt its graphics software, but Micropro had some problems because of the unusual internal structure of Wordstar.

Picture Perfect and Diagraph

Picture Perfect is a general charting program that makes it easy to do business graphics, such as bar charts with three-dimensional effects. Diagraph is a program that lets you draw flow charts, organization charts, circuit diagrams, and the like by using a library of approximately 100 primitives, including a variety of polygons and many common symbols. Computer Support Corporation of Dallas, Texas, is adapting two programs to run on the HP 150 and to use HPTouch. One program is written in Pascal and is being rewritten in the Pascal of the HP 150. The other program is written in BASIC and is being converted to the Pascal of the 150.

Michael Kallet and Jack Hudler of Computer Support explained how to use HPTouch this way: "You access the touchscreen through the HP 150's AIOS (alphanumeric input/output system), which is in ROM. You need to write some low-level assembly-language routines to access the AIOS. The same linker on the 150 handles both assembly language and Pascal.

"In fact, you access the touchscreen just as if it were any other input/output device. You call an assembly-language routine to set up the mode of the touchscreen that you want to use. We thought we would always be defining touch areas in terms of coordinates, checking for touches within the coordinates, and then going to commands that were mapped to the touch areas. But the touchscreen has a mode that makes the touch areas actually return a particular value just as if they were keys on the keyboard. Another mode lets the user define an object by touching the screen.

"The most useful mode is the one that makes the touch areas return keycodes. You turn that mode on and then set the touchscreen to report when it's been touched. Then you read the keyboard and wait for the return of a keycode and a qualifier. The qualifier says whether the code came from the keyboard or the touchscreen.

"Because of its different modes, the touchscreen is more powerful than you realize at first. You could use the row-column mode all the time if you wished, but you end up using the mode that directly returns a particular value.

"In interpretive BASIC, you can turn screen areas on and off with escape sequences. In Pascal, you make assembly-language calls to AIOS routines. Interfacing to HPTouch is easy in either case."

Adapting Wordstar

Since Wordstar is written in assembly language, Micropro is using assembly-language calls to the AIOS to adapt the popular word processor for HPTouch. Programmer Joe Masters reports, "For every character typed or screen area touched, you get information about which device the returned value came from, whether shift or control was pressed, and so on. This information comes to you from the console I/O portion of the operating system. The AIOS interprets the information.

"There are routines in the AIOS for writing a line to the screen, writing an entire screen, getting coordinates from the screen, and so on. The existing Wordstar makes multiple single-character output calls to write a line. With the HP 150, you can write an entire line at a time by calling a routine in the AIOS. It's difficult to get used to, but once you do, development goes much faster."

Kirk Hurford, manager of Micropro's OEM support group, explained the method of adaptation: "In adapting Wordstar for specific hardware, we go directly toward the I/O. Wordstar operating in the MS-DOS environment has much slower I/O because of the path that it takes through the operating system. In the case of the HP 150, the effect that we put forth to improve performance is sophisticated. There is a fair amount of intelligence with which Wordstar makes decisions as it's up-

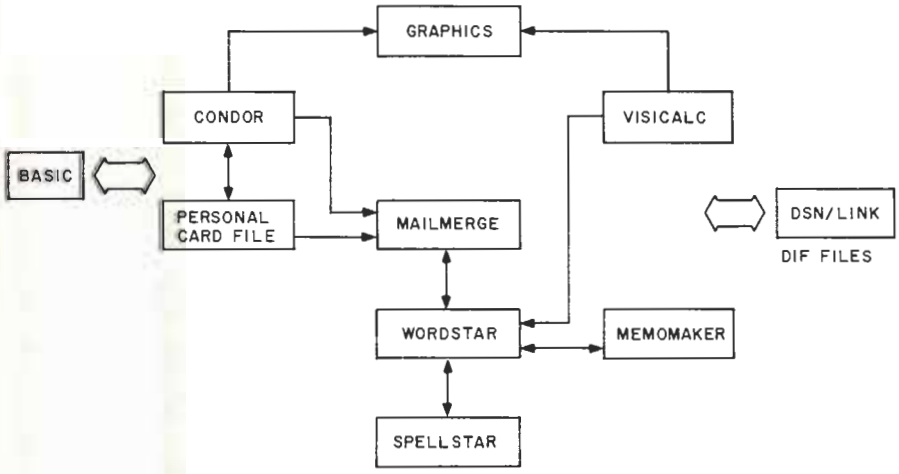


Figure 1: *This map shows the data transfer possible among the HP 150 applications.*

desktop cardfile. (The "At a Glance" box on page 41 shows the card file in action.)

PCF is designed to keep a handy list of names and addresses (which can be used in Wordstar form letters via Mailmerge). Key-field data appears on the tabs, and the file is sorted in key-field order. To search and select data for an abbreviated version of the card file, select fields and type the criteria.

PCF is not meant to be a database program, so all the data from PCF can be transferred to Condor, a relational-database program, for more sophisticated manipulation.

Graphics

With the high-resolution screen and a good selection of plotters

dating the screen. Wordstar uses different AIOS calls based upon what it knows it's going to do. When it's updating characters, for example, it uses different functions from those it would use when it knows it's going to write an entire line.

"Without the AIOS, we would not have achieved the high level of I/O performance that we have on the 150. The only other version of Wordstar that is as fast is the latest version for the IBM PC. That took three programmer months. Achieving the same speed on the HP 150 took five programmers weeks."

Masters made it clear that the hard part of adapting Wordstar for the 150 was on the Wordstar side, not the HP side. "The part of the adapting that has to do with the touch interface itself is self-explanatory. The documentation and a test program have examples of using the AIOS. But what we're doing on the Wordstar side is difficult."

