Keyboard

May-Jun/80 A Publication of Hewlett-Packard Desktop Computer Division

Keyboard

May-June 1980

Cover

Determining the performance of stereo system components has become more and more challenging as the overall level of quality has risen for these products. More sophisticated capabilities are continually added to stereo equipment, like the strobe light which helps precisely time the frequency of the turntable shown on the cover. At *Audio* magazine in Stuttgart, West Germany, a desktop computer and a collection of sophisticated instruments are necessary to determine relative stereo component quality.

Beginning with this issue, *Keyboard* will include in any article originating from a non-English-speaking country a paragraph describing the article in the native language of that country. On the first page of our cover article (facing page), you will find a short paragraph describing the article in German; that is, "auf Deutsch."

Photo and artwork credits

Cover, pages 6,9,10,11 — Hal Andersen and Paula Dennee

Pages 1 and 2 — Jurgen Tries, *Audio* magazine, Stuttgart, West Germany

Pages 3,4,5 — Caterpillar Tractor Company, Decatur, Illinois, U.S.A.

Page 7 — The figure on this page is reproduced with permission from American National Standard X3.4 — 1977, Code for Information Interchange, copyright 1977 by the American National Standards Institute, copies of which may be purchased from the American National Standards Institute at 1430 Broadway, New York, New York 10018, U.S.A.

Testing hi-fi hardware

by Jürgen Tries

Audio, Deutschlands groesstes HiFi Magazin, setzt den HP 9825A im Messlabor ein. Dort werden die Testgeraete (Plattenspieler, Verstaerker, Cassettenrecorder und aehnliche). Gemessen, Messprotokolle gedruckt und Frequenzgangkurven geplottet.

Since the stereo system began replacing the automobile as a European status symbol, the high-fidelity (hi-fi) trade journal has become increasingly important. *Audio*, the largest German hi-fi magazine, is well aware of its responsibility to its readers.

In order to meet our high demands, we have set up our own automated test system for stereo equipment. Now, nothing is left to chance in our tests of high-quality hi-fi components.

Desktop computer

In January 1978, we decided that we needed an automated measurement system to monitor tests we were making on hifi equipment. In particular, we wanted to eliminate any source of reading errors.

It didn't take long to realize that a computer was needed, but we wanted more than that. We wanted something that was easy to program and fast, since our measurements tend to have a lot of steps. At the same time, we wanted a computer capable of controlling a wide range of measurement equipment. Utilized to its maximum storage capacity and equipped with all available ROMs, the 9825A represents the heart of the *Audio* measurement laboratory. It controls, among other things, the HP 3582A Spectrum Analyzer, which is used in virtually all measurements.

Hi-fi products should reproduce music so that what you hear is as close to the original sound as possible. Although this is no simple task, today's products have reached a standard of quality that sometimes pushes test equipment to the limits of measurement. For this reason, the test editors at *Audio* are constantly looking for new ideas to keep up with technical developments.

Test equipment

For example, using the spectrum analyzer together with the HP 339 Distortion Measuring Set, distortion factors down to 0.000 001% can be measured. A 1-kHz signal is fed to the unit to be tested (an amplifier or tuner) with an extremely low-distortion sine wave signal generator.

The output signal is fed to the distortion measuring set, which filters out the first harmonic. Remaining distortions are fed to the analyzer through the monitor jack of the 339. There, even the 24th harmonic can be measured.

Frequency response is also an important criterion of quality. In amplifiers it is normally a straight line. In pick-ups, tape recorders, cassette recorders and loudspeakers, a frequency response curve reveals the limits of technology.

Audio uses several methods to

Audio's measurement laboratory is equipped with an array of HP instrumentation.

determine frequency response. To test equalization, amplifiers and tape recorders are attached to the HP 3325A Function Generator/Frequency Synthesizer/Sweeper. It produces more than 400 successive frequencies in the range of 10 Hz to 20 kHz.

Next the analyzer gives to the computer the amplitude values produced by the test unit. The HP 9872A Four-Color Plotter reproduces the resulting curve as a diagram which can be printed in the magazine. This has an immeasureable advantage over traditional methods: the frequency response curve can be stored and reproduced at any time, even when the test unit is not on the premises.

Noise

The procedure does not apply for loudspeaker measurements, since sine wave signal oscillations at various frequencies, depending on the characteristics of the reproduction room, form standing waves and thus contaminate the measurements. These waves can be prevented by a signal which is usually regarded as undesirable in hi-fi equipment: noise.

Our deliberate noise is composed of a wide-band frequency mix in which individual frequencies occur in a statistically equal distribution. This mix is provided by the analyzer and directed to the loudspeakers through a measurement amplifier. A highly sensitive precision microphone picks up the actual sound and the analyzer evaluates it.

Wow and flutter

In mechanical instruments such as turntables or cassette recorders, wow and flutter play an important role. They provide information as to how uniformly the drive motors are running at their rated speed.

An optical view of the wow and flutter is obtained when a turntable is attached to the analyzer. A calibration disc with a 3-kHz sine wave signal produces a wider or narrower spectrum on the screen according to the quality of the unit. At the same time, the deviation from the ideal speed is measured.

In order to determine exact values for wow and flutter, however, *Audio* developed a phase-locked loop circuit which translates fluctuations into amplitude changes in a d.c. current. The result is read by the HP 3437A High-Speed Digital Voltmeter, which can take about 5000 measurements per second. This is sufficient to test even the most optimistic specifications from hi-fi equipment manufacturers.

Results

When all measurement values have been determined, they are printed out by the HP 9871A Printer in the form of a measurement protocol. This assures that the measurement results remain free of reading and transcription errors. The protocol is printed in the magazine as part of the corresponding test report.

These measurements are taken completely automatically. This is possible because the measurement instruments attached to the computer can be programmed through the computer. The software is constructed and continually expanded by *Audio* engineers. Thus a comprehensive program library, stored on six cassettes, was developed in less than two years.

Automation of the measurement set-up means not only a reduction in errors but also a considerable savings in time. The test editors are developing new test procedures constantly and devoting themselves intensively to another question: What conclusions about the sound produced by hi-fi equipment does this technical data allow?

