# HEWLETT-PACKARD JOURNAL

# HEWLETT·PACKARD JOURNAL

Technical Information from the Laboratories of Hewlett-Packard Company

FEBRUARY 1983 Volume 34 ● Number 2

## Contents:

## In this Issue:

Last month's issue featured the Hewlett-Packard Interface Loop, HP's new two-wire serial interconnection system for low-cost battery-powered devices. In this month's issue you'll find articles on the first two HP instruments designed to operate on the HP-IL. Model 3468A, the subject of the article on page 3, is a digital multimeter, a basic instrument that measures voltage, current, and electrical resistance. Besides its HP-IL connection, the 3468A is interesting for its production process. The subassemblies that go into it aren't tested and adjusted individually before they're put together. Naturally, this lowers the production cost quite a lot, but by conventional manufacturing wisdom it's risky to the point of foolhardiness. To make the risk acceptable, new approaches were taken in both the design and the manufacturing processes. Currently more than 80% of the 3468As pass the acceptance tests when they're first turned on at the end of the production line. The long-term goal is 96%.

The other new HP-IL instrument is the 3421A Data Acquisition/Control Unit, pictured on this month's cover and described in the article on page 10. With the 3421A and an HP-41 Handheld Computer, you can have a portable system that will scan up to 30 channels, make a measurement on each one, and analyze the data for you. You can also set up the 3421A to actuate valves or relays to control the process it's measuring. The cost of this system is so remarkably low that many applications thought to be beyond the reach of automation can now benefit from it. One or two 3421As controlled by an HP-85 Personal Computer form an even more powerful system, the 3056DL Data Logger; it's described in the article on page 13.

To make it as easy as possible to write HP-41 programs to control the 3468A and the 3421A, many often-needed routines are available in a new read-only memory module that plugs into the HP-41. The ROM's story is on page 16.

Also in this issue are two articles on software packages for the HP 3000, HP's largest computer system. HPMAIL is an electronic mail system. Picture yourself at a terminal connected to your company's HP 3000 Computer, which might be just one node in a multicomputer network. With HPMAIL, you can use your terminal to exchange messages with any other terminal in the network. The system gives you an in tray, an out tray, a pending tray, a work area, and file folders—all electronic. It's a big step towards the all-electronic office, and its story begins on page 20.

While HPMAIL is for computer users, HPToolset is designed by programmers for programmers. Many software subsystems are needed in program development. A typical list includes an editor, a compiler, a segmenter, a symbolic debugger, and various utility programs. HPToolset ties all of these together so they look like one subsystem, saving programmers a great deal of time and thereby improving their productivity. The story of HPToolset begins on page 30.

*-R. P. Dolan*

# A Portable, Low-Cost, High-Performance Digital Multimeter for the HP-IL

*HP's first HP-IL instrument is the result of new design and manufacturing approaches. This DMM electronically calibrates itself, measures ac and dc voltages and currents, makes four-wire and two-wire resistance measurements, and uses a liquid-crystal display to output data, measurement units, and alphanumeric messages.*

by Jack P. Trautman and Lawrence A. DesJardin

**P**ORTABLE, ACCURATE MEASUREMENTS of voltages, currents, and resistances are a necessity for electronic service personnel and other users operating in the field. Large amounts of data often must be collected and analyzed. If this data can be output by the measuring device directly to a portable computer, the need for considerable manual effort to record the data and later enter it for analysis can be avoided. Low instrumentation cost is important because, unlike a laboratory situation where many persons can have convenient access to one piece of equipment, field personnel each require a dedicated set of equipment. High reliability can also reduce field support costs.

To satisfy these needs, HP developed a low-cost serial interface, the Hewlett-Packard Interface Loop (HP-IL).[1] This interface allows bench measurements by a high-performance digital multimeter to be coupled to the advanced computational power of a handheld computer. The

3468A Multimeter (Fig. 1) is the first HP instrument designed for use on the HP-IL. It is a low-cost, autoranging, 5½-to-3½-digit, five-function DMM with 1-$\mu$V sensitivity designed for bench and portable applications. A rack-mount version with identical features, the 3468B, is also available. This microprocessor-based instrument can measure dc volts, true-rms ac volts, two-and four-wire ohms, and ac and dc current. Its simple, fast, electronic calibration eliminates manual adjustments to provide a lower cost of ownership. The standard HP-IL interface on the 3468A allows complete programmability with small computers such as the HP-41 Handheld Computers or the more powerful HP Series 80 Personal Computers.

## Features

Several important features are contained in the 3468A's small package. These include:

■ **HP-IL for low-cost automation.** When the 3468A is con-
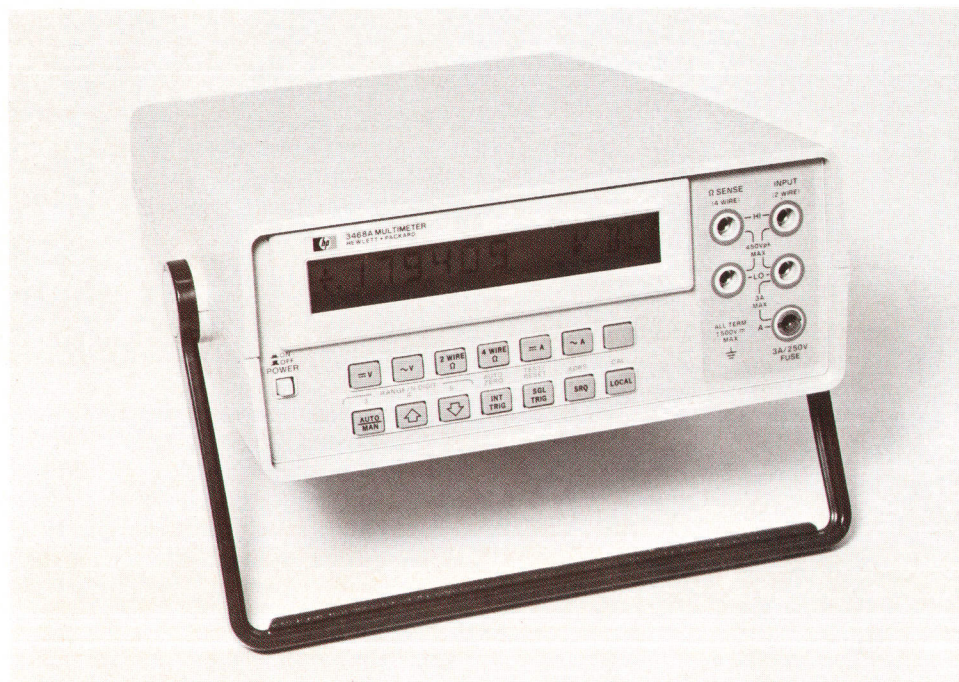


**Fig. 1.** *The HP Model 3468A Multimeter is HP's first HP-IL instrument. It has an alphanumeric liquid-crystal display, measures ac and dc voltage and current, and determines resistance by both two-wire and four-wire methods.*

nected to a handheld computer through the HP-IL, software can easily be developed that stores and analyzes data for many applications. For example, to measure temperature, an HP-41 can be used to linearize the output of a transducer device measured by the 3468A. Then, the HP-41 can be programmed to output the results in degrees Celsius or degrees Fahrenheit on the display of the 3468A. For audio and telecommunication applications, the 3468A can display power in dBm referenced to any impedance by using a simple HP-41 program. For applications such as testing resistor tolerance or performance of a device, the HP-41 can easily get data from the 3468A, perform a percent-error calculation, and display the results in percent on the 3468A. The software and documentation for the 3468A include bar-code programs for the HP-41 that reduce system programming time. Peripheral devices, such as the 82162A Thermal Printer and 82161A Cassette Drive, are available to expand an HP-IL system to provide data recording and mass storage.

■ **Interrupt capability.** The front-panel **SRQ** key on the 3468A allows the user to interrupt a running program on the computer controlling the 3468A and cause it to execute another program.

■ **High-performance bench measurements.** The 3468A can measure dc and true-rms ac voltage in several ranges from 0.3V to 300V full scale, with 1-$\mu$V to 1-mV sensitivity, respectively. This wide dynamic range enhances the number of measurements that can be made with the 3468A. The 3468A's ac converter is true-rms-responding, and provides accurate detection of nonsinusoidal waveforms with crest factors up to 4:1. With a bandwidth of 100 kHz on all ranges and an extended bandwidth of 300 kHz on the 30V range, the 3468A provides a broad range of ac detection. Resistance measurement ranges from 300 ohms to 30 megohms with respective sensitivities of 1 milliohm to 100 ohms are available for both two-wire and four-wire measurements. The four-wire ohms function can be used to reduce errors caused by lead resistance and the two-wire ohms function is convenient for bench measurements. Both dc and true-rms ac current measurements are provided up to 3A. All functions on the 3468A incorporate fast autoranging to provide answers quickly and accurately.

■ **Smart display.** The 3468A DMM uses an easy-to-read, 14-character, alphanumeric liquid-crystal display that provides measurement units as part of the reading for unambiguous answers. Annunciators below the reading show instrument status and the status of the HP-IL interface. With the HP-IL interface, words or messages can be arbitrarily displayed on the 3468A for prompting or displaying answers in calculated units such as degrees or revolutions per minute.

■ **Rechargeable battery.** The optional battery pack includes a rechargeable battery and charging circuitry to allow up to five hours of continuous portable measurements. For applications in the field, the 3468A and other battery-powered HP-IL devices can be set up to provide a completely portable measurement system.

■ **Electronic calibration.** Complete calibration of the 3468A is done electronically without any internal man-
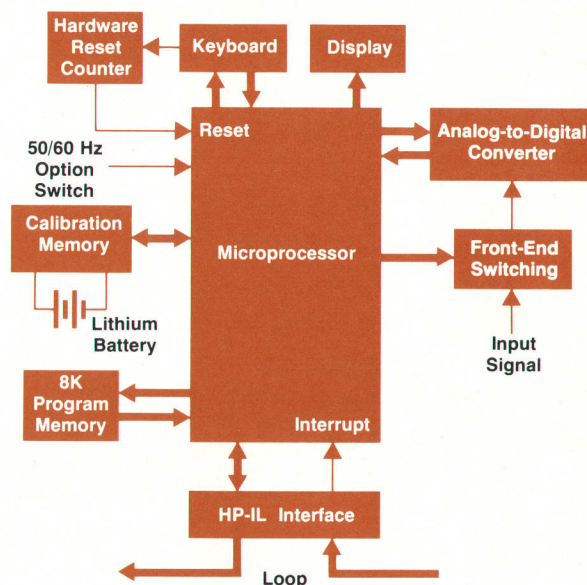


**Fig. 2.** *Simplified block diagram of 3468A.*

ual adjustments. All functions can be calibrated either from the front panel or remotely by an automatic calibration system.