Hurford explained why: "Wordstar is 14,000 lines of assembly-language code. The customization of the HP 150 is complex even though we get the information that we need from the AIOS instantly. It tells us where to go on the screen. The hard part is telling Wordstar how to get there."

"Wordstar identifies a screen location," Masters said, "not by referring to coordinates on the screen but as a 24-bit position in the file. There are lots of translations that have to take place to make Wordstar understand what the AIOS has told it."

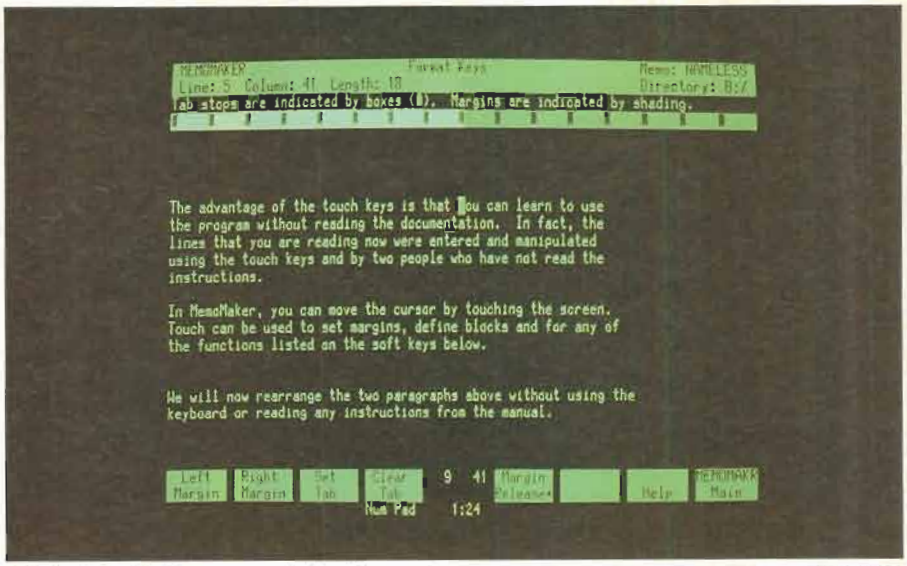


Photo 20: *All the commands for Memomaker are on softkeys. The cursor is positioned for a new right-margin setting.*

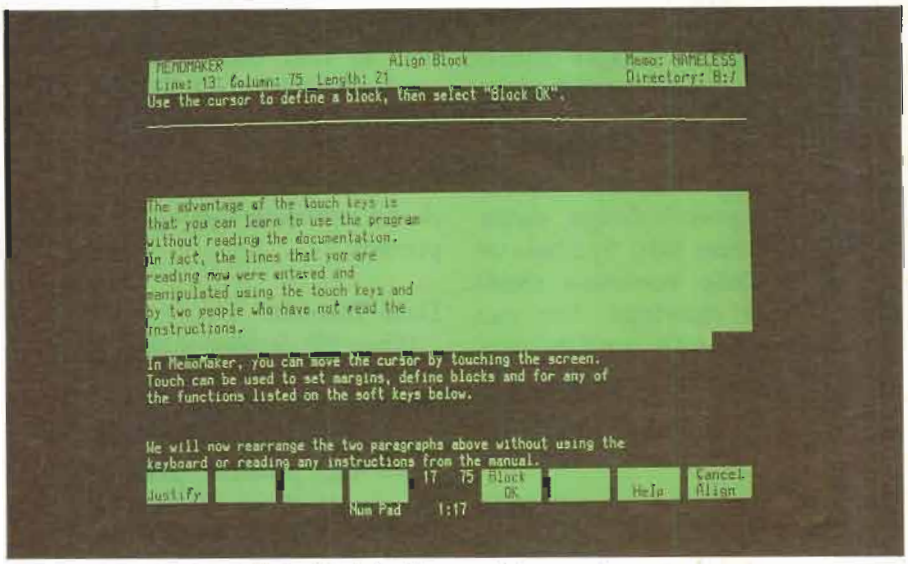


Photo 21: *Realigning a block of text for the new right margin can be accomplished entirely with touch.*

available from Hewlett-Packard, it's not surprising that HP will be offering three graphics programs for the 150—its own Series 100 Graphics plus two from Computer Support Corporation (Picture Perfect and Diagraph).

The Series 100 Graphics can use data from Visicalc or Condor to plot bar charts, line graphs, pie charts, and scattergrams on paper or slides. You can transfer data in, select pen colors and shading, choose options such as horizontal or vertical orientation, and pick paper type all by using touch. Photo 23 shows the Series 100 Graphics screen.

Photo 24 shows a bar chart created with Picture Perfect. Because of the HP 150's high resolution, the bar chart assumes an almost three-