Despite this extensive array of instruments, the "listening test" is still the most important one. Nonetheless, the goal which the *Audio* test division has set for itself justifies the expense. That goal is determining a clearly defined correlation between the listener's impression and the technical evaluation. And when this correlation can be consistently and precisely reproduced, it will permit an even more objective evaluation of hi-fi systems.

Jürgen Tries, an engineer on the editoral staff of *Audio* magazine, is responsible for *Audio's* test equipment handling and software development.
Audio
7000 Stuttgart 1
Postfach 10 42
Stuttgart
West Germany

Earthwork

by John Monahan and Bill Sharp
Hewlett-Packard Company
Desktop Computer Division

The lumberjack handling logs in some steamy Malayan jungle or the operator hauling ore from a Montana mine appreciates Robert E. Gilmore's assertion that, "quality is the cornerstone of everything we do at Caterpillar."

Equipment failure in the distant, or even not-so-distant, places where Caterpillar's yellow machines dig, scrape or haul can lead to lost time and money.

And so the company, which *Dun's Review* recently called one of the five best-managed in the United States, concentrates on the "development, adaption, and refinement of new technology," says Gilmore, who is president of Caterpillar Tractor Company.

Improving products

Engineers at Caterpillar's plant in Decatur, Illinois, U.S.A., are seeking greater fuel efficiency, enhanced machine performance, and better control over exhaust emissions and noise for the machines they make. These include several models of motor graders, wheel tractor-scrapers, off-highway trucks, and towing winches.

A tool in this search for improved technology is a Hewlett-Packard System 45A Desktop Computer, along with an HP 9872A Four-Color Plotter and a Bendix large-format digitizer.

Choosing a system

For many purposes the System 45 has augmented a time-shared terminal interfaced to a CDC Cyber 173 computer located miles from the Decatur plant. The committee of Caterpillar engineers who selected the desktop computer wanted a system that did not require punch cards or paperwork.

The committe also sought a system more powerful than a calculator but smaller than a minicomputer.

The System 45, according to computer engineer Terry Maurizio, provided interactive programming, graphics, hard-copy capability and permanent storage of data and programs.

Maurizio, who keeps a logbook of the System 45 usage, says the computer is operated more than eight hours per day — often after regular hours — by some 35 users.

Friendliness

In order to help introduce people to the System 45, a large library of programs is maintained. Maurizio has devised routines that describe different

Computing nine positions took half a day using a handheld calculator. Now, 181 positions are computed in 10 minutes.

programs available to the user, who can choose the desired program by pressing the Special Function Key that loads the selected program. Each of these programs contains prompts, which are interactive cues that lead the user through the program.

As Caterpillar engineers strive to improve their products, they have switched to the desktop computer many of the smaller jobs once done using the terminal, especially those performed frequently.

Designing

Gail Westendorf has used the desktop computer to help design new steering parts and the geometry of a hitch casting that connects the halves of an articulated scraper. Computing nine positions on a particular assembly took him half a day using a handheld calculator.

Now, 181 positions are computed in ten minutes. Westendorf, a mechnical engineer, wrote the hitch geometry program.

Drafting

Jack Stone, who is a drafting technician, had never used a computer prior to the System 45, he says. To compute, for instance, the center of gravity, section modulus and weight of a flat, irregularly shaped part once took an hour and a half using a handheld calculator, and an hour with the terminal. Now, Stone says, by using the digitizer to describe the part, he can have his answers in a matter of minutes.

Inspection analyst Donald E. Runyon (left), and quality control analyst Minor Oakley conduct noise level tests on a Motor Grader. Test comparisons between machines help identify defects.

Drafting technician Jack Stone digitizes data points into the System 45 from scale drawings.

Reliability

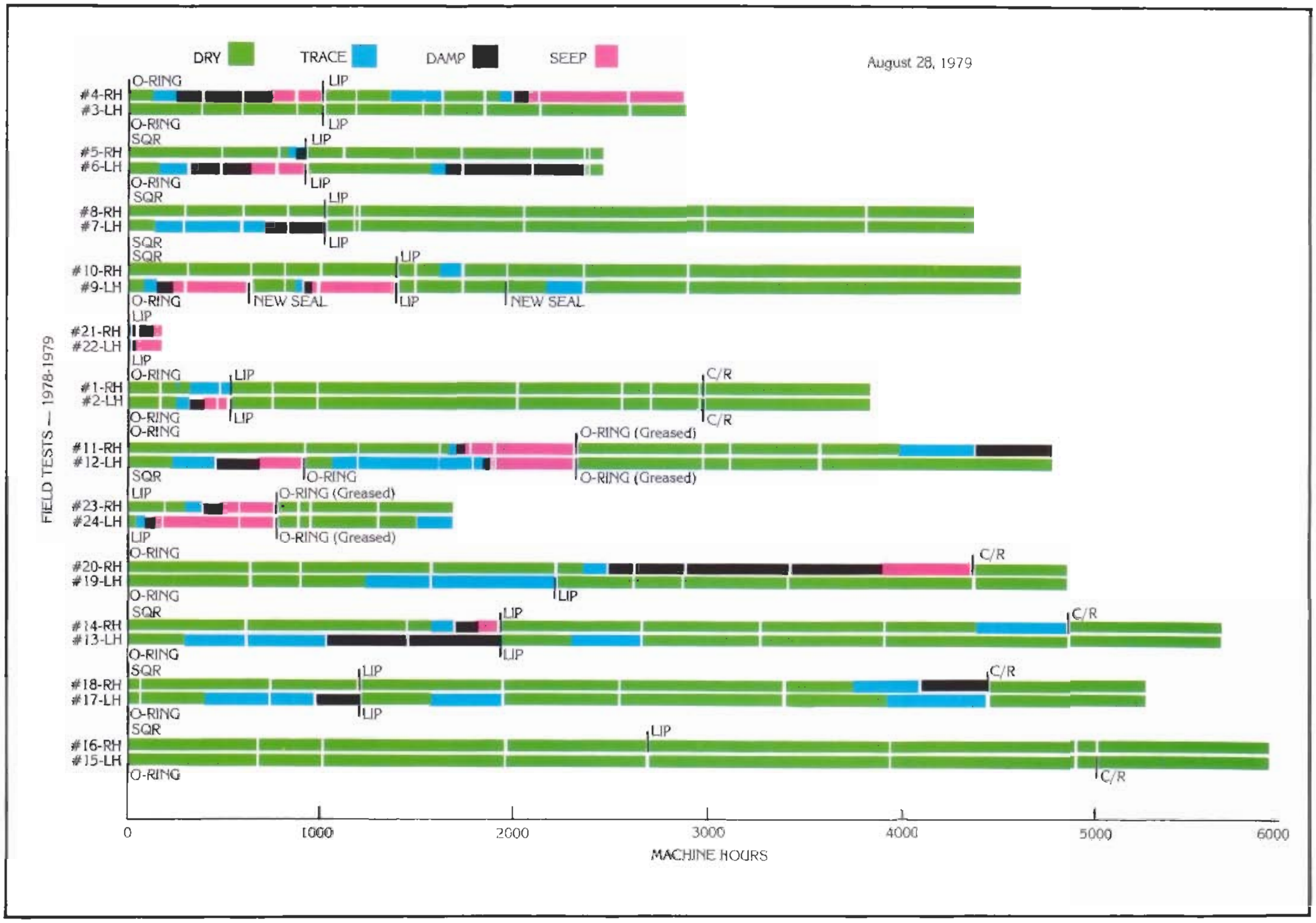
In performing failure studies for parts such as bearings, project engineer Jesse Matheny develops plots on the System 45 and the 9872 Plotter to help management determine warranties and product obsolescence.