■ **Self-test.** The 3468A has a built-in self-test feature initiated by pressing the **TEST/RESET** key on the front panel. A comprehensive check of both analog and digital circuits is performed while activating all display segments. Upon successful completion of self-test, the message SELF TEST OK is displayed.

## A Look at the Design

The 3468A achieved its goals of low cost, reliability, and high performance by a conscious effort to select each component of the voltmeter not only in terms of performance, but also in terms of producibility.[2]

A look at Fig. 2 reveals that the 3468A's block diagram is similar to the block diagram of a conventional voltmeter. The input signal is transformed by the signal conditioning circuitry of the front end to a voltage between +10V and −10V (±3V in the ac modes). This voltage is digitized by the analog-to-digital (A-to-D) converter to the resolution selected by the user. The microprocessor controller linearly corrects this value using constants that have been stored in the nonvolatile calibration memory for this particular function and range. The corrected value is shown on the liquid-crystal display of the front panel and can be output onto the HP-IL.

The photograph of the interior of the 3468A (Fig. 3) shows that the design of the instrument is much less conventional. A single printed circuit board houses all the electronics but the display. Several contrasts with conventional voltmeters can easily be noted. The first is the absence of Teflon™ standoffs, previously considered a necessity to ensure high impedances on precision analog boards. A special process was developed that guarantees high impedances across the board's surface as long as the surface remains clean. A vigorous program of instituting clean handling was adopted to safeguard the integrity of the

# HP-IB Version of 3468A

The 3478A Multimeter (Fig. 1) is a low-cost HP-IB (IEEE 488) programmable instrument for system measurements. A companion product to the 3468A with similar display and measurement features, the microprocessor-based 3478A is an autoranging, 5½-to-3½-digit, five-function digital multimeter designed for automated testing in production or research and development. Its simple, fast electronic calibration eliminates all manual adjustments to provide a lower cost of ownership.

### High Performance at Low Cost

The 3478A can measure dc voltage with 100-nV sensitivity on the 30-mV range up to a maximum reading of 300V. It can measure ac voltage from 300-mV full-scale range with 1-$\mu$V resolution up to 300Vrms. This wide dynamic range minimizes the amount of signal conditioning required to detect low-level signals and higher voltages. Either two-wire or four-wire resistance measurements can be selected with a maximum range of 30 megohms and a 100-$\mu\Omega$ sensitivity on the 30-ohm range. Both 300-mA and 3A ranges of ac and dc current are provided, completing the function capability of the 3478A. Fast autoranging improves system throughput by minimizing the time to select the correct range.

The 3478A can perform production tests or acquire experimental data at 71 readings/second with 3½-digit resolution, or obtain 33 readings/second with 130 dB of noise rejection using 4½-digit resolution. Increasing the resolution of the 3478A to 5½ digits reduces the measurement speed to 4.4 readings/second.

### Designed for HP-IB Systems

All of the performance of the 3478A can be controlled over the HP-IB by a wide range of compatible computers. Complete programmability of functions, ranges, and other modifiers on the 3478A makes it ideal for use in a fully automatic test system.

With an extensive set of service request (SRQ) interrupt capabilities, including data ready, front-panel **SRQ** key, and power-on SRQ, the 3478A allows more flexibility in a system. The data-ready service request of the 3478A can be programmed to interrupt the computer after a measurement has been taken, allowing more efficient use of computer time. The front-panel **SRQ** key lets a system operator interrupt the computer from the 3478A's front panel. The 3478A can also flag the computer at power-on to indicate loss of power.

Switchable front and rear terminals on the 3478A permit convenient measurements at a bench using the front terminals or system inputs using the rear terminals. The setting of the front/rear terminal switch on the 3478A's front panel can be sensed remotely over the HP-IB, so both sets of terminals can be effectively used in any system configuration.

A **VOLTMETER COMPLETE** output on the rear panel of the 3478A provides a signal after a measurement is completed. This output can be used to advance a scanner to the next channel without computer intervention, thus enhancing scanning speed. An **EXTERNAL TRIGGER** input is also provided on the 3478A to initiate measurements and synchronize with an external device.

### Fast Electronic Calibration

Like the 3468A, complete calibration of the 3478A is done electronically without any internal manual adjustments. All functions can be calibrated either from the front panel or remotely in an automatic calibration system.



**Fig. 1.** The HP Model 3478A Multimeter is the HP-IB version of HP's new, low-cost, high-performance digital multimeter design. It combines the display, measurement, and electronic calibration features of its companion instrument, the 3468A, with HP-IB operation capabilities and higher performance.

boards. Elimination of the standoffs significantly reduces the amount of manual labor required to assemble the voltmeter, and thus lowers its cost.

### Electronic Calibration

Even more conspicious is the absence of any manual adjustments or "tweaks." This is achieved by a design that forces significant imperfections (e.g., voltage offsets, imperfect resistor matching) to be linear errors, that is, errors of offset and gain. This allows readings to be corrected by simple math routines that subtract the offset value and then perform a multiplication to scale the final answer correctly as follows:

$$R = (R' - Y)G$$

where R is the final calibrated reading, R' is the raw uncalibrated reading, Y is the offset correction factor, and G is the gain correction factor.

The correction factors Y and G are unique for each dc and ohms range, and are stored in a CMOS memory that is kept continuously powered by a lithium battery (seen in Fig. 3). Even when the rest of the instrument is not powered, the memory contents can be kept valid for over ten years by the lithium battery. The values of Y and G are derived by the 3468A's microprocessor and loaded into the memory by an electronic calibration procedure. This allows the user to calibrate the instrument without removing the covers or turning a single screw.

Initially, the user sets the instrument to the function and range to be calibrated. The offset must be calibrated first, so a short is placed across the input terminals and the user

**Lithium Battery**
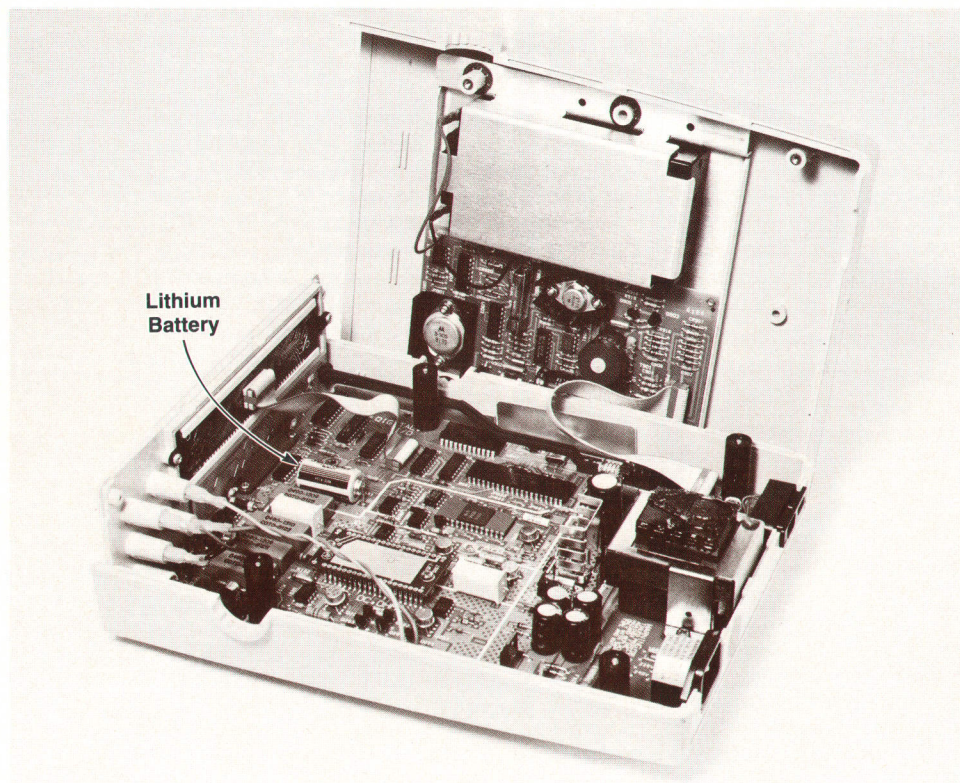
**Fig. 3.** *Interior of 3468A. The electronics system, with the exception of the liquid-crystal display, is mounted on a single printed circuit board. No Teflon™ standoffs or potentiometers are used.*

enables the **CAL** function. The 3468A takes a single measurement, recognizes that the reading is close to zero, assumes that the user wishes to calibrate the offset, and displays 0.00000? The user now presses the **SGL TRIG** key to verify that zero was indeed the user's target value. The 3468A averages ten 5½-digit readings to derive the new Y value, and stores this constant in the calibration memory. The display then reads CAL DONE, and returns to normal operation.

Using an external standard, the user can now apply the full-scale value of that particular function and range to the meter and repeat the previous procedure. Again, ten 5½-digit readings are taken and averaged. However, this time a divide operation is performed to determine the value of G, and the new value is stored into the calibration memory.

It is possible that a user may own a standard that is very stable and known in value, but not exactly equal to the 3468A's full-scale value. Furthermore, many standards are in cardinal values of 1 or 10 instead of 3, the 3468A's full scale. To accommodate both of these common situations, the 3468A quickly measures the applied signal and displays a target value of 100000? or 300000?, whichever is closest to the measured value. Using the front-panel's ⇑ and ⇓ keys, the user can then increment or decrement the display to the exact value of the standard used. Pressing the **SGL TRIG** key then commands the instrument to calibrate to the altered target value.

Calibration of the ac modes is done in a slightly different manner. On ac ranges, the nonlinearity of the true-rms converter allows it to operate accurately only from one-tenth to full scale. A shorted input would give an inaccurate Y value, as can be seen in Fig. 4. In many conventional voltmeters, the ac calibration procedure requires that the

applied signal be varied between one-tenth scale and full scale while a technician adjusts two interactive controls. To calibrate all the ac modes on the 3468A, a single 3Vac signal is applied. The target value can be adjusted to the exact value of the signal, using the ⇑ and ⇓ keys, and then pressing the **SGL TRIG** key will calibrate all ac functions.

Internally, the 3468A switches the ranges so that it has one full-scale reading and one tenth-scale reading, much like the technician calibrating a conventional voltmeter. However, instead of having to converge on the correct constants iteratively, the 3468A solves two simultaneous equations exactly for Y and G. This can be done because most of the uncalibrated inaccuracies are common to all ac voltage and current ranges. The uncalibrated reading taken on the 3Vac range will be close to full scale and is labeled FS. The uncalibrated reading taken on the 30Vac range will be close to one-tenth scale and is labeled TS. T is the target value. With the correct choice of Y and G, the following equations must be true:

$$T = (FS-Y)G$$
$$T/10 = (TS-Y)G$$

The 3468A solves for G first. Thus,

$$G = 0.9T/(FS-TS)$$

and Y is then solved for by:

$$Y = FS - T/G$$

When these constants are loaded into the calibration memory, the display reads CAL DONE and returns to normal operation. All of the calibration procedures can also be performed over the interface loop.