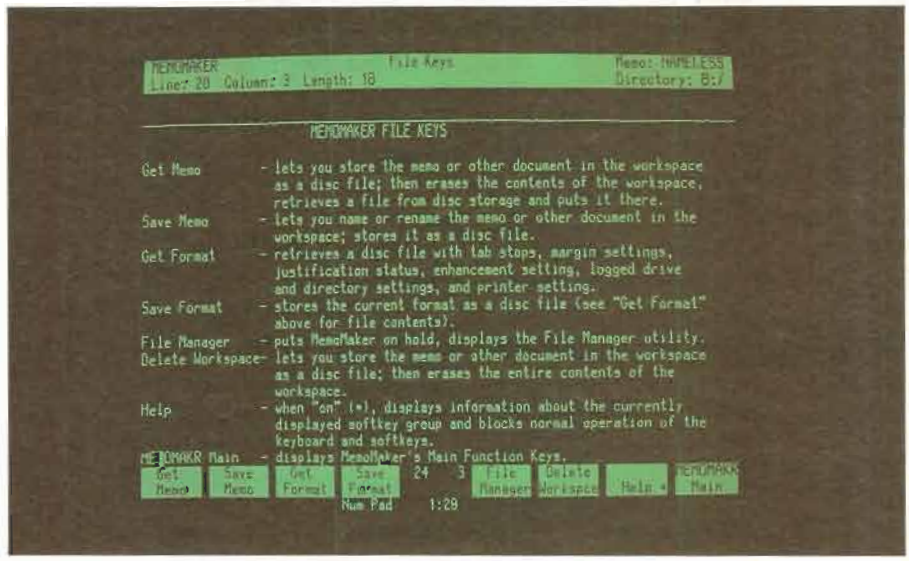


Photo 22: *You can store memo formats in a separate file and read them into Memomaker. The asterisk in the Help softkey shows Help is toggled on.*

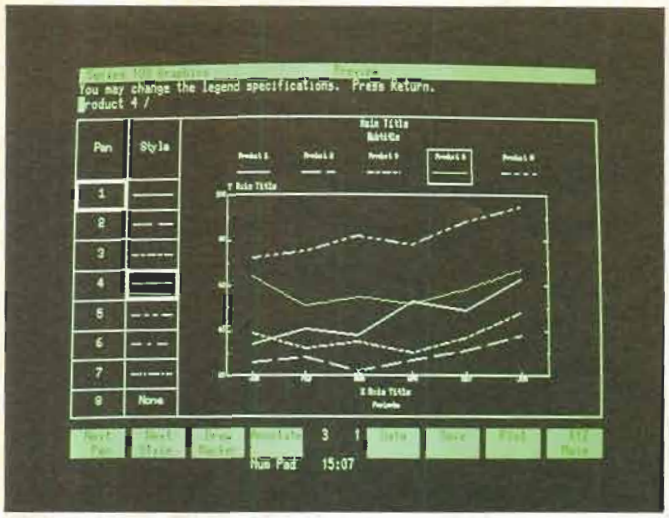


Photo 23: *Series 100 graphics from HP use touch to select pen color, style, and the functions you see on softkeys.*

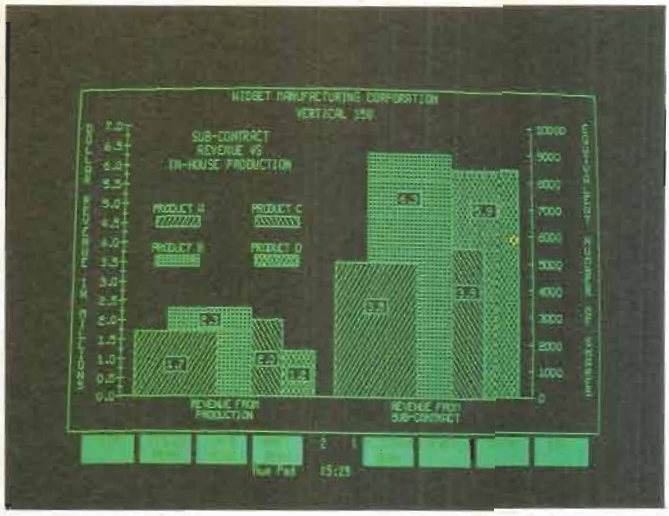


Photo 24: *A Picture Perfect bar chart demonstrates the high character resolution on the HP 150.*

dimensional appearance. Softkey menus can be used to define and adjust charts. Help messages contain key words that can be touched for additional help.

Diagraph can be used for flow diagrams, organizational charts, schematics, network design, slides, and presentation aids. Symbols on the screen help nonartists create sophisticated drawings. You can move, connect, and expand the symbols by touching the screen.

Communications and File Transfers

To use the HP 150 as a terminal for communications, simply boot up P.A.M., touch the Terminal softkey to access the four configuration menus, then, using touch or the keyboard, enter or change information to set a configuration.

Instead of typing configuration data, you can use touch to select data on the screen. For example, the "baud rate" prompt displays a new bit-per-second rate with each touch.

For file transfers, Hewlett-Packard offers the DSN (Distributed Systems Network)/Link program. DSN/Link can transmit both ASCII and binary data.

Transfers to the HP 3000 are easy and include error checking. Name the file you want to send or receive and initiate the transfer by touching the screen. Transfers to other computers require customization, but logon procedures and repetitive commands can be stored in a file, and a softkey may be assigned for those files so they can be accessed by touch. Transfers to HP 120s, 125s, and 150s can be made without a host-computer connection. Transfers can be unattended, and the program includes automatic logging to disk and printers.

Data Transfer

When HP decided to offer popular software packages on the HP 150, the company worked on facilitating data transfer between them. Figure 1 illustrates the possible data transfers among Wordstar, Condor, Visicalc, Memomaker, PCF, and graphics. (Jim Sutton and John Lee explain how they accomplished data transfer on page 51.) The data transfer isn't as extensive as it would be in a fully integrated program designed for this purpose, but there is probably enough movement possible for most applications, and most transfers can take place using softkeys alone.

What Makes It Magic?

It's not just the touchscreen that makes this machine magical. It's the combination of ease of use, sophistication, and low price.

The decision to use a standard MS-DOS operating system means software houses can quickly modify their popular software packages for the HP 150. Users familiar with these programs can easily switch to touch. Novice users will find difficult programs are easier to learn with the touch interface.

The modification of applications to use a common user interface gives a cohesiveness and predictability to all the programs on the HP 150. P.A.M., HP's solution to the MS-DOS A > interface, gives new users a place to start and makes utility commands convenient for everyone.

