

BYTE[®]

the small systems journal

JANUARY 1981 Vol. 6, No. 1
\$2.50 in USA/\$2.95 in Canada

• A McGraw-Hill Publication



HAND-HELD COMPUTERS

UCSD*

PASCAL

FORTRAN

PORTABLE

Develop on a Z-80†,
run on LSI-11‡, T. I. 990,
6800 or vice versa

EFFICIENT

Structured, readable
Speeds development X5
Easy maintenance

POWERFUL

Full standard Pascal
or ANSI '77 FORTRAN
plus extensions.
Concurrency, multiple
users soon.

COST-EFFECTIVE

Complete system with
compiler, editors, inter-
preter, assembler and
utilities from \$350.

NOW AVAILABLE

Operating System
Compilers
Programmer Utilities
Data Base Management
Word Processing
Business and
Medical Software
Games, Home & Hobby

POPULAR MICROS SUPPORTED

DEC, Radio Shack, Apple,
Microengine, CP/M§ based
and others

PCD SYSTEMS

P. O. Box 143
Penn Yan, NY 14527
315-536-3734

*TM Univ. of Calif.
†TM of Zilog
‡TM of Digital Equipment
§TM Digital Research

Editorial

The Hand-Held Computer

Chris Morgan, Editor-in-Chief

There's a new trend in personal computing today—the HHC (hand-held computer). For years computer aficionados have dreamed of a computer small enough to fit in one's pocket, yet powerful enough to do the sorts of jobs that full-size microcomputers do today.

Amazingly enough, the dream is coming true. There are now no less than four models (the Radio Shack/Sharp, the Panasonic/Quasar, the Hewlett-Packard HP-41C, and the Sinclair ZX80) that fall roughly into the ultra-small computer category. One might quibble with calling the HP-41C a "computer" rather than a programmable calculator, but it has all the necessary elements to qualify: memory, processor, I/O (input/output), and a full line of peripherals. Each of these computers is discussed in this issue.

Among the new crop of HHCs, the Panasonic/Quasar (reviewed on page 34) is perhaps the most impressive in terms of engineering innovations; it sports some features that many full-size personal computers don't have, such as the ability to run for long periods from battery power alone—an impressive achievement when you realize that the unit uses, not a CMOS (complementary metal-oxide semiconductor) processor, but a standard 6502! It also has such niceties as user-definable keys, a built-in real-time clock, uninterruptible storage of user programs, and the ability to produce color images on a color television (with the addition of an optional interface unit).

The Radio Shack HHC has its own attractions, including its (relatively) low price of \$250 and its surprisingly complete BASIC interpreter. The first time I saw the Radio Shack unit was at the West Coast Computer Faire last spring, where it was being shown in its original form from Sharp. I was intrigued, but I quickly concluded it was just a passing fad. Not until I used the computer at length did I begin to realize its potential. Here was a machine capable of running complex BASIC programs—and it was truly portable! (I have to admit that a lot of the fun connected with these units is taking them out of one's pocket and showing them to noncomputer people.)

What about the practical considerations of typing programs on such a tiny keyboard? Well, at first it felt awkward, but I quickly adjusted to it. (The Panasonic/Quasar is a bit better in this regard, because the keys are spaced more widely apart.)

Speaking of attractive prices, the Sinclair ZX80, for \$200 or so, has its own appeal. Strictly speaking, it's not a hand-held computer because it uses a separate AC adapter. Still, it's tiny and can be easily transported. It has become an overnight sensation in England. As our review on page 94 points out, the ZX80 has some bad characteristics, such as screen blackout during execution of programs. Even so, a student or other beginner in computer programming could learn a lot with this machine in conjunction with its introductory BASIC book (included in the purchase price), which seems to be very good.

Why all the sudden interest in miniaturization? In part, it's the logical culmination of the never-ending battle to put more and more capability into less and less space. Combine that with the recent Japanese trend toward miniature hi-fi components, and you begin to see the driving forces involved.

The Panasonic and Quasar Hand-Held Computers

Beginning a New Generation of Consumer Computers

Gregg Williams, Editor
Rick Meyer, Friends Amis
c/o BYTE
70 Main St
Peterborough NH 03458

Arthur C Clarke talked about them in his futuristic novel *Imperial Earth*. Jerry Pournelle and Larry Niven talked about them in *The Mote in God's Eye*. The subject is hand-held computers that can run programs, remind you of upcoming appointments, and serve as portable intermediaries between you and large, immobile, mainframe computers. Are they still science fiction? No, the hand-held computer is here—and for less than the price of some color televisions.

The HHC (hand-held computer) is a device about the size of a standard paperback book with two inches added to its longest dimension (see photo 1). Its weight is under a pound, yet it has the capabilities (when extended with portable peripherals) to do anything that existing personal computers do. The device, developed jointly by the Japanese corporation Matsushita (pronounced mat-SOOSH-ta) and Friends Amis of San Francisco, is being marketed in America by Panasonic and Quasar. Photographs in this article show both

It is impossible to lose the work you are doing by pressing the OFF key.

the Quasar and the Panasonic versions.

Description of the HHC System

The Quasar/Panasonic HHC is an integrated package of hardware and software that has the ability to do anything that other personal computers do. The HHC unit has the following characteristics:

- Dimensions: 22.7 by 3.0 by 9.5 cm (8 $\frac{1}{8}$ by 1 $\frac{1}{8}$ by 3 $\frac{3}{4}$ inches);
- Weight: 397 grams (14 oz.);
- 6502 microprocessor running at 1 MHz;
- Sixty-five-key keyboard with two-key rollover;

- 159 by 8 dot low-persistence LCD (liquid-crystal display);
- Uninterrupted storage of all user programs and other data through use of a unique "power-down" circuit;
- Redefinition of all keys during execution of an application program;
- Redefinition of all characters displayed on the LCD display and printer during execution of an application program;
- 2 K bytes of programmable memory, expandable to 4 K bytes internally or any practical limit (up to a theoretical limit of 4 megabytes) externally, by adding programmable memory peripherals;
- 16 K bytes of internal ROM (read-only memory) with sockets for four program capsules containing up to 64 K bytes of application programs or data (additional ROM, up to a theoretical limit of 4 megabytes, can be added externally through ROM peripherals);
- An internal real-time clock with a resolution of $\frac{1}{256}$ second;
- A built-in nickel-cadmium battery

pack that supplies all power to the unit;

- Internal shielding against RF (radio-frequency) interference in compliance with the new regulations from the Federal Communications Commission;

- An internal set of application programs that includes a four-function calculator, a free-form file system and editor, as well as several other functions.

In addition, the capabilities of the HHC are greatly extended by an integrated system of intelligent peripherals that include:

- A bus expander through which other modules are connected to the HHC;

- A portable thermal printer that prints 16 characters per line;

- A ROM extender that allows you to attach an additional four program or data capsules;

- A programmable-memory extender that allows you to add additional memory to the HHC;

- A 110/300 bps modem and telecomputing program through which the HHC can act as a remote terminal to other computers and to large information utilities and data bases;

- A cassette interface module that transfers data to a microcassette recorder at 1200 bps;

- A color television interface that allows a display of 16 lines of 32 characters each or up to 48 by 64 pixel (picture element) graphics in eight colors and black.

When connected to the HHC, all of the above peripherals can fit in a custom case the size of an average attaché case, or they can be interconnected to make a flat, rigid, easily portable combination. With the exception of the color television interface, the HHC and the peripherals can operate without connections to any outside power source, thus making the system truly portable and hand-held. Photo 2 shows the HHC and several of its peripherals.

Innovations in the HHC

The Panasonic/Quasar HHC embodies several technical breakthroughs. Without these developments, a computer as small and as powerful as the HHC could not have been built.

One of the most important innova-

tions in the HHC is the proprietary "power-down" circuit that allows the HHC to use the popular 6502 microprocessor in a hand-held device. In the past, manufacturers have designed hand-held products around microprocessors like the 1802. Such devices use a very small amount of current and can be powered by batteries, but they force the designer to use a slow microprocessor with a weak instruction set.

Designers have been prevented from using the more popular micro-

processors because of their high current drain: a conventional 6502-based circuit (using the same batteries as the HHC) would discharge them in about two hours. But, with this power-down circuit and additional hardware innovations, the amount of current needed to power the HHC in both its fully functioning and "off" (powered-down) modes is drastically reduced.