The desktop computer has enabled him to tailor a program to his needs, since the "populations" he studies continually change. Thus, he can easily alter his statistics program as demanded by his changing subject matter.

Acoustics

Engineering technician Carl Higar's job is to ensure that sound-suppressed machines leaving the Decatur plant operate below a specified noise level. He uses a Bruel and Kjaer sound analysis system, including a 2218 sound meter, 2306 graphic level recorder and a 1621 tunable filter.

If a machine exceeds the noise limit, Higar records its sounds, and enters a digitized strip chart of the sound levels into the System 45, which in turn produces breakdowns of the



Life test data about oil seals is presented in this plot. Project engineer Jesse Matheny uses the plots to display statistical data.

total sound into various frequency ranges. By comparing breakdowns for noisy machines against those of quieter machines, Higar can determine what parts are causing the problems.

A noisy machine, he says, "once cost a week in test and analysis time; by using the System 45 to analyze the test data, we have reduced this time to about 2 days." In the future, Higar hopes to complete all testing within a day, by running acoustic test data directly into the desktop computer.

Finance

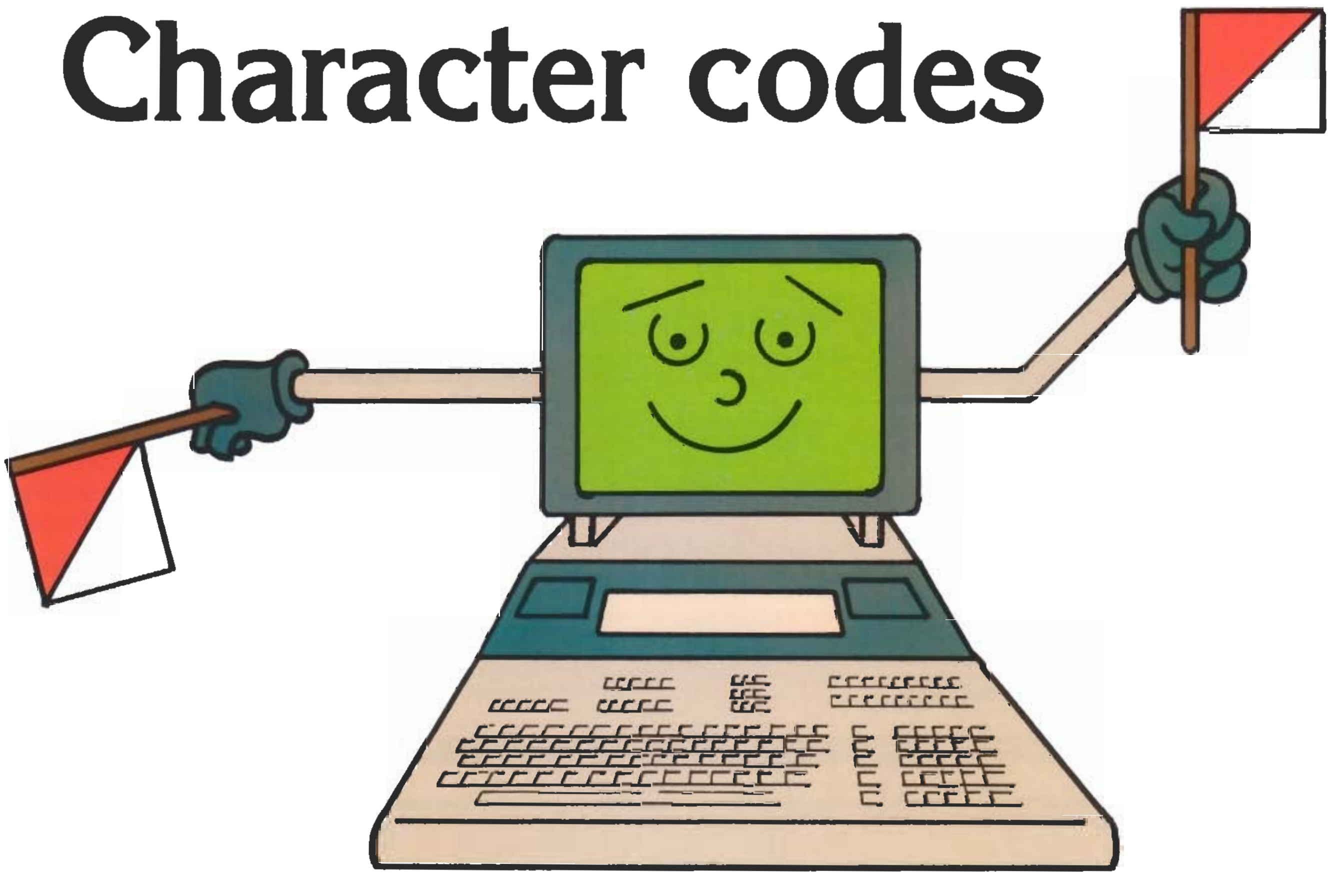
Unlike the engineers above, Dennis Kelly, supervisor of budgets and special projects, has used the Sytem 45 only to prepare chart presentations and statistical analyses. He has been working on a file that contains historical data which will enable him to perform regression analysis and graphically display trends. He also uses the computer to reduce clerical time when preparing budgeting worksheets. He wrote the worksheet and chart programs himself.