But users who wish to can ignore the touchscreen in P.A.M., or any other features that don't appeal to them. Few, however, will ignore touch. Pointing to a spot on the screen is natural, and HP has paid careful attention to preventing accidental selection by touch. Mistakes are less likely than with a traditional system that uses keyboard keys to issue commands. Anyone can quickly begin using this system and its applications.

The HPTouch interface isn't flashy. It lacks spectacular effects such as desktops full of icons and overlapping windows. These omissions don't cause any sense of deprivation; in fact, they help achieve a simplicity that contributes to the system's elegance.

Although elegance can suggest formality that often signals "don't touch," the HP 150 invites us to touch. That makes the HP 150 as inviting as it is elegant. Certainly the HP 150 represents real progress toward the goal of putting high technology at the disposal of ordinary people.

—Barbara Robinson ■

An Interview: The HP 150's Design-team Leaders

by Phil Lemmons and Barbara Robertson

Jim Sutton and John Lee led the design team for the "Magic" personal computer system, now known as the HP 150. Sutton is a research and development section manager in the personal software division. Lee is a research and development project manager in the personal office computer division. Both talked to BYTE West Coast editors Phil Lemmons and Barbara Robertson at Hewlett-Packard's offices in Sunnyvale, CA.

BYTE: What are the most innovative things about the HP 150's hardware?

Lee: The touchscreen and the compactness. The touchscreen is not a new input device, but getting it into the box and lowering manufacturing costs took a major effort.

BYTE: How did you go about doing that?

Lee: We researched the touchscreen and did some preliminary models, trying to get it to work with very inexpensive standard LEDs and photodiodes.

BYTE: You used the optical approach because of inexpensive parts?

Lee: Well, the optical approach means we don't have to put a special film on the CRT that reduces the contrast and visibility. Also, with the optical approach, you usually don't have the radiation problems that you get with other types of signals going across the screen. We also made sure our design provides ways of loading the parts so that each individual cell doesn't have to be adjusted separately. That reduces our production cost.

Sutton: One of the other issues in selecting that particular mechanism for having a touchscreen was to make sure it would be at least as reliable as the other components in the system. Some of the technologies that we looked at early on didn't seem to promise that kind of reliability. With this particular technology, the touchscreen would not be the first point of failure in the system.

BYTE: How did you decide what resolution the touchscreen should have?

Lee: As far as resolution is concerned, it's really the pointing device's resolution that matters more than the screen's resolution. The sharper the pointing device, obviously, the more precisely you can point. A light pen can get close to a pixel

resolution, but we considered having to pick up a special device with a wire attached to it less friendly. So our resolution requirement was the end of a pencil. That was the smallest size we could deal with. Normally you deal with finger size, which is larger than pen size.

BYTE: Did you decide to use touch and begin thinking of how it might be used, or did you originally look for another input device?

Lee: We were really looking for a friendlier way of interacting with the computer. We had already used labeled softkeys on the screen that match a top row of function keys. The softkey labels could change when the keys changed their functions, but their use was limited by the keyboard.

Sutton: Originally, touch was an optional feature. Over the course of the development of the product, we convinced ourselves first of the value of touchscreen and second of our ability to manufacture it for a cost low enough to make the touchscreen a standard feature of the product.

We also had some customers directing us toward the touchscreen as an input device. HP has a partner's program in which we work with certain major customers in a very active role and reveal to them some of our future products with an agreement that they won't reveal them. In return, they give us some valuable insights into the products. Some of these customers played significant roles in deciding on the touchscreen's role in this product.

BYTE: Did anyone argue for eliminating cursor control or programmable function keys, forcing people to go through the touchscreen interface?

Lee: I think it was originally designed so that you were able to do anything from the keyboard because at that time the touchscreen was optional.

Sutton: Nothing restricts you from using the keyboard as the only mode of cursor positioning. Customers who choose to do coding might use the keyboard rather than the touchscreen mode. But in some applications—for example, on a shop floor—there might be no more than a couple of dozen possible input choices. Then it would be perfectly reasonable to put up the 24 possible choices so that someone wearing a heavy leather glove could poke at the touchscreen and use it as the total input system.

BYTE: What was the process by which you designed the keyboard?

Sutton: A group was formed to try and unify HP's planning for keyboards, and it did a thorough study on the ergonomics of the keyboard.

BYTE: How did you do that study?

Sutton: I understand it was done by reviewing all of the external studies that have been done as well as reviewing the ergonomic standards established and particularly the European standards.

Our resolution requirement was the end of a pencil—the smallest size we could deal with.

BYTE: According to the International Standards Organization?

Sutton: Right. We also did a lot of testing with a number of mock-ups that we built. The tilt and swivel features of the main unit are because of the same international ergonomic considerations.

BYTE: Does moving to a different part of the world affect the touchscreen?

Lee: There are differences as you move into the Southern Hemisphere, where the magnetic fields are different and affect the convergence on the CRT. After we have converged it and centered it for the Northern Hemisphere, there is a shift when we take it to South America. The user can realign the touchscreen by using one softkey that brings up an alignment grid on the screen.

BYTE: What are you aligning at that point?

Lee: There is a set of holes you can see on the side of the plastic bezel. That's where the light beams come through. We put a grid on the screen in graphics, and you position the whole graphic screen to align with the holes. In fact, that would probably be necessary in the Northern Hemisphere if you dropped the machine and the yoke at the back moved a little.

Sutton: Our production engineers in manufacturing have been quite concerned with making sure we had an adequate solution to this problem. The need for adjusting the screen has been minimized, but if it becomes necessary it's easy.

BYTE: What was the most difficult part of the hardware design?

Lee: The compactness, fitting everything into the box. That—and also meeting all the regulatory standards set up by FCC and DBE.

BYTE: DBE is the German equivalent of our FCC?