Much design effort was expended to insure the integrity of the calibration memory once the instrument is calibrated. To prevent unintentional writes into the memory, a switch located on the rear panel must be placed into the enable
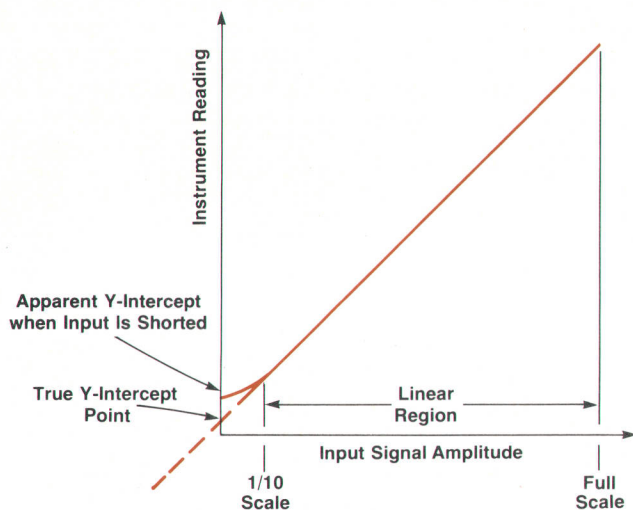
**Fig. 4.** *On ac ranges, the response of the true-rms converter is not linear below one-tenth full scale as shown above. The electronic calibration technique used in the 3468A takes two values corresponding to the one-tenth full-scale and full-scale readings and then solves two simultaneous equations (see text) to derive the true Y-intercept and gain (slope).*

position to allow the processor to write to the memory. This hardware failsafe prevents the user from unintentionally calibrating the instrument or altering the memory via the interface loop.

Although the 3468A is designed to withstand large electrostatic discharges (ESD), secondary safeguards are instituted for the calibration memory. Because the $256 \times 4$-bit memory selects a row of data at a time, it is possible that an entire row of data could be altered during ESD. For this reason, the data is addressed so that the calibration constants of any particular function and range are all located within the same row. If this row of data is altered, only that function and range will be uncalibrated, thus avoiding the need for recalibrating the entire instrument. Whenever the instrument is placed in that function and range, the CAL annunciator on the display is enabled, warning the user that the reading is no longer valid. Simultaneously, the appropriate bits are set in the 3468A's status bytes to notify the interface loop controller.

Besides ESD, there is also a small chance that an alpha particle or a cosmic ray may alter a single bit of the memory. To protect against this rare occurrence, the two constants for each function and range have parity bits associated with them so that the 3468A's microprocessor can locate single-bit errors and detect double-bit errors. The processor can correct a single-bit error in a calibration constant and the instrument will remain calibrated. If the processor detects more than a single-bit error, the CAL annunciator is displayed and the error bits are set to warn the user that the readings are uncalibrated. When this type of error is detected, the processor uses default values of zero for Y and one for G.

Electronic calibration not only simplifies calibration for the user but, by eliminating manual adjustments, the long-term stability and reliability of the product are improved. Furthermore, since the calibration routines can compensate

for gain errors as high as 4.5%, many components can be chosen for their good stability and initial accuracy becomes noncritical. Potentiometers are not the only parts to be eliminated; there are neither jumpers to be inserted or removed nor components with individually selected values in the 3468A.

## Microprocessor Design

It is principally by extensive use of the 3468A's microprocessor that such high performance is realized at low cost. Since the entire instrument is isolated from the HP-IL at the loop interface, there is no electrical reason why the system processor must be separate from the signal processor, as required in a guarded HP-IB (IEEE 488) instrument. In the 3468A, these tasks are combined and executed by a single 8-bit microprocessor.

Fig. 2 also shows how the processor interfaces with other sections of the multimeter. Besides executing instructions as they appear over the interface loop, monitoring the keyboard, and operating the display (typical system operations in a conventional system voltmeter), the processor implements a multislope analog-to-digital (A-to-D) algorithm,[3] and configures the front-end switches for proper signal conditioning (typical signal functions). The processor also performs all the electronic calibration procedures and monitors the 50/60-Hz switch so that it chooses the A-to-D algorithm having the proper power-line-frequency rejection.

A hardware failsafe counter is added to reset the processor if it hangs up (possibly done by erroneously executing code in a lookup table). If the keyboard is not scanned at least every 2.5 seconds, this counter resets the processor to its initial state.

Since many functions of the 3468A need dedicated control for finite periods of time, many tasks of the processor appear to be in conflict with each other. The most difficult conflict to resolve is that of executing the A-to-D algorithm (a dedicated operation) while still monitoring the keyboard and interface loop for new commands, and executing them. Since a high-precision A-to-D conversion can take up to 400 milliseconds, a key press can be missed entirely if it occurs during a conversion. Also, to have to wait 400 ms before a loop command can be processed is unacceptable.

This conflict is resolved by adding a constraint so that any key presses or loop activity abort the analog-to-digital conversion, and the command is executed before another conversion begins. To guarantee that both the loop and the keyboard are monitored during a conversion, the HP-IL IC chip is allowed to interrupt the processor when loop data is available, and the A-to-D algorithm firmware was modified so that the keyboard is scanned between integration periods. Thus, the maximum time between scans does not exceed 20 milliseconds, which is less than the minimum duration of a manual key press.

To increase the usefulness of the 3468A in an HP-IL system, two other features are included in its design. A front-panel **SRQ** key allows the user to initiate a service request to the loop controller from the front panel and have the ability to write alphanumeric characters to the 3468A's display remotely from the loop controller.

# The Philosophy Behind The Design

Early in the definition phase of the 3468A Digital Multimeter, very aggressive goals for low cost, reliability, and quality were set without sacrificing any of the measurement performance objectives. This was done to satisfy the cost pressures of the marketplace, and because high quality and reliability are performance measures that users are demanding. At this same time, HP's Loveland, Colorado facility was taking a close look at its design and manufacturing philosophies, trying to better understand how to design more quality and easier producibility into its new products and manufacturing processes.

The 3468A was chosen as a vehicle to change design and manufacturing strategies and demonstrate that quality is truly the key to achieving low manufacturing cost and high productivity simultaneously without sacrificing the product's measurement performance. The goal of this project was not only to develop a product that made a significant technical contribution, but to apply what was learned to improve future product design and manufacturing.

## Definition Phase

Crucial to meeting these goals and increasing the performance/price ratio is a proper instrument definition. In a conventional definition phase, technologies and design alternatives are investigated with the primary goal of meeting the measurement objectives. At this time, there is usually less emphasis placed on quality and manufacturability issues. However, in the case of the 3468A, after a technique was demonstrated to meet the measurement objectives, it was also carefully scrutinized for manufacturability. If it did not meet all of the project objectives, it was rejected. This ensured that the 3468A definition was not only well suited for the marketplace, but also an ideal vehicle for improving product development and production philosophies.

Out of this definition phase came several dominant objectives that are the essence of the 3468A's design. The first objective was to minimize the number of components used in the product. It was felt that minimizing the component count would help maximize reliability, since overall product reliability is the sum of the reliability of the individual components.

A goal of designing the 3468A with less than 100 parts was set in the early investigation phase. To design a 5½-digit DMM with 100 parts was an aggressive goal and required a totally new design approach. The final 3468A has about 160 components, still a remarkable achievement!

Another objective that originated in the definition phase was to increase the use of automatic production steps and hence reduce the manual labor content in the product. However, minimizing the amount of labor is not universally the right thing to do. For example, machine-insertable components are sometimes more expensive than their manually inserted equivalents. In such cases, a careful analysis was done to select the most cost-effective component, taking into account the cost of labor, loading errors, and product reliability.

With the goals of minimizing part count and labor in mind, a closer look at the internal definition will reveal how these goals were met while simultaneously providing performance and reliability benefits to the user.

While a conventional multimeter design incorporates at least two microprocessors, by using clever algorithm partitioning and an interrupt-driven structure, it was found that a single microprocessor design would meet the 3468A performance requirements. A reduction in parts count, complexity, power consumption, and cost, and an increase in reliability are direct results.

The 3468A makes extensive use of custom hybrid circuits. Both the analog-to-digital hybrid and the input hybrid incorporate high-voltage CMOS ICs and precision integrated resistor networks to do signal switching and conditioning, logic decoding and measurement timing. The ICs reduce circuit complexity (parts count) and increase the multimeter's performance by using the matching and temperature tracking properties inherent in adjacent devices on the ICs.

One of the biggest decisions made in the definition phase was to design the 3468A without any manual adjustments. Instead, calibration is done electronically. Not only does this feature greatly reduce the parts count and assembly costs by eliminating hand-loaded potentiomenters and associated circuitry, it enhances reliability and reduces drift caused by mechanical and thermal shock. Production-line labor is reduced by designing the instrument to be calibrated automatically under computer control via the HP-IL. And equally important, the user benefits by having front-panel calibration available.

A liquid crystal display (LCD) was chosen for the 3468A, primarily for its pleasant appearance, readability and alphanumeric capability. However, it also meets the product definition criteria by reducing the number of parts and assembly labor. An equivalent display using light-emitting diodes (LEDs) would need 31 more components than the LCD, all of which would require manual insertion into a printed circuit board. Furthermore, the LED display would consume more power (about 1.5W) than the LCD.

A one-piece, silicone-rubber keyboard was selected over a conventional keyboard array of individual, manually loaded keys. Not only does this silicone-rubber keyboard eliminate 23 parts, it provides an easy-to-actuate switch that also forms a gasket seal with the front panel, keeping dust and moisture out of the internal circuitry.

One final technology developed to meet the definition phase objectives is a custom printed circuit board process designed to maintain high impedances under all specified environmental conditions. This process eliminates the need for labor-intensive Teflon™ standoffs and allows direct automatic insertion of components in the printed circuit board.

## Design Phase

In addition to striving to meet the objectives set in the definition phase, the laboratory project engineers added another goal in the design phase of creating a "perfect" design. What was meant by "perfect" was a design that not only met all its instrument specifications, but also was designed with components that were completely characterized, specified, and guaranteed to operate properly under the worst-case tolerance conditions.

To achieve this "perfect" design goal, the team placed significant emphasis on extensive testing of the early prototypes. Complete environmental, safety, abuse, margin, and accelerated life tests were run with the intention of wringing out all design problems as early as possible. When the prototypes were not being evaluated by engineers, they were placed in an oven at 65°C and operated in all possible modes in an attempt to induce failure. Periodically, the instruments were removed from the ovens and vibrated on a shake table to try to uncover sensitivities to mechanical shock and vibration. All failures and anomalies were carefully documented, and with the help of the quality assurance department and its component analysis group, an attempt was made to autopsy all failed components and throughly understand every problem uncovered.

A large amount of data was collected on instruments that were

run through the environmental tests. These units were subjected to temperatures ranging from −40°C to +75°C and relative humidities up to 95% at 40°C. The data collected at these various conditions was statistically analyzed and compared against the theoretical specifications to ensure that all interactions of the design were thoroughly understood.