From Asia to Africa, America to Europe, Caterpillar machines are helping to do the world's work. But a machine, like any tool, is only as good as the people who designed it and the people who use it.

The men and women who move the earth can thank the men and women of Caterpillar's Decatur plant for the quality built into their machines. K

Caterpillar Tractor Company
N. 27th and Pershing Road
Decatur, Illinois 62525
U.S.A.

Character codes



by Steve Leibson
Hewlett-Packard Company
Desktop Computer Division

Language is quite possibly the most powerful component of civilization. Humans could not purposefully organize without shared language. Furthermore, the roots of all major human languages are verbal rather than visual.

Speech, our verbal use of language, would not be possible without the evolutionary heritage humans share that has produced our marvelously complex vocal tract, with lips, teeth, tongue, larynx and other organs we need to produce sound which others may understand. But the hardware of speech is not sufficient for shared understanding — a common language is also required.

Computer's alphabet

This series has been discussing the hardware components with which computers are built, allowing them to communicate with other machines. It is now time to discuss the languages computers use to communicate with other machines, rather than the equipment they use to do it.

Wanted: standard code

As covered previously, digital computers use a binary language for their internal communication. There are several methods for representing data internally in a computer, however, and it would be advantageous if there were some standard language that computers could use for communicating with other equipment.

In addition, it is important that such a language be compatible with human communications, since some of the devices that the computer will be communicating with are intended to interact with people. Printers and CRT terminals are examples of this type of equipment.

History of codes

The problem of creating a code, or computer language, that corresponds to an alphabet existed prior to the advent of computers. Even before electricity was harnessed for communications, man-made devices such as flags and semaphores were used to send messages.

Samuel Morse perfected the first code for electric data transmission, the Morse code. This set of dots and dashes is capable of representing the English alphabet and Arabic numerals so that intelligible messages may be interchanged between remote stations.

Early in this century, interest developed in replacing human telegraph operators with machines. Morse code was too difficult to mechanically decode, due to it's variable length per character.

But the idea of a standardized code was retained. The dots and dashes evolved into the concept of bits. Each bit could either be a "1" or "0",

b_4	b_3	b_2	b_1	Row ↓ / Column → ($b_7 b_6 b_5$)	0 (000)	1 (001)	2 (010)	3 (011)	4 (100)	5 (101)	6 (110)	7 (111)
0	0	0	0	0	NUL	DLE	SP	0	@	P	\	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Mnemonic and Meaning[1]		*Mnemonic and Meaning*[1]	
NUL	Null	DLE	Data Link Escape (CC)
SOH	Start of Heading (CC)	DC1	Device Control 1
STX	Start of Text (CC)	DC2	Device Control 2
ETX	End of Text (CC)	DC3	Device Control 3
EOT	End of Transmission (CC)	DC4	Device Control 4
ENQ	Enquiry (CC)	NAK	Negative Acknowledge (CC)
ACK	Acknowledge (CC)	SYN	Synchronous Idle (CC)
BEL	Bell	ETB	End of Transmission Block (CC)
BS	Backspace (FE)	CAN	Cancel
HT	Horizontal Tabulation (FE)	EM	End of Medium
LF	Line Feed (FE)	SUB	Substitute
VT	Vertical Tabulation (FE)	ESC	Escape
FF	Form Feed (FE)	FS	File Separator (IS)
CR	Carriage Return (FE)	GS	Group Separator (IS)
SO	Shift Out	RS	Record Separator (IS)
SI	Shift In	US	Unit Separator (IS)
		DEL	Delete

In this table of ASCII characters, the most significant three bits are shown at the head of each column, in both binary and hexidecimal. The least significant four bits are shown for each row. To determine the ASCII code for an upper case H, use the most significant three bits of column four, 100; and the least significant four bits from row eight, 1000; to form the binary code 1001000. Columns 0 and 1 are non-printing control characters. The rest are printable except for the last character, DEL, which is the delete character.

represented by either the presence or absence of an electrical signal.

The first code to use bits for machine communications used five bits to encode the alphabet. Five bits allowed 32 characters to be represented. Since the English alphabet has 26 characters, and at least ten numerals also need to be represented, it seemed that there would be a problem. There were not enough codes to go around.

Shift codes

The problem was solved by having two special codes that did not represent characters. Instead, these codes were shift codes. One code, called Letters, caused all following codes to be interpreted as letters of the alphabet. The other code, called Figures, caused all following codes to be interpreted as numerals and punctuation marks.

Such special codes were called shift codes because they shifted between different character sets. There were two five-bit codes that were in wide use, called Baudot and Murray.

These codes were very similar in concept, but varied in some assignments of codes to characters. The existence of two competing codes led naturally to the first I/O incompatibility problems.

A flaw

Character codes that relied on shift characters for proper operation were troublesome, because the interpretation of the incoming codes relied on the previous history of the message. Unless the receiving device knew which character set to use, there was a 50% chance of erroneous decoding.

Clearly, five bits were not enough. A code that could represent all the printable characters, and which did not rely on previous transmissions for unabiguous decoding, was needed. In addition, some sort of expandability was desired to prevent another dead end system.

Modern codes

By the time the need for this new code was felt, and the technology which could handle more complex codes became feasible, many manufacturers were involved in constructing electronic equipment which might also use the code. Whenever the need arose for such standardization, there were two methods of satisfying it.

A single manufacturer could simply go out and invent a solution and expect the rest of the industry to follow. This was the route taken by IBM, which invented the EBCDIC (Extended Binary Coded Decimal Interchange Code) character code. EBCDIC is an eight-bit code allowing 256 characters to be represented. Since there aren't that many printable characters, there are some unused codes in EBCDIC.

ASCII

The other method for obtaining a standard was through compromise in a committee. Other manufacturers did meet in order to develop a national standard called ASCII (American Standard Code for Information Interchange).