Lee: Yes. HP also has its own environmental standards to meet.

BYTE: Those are more stringent than the FCC standards, or unrelated to the FCC?

Lee: The FCC deals only with radio interference. We have standards dealing with shock test, transportation, and electrostatic discharge (so that if you are charged and touch the box, you will not cause the system to reset). Actually, the most important thing for any computer user is data integrity. We try to make sure that nothing that the user can do will cause loss or corruption of data. Every bit counts.

Sutton: You were asking about the keyboard earlier. In fact, although we get our key switches from a very good vendor, and the vendor does extensive testing, we subjected all key switches to millions of keystrokes with mechanical hands, off-center keystrokes with millions of repetitions, and so on. This testing identified some problems so we went back to the vendor and collaborated until we got solutions to those problems. That's fairly typical. Usually we have standards internal to HP that are higher than any of the standards that come from regulatory agencies.

BYTE: What did you have to do to get everything into that little box?

Lee: Well, there are some design trade-offs. For example, not using an 8086, which would require a 16-bit bus instead of the 8088's 8-bit bus. In

general, when you do logic design you have to pick parts very carefully to make sure you don't use too many SSI (small-scale integration) parts.

Also, you make sure that the chips will be in sync. The 8088 needs a lot of support chips, and the major functional chips must talk well to one another so you can massage the signals going from one to the other.

BYTE: What sort of a CRT is necessary to get the resolution so high on a 9-inch screen?

Lee: The CRT itself is not the major problem—it's the electronics that drive it, the discrete analog circuitry. We run at about twice the frequency that people normally run, so our analog circuitry has higher frequency response requirements. That usually creates the much higher resolution you see on the screen.

BYTE: The graphics are very fast. What did you do to make them that way?

Lee: There are both hardware and software factors. First, we have one gate array that handles our graphics display. Second, there is a careful design of the algorithms that do the graphics. We have a very strong graphics group that understands the algorithms of doing vector drawing, area fill, and so on.

BYTE: The gate array plays the role of a video-controller chip?

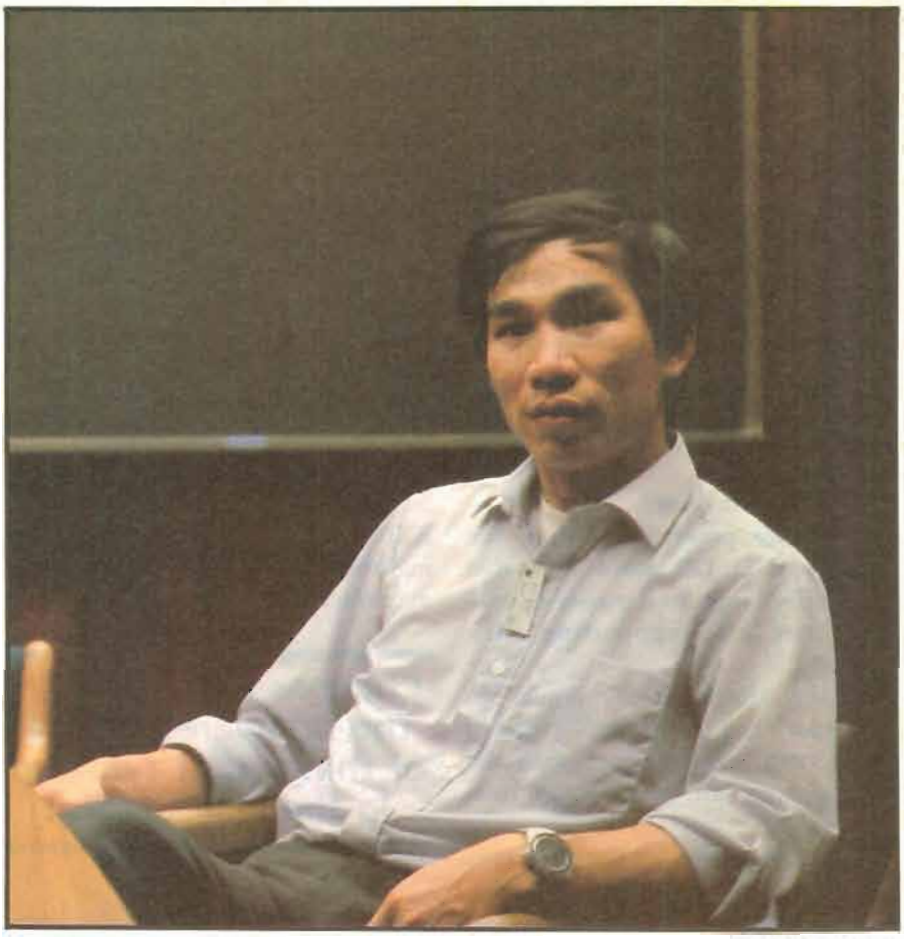
Lee: That's right, for the graphics.

BYTE: What about character-oriented I/O?

Lee: There's a separate controller. We have two planes, a graphics plane and an alphanumeric plane. A Standard Microsystems 9007 does the alpha control. It's a highly functional chip that replaces a lot of peripheral chips needed with the other controllers we have been using. The 9007 also does a lot of things itself without requiring the processor to support it. Even if the processor stops, the screen will not. The screen will be refreshed properly.

BYTE: Is some RAM (random-access read/write memory) dedicated to graphics storage, or is it all one continuous address space?

Lee: It's one continuous address space, but there are separate RAMs for the display because they need much faster access. We use some static RAMs.



John Lee, project manager for research and development in HP's personal office computer division.

Sutton: One of the other things that makes graphics fast from an applications point of view is the additional level of interface to graphics that is accessible without going through some of the overhead associated with the operating system.