An innovative feature of the 3468A design phase, which proved crucial to meeting the quality goals, was the full-time assignment of a materials engineer to the project. This individual's job was to understand the design requirements of every purchased component thoroughly and use that knowledge to select the best vendor. The materials engineer discussed the expectation of "perfect" components with each vendor and then worked with the vendor to establish methods for achieving this goal. Overall, the response on the part of the vendors has been quite favorable, and work continues to further improve vendor relationships.

A final departure from the conventional design sequence was the early involvement on the part of manufacturing engineering. The team of manufacturing engineers established their own department objectives to meet the overall product goals. These objectives were to ensure that 100% good parts were delivered on time to the production line, and that the product went through an error-free manufacturing process. Given that a production cycle begins with 100% good parts, and no defects are introduced through the manufacturing process, it can be expected that all of the instruments will turn on and function properly without any intermediate testing. This is the goal of the 3468A manufacturing team, which represents a significant departure from conventional manufacturing philosophy.

To achieve their goals, the manufacturing engineers played an active role in working with the design engineers to maximize the producibility of the design. The design engineers consulted with the manufacturing engineers at each prototype phase to get suggestions on how to improve the producibility of the 3468A. Under the scrutiny of the manufacturing engineers, production-line workers assembled the prototypes and commented on assembly-related problems. These comments were compiled and a manufacturability review was held. The appropriate changes were incorporated in the next prototypes.

The goal of a "perfect" production process requires the complete characterization of major process steps. Machinery such as the automatic insertion equipment and the wave-soldering machine were statistically characterized and are monitored through the use of control charts. Conductive work surfaces and ground straps were added to work stations. Air ionizers, grounded soldering irons, and antistatic packaging are installed throughout the manufacturing area. New handling procedures help ensure that assemblies are not inadvertently contaminated during fabrication. Adequate documentation and training are a necessary part of these changes.

The entire production process is designed with manufacturability in mind. All intermediate tests and repair loops are eliminated. Instead, a single, fully automatic test and calibration system turns on, calibrates, and characterizes each instrument only after it is completely assembled. The HP-IL interface and electronic calibration of the 3468A made the design of this automatic system possible. Another feature is the automatic collection of data on each instrument. This data is analyzed on a run-by-run basis to look for trends or shifts in specifications. Thus the final test system also acts as a process monitor to detect problems as they arise in the production process.

## Production Phase

With the transfer of the 3468A to production, the results of these efforts can finally be seen. With defect rate, turn-on rate, and failure rate goals firmly established, the actual results were compared against predictions. The first production run yield was 70%, a little short of the initial goal of 75%. However, yields have been steadily increasing and presently exceed 80%. The long-term goal of achieving a 96% turn-on yield appears to be achievable. Each defect that is found is fully documented and presented to a task force whose job it is to identify the source of the defect and correct the problem.

The 3468A experiment seems to be a success. It is being demonstrated that by changing design philosophies, the quality and manufacturability of products can be significantly improved.

## Analog Design

The 3468A's precision analog sections are dominated by two custom hybrid circuits: a front-end signal conditioning hybrid and an A-to-D converter hybrid. Both consist of custom CMOS digital logic, low-leakage MOS switches, and precision thin-film resistor networks.

The microprocessor serially loads the front-end hybrid by using three port lines: clock, data and load. This data is transferred to a latch, which drives the MOS switches. The state of these switches, which include relay drivers, completely defines the signal conditioning and varies with the particular function and range of the voltmeter. The precision front-end hybrid requires one silicon chip and two resistor networks and is mounted on a custom high-impedance printed circuit board. Outside the hybrid circuit are the operational amplifiers that are the active components for the dc, ac, and ohms current sections. Four low-thermal-emf relays are switched by the hybrid when high voltages are a possibility. An integrated rms converter supplies a dc signal representing the ac signal value to the A-to-D converter.

The A-to-D hybrid consists of one CMOS chip and one resistor network mounted onto a standard 28-pin ceramic dual in-line package. The 3468A uses a modified Multi-Slope-II algorithm[3] for its A-to-D conversion. Five microprocessor lines into the hybrid determine the current to be summed by the integrator. A comparator on the chip measures the polarity of the integrator and sets a microprocessor input to the proper sign value. The integrator operational amplifier and two reference operational amplifiers are located off the chip.

## Acknowledgments

sign of the 3468B. Bill Miner was responsible for the 3468A mechanical design, and Doug Olsen did the 3468B mechanical design. The I/O software and hardware design was provided by Gary Stadele, whose technical maturity and guidance got us through the difficult times. Industrial design for the 3468A was done by Jon Pennington.

The project manager and pundit for the 3478A, the HP-IB version of the 3468A, was Ron Tuttle. Rich Wilson and Virgil Leenerts did the 3478A electrical design, while Ed Pennington did the mechanical design. Industrial design for the 3478A was done by Ted Crawford.

Norm Dillman designed the environmental test system and wrote the test software for both products. Si Sanders was the materials engineer for both products.

Joe Marriott was the section manager for the completion of the project.

A special word of thanks goes to our original section

manager, Roy Barker. His managment style and support provided us the work environment necessary to try something different, and then prove we could make it work.

Finally, all of the functional areas within the Loveland facility deserve recognition. The 3468A and 3478A were division-wide projects that required the unwavering efforts of many individuals outside of the lab to be successful. To all of you, we give our thanks.

### References
1. R.D. Quick and S.L. Harper, "HP-IL: A Low-Cost Digital Interface for Portable Applications," Hewlett-Packard Journal, Vol. 34, no. 1, January 1983.
2. A. Furst, "Defect-Free Design Applied to HP Voltage Meters," Electronic Business, February 1982, pp. A6-A7.
3. L.T. Jones, J.J. Ressmeyer, and C.A. Clark, "Precision DVM Has Wide Dynamic Range and High Systems Speed," Hewlett-Packard Journal, Vol. 32, no. 4, April 1981.

**Lawrence A. DesJardin**

Larry DesJardin attended the California Institute of Technology, earning a BS degree in engineering in 1977. He started working for HP that year while continuing his studies at Stanford University. After receiving the MSEE degree in 1978, he transferred to HP's Loveland, Colorado, division where he currently is a project manager. He was born in Redwood City, California and now lives with his wife and one ferocious cat in Loveland, Colorado. He is the mayor pro-tem of Loveland and a member of the city council and planning commission. When not busy with work and civic activities, he enjoys skiing, listening to rock music, running, and high-performance driving on mountain roads.

**Jack P. Trautman**

A native of Cicero, Illinois, Jack Trautman attended the University of Illinois, earning a BSEE degree in 1972, and Stanford University, earning an MSEE degree in 1973. He has been with HP since 1974 and has worked on analog MOS IC design and the 3466A Voltmeter. He was project manager for the 3468A and now is head of the Loveland Technology Center. Jack is an active supporter of Junior Achievement and is presently on its Loveland board of directors. He lives in Fort Collins, Colorado, and has a variety of outside interests: sailing, scuba diving, skiing, racquetball, world travel, and high-performance driving in his sports car.

# Low-Cost and Portability Come to Data Acquisition/Control Products

*Inexpensive, portable data logging with the flexibility of a data acquisition/control system is now within the budget of nearly everyone making transducer measurements.*

**by James J. Ressmeyer**

THE 3421A Data Acquisition/Control Unit (Fig. 1), HP's newest low-cost data acquisition and control scanner, represents the first instrument of this type using the Hewlett-Packard Interface Loop (HP-IL).[1] Battery power and the unit's light weight allow a complete

measurement system to be configured by adding an HP-41 Handheld Computer, an 82161A Digital Cassette Drive and an 82162A Thermal Printer/Plotter. This system (shown on the cover) can be easily transported and operated in the laboratory or field. Its low cost allows individual engineers
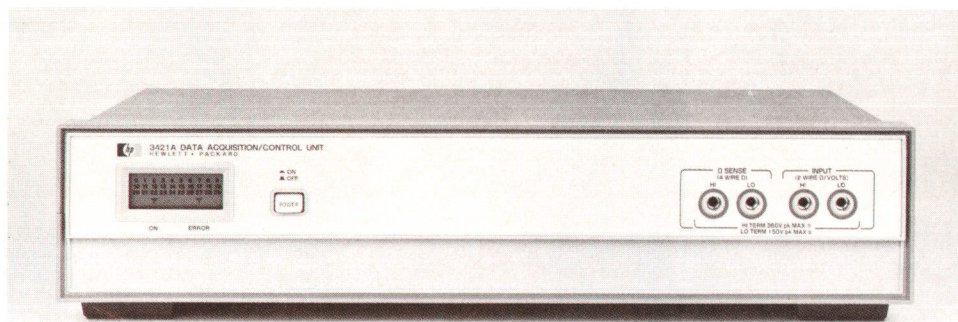
**Fig. 1.** The HP Model 3421A Data Acquisition/Control Unit is a low-cost, battery-powered, up-to-thirty-channel instrument with an internal multimeter and counter designed for use on the HP-IL.

to dedicate a system to specific measurements during the design or test phase of product development, and opens new possibilities for data acquisition applications.

The 3421A can be configured with up to 30 channels for measuring dc volts, ac volts, resistance or frequency through the use of three 10-channel multiplexing assemblies. These assemblies are directly connected to a 5½-digit multimeter contained inside the 3421A. The basic accuracy of its analog-to-digital (A-to-D) converter is 0.009%+6 counts for a 5½-digit dc voltage measurement. Selectable resolution of 5½, 4½, or 3½ digits can be traded off against measurement speed. Other assemblies are available for sensing and setting digital information, switching high-level (250Vrms) devices, and for designing custom circuits to interface directly with the instrument. Up to three assemblies can be used in each 3421A mainframe.

An important feature is the instrument's ability to operate on battery power. A lead-acid battery inside the instrument is kept fully charged during normal operation on ac line power. Should ac line power be interrupted to the unit, the battery will supply power for continued operation. The 3421A can also be used in the field away from ac line power, its battery eliminating the need and expense of a portable generator. The instrument responds to the HP-IL standby command which puts the instrument into a sleep mode to conserve battery power. The 3421A can operate up to two months in the sleep mode, waking up occasionally to take measurements, or up to twelve hours continuously on its internal battery.

The design objectives for the 3421A were focused on providing transducer-based measurement capability with low cost, low power and battery backup, portability, and ease of use.

HP offers data acquisition systems based on desktop computers capable of handling a very large number of measurement points. However, until the 3421A, no similar system existed that could take full advantage of the new serial HP-IL and the low-cost controllers made available by it.