ASCII is a seven-bit code allowing 128 characters to be represented. This includes the alphabet, both upper and lower case, the numerals 0 through 9 and punctuation marks such as the period and the question mark.

In addition, several codes exist to control the operation of the device receiving the message. Codes representing Carriage Return and Line Feed are evident to anyone who uses a typewriter. Other control codes

Just as differences in language can create communication problems for humans, character code incompatibility can render an otherwise operable interface useless.

include Form Feed, Bell and Horizontal and Vertical Tabs. These codes are clearly for control of various printing devices, although manufacturers of some products have used these codes for other machine dependent actions.

Finally, there are codes used to control how the receiving device will interpret subsequent data. There are two shift characters, called Shift In and Shift Out, used to switch between character sets (English letters aren't the only kind, you know). There are also control codes that delimit text; STX (Start of Text) and ETX (End of Text).

ASCII has been a very successful character code. Thousands of instruments and computer-related products use this code for I/O. Even IBM now offers equipment that uses ASCII. Several interfaces have been covered in this series, and all except the BCD interface may be used to transmit and receive ASCII code.

Planning the escape

The planners of ASCII tried to foresee as many different applications as possible. That is the reason for including the various control codes. They recognized that technology's advance could not be totally predicted and therefore gave themselves an escape clause.

One of the ASCII characters is called the "escape" character. This character designates that the characters following have a special meaning.

The intent in creating the escape sequence was to extend the range of the character set by selecting from a range of available sets. Graphics, nationalized character sets, and special application sets have been developed for selection with certain escape sequences. Escape character sequences allow for a much richer variety of characters than the simple shift in/shift out scheme of the five-bit codes.

The now common CRT terminal has provided the escape sequence its widest application, however. The inclusion of microprocessors in terminal design has greatly augmented CRT capabilities. The serial communications link to these terminals has not been changed in years. One data channel to the host computer is all that is available.

Ordinarily, any characters that are received via this channel are printed on the terminal screen. But capability for character and line deletion, display enhancements such as inverse video and underlining, and even control of tape drives in the terminal does not exist in the ASCII standard. The escape sequence allows for these new capabilities.

Creativity

Manufacturers of CRT terminals are now adding increased performance to their products via escape sequences. Unfortunately, since the actual effect of these sequences is not covered in the ASCII standard, the terminal designers have felt free to create their own standards.

For example, one major feature now found on most CRT terminals is cursor positioning. The ability to place a cursor anywhere on the screen directly is important for many types of form-filling applications. There are about as many escape sequences for performing this task as there are CRT terminal manufacturers.

They all work similarly. The host computer sends the terminal an escape character. This is followed by a second character indicating that the escape sequence is for cursor positioning. Two more characters follow, giving the X and Y positions for the cursor. Usually the sequence is self-terminating, meaning that four characters including the escape are all that the computer need send.

After receipt of the fourth character, the terminal performs the action requested and prints any further characters received. Note that the ASCII characters in the escape code sequence are not interpreted as printing characters, but as control characters. The escape character has the effect of temporarily converting all ASCII characters to control.

Code conversion

The majority of computer equipment uses ASCII character representation today. Unfortunately, some of the older equipment still in use may not.

Interfacing to these devices requires that the ASCII characters the computer would like to send must be converted to the characters that the peripheral would like to receive. Here we are assuming the hardware interfacing requirements have already been met.

In addition, some modern peripherals may have odd requirements that can be met only through code conversion. An example is a printer that automatically inserts line feeds whenever it receives a carriage return. Unless the application calls for double spacing, the printout won't be as desired, since many computers send both carriage return and line feed to denote the end of a line of text.

One solution to this problem is to have the computer convert all line feeds to non-printing characters, such as "nulls." Most peripheral devices ignore the null character, which is the ASCII zero.

Character codes are yet another source of incompatibility in the world of I/O. Just as differences in language can create communication problems in humans, character code incompatibility can render an otherwise operable interface useless.

Fortunately, if the computer has a language that is rich in I/O capability, even this language barrier can be overcome. K

Accounting for costs

by Bill Sharp
Keyboard editor

Marilyn Nierman of the Cost Accounting Department at Hewlett-Packard's Desktop Computer Division has been using desktop computers since 1970, when she started with a 9100A.

"The 9100 performed work that can now be done on HP programmable calculators; so there's really no comparison with the current desktop computers," she says.

The department began using the 9830 soon after its introduction, and has since added the System 45 to its work force. They now use two of each, and borrow more computers from the division's "loan pool" when the work load is especially heavy.

Variancing, costing and targeting

This does not mean that all accounting work is done on desktop computers. The department has its own HP 3000, programmed to handle the majority of accounting functions. Cost accounting efforts fall primarily into three areas:

- Variancing, a coined term, the calculated deviation of actual production costs from expected (or standard) costs. The object is to value inventories at a computed standard.
- Costing, the process of determining the cost of producing a product, based on labor trends, material requirements and yields, where applicable. This data is instrumental in product pricing and variancing.
- Targeting, the process by which managers determine what their financial needs will be, based on sales projections, cost trends and department objectives. This data is used for monthly department comparisons to actual expenditures.

A yield is the success rate of a process, particularly when the process has a significant level of mortality due to its complex nature. For example, a reasonable yield for IC chip production might be about 12%; that is 12 of every 100 produced performing according to specifications.

Account clerk Chris Habel runs the IC costing program on the System 45.

Many of the tasks in variancing and costing categories are performed using the HP 3000 computer, which is designed to work with the huge data base required. But existing HP 3000 programs do not accomodate the yields figures that are essential to variancing and costing at HP divisions manufacturing ICs.

Because of this, DCD accounting personnel have written and enhanced variancing and costing programs. These were originally written on the 9100. As the programming complexities grew, the 9830 was used.

"Programming experience was not a prerequisite for me to learn how to program the 9830," says Marilyn. "I began by reading the manuals and then just sat down and punched in programs until I got what I wanted.

"In most areas of our work, it's much quicker to write a program to do a new job on one of our desktop computers than to try to create the same capability in the big mainframe."

"And, when compared with our old manual methods using working papers for variancing and costing, the 9830 greatly increases the volume of work that can be done in a given period. It takes out the human errors in calculations, and makes it easier to control the quality of information input."

Costing

Bob Morse, like Marilyn a cost accountant, has just transferred the costing program from the 9830 to the System 45.