BYTE: The GIOS?

Sutton: That's right. The graphic I/O system allows an application using this level of interface to do so without incurring some of the overhead that would otherwise be necessary.

Lee: That software interface also has some other very important purposes. Most computers actually go directly to the hardware in order to enhance their performance. That poses a very serious compatibility problem in the future, because hardware will change. The GIOS and AIOS (alphanumeric input/output system) interfaces give you high performance but are software, so future products can be made compatible.

BYTE: Looking at other controllers, what microprocessor is in the keyboard?

Lee: I guess it would depend on what you call the keyboard. The keyboard itself has no microprocessor in it, just a few random logic chips. We use an 8041 on the main processor board to handle the touchscreen and the keyboard.

BYTE: Tell us a little bit about the HPIB (Hewlett-Packard interface bus) and how that affects adding peripherals to the system.

Lee: The HPIB has been accepted as an IEEE standard and we conform to it. That bus provides the flexibility to add a lot of peripherals through one connector in the main unit. You can just tie in the whole daisy chain without any restrictions, except for the electrical and loading restrictions. You can add disk drives and plotters and printers through the same bus.

BYTE: What CPU chips did you consider, and why did you choose the 8088?

Lee: The personal computer market tends to be mainly concentrated on the Intel processor family. The Intel family was chosen so we could run an industry-standard operating system. Other HP divisions are developing systems based on other processors, focusing on the Intel family.

BYTE: Will the hardware be an open system? Will you cooperate with companies that want to make add-on products?

Lee: Definitely.

BYTE: Does that apply to software as well?

Sutton: Yes, it does.

BYTE: The power supply in the main unit is 120 watts. Isn't that unusually large for a power supply that isn't also supporting disk drives?

Lee: We computed the power requirement for up to 640K bytes of RAM, with all the processors, the video, plus an internal thermal printer. . .

BYTE: The power for the printer is in there, too?

Lee: Yes. The power supply will support the printer and we have some left over for the option cards. In fact, some of the option cards could take quite a bit of power.

BYTE: Can you print the screen image at any time with this system?

Sutton: Yes, that's true, but with one caveat. It's really printing the internal memory image, not the screen, as you might expect. There are separate planes for graphics and alphanumerics, and, generally, you can print the normal alpha you see on the screen and the graphics you see on screen, but if you are seeing alpha on top of graphics with both planes displayed at once, you cannot get those printed out and overlaid correctly.

BYTE: What manages the HPIB and the serial ports?

Lee: The HPIB is managed by an HPIB controller chip from Texas Instruments, and the serial ports are managed by the NEC 7201 serial protocol controller.

BYTE: Did you make changes in hardware because of things that came up in the software? Jim, was there anything you asked John to do so that you could do something you were having trouble with?

Sutton: Clearly we made changes in the firmware because of things that came up in the applications. John's role has been primarily the firmware. He's been very helpful to us in terms of being able to provide the right kinds of AIOS and GIOS calls for us to be able to get the very high screen performance that we want.

BYTE: John, you actually did the AIOS and the GIOS?

Lee: A lot of people got involved in that because it involves everything from the operating system to. . .

Sutton: However, the answer's yes.

BYTE: What about manufacturing methods? Is HP active in automated manufacturing?

Changes were made in the firmware because of things that came up in the applications.

Sutton: We are increasingly interested, as are all companies, in this area, to both improve our quality and reduce our manufacturing costs through factory automation. If you walked out through the back room you would see some strange-looking robots making their way among the various devices.

We also have manufacturing representatives involved at extremely early stages in the design of any product like this, to make sure that the product is easily manufacturable with the kind of quality that we like to have associated with HP.

Lee: Manufacturability is an issue that's addressed on day one. It affects some of the things that we do—how the boards are laid out, where the holes are and how many screws we need, and other such considerations. Currently I would guess our most automated process is building the PC board. Production is automated and so is testing.

BYTE: You already have disk drives in different sizes that use the HPIB, so all the 3½-inch, 5¼-inch, and 8-inch drives are available for the 150 at its announcement?

Lee: Right, all those drives, the Winchesters and floppies.

BYTE: How compatible is the 150 with IBM PC software?

Sutton: Software that was written originally for the IBM PC is transportable directly into our environment so long as it doesn't depend on special hardware features of an IBM device and uses vanilla MS-DOS calls. If it does use vanilla calls, then it will work equally well with our vanilla MS-DOS. Of course, our graphics resolution is somewhat different, so things that are written to make specific use of our graphics may not be directly transported to the IBM PC. Conversely, things written to make use of special features on the IBM PC may not be directly transportable to the HP 150. People willing to make the effort of using special features in the IBM PC environment may well like to use the special features in our environment. For example, to be able to get the kind of screen performance that we have.

BYTE: Will you be offering a Winchester drive unit with the same form factor as the double microfloppy unit?

Sutton: Yes. Those units will be available at first shipment of the 150.

BYTE: Did you consider developing your own operating system with perhaps an iconic interface and bitmap graphics, with your own applications integrated into that?

Sutton: Yes. We did a significant investigation and a reasonable amount of work on our own interface and an operating system that provided some substantial additional features over MS-DOS. But we viewed access to software already in the marketplace as critical to the success of the product. We decided to use MS-DOS for that reason. We would like to provide an extremely wide variety of industry-standard software at the same time we provide another rich set of our own software, which will be more valuable than competitive products in the marketplace.

Lee: Users should not really have to be aware of what operating system the software is running under. What they're more concerned with is how they're interacting with the system,

and that's really through a shell or a program that's running on top of what traditionally is called an operating system. I think we achieve the added user-interface capability without having to change the standard functions that MS-DOS provides.