A related feature of the HHC is that when the OFF button has been pressed, the computer is still on. It is in a dormant state that uses only

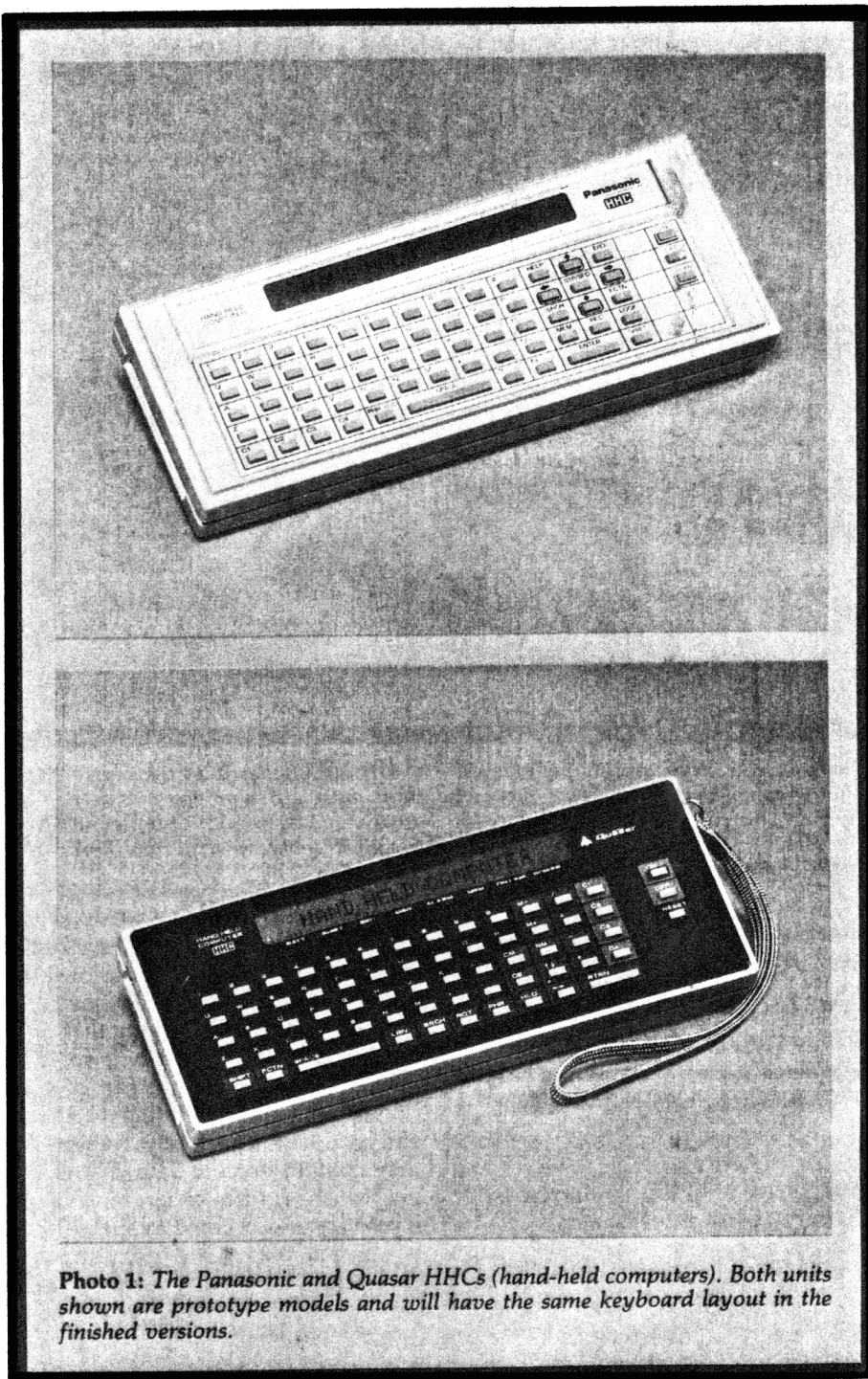


Photo 1: The Panasonic and Quasar HHCs (hand-held computers). Both units shown are prototype models and will have the same keyboard layout in the finished versions.



Photo 2: The HHC and its peripherals. The HHC computer is in the center of the photograph. The peripherals are (clockwise, from upper left): a programmable-memory extender, the color television interface, the I/O driver (a distributor of bus signals from the HHC to other peripherals), an acoustic-coupler modem, a portable printer, a cassette interface, and a ROM expander.

All functions are selected via a set of nested menus.

enough current to retain the contents of the HHC's display image and CMOS (complementary metal-oxide semiconductor) memory and to preserve the real-time clock and keyboard functions. A side benefit of this feature is that it is impossible to lose the work you are doing by pressing the OFF key; when you press the ON key, the computer resumes whatever it was doing before it was turned off.

A specially designed 44-pin bus connector allows you to connect and disconnect the HHC and its peripherals while all the components are powered up. Because of this feature, the HHC and its peripherals can join their respective data, address, and control buses without destroying data in either unit. As an additional safety feature, the piezoelectric beeper inside the HHC sounds if the HHC finds any loose connectors.

The ability to connect and discon-

nect modules while the power is on is very important because it allows the unit to be used in a variety of combinations without worrying that data will be destroyed by doing so. The HHC and its peripherals can be considered as interconnecting modules, and you can effectively forget that they contain volatile data. For example, when future program-development capsules become available, you will be able to write a program while traveling, then debug it more easily by hooking the HHC into the color TV adapter and printer. Data can also be entered into an HHC memory peripheral that may then be detached from the HHC and given to another HHC owner. He or she can plug it into another HHC and access the data that was stored.

Friends Amis has invented a particularly elegant solution to the packaging of programs in ROM (read-only memory). This solution also allows denser storage of information than was previously possible. The HHC uses 24-pin ROMs that are packaged in a plastic carrier around which the pins of the ROM are bent (see photo 3). This combination is

called an Amis Memory System Capsule (patent pending). (When a capsule is inserted into the back of the HHC, the flat base of each pin makes contact with the socket. This insures a good electrical contact without the usual fragility of integrated circuit pins.) Since a minimal amount of hardware is used to package the ROMs, more can fit inside the small body of the HHC.

These capsules have already been used in the Craig, Panasonic, and Quasar language translators (also developed by Friends Amis), and in the Friends Amis point of information display computer. Capsules can contain data to be manipulated (eg: words in a French language capsule), application software (eg: a capsule of game programs), programming languages (eg: a BASIC capsule), or any other data that the computer can act upon. Capsules can hold 2 K, 4 K, 8 K, or 16 K bytes of information. The 16 K-byte ROM allows an unprecedented amount of data to be stored in a small space. The large amount of information that can be stored in the HHC is increased by its internal use of a threaded language and by the application of a set of data compression techniques.

Human-Engineered Features

As a direct result of the manufacturers' desire to design a computer specifically for the mass market, the Quasar/Panasonic HHC was developed with a heavy emphasis on human engineering. This design philosophy is reflected in the operation and features of the HHC.

The keyboard has always been a crucial interface between the user and the computer, and the popularity of several existing microcomputers has been largely influenced by the usability of their keyboard. This fact, coupled with the small size of the HHC, makes it necessary for the HHC keyboard to be as usable as possible. We feel that the designers have achieved this objective.

[Despite my initial disbelief that a keyboard this small could be of any practical use, I was soon convinced that the HHC keyboard is easy to use and that, given some familiarity with it, I could use the keyboard without being distracted from the task at hand...GW]

Photo 1 indicates that the keys on

the HHC are arranged in the standard typewriter format. In addition, a key can be pressed without pressing any adjacent keys, so it is possible to touch-type on the HHC, regardless of individual finger width. This fact allows the HHC to be used in text applications—an area not practically accessible by any other device of its size.

Another powerful feature of the HHC is its ability within an application program to redefine any key position to any function. With the addition of a keyboard overlay, this can provide a keyboard that is completely suited to a given application. It was the intention of the HHC designers that no application, regardless of complexity, would require memorization of command language or special key sequence (like control-P for print) to perform a function available to the computer but not allotted a key. With redefinable keys and keyboard overlays, this will never happen.

Three special keys, labeled f1, f2, and f3, can be assigned to be any sequence of keystrokes, including most function keys. When one of these keys is typed, its current definition is input as if the sequence of keys had been typed by the user. The definitions are processed as interrupts and are independent of the program in use. Thus, they can be used with any present or future programs, even those written in BASIC or SNAP (the two computer languages currently planned for the HHC). For example, one key can be assigned to a sequence of calculations and/or constant values for use with the built-in calculator. Another key can be used to enter repetitive text in the memory bank text editor or to create special functions such as search-and-replace. Another definition can be used to make a commonly used sequence of menu selections to reach a frequently used program.

A unique feature of the HHC is the HELP key. When this key is pressed, you are prompted by the LCD display to press any key to find its definition. When a key is pressed, the function is given in a complete sentence of up to 80 characters. For example, pressing the HELP key followed by the STP/SPD key causes the message "STOP / ENTER 1-9 FOR SPEED" to be displayed.

Four HHC keys are used to indicate

LEFT, RIGHT, UP, and DOWN. In most programs, these keys are used for cursor control and horizontal and vertical scrolling. Since the HHC's built-in display shows only one short (26-character) line at a time, it is important to be able to "steer" the display through a larger page or list of material. The display is often used as a window into a larger virtual space (as is done in the popular VisiCalc program), and the four direction keys, which are auto-repeat

keys, move the window in any direction. Another key, STP/SPD (stop/speed), allows you to freeze and continue any program, like a run/stop switch, and to adjust the rate of information display.