"With the detailed printouts we get using this program, a manager can see present performance in one column, and his target performance right next to it," says Bob. "A manager can easily spot problems if they exist, and has the information here to determine just why there is disagreement between his performance and the established standard for the process."

Targeting

When the targeting period rolls around in late summer, the System 45 and a program written by Marilyn can be used by each manager to take some of the sting out of this task. The program prompts the manager for the different areas of expenses that must be taken into account. The manager supplies the necessary data.

This information, stored on a data cartridge, is returned to the Cost Accounting Department, where similar inputs from all the managers are transferred to the 3000, which produces a division targeting report.

"It used to be that you filled in a big matrix with amounts that were tediously computed by hand, and then you manually summed monthly columns and category rows," says Marilyn. Her System 45 program now takes a lot of the headaches out of the process for managers, and makes data transfer to the 3000 all but error-proof.

Hardware and software for desktop users

Data communications

Scientists and engineers often need files from the data base of a large computer copied down to a computer sitting on their desk. The new Hewlett-Packard Data Communications packages enable System 35 or System 45 Desktop Computers to provide stand-alone computer power, or become terminals capable of communicating directly with a large mainframe. The Data Communication packages are designed specifically for flexibility and ease-of-use in both synchronous and asynchronous modes.

Asynchronous transmission sends data, one character at a time, to the host computer, with the user interacting. The synchronous mode transmits data characters in a continous stream, allowing unattended data analysis.

Data base management

Efficient assembly of technical data for easy and speedy access is another common need of scientists and engineers. By using a data base, a user can group logically related files containing the necessary data to satisfy his needs.

Hewlett-Packard's new System 45 Data Base Management pack, called IMAGE/45, includes Query/45 software, the primary data base manipulation tool. By typing in English language-like commands and responding to Query/45 prompts, a user can read, enter, delete or modify data without writing applicaton programs.

Included in this package is a Data Base Design Kit that takes a user, step-by-step, from the definition of a data management problem to a workable solution.

Assembly language

High-level language such as HP Enhanced BASIC satisfies most users' needs. But for specialized I/O routines or tasks involving several computations, a low-level language with BASIC intermixed provides a fast alternative.

Hewlett-Packard's new Assembly Language ROM is designed to enhance the System 35 and 45 Desktop Computers by increasing speed in time-critical portions of BASIC programs. The programmer has complete control over the System 35 and 45 central processing units through the use of machine instructions and extensions to the BASIC Language. Specialized subroutines written in Assembly can be called from a BASIC program.

The biggest feature of the System 35/45 Assembly Language capability is speed. The relocating assembler generates machine code at the rate of 400 to 800 statements per second. This speed is possible because the assembly process is integrated and the source statements are syntaxed as they are entered.

For more information about HP's Data Comm, Data Base Management or Assembly ROM packages, contact your local HP sales and service office or write to *Keyboard*, 3404 East Harmony Road, Fort Collins, Colorado 80525, U.S.A.

9030 measurement and control system

Product development engineers often run up against a mechanical/electronics interfacing headache. There can be compatibility problems when these two types of equipment are interfaced in the final product.

One way to approach this is to use a system, in operating the mechanical parts under development, that is designed to simulate different controller or microprocessor boards. With such equipment, you can simulate control for a mechanical system under development, be it a washing machine or an automated manufacturing process. And, you can simulate different electronic configurations to suit changing product designs.

Hewlett-Packard now offers the 9030 Measurement and Control System, which is designed to do meet these needs. It combines products from two HP divisions, and offers:

- A wide selection of components to tailor the system to specific needs.
- A system completely pre-assembled and tested before shipment.
- Portability, to get the system to where the work is located.
- Plug connectors to allow you to easily connect and disconnect a

particular set of equipment and move to another site to connect another set of equipment.

- Software that tests for correct operation of all system components.
- Software that provides an exerciser system to make it possible for you to make the system start working right away.

The 9030 consists of a desktop computer (System 35A, System 45B in "T" configuration or 9825S), an instrumentation cabinet containing an extended-performance version of the HP 2240A Measurement and Control Processor, room for up to six trays of measurement and control electronics and a series of optional Signal Conditioning Cards to tailor the system to specific needs.

What results is a system that is compact and attractive, and, more importantly, a tool which is ready to be put to work on arrival.

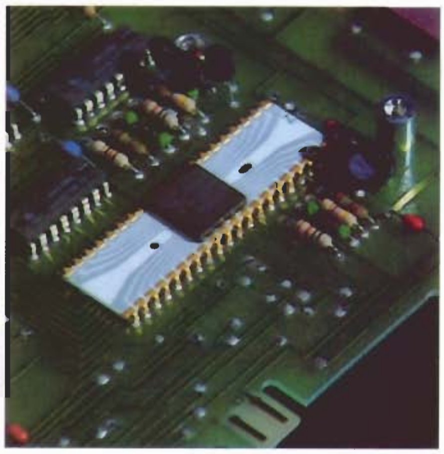
Control subsystem

A silicon-on-sapphire (SOS) microprocessor gives the 2240 its own intelligence, making it possible for the subsystem to perform its tasks independent of the desktop computer. So, while the subsystem does work previously specified, the desktop computer can be used for data reduction, computation or control functions, including communication with external peripherals. The desktop computer and 2240 are connected by an HP-IB interface.

The delivered system includes the 2240A Measurement and Control Processor and Signal Conditioning Cards mounted in the cabinet, the desktop computer of your choice, I/O ROM, HP-IB cable, manual and the system exerciser software.

For more information on the 9030 Measurement and Control System, contact your local HP sales and service office or write to *Keyboard*, 3404 East Harmony Road, Fort Collins, Colorado 80525, U.S.A.

Microprocessor development software



The discovery and development of the microprocessor has opened up seemingly limitless possibilities for industry and consumers alike. And yet, for all its unquestionable value, the microprocessor is not without problems.

While operation of the microprocessor is fairly straightforward, programming it to do what you want is relatively complicated. To simplify the task of programming microprocessors, Hewlett-Packard offers the Microprocessor Development Software Pack.

Use System 35A or 45B

This is not a microprocessor development system, but a software pack that has been designed to run on either the System 35A or 45B Desktop Computers. This makes it possible for the desktop computer user to program microprocessors through the friendly, easy-to-use features of the desktop computer.