**Key Features**

The function and feature sets of the 3421A allow most of the transducer-based measurements needed in data acquisition and logging. Many transducers, such as thermocouples, have dc voltage outputs. The 3421A has the same 300,000-count autoranging A-to-D converter used in the 3468A and 3478A voltmeters (see article on page 3). The

lowest range has the 1-$\mu$V sensitivity required for thermocouple measurements. The highest range allows a maximum input of 300 volts, giving the 3421A a wide dynamic input range. A temperature sensor on each 10-channel multiplexer card provides a reference junction for thermocouple-based temperature measurements. The type-T thermocouple conversion in the 3421A provides temperature readings directly in degrees Celsius.

The 3421A also performs resistance measurements, useful in the measurement of other types of transducers, such as RTDs (resistance-temperature detectors) and thermistors. The lowest range has a sensitivity of 1 m$\Omega$ and the highest range has a maximum input of 30 megohms. The multiplexer cards can be configured for either two- or four-wire resistance measurements. An internal zero-crossing counter allows frequency measurements (up to 10 kHz), useful for transducers with pulse outputs, such as flowmeters. Another function in the 3421A is an average-responding ac converter for measuring line-related and ac signals of frequencies up to 1.5 kHz.

All of the functions and ranges of the 3421A are electronically calibrated by the same technique used in the 3468A (see page 5). A nonvolatile CMOS RAM powered by a lithium battery is used to store all of the constants obtained during calibration of the instrument. There are separate constants for offset correction of each of the reference-junction temperature sensors. The calibration RAM is protected from accidental alteration by requiring a hardware-enable switch on the back panel (see Fig. 2) to be set for calibration. The calibration constants can also be read or written to by an HP-IL controller through the instrument's I/O interface.

The 3421A can be programmed to scan a sequence of up to 30 channels. The instrument has enough internal memory to store one scan sequence program and one set of readings. One trigger command will take readings per the specified scan sequence and store them. By using the scan commands, a set of readings can be taken, stored, and later read back into the controller. In this manner, I/O time is not required between each reading.

The scanner/multiplexer capability of the 3421A resides on up to three optional printed circuit boards accessible through removable covers on the rear panel. Each board comes with removable printed circuit board edge connectors for convenient termination of the user's field wiring. Terminations may be made permanent by purchasing extra
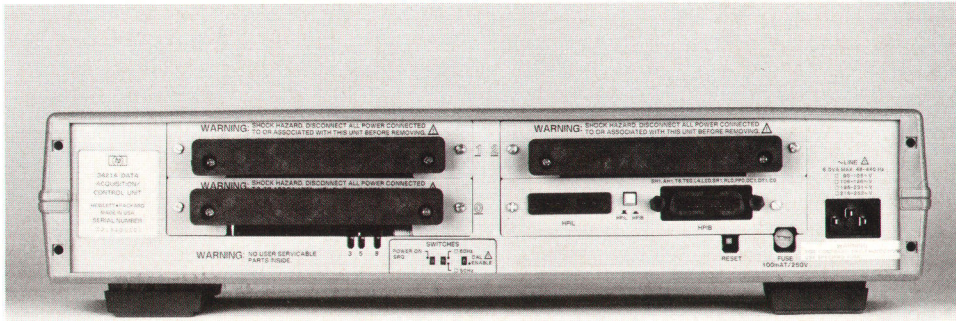
**Fig. 2.** *Back panel of the 3421A. The three slots can be used to insert 10-channel latching-relay cards, digital I/O cards, or breadboard cards customized by the user for specific applications. This 3421A also has the optional HP-IB interface.*

terminal blocks.

Three different plug-in assembly options exist, any of which can be placed in any or all of the three slots. These include the 10-channel relay multiplexer/actuator, a digital I/O interface, and a breadboard option.

### Multiplexer/Actuator Assemblies

Each 10-channel multiplexer/actuator board contains ten latching, low-thermal-dissipation, armature relays giving the 3421A a maximum capacity of thirty relays when three boards are installed. Latching relays were chosen because of the small drain on the battery and their inherently low thermal dissipation. The ten relays can be configured by jumpers in a number of ways to suit the needs of the user. Two of the relays can be used as either two-ampere actuators or multiplexer relays. The other eight are always multiplexers, but series and/or shunt elements can be added to them. One channel comes with a removable 10:1 divider intended for use in monitoring power-line voltage, using the 3421A's internal 30Vac range. The board also can be jumper-selected to multiplex the sense terminals for four-wire resistance measurements. The **HI** and **LO** front-panel terminals are connected to the multiplexer's common bus. This bus can be disconnected from the 3421A's internal multimeter to use external instruments or to allow single-ended scanning of up to 56 points with a common ground.

### Digital I/O Assembly

The digital I/O board has eight floating inputs and eight floating outputs. The inputs are protected by optoisolators. Because each input can detect the presence or absence of current flowing, either 5-volt TTL or 24-volt CMOS signals will trigger the input equally well. Each bit can be read individually, or the eight bits can be read as one word. The 3421A can be set up to trigger a multiplexer board scan sequence when a specific digital word occurs at the inputs of a digital I/O board. The outputs of the digital I/O board are protected against shorting and use isolated DMOS FET switches. An on switch is capable of sinking up to 300 mA at less than two ohms. An off switch has an impedance greater than 10 MΩ. These switches are used as open-drain outputs. Each switch can be written to individually or all eight can be written to as one word.

### Breadboard Option

The breadboard option has circuitry for decoding the

instrument's backplane and an area for the user to add custom circuitry. For example, this card could be configured as a multiple-channel, continuously sampling totalizer card to monitor a few channels of pulsed outputs.

### HP-IB Option

To fit into more traditional instrumentation systems, the 3421A has an optional HP-IB (IEEE 488) interface. The HP-IB option does an internal conversion to the HP-IL. The internal HP-IL interface is used for isolation between the HP-IB and the user's measurement potential. A pushbutton on the rear panel (see Fig. 2) is used to select between HP-IL and HP-IB operation.

### Low-Cost Design

Keeping the cost low while keeping the capabilities high was a major goal for the design team. One of the largest cost savings is realized through the mechanical packaging. Since the instrument is intended to be low-power and lightweight, plastic packaging fits the definition perfectly. The package is a two-piece full-rack-width clamshell with separate plastic front and rear panels.

The 3421A is designed to be used with a controller, so it was not necessary to include costly front-panel controls for manual operation. Front-panel-emulation programs for user convenience in setting up data acquisition applications are available for both the HP-85 Personal Computer and the HP-41 Handheld Computers. The programming for the HP-41 is achieved by adding a special ROM and keyboard overlays (see article on page 16). For use with the HP-85, the 3056DL system was developed (see box on page 13).

The block diagram of the 3421A (Fig. 3) illustrates other cost savings. Since most pulse output transducers do not use frequencies higher than 10 kHz, and ac voltage measurement at other than line frequencies is normally not done in data acquisition, the dc front end can also be used for both ac voltage and frequency counter measurements. All the protection and signal amplitude conditioning already exist in the dc front end. The additional circuitry required for ac measurements and the counter is minimal. The 3421A's microprocessor accumulates the pulses for the counter.

Another major area of cost savings is in the power supplies. Portability, and therefore battery power, was one of the major design goals. By optimizing the power

# Data Acquisition and Control Software for the 3421A Using the HP-85 Computer

by David F. Leonard

To simplify the use of the 3421A Data Acquisition/Control Unit for data logging applications, the 3056DL System is available. One or two 3421As and an HP-85 Personal Computer to control them are housed in one cabinet (Fig. 1) and supplied with an appropriate package of software routines.

Most dedicated data loggers are self-contained units with the analog front end and the digital user interface portion housed in one box. Because of the low cost and flexibility of both the 3421A and the HP-85, the total 3056DL system is comparable in price to that of most single-box data loggers and is more versatile. The 3056DL software has been written to make the system easy to use for the person who is not familiar with computers or instrumentation. Thus, a user can take advantage of the flexibility of the computer without paying the price of the long startup times for which computer-based data acquisition systems have been known.

The 3056DL software package includes a data logger routine, a program that emulates an instrument front panel, instrument diagnostic programs, and graphics programs to plot data from the data logging results. Subprograms that linearize temperature transducers are also available on the tape for the user who needs to write a more specialized program.

## Data Logging Setup

The data logger setup is divided into two major parts: channel setup and timing setup. A setup is done for up to 20 groups. A group is defined as a group of channels with the same label, function, data conversion equation, and limits.

A label can be up to 26 characters long. The function can be

```
SELECT LIMIT ACTION

OpnChn =OPEN ACTUATOR/CLEAR
              DIGITAL CHANNEL
ClsChn =CLOSE ACTUATOR/SET
              DIGITAL CHANNEL
DeactGr=DEACTIVATE GROUP
Act Gr =ACTIVATE GROUP
DspWarn=DISPLAY WARNING
PrWarn =PRINT WARNING


------------------------------------
OpnChn   DeactGr   DspWarn
ClsChn   Act Gr    PrWarn      Beep
```

**Fig. 2.** *Limit display for the 3056DL system.*

one of the following:

- dc volts
- ac volts
- Ohms (two or four wire)
- Frequency
- Thermistor temperature measurement
- RTD temperature measurement
- Thermocouple type J,K,T,E,R, or S temperature measurement
- Digital input
- 4-to-20-mA input
- Close or open an actuator or digital channel.

Data conversion software will convert inputs according to linear or fifth-order polynomial equations, or for greater flexibility, the data logger will call a subprogram (the name of which is entered during setup) at the time data conversion is done. The user writes the subprogram and stores it on the software tape. Units for the converted data can be specified. These units can be up to five characters long.

Limits are used to initiate an action when a reading or converted reading reaches a value specified by the user. Limits can be one of three types: high, low, or delta. The action will be initiated when

```
GROUP NUMBER 1
LABEL:FURNACE TEMPERATURE
FIRST CHANNEL: 2
LAST  CHANNEL: 19
FUNCTION: THERMOCOUPLE-R
CONVERSION TYPE: NONE
LIMIT# 1 LOW 1000
LIMIT# 2 HIGH 1390
LIMIT# 3 NONE
LIMIT# 4 NONE
LIMIT# 5 NONE


------------------------------------
clear                        /|\       |
EDIT       DONE               |      \|/
```

**Fig. 3.** *Channel setup display for the 3056DL system.*



**Fig. 1.** *A powerful, compact, yet low-cost data acquisition system, the HP 3056DL, is formed by combining one or more 3421As with an HP-85 Computer.*

```
CHANNEL:5                          CHANNEL|20|21|22|23|24|25|26|27|
                                   INPUT  | 1| 1| 1| 0| 1| 1| 1| 0|
FUNCTION:TC-R Deg C                DECIMAL VALUE OF INPUT IS 119

READING:38.89       ▨             K1-VIEW NEXT DIGITAL CARD

                                   K2-OUTPUT TO DIGITAL CARD

                                   K4-RETURN TO FRONT PANEL MENU


---------------------------------  ---------------------------------
RESET    CHANNEL DIGITAL  done
FUNCTION FORWARD REVERSE  CMD      NEXT    OUTPUT         RETURN
```

**Fig. 4.** *Basic front-panel display on the HP-85 Computer for the 3056DL system.*

**Fig. 5.** *Digital card display for the 3056DL system.*

a high limit is exceeded, the reading is less than a low limit, or a reading exceeds the last reading for that channel by a specified amount (delta limit). The limit actions are indicated on the HP-85's display as shown in Fig. 2. Up to five limits can be specified per group.