BYTE: How would you describe the operating-system architecture? Where does the touchscreen fit into that?

Lee: The touchscreen is just another device in the system.

BYTE: Handled the same way the keyboard is?

Lee: Handled similarly to the keyboard.

BYTE: Where do the AIOS and GIOS fit in?

Lee: You can view them as an extension of MS-DOS functions. What we want to do is design compatibility for future products at the operating-system level. MS-DOS is a standard operating system and changing it would make it nonstandard. But we still want to define a variety of new functions that a standard system cannot provide. So all the new functions fit within the MS-DOS definition. What we do is go through the I/O control path of MS-DOS system calls to access all the AIOS and GIOS functions.

BYTE: What is the AIOS?

Lee: In a nutshell, what AIOS does is replace what current systems do to write directly to video. That's the main goal of AIOS as far as output is concerned. We want high performance and multifunction capability to write to video and much tighter control of what people see on the screen. We provide that functionality while at the same time hiding all the hardware dependencies of the system. We defined a logical interface that can be transferred.

Sutton: You can think of AIOS and GIOS as being at a lower level than MS-DOS. They bypass some of the things that would normally go on in terms of typical device output in MS-DOS. At the same time, the AIOS and GIOS are in essence at a higher level because instead of being character-oriented in terms of its output, for example, they can deal with large blocks of text going to the screen at once. As a consequence, you can bypass many of the inefficiencies of the usual way of getting to the screen, and at the same time do things in large blocks of data rather than a character at a time. The net result is multiplicative, making the actual screen transfer rates very high.

Jim Sutton, section manager for research and development in Hewlett-Packard's personal software division.

Lee: The input our system allows is very important. In a standard system input is single-character-oriented; hit one key, and you get one character. That is very limited information. For example, the up-arrow key is a non-ASCII (American National Standard Code for Information Interchange) key, so you have no standard ASCII representation of it in one byte. To give the application better control of how the keyboard is actually used, you have to enhance input capability.

BYTE: How is transfer to the screen handled under the GIOS?

Sutton: There isn't blocking of data, but there are high-level operations such as polygonal fill and vector drawing. These are all handled as single operations, so logically you might consider that blocking of data. I can specify a polygon and fill it, and I don't have to send all the vectors to do the fill. A variety of fill patterns and line styles are all built directly in as GIOS functions.

Lee: Most operating systems are basically designed so that you have a computer and two wires talking to a terminal. You view console I/O as talking serially to a terminal outside of the computer, and that basically is a low-bandwidth path. We wanted a path whereby data goes out in a parallel fashion. The AIOS and GIOS achieve that.

Sutton: We can frequently write the entire screen in the blink of an eye. With that capability I don't have to wonder about algorithms that will rewrite the precise part of the screen that's changed. We've found that using the I/O structure simplifies a number of the programs that we do. For example, in our extended version of Visicalc, we don't write the whole screen; we're more intelligent than that. We write columns when the screen scrolls because we can write a column as a block on the screen. And we can do the things very rapidly. A trick we haven't made use of, but which I'm sure is awaiting exploitation in some future product from us or an outside vendor, is to make use of the two separate planes

of screen memory. If you are doing something in a graphics application and all your menus are completely active in the alpha plane, bringing up a menu just consists of turning on the alpha plane. This means you can fill an entire screen with a touchscreen menu instantaneously. And as quickly as the person touches the item of interest on the menu, the menu can disappear, leaving the graphics intact on the screen. The I/O structure suggests a number of novel uses.

BYTE: You did your work in Pascal. Did you consider other languages?

Sutton: Actually, we did consider and used other languages for that matter. We actively considered using Pascal from a number of independent vendors instead of the Pascal from the HP 3000 and considered using C as a language.

BYTE: Why did you choose Pascal over C?

Sutton: There are a variety of reasons. We like to get a great deal of the work done inside HP in a standard language so that it's easy to port around to other environments. For example, our Visicalc will be available on the 3000. And that's because we were able to write the program once. Then with changes to the I/O structure going to the screen, we were able to make that product run on both machines.

To be frank, we also wanted to have control over the quality of the language. If things turned out to be generating code that was too large for us, we knew we could go in and optimize the compiler.

Some of the utility programs in the system are written in C, but for utilities specific to the 150, it wasn't important to have a language with great porting capability. It made more sense to choose a language for other considerations, such as generating smaller or faster code.

BYTE: Is P.A.M. the utility that you're talking about?

Sutton: Part of P.A.M. was written in C and part in HP Pascal.

BYTE: Will you be able to write programs that use the touchscreen in any language?

Sutton: Any language can use touch already. There's nothing magic about touch. The touchscreen is controlled at the simplest level by escape sequences. I send those escape sequences just as if I were, if you will, printing them. And if I know what the escape sequences are, I read in my manual the escape sequences to activate a touch area at some certain area, then I can do that in any language.

BYTE: How did you decide what percentage of the resources, including memory, would be devoted to ease of use? How important was ease of use?

Sutton: We wanted to differentiate the 150 from other products in this marketplace, and ease of use was one of the critical differentiating factors, which is one of the reasons, of course, that we settled on the touchscreen.

It has always been our goal to use popular software packages and to develop some of our own.

We have a fairly formal procedure at HP for passing external specifications through a rigorous review process before the product is actually started. But I don't think we made a conscious decision during that process as to what amount of resources we would devote to ease of use. Our most important constraints, I think, are memory and disk size because we want everything to run on our smallest-capacity disks. Those are important constraints, and some ease-of-use features do take machine resources. For example, we wanted to prevent the possibility of invalid input. I think it's quite easy to enter invalid inputs and have the machine not object to them in some of the products we see on the market. The machine may not blow up, but it will do something that is inconsistent with the inputs. To us, ease of use is important in things as small as that to things as large as using the touchscreen. It all takes effort and code and size.