Four modules

The software pack includes four modules:

- INITIALIZE, a routine that helps the user specify parameters such as printer select code, paper width and source code storage unit.
- EDITOR, which enables the user to write, modify or correct the source code in the language of the chosen microprocessor. The module also includes a syntax check at the time of input, automatic numbering of lines and FIND/CHANGE statements to permit automatic modification of specified characters or strings in code.
- ASSEMBLER, which translates the user-written source code into an absolute machine language object code.
- CONSOLE, which conducts all debugging within a microprocessor system which is supplied by the user. The physical connection can be an HP RS-232-C Interface. Communication between the desktop computer and the microprocessor system is handled by the CONSOLE module.

For the System 35A, cartridge 09835-12530 is the Motorola 6800 version; 09835-12540 is the Intel 80/85 version; and 09835-12550 is the Zilog Z80 version. All three 35A cartridges are included in 09835-12560.

With the System 45B, 09845-12530 is the Motorola 6800 version; 09845-12540 is the Intel version; 09845-12550 is the Zilog Z80 version. All three 45B tape cartridges are included in 09845-12560.

For more information on HP's Microprocessor Development Software pack, contact your local HP sales and service office or write to *Keyboard*, 3404 East Harmony Road, Fort Collins, Colorado 80525, U.S.A.

Programming Tips

Tricks with 9825A/S string buffers

When there is a need for an I/O buffer, you may choose to allocate it as a string to maximize I/O speed.

Unprocessed data in the I/O buffer can be stored as a string on a tape cartridge or disc without first converting data to the 9825's internal format. Later, when there is time to process the data, the string can be loaded back into the string buffer area to be accessed with read or read binary statements or string statements.

A problem will arise when the data is stored on a floppy disc. You will find that when reloading the data into the string the buffer pointers are lost, which makes it impossible to read data from the buffer using read or read binary (i.e., red and rdb) statements.

The technique that follows can be used to save buffers without losing pointers. It also can be used to write a single buffer to several peripherals.

Restore the pointers

To restore the buffer pointers, you must first store the status (length) of the buffer with the string. Then, when the data is to be retrieved, load the string and the buffer status from the disc. The buffer pointers can be restored by writing the string to buffer using the wtb statement.

Here is example code to save the buffer:

```
dim A$[36]
```

Dimension a string 16 characters longer than the needed buffer.

```
buf "Buffer",A$,1
```

Allocate a buffer of any type.

```
rds ("Buffer")→L
```

Save the number of characters in the buffer.

```
sprt F, A$, L
```

Write the string data and buffer to disc, using either random or serial print.

Here is example code to retrieve the data and restore the buffer:

```
dim B$ [36]
```

Dimension a string 16 characters longer than the buffer.

```
buf "New", B$, 1
```

Set up a buffer of any type.

```
sread F,B$,L
```

Recall data from disc using either random or serial reads.

```
wtb "New",B$[1,L]
```

This statement will reset the buffer pointer for any length buffer.

NOTE that if you can guarantee your buffer will be completely full, then you do not need to store the status of the buffer on disc, and the buffer pointers can be restored with the following line of code:

```
wtb "New",B$
```

Another method to retrieve numeric data from the string is to use the string value and numeric functions.

This technique is slow because it uses HPL rather than the I/O ROM assembly-level internal formatter to convert the string data to internal numeric form. It also can involve some very complex programming techniques, depending on the format of the string data.

The first technique, restoring the buffer pointers, is a faster and cleaner way to retrieve the data.

Background

The internal representation of a simple string consists of two parts — the data area and a four-word string pointer area (see Figure 1). When the string is defined as a buffer, several things occur.

First, 16 bytes (8 words), are taken from the end of the string data area and used for buffer pointers. Second, the string pointer DLEN is shortened by 16 characters. Third, the string is filled with blank characters, an action which also sets the current length to the defined length, that is DLEN=CLEN (see Figure 2).

When the string buffer is saved on the 9825 internal data cartridge, the record file statement (rcf) uses the string pointer, SIZE to determine how many words should be written to the tape. Since SIZE points to the end of the buffer pointer area, the buffer pointers are also saved on the tape and retrieved when the string is reloaded.

However, the reason that the buffer pointers are not stored on disc is that the disc print commands (sprt and rprt) use the string pointer DLEN to determine how many words should be written to the disc. Since DLEN only points to the end of the string data area, the buffer pointers are never recorded.

Writing one buffer to several peripherals

Recall that transferring data from a buffer to a device will clear the buffer; that is rds ("Buffer")=0. This is because the buffer pointers have been reset, not because the data is erased. By writing the string back to the buffer, the buffer pointers can be restored. Again, wtb "New", B$ can only be used if the buffer is completely filled.

Example programs

For the following programs, a separate program partially filled a string buffer with 1000 values of format XX.XX. Each value was separated by a comma (,) and the values were terminated with a CR LF. Here are sample programs that will recover data with either string commands or with buffer techniques.

Recover data with string commands:

```
0: files big
1: dim B$[12016]
 ,B[1000]
2: buf "New",B$,
 1
3:
4: "read the  ":
5: "the string":
6: "from disc ":
7:
8: sread 1,B$,L
9:
10: "98035     ":
11: "real time":
12: "clock     ":
13: "commands  ":
14:
15: wrt 9,"U2H/
 U2C/U2G/U2H/
 U2V";red 9,C
16: wrt 9,"U2C/
 U2G"
17:
18: "loop to   ":
19: "recover   ":
20: "1000      ":
21: "values    ":
22: "of format":
23: "XX.XX     ":
24:
25: for I=0 to
 999
26: I*6+1→J
27: val(B$[J,J+
 11])→B[I+1]
28: next I
29:
30: "98035     ":
31: "real time":
32: "clock     ":
33: "commands  ":
34:
35: wrt 9,"U2H/
 U2V";red 9,V
36:
37: "print     ":
38: "loop time":
39:
40: wrt 701,"str
 ing loop time:"
 ,V-C
41: end
*18090
```