The HHC also has INSERT and DELETE keys that allow text material to be changed. The HHC normally displays a solid rectangular cursor, but when you enter the insertion mode, the cursor changes to a blinking checkerboard cursor. Similarly,

WORD	FIRST NUMBER*	LETTERS BORROWED FROM LAST WORD	FIRST LETTER NOT COPIED	SECOND NUMBER* (COUNT FORWARD)	NEXT LETTER OF NEW WORD	REMAINING LETTERS OF NEW WORD*
SLOW	--	--	--	--	--	--
SLUMP	2	SL	O +	6 =	U	MP
SLY	2	SL	U +	4 =	Y	--
SMALL	1	S	L +	1 =	M	ALL
SMART	3	SMA	L +	6 =	R	T

Figure 1: Compression of an alphabetized list. The tables of alphabetized lists within the HHC are kept as small as possible by using numbers to keep track of the number of letters shared from the previous word and the number of letters between the first different letter in the new word and its counterpart in the previous word. Note that the shaded letters on a line make up the word being encoded, but only the two numbers and the letters in the last column (all marked with an asterisk in their table headers) are actually stored in the encoded table. The dashes indicate an empty entry (as in the line for the word SLY). The first line is all dashes because it does not have a previous line to refer to; in practice, all the letters of the first entry must be normally encoded.

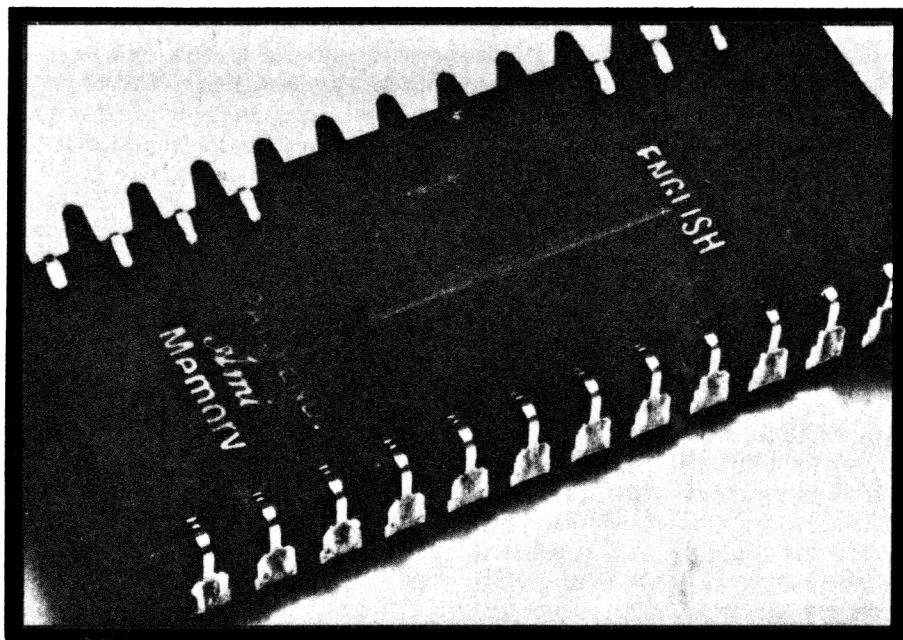


Photo 3: Close-up of an HHC program capsule. The program capsule is actually a standard 24-pin integrated circuit with its pins curled around a plastic harness. Its length is 3.65 cm (1¹/₁₆ inches).

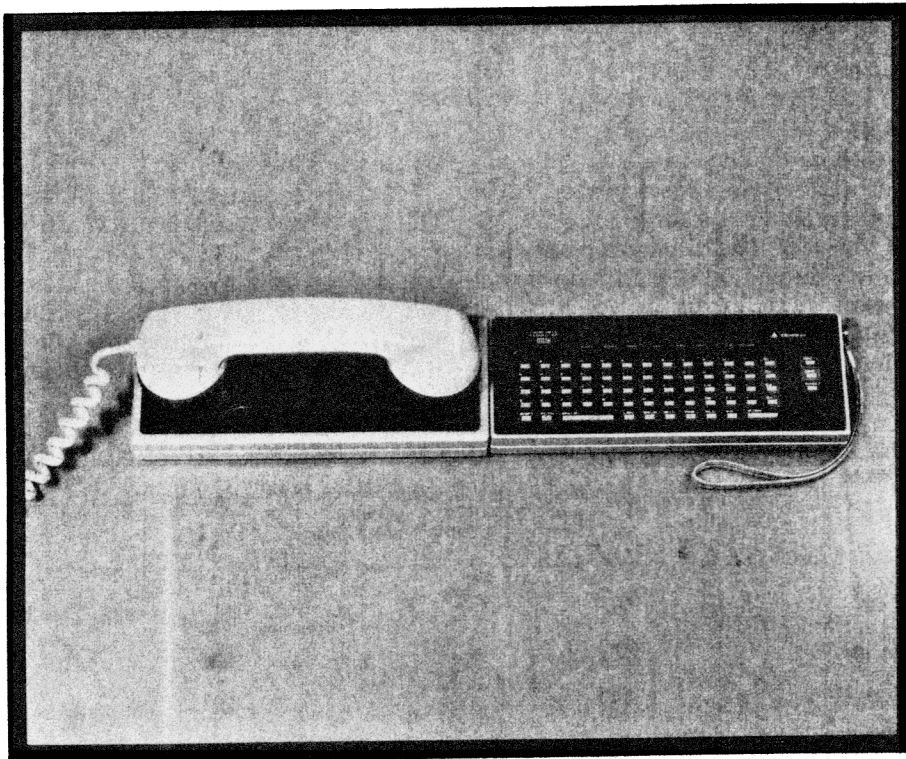


Photo 4: The Quasar HHC connected directly to its acoustic coupler. The combination, which is also available in the Panasonic HHC system, is a self-contained portable computer terminal.

The computer executes a FORTH-like language called SNAP.

when you enter the deletion mode, the cursor changes to a rectangular outline cursor. These useful features give you visual feedback regarding the mode that the computer is in.

Other keyboard-related features are the search and locate commands available within the memory bank electronic file system. These features are available in two modes—*context* and *initial search*. A context search searches for a match to the given character string anywhere in the file, while an initial search searches for a match beginning with the first character of each record in the file. The former method allows maximum searching power, but the latter provides a faster search when the position of the string to be matched is at the beginning of each record (eg: when the file contains last names and telephone numbers and you are given the last name).

Other strong keyboard features of the HHC are the size and placement

of certain keys. The SPACE and ENTER keys are in their traditional positions, and both are wider than the other keys for ease of use. Also notice from photo 1 that the CLEAR, ON, and OFF keys are located five rows to the right of the rightmost letter key, and at least two rows to the right of any other key. Although the consequences of hitting these keys by accident are less critical than on other personal computers (more on that later), the keys were placed there to minimize the danger.

Finally, the behavior of the SHIFT and LOCK keys should be mentioned. In applications where the program differentiates between uppercase and lowercase letters, an uppercase letter is obtained by hitting the SHIFT key, followed by the key to be shifted. The HHC is locked into uppercase by hitting the LOCK key after the SHIFT key. You can return to lowercase by hitting either the SHIFT or LOCK keys. The LOCK key can also lock the four cursor-control keys and the INSERT and DELETE keys.

The Menu and Other Features

To allow for use of the Panasonic/Quasar HHC with minimal prior knowledge of the machine, all func-

tions are selected via a set of nested menus. The first menu that appears when the computer is turned on is called the *primary menu*. It displays the available internal and capsule program choices (eg: clock/secretary, program capsule, etc) with a 1-digit number assigned to each. A choice is selected by pressing the corresponding digit key. If the selected application allows choices of its own, its menu is displayed in the same way. This process is repeated until an executable program is reached. Pressing the CLEAR key causes the HHC to display the second menu (the one immediately after the primary menu). Pressing the CLEAR key twice causes the HHC to return to the primary menu.

The HHC computer contains a piezoelectric beeper that can produce either a click (to provide audible feedback to an event, usually a keypress) or a tone within a four-octave range.

Squeezing More into Less

There has been recent publicity on threaded languages—most visibly FORTH. (See the special language issue on FORTH, August 1980 BYTE.) Threaded languages offer program compactness and speed of execution halfway between those of machine language and a high-level language like BASIC, while offering the programming ease and language transportability of high-level languages.

The Quasar/Panasonic HHC is actually a hardware machine that executes a FORTH-like language called SNAP, in addition to 6502 machine code. The HHC uses SNAP for every function that it performs, from the display of characters on the LCD readout to the handling of interrupts from the peripherals. When timing is critical in a specific routine, such as interrupt handling for high-speed peripherals, SNAP allows any portion of itself to be coded in assembly language for maximal speed.