BYTE: Did you know from the beginning that you wanted to use popular software packages on your machine?

Sutton: Yes; that has always been a goal.

BYTE: When did you decide to also develop some of your own software?

Sutton: That's always been involved in the machine also.

BYTE: To what degree do the applications transfer data among them?

Sutton: We have modified existing applications or worked with vendors to modify the applications to use the high-speed I/O and touchscreen features, and also, in a more limited way, to have the ability to move data back and forth. We've often reworked our own applications to facilitate the data transfer. For example, our own card-file package has been tested to make sure that its data can be transferred to and from the Condor database package. In the same way, we've made sure that our own Memomaker editor is able to transfer data between itself and word-processing packages from outside vendors. We facilitate the transfer by using outside vendors' file formats directly or by having easy ways to get between our format and theirs.

BYTE: You can move information between the Personal Card File and Condor and between the Personal Card File and Wordstar's Mailmerge program. Did you try for greater data transfer?

Sutton: Yes. Our inability to do a greater degree of data transfer is in some sense a limitation of MS-DOS. One reason we thought about building our own operating system was that we knew that with it we could give the appearance of integrating separate, independently written programs. We may yet be able to accomplish that through other mechanisms or through future releases of MS-DOS.

To transfer data among our own applications, we have a mechanism that might be considered similar to Unix pipes.

BYTE: What functions does the file manager provide for applications?

Sutton: Well, fundamentally, it provides consistency across all applica-

tions, whether in-house or proprietary. The user has a consistent way to get at data and files on the system. So, for example, using the file manager, a person can readily view any directory on any disk, see that directory sorted in alphabetic order, and by simply pointing to a file cause that file to become the object of interest to the application. In addition it provides more functions—in fact, better functions—than conventionally available in packages like Visicalc or our previously existing graphics packages.

BYTE: How did you decide what enhancements to add to Visicalc?

Sutton: We looked at a number of spreadsheet packages that are available in the marketplace today, including the Advanced Visicalc package. Then we compiled a prioritized list of features we would like to see in the package. From that list we chose the features that we could accomplish in the time frame that we had for the product. We expect to continue adding important extensions to Visicalc. Our special interest right now is how to tie spreadsheet packages to a larger network environment.

BYTE: How did you speed up software written in high-level languages?

Sutton: Our approach has been to develop the languages in a fairly straightforward approach in the 3000 environment and to bring those languages across to the 150. We used the results of that as a basis for analysis on where we could tune and improve performance. First and most obvious was to modify all the I/O to use the AIOS/GIOS to improve performance. In addition, monitors tracked time we spent on particular blocks of code. We used this to determine where, for example, our run-time library might need to be optimized to make it faster. Sometimes the compiler itself needed to be optimized in some ways. Sometimes we could change our coding practices and make use of particular algorithmic features. For example, our first version of Visicalc wrote the full screen in MS-DOS vanilla screenwriting. Our second version of Visicalc wrote the full screen using AIOS. The third version would rewrite only the column that would move the whole screen over and then write only the new column, both of which are AIOS functions.

BYTE: You've been working with lots of independent software vendors. Do you have a mechanism in place for looking at proposals from ISVs (independent software vendors) and from independent, one-man programming firms that want to write for the HP 150?

Sutton: We have a mechanism in place and a better one will certainly be on the way by the time this article appears. We are doing two things simultaneously. One is actively soliciting all the vendors of top software packages in the marketplace, and the second is being receptive to the ideas of smaller vendors who desire to participate. At the moment, we have an ISV cookbook that helps vendors use all the features we've described. We also offer vendors technical support and the physical hardware.

A special interest is how to tie spreadsheet packages to a larger network environment.

In the future we will have parallel R&D and marketing activities: a complete R&D lab within the personal software division, whose entire orientation is toward helping ISVs develop software for this machine, and a parallel marketing organization helping from the marketing side.

BYTE: What was the hardest part of the whole project?

Sutton: The hardest part of the whole project for me was the fact that a lot of things that are normally done serially were being done in parallel. There were huge numbers of critical and interrelated decisions going on at once.

BYTE: When were you convinced it would really work as you had imagined it would all work?

Sutton: I'm a true believer; I always believed it would. The only variable was the time.

BYTE: Did this project start before or after the consolidation of the personal computer division?

Lee: Before.

BYTE: And is this the first project of that division, or is this a project the division inherited?

Sutton: In a sense the division inherited it, but I think that the notion of the project was one of the things that caused the organizational restructuring of the company. I think that trying to decide your placement in the marketplace goes a long way toward dictating the structure of your organization, and that was true here.

BYTE: Was there a specific group of people who worked throughout the project, or were people pulled in and out of the Magic team?

Sutton: It was a little bit of both.

BYTE: How many people have worked on this project?

Sutton: About 50 R&D people are working on it now.

BYTE: Did people take the machine home to continue work, or did you keep it pretty well locked up?

Sutton: Lately people have been taking it home to work on. I know I took it home. My kids had a wonderful time playing with it. We will achieve great penetration in the 4- to 7-year-old market as well as in the business market.

BYTE: Was there a kernel of the team, a group of people who've been working only on this who will go on to another project as a team?

Lee: Not necessarily.

Sutton: I think in the software area that will probably be true. Our view of the software area has a longer range than this particular set of hardware. One of the purposes of forming the personal software division is to have compatible, consistent software across a range of products, both the products that are available now and in the future.■

Phil Lemmons is West Coast Bureau Chief for BYTE. Barbara Robertson is West Coast Technical Editor. They can be reached at McGraw-Hill, 425 Battery St., San Francisco, CA 94111.