Fig. 3 shows the channel setup display. Specifications for the channel setup are entered by moving the line (shown under limit #1 in Fig. 3) to the line that is to be changed. The line is moved up and down by pressing the softkeys labeled by arrows. When the EDIT key is pressed, the display instructs the user on what entries can be made, usually by relabeling the softkeys. After the entry is made, the CRT returns to the original display, which then shows the entry just made.

Once the channel setup is complete, the timing setup for each group is done. The timing setup specifies whether or not the group of channels is to be scanned at specified intervals or at a time of day, the number of times the group is to be scanned, and finally how the resulting data is to be handled. Data can be stored on tape, displayed on the HP-85's CRT, or printed on its printer.

### Front-Panel Emulator

The basic front-panel display is shown in Fig. 4. The first line shows the acquisition channel that is closed. The function is shown on the second line. The third line shows the last reading read from the 3421A. The shaded rectangle, shown to the right of the reading, blinks when a new reading is displayed.

■ When the FORWARD or REVERSE softkey is pressed, the 3421A will step forward or backward through the acquisition channels.

The reading shown will then reflect the input on the new channel chosen.

■ The CHANNEL softkey allows the user to close or open an acquisition, actuator, or digital channel.

■ When the FUNCTION softkey is pressed, the user can change the function to one of the fifteen available.

■ The CMD softkey is used for outputting an ASCII command to the 3421A.

■ When the DIGITAL softkey is pressed and a digital card is present in the 3421A, the display shown in Fig. 5 appears. Here the display shows the eight-bit input of a digital card in slot 2. When the OUTPUT softkey is pressed, the display prompts the user for a decimal number to be output to the digital card, and the 3421A's liquid-crystal display shows the value of the digital outputs.

**David F. Leonard**
Dave Leonard received his BSEE degree from the University of Colorado in 1978 and began working at HP that year. He worked on the 3497A Data Acquisition Unit before joining the 3421A and 3056DL design teams. In his spare time, he is working on an MSEE degree at Colorado State University. He is married, has two children, and lives in Westminster, Colorado.
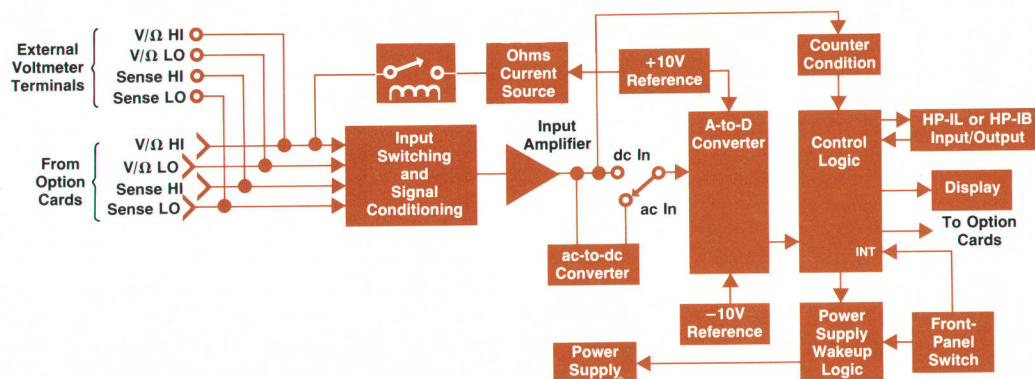
**Fig. 3.** *Block diagram of the 3421A.*

## Table I
## 3421A Instruction Set

### Standard Instruction Set:

| | |
|---|---|
| **DCV** | dc volts [channel list] |
| **ACV** | ac volts [channel list] |
| **TWO** | Two-wire ohms [channel list] |
| **FWO** | Four-wire ohms [channel list] |
| **TEM** | Temperature [channel list] |
| **FRQ** | Frequency [channel list] |
| **TOT** | Totalize [channel number] |
| **REF** | Reference junction temperature [channel number] |
| **CLS** | Close a single channel or digital bit ⟨channel number⟩ |
| **CLP** | Close a pair of scanner channels ⟨channel number⟩ |
| **OPN** | Open a channel or digital bit [channel number] |
| **WRT** | Write a digital byte ⟨slot number, decimal value 0-255⟩ |
| **RED** | Read a digital byte ⟨slot number⟩ |
| **BIT** | Read a digital bit ⟨bit list⟩ |

### Advanced Instruction Set:

| | |
|---|---|
| **RS** | Reset |
| **F** | Function ⟨function number⟩ |
| **R** | Range ⟨range exponent⟩ |
| **RA** | Range auto ⟨1 if on, 0 if off⟩ |
| **G** | Gate time ⟨second exponent⟩ |
| **N** | Number of digits of resolution ⟨3,4, or 5⟩ |
| **T** | Trigger mode ⟨number⟩ |
| **Z** | Autozero ⟨0 if off, 1 if on⟩ |
| **C** | Calibrate [cal value] |
| **SR** | Status registers |
| **DN** | Display number [number 0-29] |
| **DI** | Display digital inputs |
| **DO** | Display digital outputs |
| **M** | SRQ mask [number] |
| **DT** | Digital trigger ⟨digital channel⟩ |
| **RL** | Read channel list |
| **LS** | Load single scanner channels into list ⟨channel list⟩ |
| **LP** | Load pairs of scanner channels into list ⟨channel list⟩ |
| **SI** | Scanner increment ⟨0 or 1⟩ |
| **UC** | Unconditional close ⟨channel number⟩ |
| **MH** | Monitor high ⟨digital bit⟩ |
| **ML** | Monitor low ⟨digital bit⟩ |
| **AN** | AND mask [decimal value 0-255] |
| **XR** | XOR mask [decimal value 0-255] |
| **MN** | Monitor ⟨slot number⟩, used with **AN** and **XR** |
| **DS** | Digital set ⟨slot number, decimal value 0-255⟩ |
| **DC** | Digital clear ⟨slot number, decimal value 0-255⟩ |
| **CAL** | Calibration RAM ⟨1 if read, 2 if write⟩ |

[ ] if optional
⟨ ⟩ if not optional

---

supplies, knowing that a battery will be always present, a significant savings is realized over traditional battery options in other instruments that use more expensive center-tapped transformers and linear regulators for direct ac operation in addition to battery-powered supplies like those used in the 3421A.

## Instrument Friendliness

Our ability to use instruments in systems has taken gigantic steps since the first programmable instrumentation was introduced in the 1960s. We feel that because of its low cost and high capability, the 3421A will find its way into the hands of some less electronically sophisticated users. Therefore, we made it easy to program. The first major hurdle in programming most instruments is remembering to write to the instrument everything it needs to know to complete the task. The 3421A's standard instruction set (see Table I) includes most of the normal measurement tasks made by an instrument of this type. By sending it just one simple command, everything is handled by the instrument to complete the entire task. For example, to measure dc voltages on channels 2, 3, 4, and 8, the user merely sends DCV2-4,8 or DCV2,3,4,8,. If this instruction is compared to those for earlier instruments capable of making the same set of measurements, it can be seen that the 3421A is much easier to use. The 3421A also contains more traditional instructions which we have chosen to call the advanced instruction set. To take the same set of readings using the advanced instructions, the following commands are sent to the 3421A:

LS2,3,4,8     (Loads scanner list)
F1RA1Z1N5T3   (Configures 3421A; F1=dc volts, RA1=auto-
              range, Z1=autozero, N5=5 digits, T3=trig-
              ger through scanner list)

Besides the software programs available for the HP-41 and HP-85 controllers, other features include an internal type-T thermocouple conversion. Thermocouple measurement with the 3421A is easily accomplished by connecting the thermocouple to the appropriate channel and sending TEM(channel number). The measurement reads the temperature of the reference junction and the voltage across the thermocouple. A software compensation is performed on these readings to give the corrected temperature. A test of friendliness devised early in the definition phase of the project centered around how easily a temperature measurement could be made using a thermocouple. For most instruments, it is a complicated task. However, with the 3421A, all the work is done for the user.

## Low-Power Operation

To do a good job at remote data logging and to make full use of the battery, low-power design techniques are used throughout the instrument. One of the major power savings is realized through the use of latching relays in both the A-to-D front end and in the multiplexer. Each relay is magnetically latched either on or off; all that is required to set or reset a relay is a 10-ms current pulse through the appropriate winding. An additional advantage of these relays is the low thermal voltage offsets because of the lack of power dissipation.

Another major contributor to keeping the power consumption low is the power supply design. The ac line input charges the battery, and the ±15-volt supplies are derived using an efficient switching technique. The 5-volt supply is linearly regulated directly from the battery. The liquid-crystal display and extensive use of CMOS logic also con-

tribute to reducing power consumption.

Implementation of the HP-IL power-down mode provides the user with greatly increased battery life. In this sleep mode, only a small fraction of the circuit is kept powered up. At times specified by the user, the HP-IL system can wake up and perform its function. For example, in continuous operation, the battery life of the 3421A is specified at twelve hours minimum. Use of the HP-IL power-down function extends the battery lifetime, allowing the user to take a set of thirty readings once an hour for a period of more than one month. Since the HP-41, the 82162A Printer, and the 82161A Cassette Drive also implement the power-down function, a powerful battery-powered system, which can be controlled by the 82182A Time Module (which plugs into the HP-41), can be assembled for unattended data logging.

### Acknowledgments

The author wishes to acknowledge the following persons for their contributions to the design of the product. Chuck Clark, multiplexer, digital I/O and breadboard options, Rob Leiby, strife testing, Dave Leonard, power supplies and 3056DL software, Rob Saffer, packaging and mechanical design, Vicky Sweetser, logic and firmware, Kevin Thompson, HP-41 ROM, Dave Wolpert, HP-IB and HP-41 ROM, and Paul Worrell, A-to-D converter and signal conditioner.

**James J. Ressmeyer**
Jim Ressmeyer was born in Pipestone, Minnesota and attended the University of Minnesota, receiving the BSEE degree in 1972. Jim came to HP in that same year and has worked on thin-film hybrids and analog design for voltmeters, including the 3456A. He was responsible for the ohms, input filter, and front-end switching sections in the 3456A and now serves as a project manager on the 3421A and 3056DL. Between work and outside activities, he attended Colorado State University, earning an MSEE degree in 1975. He and his wife and two daughters live in Loveland, Colorado. Jim enjoys bowhunting, playing racquetball, and backpacking and camping with his family.