SNAP, like other threaded languages, is defined in terms of a given set of operators (which are analogous to the operation codes of a given microprocessor). SNAP programs are simply lists of these operators, so these programs (including applications programs embedded in program capsule ROMs) may be used without change on any machine that executes the SNAP language, provided no ma-

chine code is used. This protects the sizable programming effort put into the HHC against hardware innovations in future versions of the HHC, while maintaining a body of programs that execute quickly and use little memory.

Another way in which the execution time of programs is decreased is through the use of interrupts for the HHC keyboard and all peripherals. In contrast to other computers which use *polling* (ie: they periodically check the device to see if it needs computer time), the HHC peripherals and keyboard generate *interrupts* when they require attention from the 6502 microprocessor. In this way several peripherals can be serviced at once. The HHC slows down only when it is interrupted to do specific work and is therefore faster than computers that waste time polling inactive devices. The HHC peripherals that require serial data all use separate UART (universal asynchronous receiver-transmitter) integrated circuits for this purpose.

Given the 64 K-byte maximum addressing ability of the 6502 microprocessor, the HHC must somehow pack more memory into less space. It does so, using the familiar technique of *bank-switching*. Three banks of memory, hexadecimal 2000 to 3FFF, 4000 to 7FFF, and 8000 to BFFF, are bank-switched. This means that several blocks of up to 16 K bytes of memory could be assigned to one of the above address areas, with electronic circuitry enabling only one such block to be active at a time.

The program capsules that insert into the back of the HHC all map into the same 16 K-byte address area: hexadecimal 4000 to 7FFF. Only one capsule is active at a time and is selected from the HHC primary menu. This area is also used for user data and programs.

The 16 K-byte area from hexadecimal locations 8000 to BFFF is used for external programmable memory banks. Since this bank is in a different address area from ROM banks, many ROM-based programs can reference data in programmable memory without bank-switching.

The 8 K-byte address area (from hexadecimal locations 2000 to 3FFF) is used by the specialized firmware that is contained in each HHC peripheral. When a given peripheral is being used, the firmware that con-

Letter	Huffman Code	
E	000	(a)
T	001	
I	010	
O	0110	
N	0111	
S	1000	
•	•	
•	•	
•	•	

Code	Bit(s) to Be Matched	Comments	
<u>0</u> 100111	0	--no match	(b)
<u>01</u> 00111	01	--no match	
<u>010</u> 111	010	--matches I	
(I) <u>0</u> 111	0	--no match	
(I) <u>01</u> 11	01	--no match	
(I) <u>011</u>	011	--no match	
(I) <u>0111</u>	0111	--matches N	
(I) (N)	done	--message is IN	

Table 1: An example of Huffman coding. Table 1a shows an example Huffman code for several letters. Table 1b shows how the code 0100111 is decoded into the letters I and N. Bits are taken from the left side of the remaining binary string until the sequence of bits matches one of the table entries. Notice in table 1a that the code for no letter is a beginning substring of the code for another letter. (This, for example, accounts for the fact that no letter is given to the bit string 011—it would conflict with 0110, the code for the letter O.) Every Huffman code (of which there are an infinite number) is constructed so that no two letters can be confused with each other. If the letters are assigned codes in the order of their decreasing frequency for the text to be decoded, a Huffman code permits the maximum data compression possible.

Table Rank (N)	Number of Elements In Table (= 2 ^N)	Number of Bits In New Permutation Algorithm (F(N) = 2 ^N + 2F(N - 1))	Number of Bits In Ordinary Look-up Table (= N2 ^N)
1	2	1	2
2	4	6 = 4 + 2(1)	8
3	8	20 = 8 + 2(6)	24
4	16	56 = 16 + 2(20)	64
5	32	144 = 32 + 2(56)	160
6	64	352 = 64 + 2(144)	384

Table 2: Efficiency of the permutation algorithm given in figures 2 through 4 and the text box. As can be seen from the last two columns, this algorithm uses fewer bits to define a given permutation. The ordinary look up table uses a table 2^N entries long by N bits long to look up the value (from 0 to 2^N - 1) that a given element (in the same range) is permuted to.

trols its communication with the HHC is selected and used. This area also contains the memory-mapped contents of the video display when the HHC is connected to the color TV interface.

In both 16 K-byte bank-switched areas it is possible to reference a program or a program/data combination that is more than 16 K bytes long. The program (or program and data)

is divided into 16 K-byte blocks, all of which map into the same area. Under program control the software can then jump between 16 K-byte blocks by writing the appropriate value to a location in the HHC that determines which block is currently selected.

Text Compression in the HHC

The increase in data storage caused

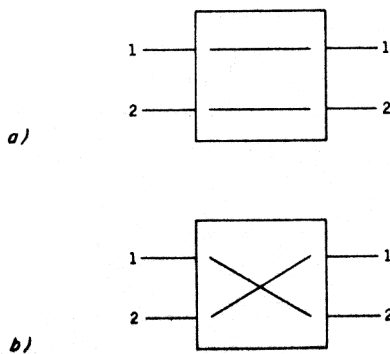


Figure 2: Two possible outcomes for the permutations of a two-element list. See the Mapping Algorithm text box for further details.

by the use of SNAP and 16 K-byte program capsules is significant. But the increase caused by the use of data compression techniques is even more significant, almost doubling the amount of information that can be stored in an HHC data capsule. A variable word-length code and increased data compaction through context are the two techniques used.

In traditional data storage, one character of information is stored in a byte (or 8 bits or binary digits) of computer memory. Letters, numbers, and punctuation are stored in the ASCII (American Standard Code for Information Interchange) format, which uses 7 bits per character. Using a method developed by Friends Amis that modifies what is called a *Huffman code*, variable bit-length codes can be devised for the characters to be encoded such that frequently used characters will be given shorter codes (called *codons*), thus decreasing the average number of bits used per character. Table 1 shows an example of a standard Huffman code (there are an infinite number of such codes).

Because of this variable-length coding, the computer's memory is seen as a long string of bits. Bits are read from left to right (figuratively speaking) until the bits read match the codon for any character in the set. (Codons are generated by rules that guarantee that a beginning string of bits can match the codon of only one letter in the set.) Codons are also devised so that the most frequently used letters have the shorter representations and are also near the top of the look-up stack. Because the number of look-up entries read before a match occurs is kept to a minimum (on the average, slightly more than eight entries), the decoding process

does not slow the machine down.

A further measure of compression is made by modifying the look-up procedure to be sensitive to the context of the previous letter. For example, even though the most frequently used letters in normal English text are (in decreasing frequency) E, T, I, O, N, and so on, if the previous letter looked up was Q, then the letter U is most probably the next letter and so should be close to the beginning of the look-up table. Within the HHC, the letter-decoding routine uses the previously decoded letter to index one of several look-up tables. In this way, encoded characters can be represented in even fewer bits than would otherwise be possible using straight frequency-determined codons.

Two more techniques are used within the HHC to decrease the number of bits used to represent character information to a final density of just over 4 bits per character. Although these techniques

were developed to deal with alphabetized lists of words (for the Friends Amis language translator), it is possible to use them to compress nonalphabetized text in some situations.

The first technique replaces the beginning of each word (except the first word in a list) with two numbers. The first number tells how many letters to borrow from the previous word. The second number tells how many letters away the first non-matching letter is from its counterpart in the previous word. For example, if the words are SMALL and SMART, the following is stored for the word SMART: 3 (telling the computer to borrow SMA from the word SMALL); 6 (telling the computer to count forward six letters from the L in SMALL to arrive at the R in SMART); the encoded letter T (ending the encoding of the word SMART). (See figure 1 for other examples.) Because the two numbers (contained in 3 and 4 bits, respectively) take up fewer bits than the letters

The Mapping Algorithm

It is sometimes profitable to maintain a list of words in alphabetic order but to be able to retrieve them in some other pre-specified order. The problem then becomes one of finding the most compact way of specifying a permutation of N elements from $(1, 2, 3, \dots, N)$ to some other ordering.

The algorithm used within the Panasonic/Quasar HHC requires that the list be a power of 2 (ie: have 2, 4, 8, 16, 32, 64, ... elements). The algorithm can be considered as a recursive set of pair switchings. The permutations of a list of two elements can be represented by 1 bit of information—say, a 0 to represent that the elements are not switched, eg: $(1, 2)$ becomes $(1, 2)$; and a 1 to represent that the elements are switched, eg: $(1, 2)$ becomes $(2, 1)$. This is represented pictorially in figure 2, where a box represents 1 bit of information.

The diagram in figure 3a is used with a list of four elements. The upper-lefthand box is always filled in with an equal sign (=). The input arrangement, usually $(1, 2, 3, 4)$, is substituted for IN1 thru IN4, and the desired permutation is sub-

stituted for OUT1 thru OUT4. The boxes in the first and third columns are filled in with either equal signs (=) or cross signs (X), leaving the boxes in the second column for last.