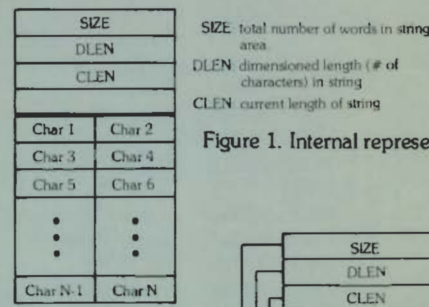


Figure 1. Internal representation of a string

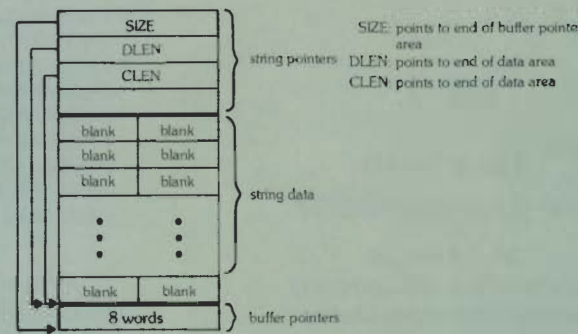


Figure 2. Internal representation of a string buffer

string loop time: 5549.00 msec†

Recover data with buffer techniques:

```
0: files big
1: dim B$[12016]
 ,B[1000]
2: buf "New",B$,
 1
3:
4: "read the  ":
5: "the string":
6: "from disc ":
7:
8: sread 1,B$,L
9:
10: "98035     ":
11: "real time":
12: "clock     ":
13: "commands  ":
14:
15: wrt 9,"U2H/
 U2C/U2G/U2H/
 U2V";red 9,C
16: wrt 9,"U2C/
 U2G"
17:
18: "restore   ":
19: "the       ":
20: "buffer    ":
21: "pointers  ":
22:
23: wtb "New",‡
 B$[1,L]
24:
25: "loop to   ":
26: "recover   ":
27: "1000      ":
28: "using red":
29:
30: for I=1 to
 1000
31: red "New",
 B[I]
32: next I
33:
34: "98035     ":
35: "real time":
36: "clock     ":
37: "commands  ":
38:
39: wrt 9,"U2H/
 U2V";red 9,V
40:
41: "print     ":
42: "loop time":
43:
44: wrt 701,"buf
 fer loop time:"
 ,V-C
45: end
*24216
```

buffer loop time: 3632.00 msec‡

†Times for programs without comment or blank lines.
‡Requires approximately 1 msec.

by Sue Bodoh,
Sales Development Engineer,
Hewlett-Packard Company,
Desktop Computer Division

Update

How fresh are your manuals?

by Ed Brovet
Hewlett-Packard Company
Desktop Computer Division

Your desktop computer manuals have a coded "freshness" date. The date is located on the rear cover, under the printing location message. The month and the year signify when the manual was printed, while the day of the month shows which edition of the manual you have (February 1, 1980 means a first edition printed in February, 1980, while October 3, 1979 means a third edition printed in October, 1979).

Manuals are updated periodically to include new or changed information, or to correct errors. When this happens, the printing date is changed. The following tables show the printing dates for the most current manuals for the System 35 A&B and System 45 A&B. If you feel that some of your manuals need to be replaced, contact your local HP sales & service office to order new manuals.

System 35 A & B

Part number	Manual Title	Printing date
09835-90000	Operating and Programming	August 3, 1979
09835-90001	Beginner's Guide	November 2, 1979
09835-90002	Preview	August 1, 1978
09835-90003	Read Me First	April 1, 1979
09835-90005	Owner's Manual	January 2, 1979
09835-90010	Reference Guide	January 2, 1979
09835-90015	Quick Reference	November 1, 1978
09835-90040	System Exerciser	April 2, 1979
09835-90050	Plotter ROM	August 1, 1978
09835-90060	I/O ROM	July 2, 1979
09835-90065	Advanced Programming ROM	June 1, 1979
09835-90066	Structured Programming ROM	November 1, 1979
09835-90070	Mass Storage ROM	January 2, 1979
09835-90080	Assembly Quick Reference	May 1, 1979
09835-90082	Assembly Execution ROM	April 2, 1979
09835-90083	Assembly Development ROM	February 1, 1979
09835-90600	Interfacing Concepts	September 1, 1979

System 45A

Part number	Manual Title	Printing date
09845-90000	Operating and Programming	September 3, 1978
09845-90001	Beginner's Guide	July 2, 1978
09845-90002	Programmer's Introduction	November 3, 1978
09845-90010	Reference Guide	February 3, 1978
09845-90040	System Test (replaced by 09845-90042)	
09845-90042	System Exerciser	October 1, 1978
09845-90050	CRT Graphics	July 2, 1978
09845-90060	I/O ROM	June 1, 1978
09845-90070	Mass Storage ROM	October 3, 1978
09835-90600	Interfacing Concepts	September 1, 1979

System 45B

Part number	Manual Title	Printing date
09845-91000	Operating and Programming	May 2, 1979
09845-91001	Beginner's Guide	May 2, 1979
09845-91005	Owner's Manual	May 2, 1979
09845-91015	Quick Reference	May 3, 1979
09845-91040	System Exerciser	May 2, 1979
09845-91050	Graphics ROM	May 2, 1979
09845-91060	I/O ROM	August 2, 1979
09845-91065	Advanced Programming ROM	September 2, 1979
09845-91070	Mass Storage ROM	October 3, 1979
09845-91080	Assembly Language Quick Reference	February 1, 1980
09845-91082	Assembly Execution ROM	February 1, 1980
09845-91083	Assembly Development ROM	March 1, 1980
09835-90600	Interfacing Concepts	September 1, 1979

Keyboard

Hewlett-Packard, 3404 E. Harmony Road
Fort Collins, Colorado U.S.A. 80525
Phone: 303/226-3800

Publications Manager - Ed Bride
Editor - Bill Sharp
Feature Writer - John Monahan
Editorial Assistant - Ms. Chris Stumbough
Art Director - Hal Andersen
Typography - Ms. Paula Dennee

For further information on HP products or applications, please contact your local Hewlett-Packard Sales and Service Office or write to Keyboard

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address. Send changes to Hewlett-Packard Keyboard, 3404 E. Harmony Road, Fort Collins, Colorado 80525, U.S.A. For Europe, send changes to Hewlett-Packard N.V., Post Box 529, 1180 AM Amstelveen, Holland.