# Low-Cost Instrument Control: A New ROM for the HP-41 Handheld Computers

**by David L. Wolpert**

HP'S NEW SERIAL INTERFACE LOOP (HP-IL) and 82160A HP-IL Module make it possible for an HP-41 user to control instruments such as the 3468A Multimeter and 3421A Data Acquisition/Control Unit, thus forming low-cost measurement systems for bench or field applications. To reduce the amount of programming and simplify use, a custom ROM and keyboard overlays (Fig. 1) are available for HP-41 users. The HP 44468A Data Acquisition and Control Package (DA/C Pac) is based on the instruction set for the 3421A. Each of these instructions performs a complete measurement task or a complete operation.

To illustrate the capability and friendliness provided by the 44468A DA/C Pac ROM, consider the measurement of ac voltage by a 3468A Multimeter on the loop. Even without DA/C Pac ROM, this can be done by a fairly simple program for the HP-41:

```
01*LBL "MEAS2"
02 "HP3468A"
03 FINDID
04 SELECT
05 "F2N5RAZ1T2"
06 OUTA
07 IND
08 END
```

The above program finds the voltmeter's address on the loop, selects it, sets function, range and trigger, and takes a reading. But if there is no 3468A on the loop, there will be an address error at step 04 (ADR ERR), and if a syntax error is noted by the 3468A, step 07 may produce a TRANSMIT ERR. To trap most of these likely errors, a more complex program is necessary. The following example handles most foreseen anomalies, and responds with friendly prompts on the HP-41's display such as NO INST (no such instrument found) and INST ERR (instrument error).

```
01*LBL "ACV"
02 "HP3421A"
03 FINDID
04 X=0?
05 GTO "F68"
06 SELECT
07 "ACV"
08 OUTA
09 INSTAT
```

First look for a 3421A. The ROM actually looks for a measurement device, using accessory poll.

Tells the interface which device to talk/listen to.

This is the standard 3421A instruction for measuring ac voltage.

Read instrument status byte.

```
10 FS? 05                    Error occurred if flag 5 is set in status
11 GTO "ER2"                 byte.
12 IND
13 RET
14*LBL "F68"                 Get the reading.
15 "HP3468A"
16 FINDID                    No standard instruction set instru-
17 X=0?                      ment was found, look for a 3468A.
18 GTO "ER1"
19 SELECT                    This is the 3468A instruction string
20 "F2N5RAZ1T2"              for a single ac voltage reading.
21 OUTA
22 IND                       Get the reading.
23 RET
24*LBL "ER1"
25 "NO INST"                 3421A or 3468A not found—cannot
26 AVIEW                     measure ac voltage.
27 STOP
28 *LBL "ER2"
29 "INST ERROR"              Syntax or calibration or other error.
30 AVIEW
31 STOP
32 END
```

By contrast, the 44468A DA/C PAC ROM reduces this program to three simple lines:

```
01*LBL "MEASURE"
02 ACV
03 END
```

This trivial program returns an ac voltage to the HP-41's X register, and handles all the commands that are necessary for using the 3421A or 3468A.

## DA/C Pac ROM

The custom ROM is divided into seven basic parts:

- Microcode for friendly operation of instruments. This makes it easy to use the automatic addressing features of the HP-41 and HP-IL. Included are some basic instrument operations for digital multimeters, scanners, and combination devices such as the 3421A. These include MEASURE dc VOLTAGE, CLOSE CHANNEL, et cetera.
- Microcode for polynomial (and other) expansions of temperature functions. This makes the HP-41 significantly faster in performing temperature conversions, a typical user task for the 3421A and other instrument systems. Conversion functions for thermocouples (T,J,R,E,K, and S types), thermistors (2252 or 5000 ohms at 25°C), and platinum resistance-temperature detectors (RTD, 100 ohms at 0°C) are included, with software cold-junction compensation for the thermocouples.
- User code for complete temperature measurements, including cold-junction measurement and software compensation.
- A data logger application program. This simple data acquisition program can print or store on an 82161A Digital Cassette Drive the data from up to 30 channels of inputs of various kinds (voltage, resistance, temperature, frequency) taken at time intervals specified by the user during setup.
- Front-panel program to allow the HP-41 to manipulate the 3421A, acting as its front panel.
- Miscellaneous routines which include an alarm setup program (prompting), a routine to find an instrument, DVM or scanner on the HP-IL, a routine to output a command to an instrument, and a decode routine to return measurement functions and units to the data logger.



**Fig. 1.** The 44468A Data Acquisition/Control Package ROM comes with overlays for the HP-41 keyboard to simplify HP-IL measurement system programming.

- Miscellaneous microcode for easy programming, such as a key assignment routine and programmable timer alarm clearing.

Because there are certain things that cannot be done by the standard HP-41 instruction set, the DA/C Pac ROM extends the instruction set by microcode to use the Accessory ID provisions of the HP-IL. Also, the thermocouple/thermistor/RTD conversions are written in microcode. Although these could have been done with standard HP-41 instructions, the microcode executes much faster and does not require the use of the stack or other registers. Below is a listing of some microcode routines in the measurement portion of the DA/C Pac ROM:

| | |
|---|---|
| DCV | Measure dc voltage |
| ACV | Measure ac voltage |
| TWO | Measure resistance (two-wire) |
| FWO | Measure resistance (four-wire) |

These four commands differ only in the string that is sent to the measurement device. The command flow is as follows, where (cr) indicates a carriage return and (lf) indicates a line feed operation.

1. Search for measurement device (detector)
   A. If instrument type 51 (e.g., 3421A) is found:
      a. Address device to listen
      b. DCV: send "DCV (cr) (lf)"
         ACV: send "ACV (cr) (lf)"
         TWO: send "TWO (cr) (lf)"
         FWO: send "FWO (cr) (lf)"
   B. If 3468A is found:
      a. Address device to listen
      b. DCV: send "F1N5RAZ1T2 (cr) (lf)"
         ACV: send "F2N5RAZ1T2 (cr) (lf)"
         TWO: send "F3N5RAZ1T2 (cr) (lf)"
         FWO: send "F4N5RAZ1T2 (cr) (lf)"
   C. If instruments are not found:
      a. Display "NO INST"
      b. Return
2. Get device status byte
   A. If bit 5 is set:
      a. Display "INST ERR"
      b. Return
   B. If bit 5 is clear:
      a. Input device data to X register
      b. Return

Some examples of the mathematical routines available in the DA/C Pac ROM are listed in Table I.

The 3421A front-panel program is another part of the DA/C Pac ROM; it makes it very easy to set up and test a 3421A. It prompts the user when numeric input is required and uses standard instructions to address the 3421A. For example, the user can connect the HP-IL cables, turn on the 3421A and the HP-41, and execute the following.

| HP-41 Keys | Display | Comment |
|---|---|---|
| **Xeq, Alpha, F, P, Alpha** | | The HP-41 will find and select the 3421A on the HP-IL and reset it. |
| **DCV** | | This is the **Σ+** key if the DA/C Pac ROM overlay (Fig. 1) is not in place. |
| | Channel? | The HP-41 prompts the user for a channel number. The user would then put in the channel number and press **R/S**, or press **R/S** alone to measure the current channel or at the front-panel terminals. |
| **2, R/S** | | The user has selected channel 2. |
| | 3.00102  00 | The reading from channel 2 is flashed on the display repeatedly. |
| **FOR** | | The user wishes to advance to the next channel (the **SST** key without the overlay). |
| | −3.00105  00 | The reading from channel 3 is flashed on the display repeatedly. |

When an error occurs, caused by selecting a channel that is not present or an invalid function for the current channel, the calculator beeps and displays dashes to indicate that no reading is available.

---

**Table I**
**Example DA/C Pac ROM Mathematical Routines**

| Command | Function |
|---|---|
| C-F | °C to °F conversion<br>$X' = 1.8X + 32$ |
| THM2 | Ohms to °C, thermistor is $2252\Omega$ at 25°C<br>$X' = \left[1/(a + b\ln X + c(\ln X)^3)\right] - 273.15$<br>where $a = 1.470873889E{-}3$<br>$b = 2.377905230E{-}4$<br>$c = 1.032577937E{-}7$ |
| THM5 | Ohms to °C, thermistor is $5000\Omega$ at 25°C<br>$X' = \left[1/(a + b\ln X + c(\ln X)^3)\right] - 273.15$<br>where $a = 1.285496378E{-}3$<br>$b = 2.360998857E{-}4$<br>$c = 9.324409398E{-}8$ |
| RTD | Ohms to °C, RTD is $100\Omega$ at 0°C<br>If $X < 100$:<br>$X' = a + bX/10^2 + cX^2/10^4 + dX^3/10^6$<br>where $a = -241.9967592$<br>$b = 222.5606179$<br>$c = 25.24882388$<br>$d = -5.812682625$<br>If $X \geq 100$:<br>$X' = a - \sqrt{b + cX/100}$<br>where $a = 3367.821441$<br>$b = 13,065,764.86$<br>$c = -1,723,543.606$ |

```
HP 3421A
DATA LOGGER
2-3
DCV
23-25
ACV
4-5
DCV
RECORD OFF
PRINT

PASS 1
07/12/82 2:00 PM

2: 95.9530E-3 DCV
3: 29.9090E-3 DCV

23: 34.8000E-3 ACV
24: 35.1000E-3 ACV
25: 35.1000E-3 ACV

4: 7.74500E-3 DCV
5: -96.0940E-3 DCV

PASS 2
07/12/82 2:15 PM

2: 232.888E-3 DCV
3: 154.632E-3 DCV

23: 31.8000E-3 ACV
24: 31.8000E-3 ACV
25: 31.7000E-3 ACV

4: 96.7520E-3 DCV
5: -8.97100E-3 DCV
```

**Fig. 2.** *Typical data logging printout using the 44468A DA/C Pac in an HP-41CV Handheld Computer controlling a 3421A Data Acquisition/Control Unit and an 82162A Printer. Program records dc voltages on channels 2 and 3, ac voltages on channels 23, 24, and 25, and dc voltages on channels 4 and 5 during each periodic scan sequence. The values recorded for two scans are shown.*

The data logger program is also very simple to use, yet has some powerful capabilities not usually found in such a small system. The user is guided through a setup in which the user can select channel groups and assign various functions to these groups. Functions available include six thermocouple types, dc and ac voltage, two- and four-wire resistance, RTD and thermistor measurements, frequency, user-written routines, actuator control, and digital outputs. The user is prompted to find out whether the readings are to be recorded on the 82161A Digital Cassette Drive or printed on the 82162A Thermal Printer. Next the user is asked to select a start time, a time interval for the measurements, and the number of scans to be done. Between each scan of the

channels, a loop power-down command (PWRDN) is used to conserve battery life, in case of a system which is operated away from ac line power. At the proper time, the HP-41 and the HP-IL devices are powered up and a measurement cycle is performed, with the results being printed and/or recorded as specified by the user before the system is powered down to await the next scan. Fig. 2 shows a typical data logging printout.

Fig. 3 illustrates the use of the 44468A DA/C Pac ROM in an HP-41 to control a 3468A Multimeter for dc voltage measurements in the field.