Consider the example of permuting the list $(1, 2, 3, 4)$ to become $(4, 1, 3, 2)$. Given the interconnections between boxes and the constraints given above, the only path that can be taken from 1 to 1 goes through the top middle box (in a manner not yet specified) and to the righthand side through a cross in the upper-right box, as shown in figure 3b. In figure 3c, the element 4 is traced from box A to box B. Similarly, element 3 is traced from box B to box C, and element 2 is traced from box B to box S, where we started.

Given the conditions shown in figure 3c, it is a simple task to fill in the middle columns, thus completing the diagram. The finished diagram is shown in figure 3d. Through use of this diagram, the list $(1, 2, 3, 4)$ can be permuted to the list $(4, 1, 3, 2)$ using 6 bits of information (1 bit for each of the six boxes).

Study of an eight-element list example illustrates the recursive

they replace, this method can represent the same text in fewer bits.

The last technique saves space in that it allows alphabetized lists to be used in a different order. (For example, in language lists a given set of words is mapped from the sequential order in its alphabetized list to a semantic order in a list of words of equivalent meaning available in each language list; this is done so that the computer can translate a given word to its equivalent in another language.) With this technique, a list of 2^N elements can be permuted into any other arrangement of the same elements by a relatively small number of bits of information (see table 2). Refer to the Mapping Algorithm text box for the details of this algorithm.

The Real-Time Clock

One of the most important internal features of the Panasonic/Quasar HHC is its real-time clock and event sequencer. The real-time clock exists in memory as a 40-bit number stored

method that is used to generate the final structure for longer lists. Figure 4 shows a mapping of the list (1, 2, 3, 4, 5, 6, 7, 8) to (6, 3, 8, 1, 7, 5, 4, 2). As before, box S is marked with an equal sign. Boxes in the first and last columns are then filled in; this can even be done with no knowledge of the contents of boxes X and Y. The boxes A through G are filled in alphabetically. Note that when these boxes are filled, the boxes X and Y become "black boxes" that map four-element lists into another ordering. These boxes are then solved as shown in figure 3, and the permutation of eight elements is now solved. The final solution has twenty boxes: eight as shown in figure 4, plus six boxes each for boxes X and Y.

Larger lists are solved in an analogous way, with a list of 2^N elements first filling the 2^N boxes in the first and last columns, followed by the solution of the two middle boxes, each of which permutes a list of 2^{N-1} elements. Table 2 shows the number of boxes (or bits) necessary to solve larger permutations.

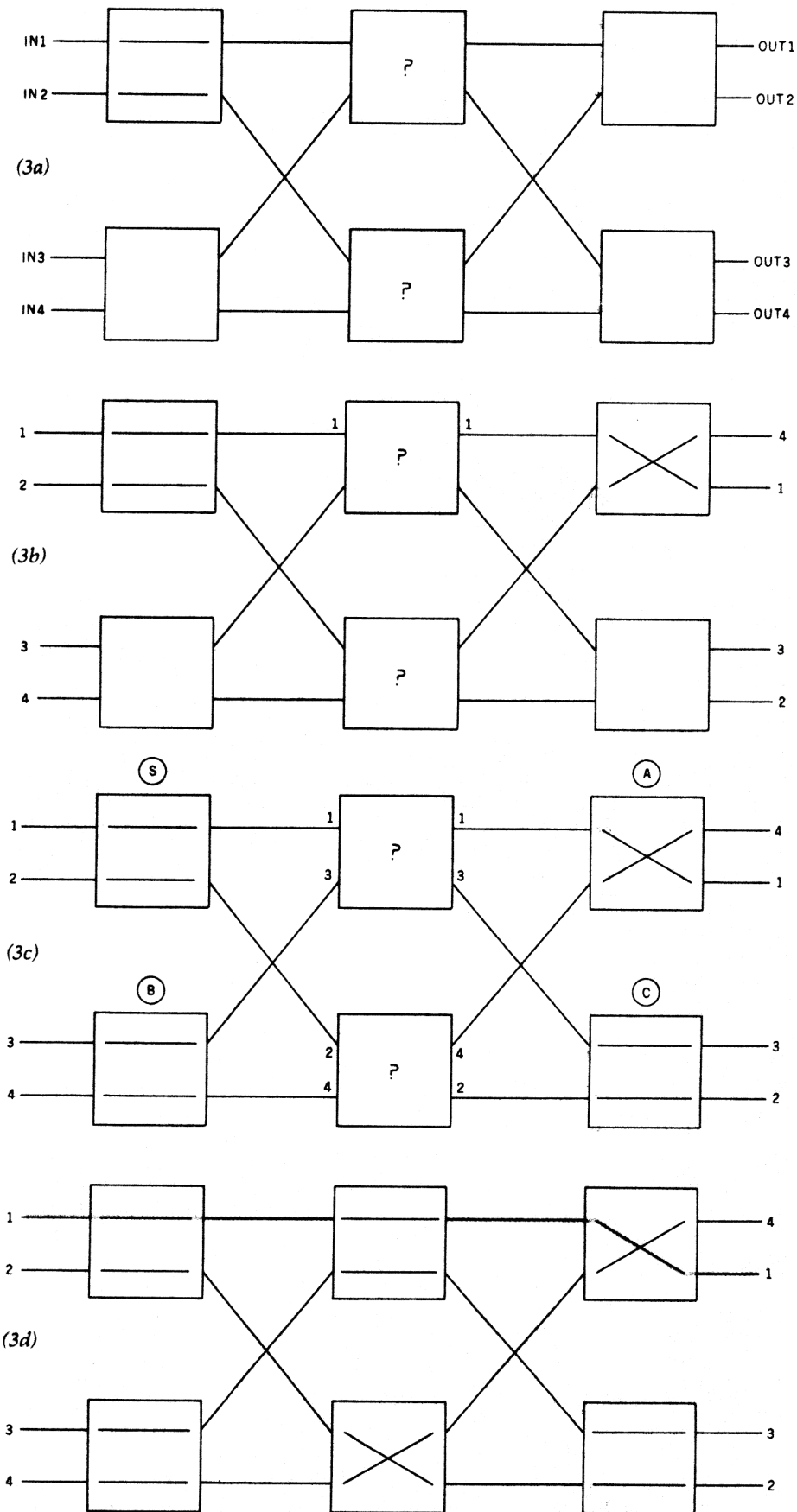


Figure 3: Solving a four-element permutation problem as a network of binary decisions. Figure 3a shows the initial configuration used in the solution of any four-element permutation. Figures 3b, 3c, and 3d show steps in the solution of this problem. See the Mapping Algorithm text box for further details.

in 5 contiguous bytes of program-mable memory, supported by a hardware counter that can be preset. An increment of one unit in this number represents a time change of 1/256 second (about 4 milliseconds), so that the 40-bit number represents the number of 1/256 second intervals that have elapsed since the computer was permanently turned on. (Given the above figures, a 40-bit number will represent a time period of approximately 139 years.)

In keeping with the design philosophy of burdening the 6502 microprocessor with as few tasks as possible, the real-time clock was designed to require the generating of as few interrupts as possible. Another area of memory contains a signed 23-bit counter circuit that automatically counts down to 0 at a rate of one count every 1/256 second. Normally, when this timer reaches 0 (once every $2^{23}/256$ seconds, or about 9 hours), it generates an interrupt that adds the same amount (about 9 hours) to the 40-bit clock number. However, if any program needs to access the real-time clock, the appropriate count based on the value in the 23-bit counter can be added to the 40-bit clock number and the 23-bit counter can be cleared, thus updating the clock to its correct value.

Associated with the real-time clock is an event queue in which future events are stored as 40-bit numbers along with instructions to be carried out when the 40-bit clock number reaches that value. Internally, the operating system software can use this event queue to manage a set of asynchronous events with a minimum of processing. Application programs can use the event queue, as can users programming on the HHC.

Design for Component Interaction

The Quasar/Panasonic HHC was designed to be compatible with both existing and future hardware and software. Because of this, the memory usage of the computer had to be planned to provide maximum flexibility.

In most microcomputer systems, there are fixed memory locations or I/O (input/output) ports assigned for specific hardware peripherals. The limitation of this approach is that the entire memory mapping must be foreseen; otherwise the ability to include

Data compression techniques in the computer almost double the amount of information that can be stored in a given number of bits.

future peripherals is questionable. The HHC does not make any fixed assignments. Instead, 4 bytes for each peripheral are dynamically assigned as I/O and status locations for all currently connected peripherals each time the clear key is pressed, so any number of different peripheral types can be accommodated without running into memory map conflicts.

This flexible system of directing input and output allows the HHC to offer a more commonsense approach to dealing with devices like printers, modems, LCD displays, and other devices. In most computers, special commands must be given to direct input and output to specific devices, and even then you may not be able to distribute it to several devices. For example, a special command, LPRINT, must be used to get either the Radio Shack TRS-80 or the Atari 400 or 800 to print information on their associated printers, and it is impossible to get a program to print on both the video display and the printer without using both PRINT and LPRINT statements. With some limitations this can be done with the Apple computer, but only with the correct interface board and the correct PR#N command.

The attitude taken by Friends Amis is that you *shouldn't have to remember* extra information (which is often complicated by being conditional on what the computer is currently doing). With the HHC computer, the use of I/O devices can be changed by pressing the I/O key and enabling or disabling the appropriate devices from a menu displayed by the HHC. You can even, for example, interrupt a running program to enable the printer, and resume the program without error; from that point on, both the current display device (the LCD display, color TV, or other device) and the printer display whatever the program tells them to. This method allows HHC programs

to be independent of the I/O devices, and it allows the use of future peripherals with current software.

Application Software

The Panasonic/Quasar HHC includes several application programs that are contained in the same built-in read-only memory devices as the operating system. These programs implement a calculator, a clock/secretary, and an electronic file system and editor. Each of these programs is called from the primary menu of the HHC.

The calculator program, when selected, transforms the HHC into a standard four-function calculator that adds, subtracts, multiplies, and divides. The calculator can store one number and has keys to add to, subtract from, clear, and recall memory. It also has a percent key.

The clock/secretary uses the real-time clock that knows the time of day, the day of the week, and the date (day, month, and year). A clock option within the clock/secretary allows the time and date to be displayed and continuously updated on the LCD display window. Otherwise, the clock/secretary can be used to keep track of future events. You can specify a time for the clock/secretary to activate itself, and include an optional reminder message. When that time arrives, the HHC sounds a musical tune regardless of its current task; you can then perform an "acknowledge" operation and see the message associated with the event. The number of events and messages that the clock/secretary can hold is limited by the amount of program-mable memory in the HHC.

The "memory bank" is the nickname of an electronic file system and editor within the HHC. You can enter lines (or *records*) of up to 80 characters of ASCII information, group them to make *files*, and modify and list these files. Any file can be edited with a powerful cursor-controlled editor that allows insertion and deletion of characters or lines at the current cursor position. With the SEARCH key, you can also retrieve records from a file based on a character string to be matched.

Memory bank files can have any number of records, with each record holding up to 80 characters. The size and number of files that can be stored depends on the amount of program-

mable memory in the HHC. The current model of the HHC has somewhat less than 1500 bytes of memory for this purpose, but the amount of memory in the HHC can be expanded with a battery-powered 4 K-byte memory extender peripheral. Future models will accept more program-mable memory in the form of capsules that fit into the same sockets as the read-only memory capsules.

The Extended HHC

The Quasar/Panasonic HHC, when combined with its line of peripherals, has the ability to perform any function that existing personal computers do, while retaining the characteristics and advantages of a hand-held unit. The following sections describe two of the most interesting peripherals—the color television interface and the modem.

The color television interface is the only peripheral that requires connection to an AC power line. But since the interface is also connected to a color TV, this is hardly a limitation. Once the interface is connected, output can be routed to the TV through the use of the I/O key.

Through the color TV, the HHC will display 16 lines of 32 characters each. Characters can be displayed in several combinations (orange or green characters on black, or black characters on either an orange or a green background). Several kinds of characters can be displayed: uppercase and lowercase ASCII letters; numbers and punctuation; graphics patterns; and katakana characters (a set of phonetic characters used by the Japanese). All characters are created in a 7 by 9 dot matrix.

The color TV interface offers two modes of color graphics: 32 by 64 pixels, or 48 by 64 pixels. The interface allows for black and eight colors (red, blue, green, yellow, orange, magenta, cyan, and buff).

The color TV interface contains a built-in RF (radio-frequency) modulator, as well as 1.5 K bytes of dynamic memory organized as two software-selectable screen images. The connection from the interface to the HHC is an interrupt-driven parallel connection.

The modem, which connects to the HHC through an interrupt-driven parallel interface, is acoustically coupled to a standard telephone handset (see photo 4). Its options—

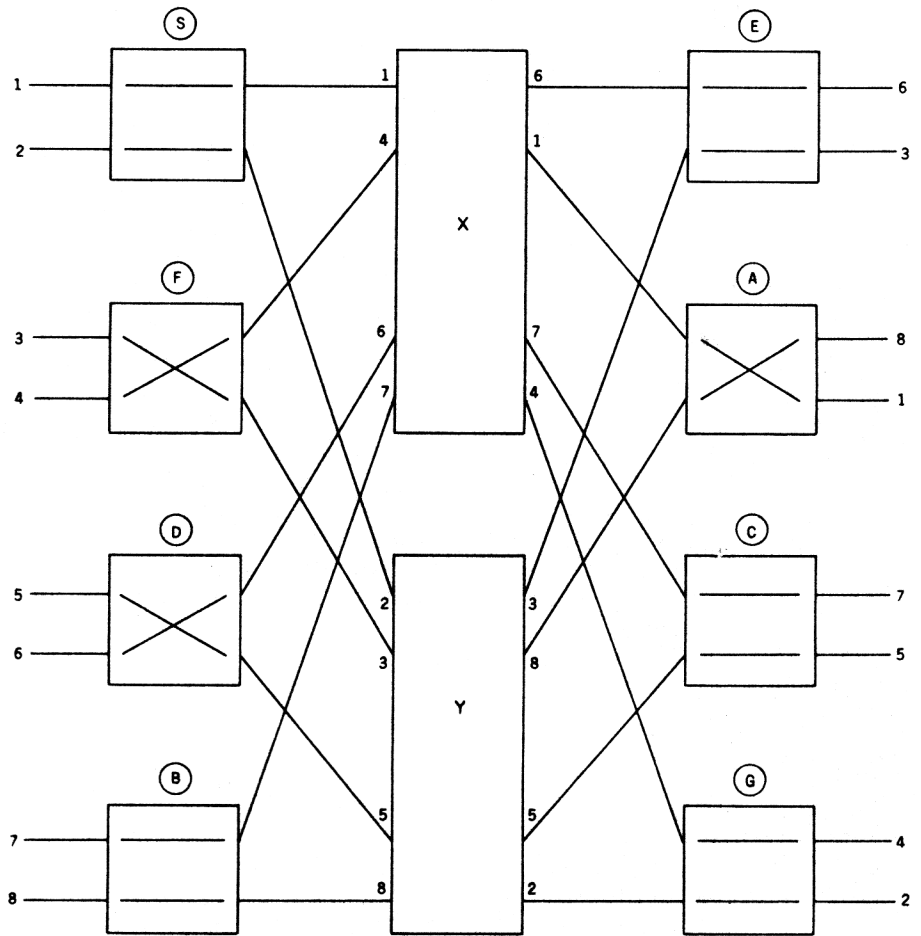


Figure 4: Partial solution of an eight-element permutation problem. Each of the boxes in the first and last columns is filled in first. The solution of this problem is then finished by the solution of two four-element permutations as given by the numbers on both sides of the boxes marked X and Y.

110 or 300 bps (bits per second) data transfer rate, full- or half-duplex transmission, answer or originate mode, number of start and stop bits, and parity—are all selected by software. In a daring departure from conventional modems, the HHC modem has no visible switches to set any of its options. This forces the software to control all the options and leaves nothing for you to worry with (or set incorrectly).

The HHC modem, like other HHC peripherals, is responsible for supplying standard input and output routines. (By using a uniform software interface for all peripherals, the HHC can be expected to work with peripherals that have not yet been designed.) Since the modem can be used in several ways, it is supplied with a socket in which to place a program capsule for a given application. The first capsule to be produced for the HHC modem is called "Telecom-

puting" and it will allow the HHC to be used as an intelligent remote terminal that is connected, through the modem, to a timesharing computer or data base. The program can be used with the small battery-operated modem directly connected to the HHC, in a hand-held configuration, or the printer and TV can be used.

The telecomputing software can use an automatic X-ON/X-OFF handshaking with a host computer so that you can regulate the rate of display to your reading speed. This protocol is supported by most popular networks such as Micronet, The Source, and Tymnet. When a printer is not connected, you can review many lines of previous interaction as they appear in the LCD display, creating, in effect, a virtual printout. Incoming lines longer than the 26-character LCD display are divided only at blanks. This "word-wrap" feature, combined with the review mode, assures

readability with the 1-line display.

Background of the HHC

The HHC was developed as a result of a unique union of Japanese and American technology. Friends Amis, with headquarters in San Francisco, contributed the best of Silicon Valley—a software-based systems architecture, circuit design, a unique operating system and SNAP language. The company's founders, who came from Atari Inc, were responsible for introducing the now widely accepted consumer video games. Friends Amis' first product was the highly successful language translator sold by Craig, Quasar, and Panasonic; this product was quickly followed by its point of information display computer and the HHC (hand-held computer).

Matsushita, the parent company of Panasonic and Quasar, in Osaka, Japan, brought its unparalleled techniques of miniaturization, industrial design, quality assurance, and the ultimate in highly

The HHC, through the color television interface, can display 16 lines of 32 characters each.

automated, high-volume, low-cost manufacturing—areas in which Japan has clearly outstripped the US in recent years. Putting the best of both worlds together has resulted in a special product that could not have been produced alone: the first hand-held computer with bus architecture, a powerful operating system, and a fast 8-bit microprocessor.

Conclusions

•The Quasar and Panasonic HHCs are certainly impressive first entries into the new market of hand-held, consumer-oriented computers. Great emphasis has been placed on human engineering. This is important for any device marketed to the general

public, even more so when so many functions are being placed into such a small package.

•The HHC was designed as a basic unit augmented by an extensive complement of peripherals. This "de-bundled" approach allows you to buy only those peripherals you want, giving you a customized computer at minimal cost.

•Several innovations in the HHC computer allow it to have the power of conventional personal computers while retaining the portability of a hand-held unit. The use of data compression techniques and program capsules enables very large amounts of data to be contained within the hand-held unit.

•The HHC is supplied with internal application programs that include a clock, an electronic secretary that reminds you of future appointments, and a file system for user data contained completely within the programmable memory of the computer. These are nice touches that add to the utility of the computer.

A Fictional Hand-Held Computer

Duncan's Minisec had been a parting gift from Colin, and he was not completely familiar with its controls. There had been nothing really wrong with his old unit, and he had left it behind with some regret; but the casing had become stained and battle-scarred, and he had to agree that it was not elegant enough for Earth.

The 'Sec was the standard size of all such units, determined by what could fit comfortably in the normal human hand. At a quick glance, it did not differ greatly from one of the small electronic calculators that had started coming into general use in the late twentieth century. It was, however, infinitely more versatile, and Duncan could not imagine how life would be possible without it.

Because of the finite size of clumsy human fingers, it had no more controls than its ancestors of three centuries earlier. There were fifty neat little studs; each, however, had a virtually unlimited number of functions, according to the mode of operation—for the

character visible on each stud changed according to the mode. Thus on ALPHANUMERIC, twenty-six of the studs bore the letters of the alphabet, while ten showed the digits zero to nine. On MATH, the letters disappeared from the alphabetical studs and were replaced by X, +, ×, −, =, and all the standard mathematical functions.

Another mode was DICTIONARY. The 'Sec stored over a hundred thousand words, whose three-line definitions could be displayed on the bright little screen, steadily rolling over page by page if desired. CLOCK and CALENDAR also used the screen for display, but for dealing with vast amounts of information it was desirable to link the 'Sec to the much larger screen of a standard Comsole. This could be done through the unit's optical interface—a tiny Transmit-Receive bull's-eye operating in the near ultraviolet. As long as this lens was in visual range of the corresponding sensor on a Comsole, the two units could happily exchange information at the rate of megabits

per second. Thus when the 'Sec's own internal memory was saturated, its contents could be dumped into a larger store for permanent keeping; or conversely, it could be loaded up through the optical link with any special data required for a particular job.

From Imperial Earth, copyright 1976 by Arthur C Clarke. Reprinted by permission of Harcourt Brace Jovanovich Inc.

[Editor's Note: The 'Duncan' referred to in the first paragraph is Duncan Makenzie, the main character in Clarke's Imperial Earth. Duncan's boyhood friend is Karl Helmer, a character whose name is a variant spelling on that of our Founding Editor, Carl Helmers. For a humorous (and somewhat eerie) commentary on the name similarity and the anticipated possibility of a hand-held computer, see Carl Helmers' editorial in the April 1977 BYTE (page 6). "How I Was Born 300 Years Ahead of My Time."]

•The HHC retains the contents of memory even when it is turned off. In addition, you do not lose what you are working on if you accidentally hit the OFF button. These are important features that indicate the amount and depth of human engineering that has been applied to the design of the HHC.

•The HHC will be marketed aggressively by both Quasar and Panasonic. The public reaction to this device, which is the first of its kind to be marketed on such a large scale, will be carefully observed by manufacturers and may determine the extent and direction of future consumer products in this area. We feel that the

Panasonic/Quasar HHC is highly qualified to receive this scrutiny and that the public response will be favorable. ■

Acknowledgment

The cover photograph and all interior photographs are by Ed Crabtree. Photo 2 is courtesy Quasar Electronics Company.

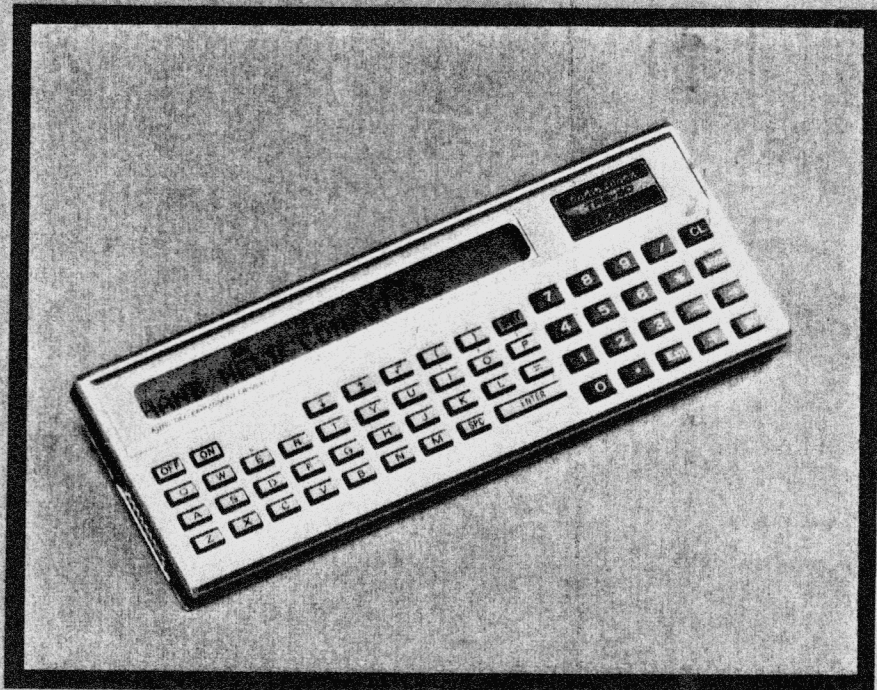
Another Pocket Computer

The internal architecture of the TRS-80 Pocket Computer is radically different from the other pocket computers now reaching the market. Instead of a single 8-bit microprocessor (such as that used in the Quasar/Panasonic HHC and the Sinclair ZX-80), the designers of the TRS-80 Pocket Computer (Sharp Electronics of Japan) decided to use two 4-bit microprocessors in a unique serial configuration.

Both microprocessors are custom CMOS (complementary metal-oxide semiconductor) integrated circuits with built-in ROM (read-only memory). The purpose of microprocessor 1 is to arrange data and make decisions. It reads the data that is keyed in or fetched from programmable memory. It is also responsible for parsing arithmetic operations and interpreting the syntax of BASIC statements. It then arranges the data and provides instruction codes to microprocessor 2 through a transfer buffer. The actual execution of an instruction is performed by microprocessor 2, which also updates the display and notifies microprocessor 1 that it has finished its function. The respective duties of the microprocessors are listed at right.

Memory Organization

The programmable memory of the TRS-80 Pocket Computer is contained in four integrated circuits. There are three memory ICs, each containing 512 bytes of programmable memory. The three ICs which drive the liquid-crystal display each contain 128 bytes of programmable memory. Putting it all together, you end up with 1920 bytes of programmable memory. After you subtract memory space used for the transfer buffer, input buffer, display buffer, fixed mem-



Microprocessor 1

Key input routine

Acknowledgment of the remaining program

One instruction to one program step incorporation

Interpreter:

Program execute statement
Cassette control statement
Command statement
Printer control (reserved)

Execution of manual operation

Power shut-off control

Clock stop control

Microprocessor 2

Display processing routine
Input buffer
Computational result
Error

Arithmetic routine

Character generator

Cassette routine

Print routine

Buzzer

Recognition of printer (reserved)

Power off

Clock stop

ories, and reserved keys, you end up with 1424 bytes of user-addressable memory. Into this space you

can easily fit a BASIC program of around 250 lines (average length)...SM ■

Numerical Analysis for the TRS-80 Pocket Computer

Mike Salem, 26A Delancey St, London NW1 7NH, England

Complicated programs can often be easily modified to fit into the new pocket computers. I've taken three programs from the December 1979 issue of BYTE and modified them to run on the Radio Shack TRS-80 Pocket Computer (sold as the Sharp PC-1211 outside of the United States). The Pocket Computer has a 24-character LCD (liquid-crystal display), twenty-six fixed variables, and 1424 bytes of programmable memory.

One of the programs I modified was the discrete-Fourier-transform program that appeared in "Frequency Analysis of Data Using a Microcomputer" by F R Ruckdeschel (December 1979 BYTE, page 10). I also combined two programs that compute the time-domain

response of a system with a given transfer function into a single program ("Noniterative Digital Solution of Linear Transfer Functions" by Brian Finlay, December 1979 BYTE, page 144). The modified programs have all of the features of the originals, with the obvious omissions of printing and plotting.

Incidentally, it is important to note that the TRS-80 Pocket Computer, in common with many machines allows BASIC lines to contain multiple statements (saving 3 bytes of programmable memory for each line number omitted). Although this feature is useful in itself, the TRS-80 Pocket Computer also has an IF statement that can control all of the remaining statements in the

Listing 1: A discrete-Fourier-transform program for the TRS-80 Pocket Computer. This program was modified from "Frequency Analysis of Data Using a Microcomputer" by F R Ruckdeschel (December 1979 BYTE, page 10). Statements entered on the same line are separated here for clarity.

```

10 :REM BYTE DEC 79
11 :RADIAN
190 :INPUT "1ST X? ";Z,"LAST X? ";Y,"#OF POINTS? ";N
250 :I=1
    :INPUT "1/P SCALE FACTOR? ";I
    :IF I<1 GOTO 250
290 :D=(Y-Z)/(N-1)
    :Q=0
    :V=π/DI
    :U=V/(N-1)
340 :FOR I=1 TO N
    :PAUSE "NEXT # = ";I
    :BEEP 1
    :INPUT "NEXT F(T) VALUE? ";O
    :A(I+26)=O
    :NEXT I
370 :B=0
    :FOR I=27 TO N+26
    :IF B>A(I) LET B=A(I)
410 :NEXT I
420 :FOR I=27 TO N+26
    :A(I)=A(I)-B
    :NEXT I
    :B=ABS B
    :T=0
    :FOR I=27 TO N+26
    :IF T<A(I) LET T=A(I)
510 :NEXT I
710 :FOR I=1 TO N
    :W=(I-1)*U
    :C=0
    :P=0
    :FOR M=1 TO N
    :X=Z+(M-1)*D
    :G=WX
770 :C=C+A(M+26)*COS G
    :P=P+A(M+26)*SIN G
    :NEXT M
800 :F=√(PP+CC)*D
    :IF I=1 LET C=C-NB
    :F=D*ABS C
810 :BEEP 1
    :PRINT U*(I-1);"RAD/S"
    :PRINT "AMPL. =";F
815 :IF C<>0 LET O=ATN(P/C)*180/π
    :PRINT "PHASE =";O
820 :NEXT I
    :BEEP 3
    :PRINT "END OF RUN"
:END

```

Payroll General Ledger Accounts Payable Accounts Receivable

Flexible and sophisticated business software that is of the highest quality. Originally developed by Osborne & Associates and rapidly becoming a standard. Accountants are pleased with our balance sheets and income statements. It has been tested and approved by secretaries.

- Complete instructions
- Up and running the same day of delivery.
- 100% guaranteed
- Unlimited free telephone assistance.
- Return within 30 days for complete refund if not satisfied.
- No licensing agreement - use as you want.

Accounts Receivable	\$95.00
Accounts Payable	\$95.00
Payroll-California	\$95.00
Payroll-Non California state	
Income tax calculations	
(add to California payroll)	\$95.00
General Ledger	\$95.00
Multiple profit center option	
for general ledger	\$95.00
Manuals (each)	\$20.00

Formats: Any 8" single or double density, TRS-80 Mod II, Northstar DD, Vector Graphics, Super Brain, many others.

Runs on Beehive, Sorac, Hazeltine, ADDS, Televideo, SOL, TRS-80 Mod II, HEATH, ACT V, Intertube II, Altos, ADM and many other terminals. All programs in CBASIC under CP/M (includes source).

Synergetic Computer Products
575 Middlefield Avenue Suite J, Palo Alto, CA 94301
(415) 328-5391

VISA • Master Charge • COD • Company Check

TRS-80 is a trademark of Radio Shack, Inc. CP/M is a trademark of Digital Research

Systems Notes

same line. Since this makes listings a bit difficult to read, I prepared listings 1 and 2 with a separate statement on each line. ■

Listing 2: A program for the TRS-80 Pocket Computer that computes the time-domain response of a system with a given transfer function. The program shown was combined and modified from two programs contained in "Noniterative Digital Solution of Linear Transfer Functions" by Bryan Finlay (December 1979 BYTE, page 144).

```

10 :REM "TF: TRANSFER FCN - BYTE DEC 79"
70 :RADIANT
:INPUT "CONST.? ";K,"#TERMS NUM.? ";E,"#TERMS
DEN.?";L
150 :IF E=0 GOTO 240
160 :   FOR G=27 TO E+26
:     O=10+G
:     INPUT "RL, NUM.? ";A(G),"IM, NUM.? ";A(O)
:     NEXT G
240 :IF L=0 GOTO 330
250 :   FOR H=47 TO L+46
:     O=10+H
:     INPUT "RL, DEN.? ";A(H),"IM, DEN.? ";A(O)
:     NEXT H
330 :FOR G=1 TO L
:   O=66+G
:   Q=76+G
:   A(O)=1
:   A(Q)=0
:   IF E=0 GOTO 450
370 :   FOR H=1 TO E
:     D=A(26+H)-A(46+G)
:     C=A(36+H)-A(56+G)
380 :     M=√(DD+CC)

```

```

:   N=ATN(C/D)
:   IF D<0 LET N=N-π
410 :   A(O)=MA(O)
:   A(Q)=N+A(Q)
:   NEXT H
450 :FOR R=1 TO L
:IF R=G GOTO 501
465 :D=A(46+R)-A(46+G)
:   C=A(56+R)-A(56+G)
470 :M=√(DD+CC)
:   N=ATN(C/D)
:   IF D<0 LET N=N-π
500 :A(66+G)=A(66+G)/M
:   A(76+G)=A(76+G)-N
501 :NEXT R
:   NEXT G
520 :INPUT "T(O)? ";O,"DT? ";S,"# STEPS? ";N
:   T=O+NS
620 :U=-S
:   FOR Q=1 TO N
:     U=U+S
:     V=0
:     W=0
:     H=1+INT((U-O)/S)
650 :     FOR G=1 TO L
:       X=A(66+G)*EXP(-UA(46+G))
:       Y=A(76+G)-UA(56+G)
:       V=V+X*COS Y
:       W=W+X*SIN Y
:     NEXT G
710 :Z=K*√(VV+WW)*SGN V
:   BEEP 1
:   PRINT "TIME=";U
:   PRINT "RESP.=";Z
730 :NEXT Q

```

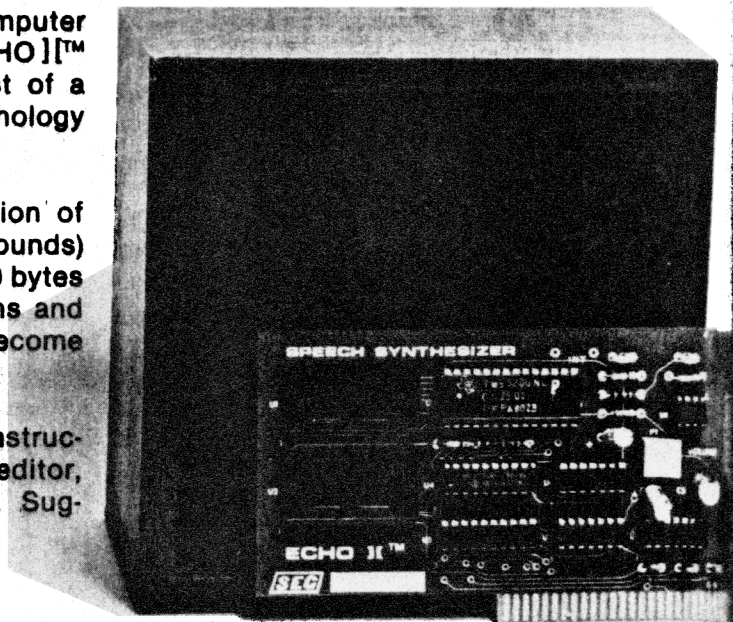
ECHO SERIES™ SPEECH SYNTHESIZERS COMPUTERS ARE SPEAKING OUT!

Now you can add intelligible speech to your computer without using vast amounts of memory! The ECHO][™ speech synthesizer for the Apple* is the first of a series of synthesizers based on the same technology that made the Speak & Spell** a success.

The initial operating system allows the creation of your own vocabulary with phonemes (word sounds) while using very little RAM memory (approx. 800 bytes + 20 bytes/word). Enhanced operating systems and vocabulary ROMs will be offered as they become available.

The ECHO][™ comes complete with speaker, instruction manual, and a disk containing a speech editor, sample programs, and a sample vocabulary. Suggested list price is \$225.

See your dealer or contact:



**STREET ELECTRONICS
CORPORATION**

*Trademark of Apple Computer

**3152 E. La Palma Ave., Suite C
Anaheim, CA 92806 (714) 632-9950**

**Trademark of Texas Instruments