**Fig. 3.** *Measuring electrical parameters in the field is now much easier using a 3468A Multimeter, an HP-41 Handheld Computer, and the 44468A DA/C Pac ROM.*

**David L. Wolpert**
Dave Wolpert has been with HP since graduating from the Georgia Institute of Technology in 1972 with a BEE degree. He started in data acquisition on the design team for the 3495A Scanner, was responsible for the digital control portion of the 3467A Logging Multimeter, and after a stint in production engineering, worked on the 3497A Data Acquisition/Control Unit and the 3054DL system. Dave was responsible for coding the temperature conversion and binary/decimal routines in the 3421A and implementing the HP-IB interface board. He is currently on loan to an HP group investigating code and format standards for programmable instruments. His outside interests include music and enjoying the Colorado outdoors.

# Electronic Mail for the Interactive Office

*Here's how electronic mail is implemented on the HP 3000 Computer System. HPMAIL lets users who aren't familiar with computer technology exchange messages effectively within their organization.*

**by Ian J. Fuller**

HPMAIL IS AN ELECTRONIC MAIL SYSTEM that operates on the Hewlett-Packard 3000 Computer System. It is designed to enable users who may not be familiar with computers and their associated technology to interchange information effectively throughout their organization. With HPMAIL, messages, documents, business charts and graphs, and HP 3000 system files can be exchanged both locally and remotely.

HPMAIL operates concurrently with other data-processing and office information-processing activities. It operates from any terminal that can be connected to an HP 3000. That includes the HP 2626W Word Processing Station, the compact HP 2382A Office Terminal, the HP 125 Personal Office Computer, and any of the large range of other HP terminals and desktop computers with data-communications connections to HP 3000s.

Using standard Hewlett-Packard Distributed Systems Network capabilities, HPMAIL manages all message routing and data communications among multiple HP 3000 Systems. A store-and-forward feature facilitates transmission through the nodes of a multiple-system network, providing flexible network paths through existing data-communications connections. These may be over dial-up, leased, direct-connect or public data network (X.25 and X.21) lines. Users need only specify the name of the recipient. Actual routing paths are invisible to the user. A general delivery capability routes messages to locations where they can be distributed manually.

Each user has a set of software tools that map onto the traditional elements of the desk. An electronic in tray keeps track of incoming messages, while the out tray organizes mail being sent. Users monitor the progress of messages in the pending tray. Senders can not only verify that their messages have arrived, but also that they have been read. A work area provides for composing and editing longer communications, and for assembling packages of information files, including graphics. A file cabinet stores messages and documents. A distribution directory lets the user construct, use and store standard distribution lists. An administration area provides the means to tailor the HPMAIL environment with passwords, autoforward instructions, autoanswer messages, and the choice of a person to handle mail on the user's behalf.

Simple commands perform everyday mail functions. Movement between HPMAIL areas is by function keys, automatically labeled on the screen when that ability is supported by the particular HP terminal.

To reduce the on-line system storage traditionally required by computer-based message systems, HPMAIL uses an HP 3000 IMAGE data base for document storage and local distribution. Documents are stored only once on each system in a network with pointers for each intended recipient.

## HPMAIL Functions

The basic facilities required of any electronic mail system are to accept mail, address users, store mail, sort outgoing mail, move the mail, sort incoming mail, deliver mail to recipients, and notify recipients. HPMAIL performs all these functions. It also tracks the progress of mail and provides a flexible and powerful user interface.

HPMAIL can transmit any information that can be stored in the HP 3000 system. This can range from a few lines of simple text to a complex report containing, perhaps, reports produced by a word processor such as HPWORD, graphics produced by DSG/3000 or HPDRAW, and even HP 3000 program and data files. We decided to adopt as flexible a structure for these items as possible, enabling the user to organize them into packages of logically related items. At the same time, we realized that the majority of HPMAIL messages would be simple text and made it very simple for this type of message to be accepted into the system.

HPMAIL can be implemented as a multicomputer system, but it is important in the office environment that users be shielded from the details of the configuration of their particular network. To this end we implemented an addressing scheme. HPMAIL users are registered with the system in such a way that senders of messages do not need to know whether recipients are located on their computer or on one many miles away, connected through a complex network. It also allows the administrator of an HPMAIL network to reconfigure it without users having to change the way in which they address other users.

The system must be able to store mail before transmission, on intermediate computers in a store-and-forward network, and on the behalf of users. Because of the potentially large space requirements for messages on a system, HPMAIL has adopted the strategy of sharing information wherever possible. Thus, when a message is delivered to several users on the same computer, only one copy is stored and pointers are set up linking the message to each user's in tray.

Mail in transit must be sorted into queues for transmission to the required destination and for distribution at the destination computer, a process analogous to the operation of a sorting office in a manual mail system. This can be a
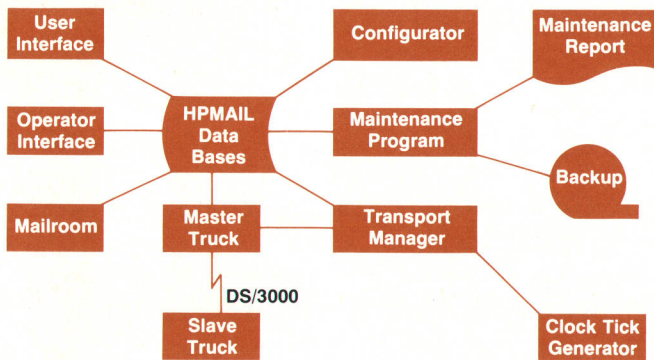
**Fig. 1.** *Major software components of HPMAIL.*

time-consuming process, involving the analysis of large distribution lists, and so we decided to have this function performed by a background process called the mailroom.

Mail must be transported between computers as economically as possible. HPMAIL uses the program-to-program (PTOP) communication facility provided by DS/3000 for this task. These potentially lengthy and error-prone operations are handled by autonomous processes known as the transport manager and the master and slave trucks.

Once arrived at its destination computer, mail must be delivered to its recipients and the recipients must be notified of their new mail if possible. This work is also handled by the mailroom program.

For users to have confidence in the operation of the mail system they must be able to track the progress of messages that they have sent. HPMAIL provides five acknowledgment levels that users can put on a message and the transport system keeps users informed of the progress of their messages according to the acknowledgment level that they have set.

The HPMAIL user interface provides users with powerful facilities to create and store messages on their electronic desk, making HPMAIL far more than a simple message system. Examples of the facilities provided are the work area in which items may be constructed before incorporating them in messages, the filing cabinet, in which users may store messages for future retrieval in folders they have named, and the distribution list area, where users can keep distribution lists of other users with whom they correspond for easier compilation of messages. When we designed the HPMAIL user interface we placed considerable stress on ease of use for the nontechnical user and compatibility with any terminal that can log on to the HP 3000. This lets users access their mail even when they are away from their office if they have a portable terminal and access to a telephone.

## Implementation of HPMAIL

One of our principal design objectives in implementing HPMAIL was that the product should be easy to maintain, test, debug, and update. To this end we decided upon a layer concept for the software, with each layer communicating via rigidly defined interfaces. This allowed development of the transport mechanism to proceed relatively undisturbed while the details of user interface design were refined through a series of tests, prototypes, and evaluations. The task of integrating these components, often a

major headache in software projects, proved to be a relatively simple task with HPMAIL and the software has proved to be very robust in use.

Fig. 1 shows the major software components of the product. The user interface program is responsible for all the user interaction in the product and provides the user with the electronic desk described previously. The mailroom is a background job streamed by the system operator and is charged with the sorting and delivery of local mail, that is, mail that is destined for users on the same computer. The transport manager is responsible for scheduling the communication of messages between HPMAIL computers according to the current message load and the availability schedule set up by the HPMAIL system administrator.

The master truck is controlled by the transport manager and is responsible for the setup and control of the DS/3000 connection with a remote computer where it uses the slave truck program to receive the mail and put it onto the remote data base. The clock tick generator runs as a son process of the transport manager and provides it with a tick every fifteen minutes to cause the transport manager to reschedule its transmission priorities as necessary.

The configurator is a VPLUS/3000 application program used by the HPMAIL system administrator to build and configure the HPMAIL data bases. Facilities are provided so that the administrator does not need to have any detailed knowledge of IMAGE/3000 to control HPMAIL.

The operator interface is a set of functions that enable the system operator to control HPMAIL and determine its status. Examples of operator commands provided are MAILON and MAILOFF to enable and disable the system, MAILSTATUS to provide a status display for the system, MAILSHOWNODE to display the number of messages awaiting transmission to remote computers, and MAILMAINT to start the maintenance program.

The maintenance program is run periodically to provide verification of the intactness of the HPMAIL data bases, to delete outdated mail items, and to generate statistics, which are formatted and printed by the report program. Regular running of the maintenance program is a vital part of maintaining a reliable HPMAIL system.

## Development Techniques

The programs described above were developed simultaneously over a period of eighteen months. For this work to proceed reliably a set of techniques was derived before detailed design began.

Three techniques we adopted in the development of the product deserve mention here as being important to the way in which the product evolved. These are the use of mail procedures to access the data bases, IPC or pipe files to communicate between the components of the product, and the standardized tracing and error reporting mechanism used throughout the product.

**Mail Procedures.** The HPMAIL software is built around two IMAGE/3000 data bases, called LOCAL and GLOBAL, which are discussed in the next section. The application programs that access the data bases, for example the user interface, mailroom, transport manager and trucks, do so through a set of intrinsics (routines) called mail procedures. These mail procedures are in a segmented library and so are

accessible to all the programs. They perform the low-level data base access functions and insulate the programs that call them from the details of the data base structure. In fact, the transport programs do not have a single IMAGE call in them; all interaction is done through mail procedures. Examples of the mail procedures include MOPEN to open the mail data bases, MCREATE to create a new item, MEXPLODE to return information about the contents of an item, for example the messages in a user's in tray, MATTACH to attach one item to another, for example, to attach a new message to a user's in tray when delivering it, MDELETE to delete an item, and MCLOSE to close the mail data bases.

Other mail procedures were implemented to perform common functions required by different modules, for example, comparing two names. In all, about thirty mail procedures exist in the HPMAIL segmented library.

These procedures were designed, tested, and made available before any coding was done on the programs that would use them, thereby providing a firm base upon which development could proceed.

**IPC Files.** Besides using the two IMAGE data bases, the various components of HPMAIL communicate with each other using the interprocess communication (IPC) or pipe facility provided by the HP 3000's MPE-IV operating system. This method of communication has the advantages of being very well defined and straightforward. It is reliable and provides an interrupt mechanism whereby a program can wait for a message on its pipe telling it what to do next. Once again, this convention aided the product implementation. Modules were tested using simple test harnesses before attempting to integrate them.

Three IPC files are defined in HPMAIL. MRIPCIN is the mailroom input pipe, which is written to by the user interface when a user MAILs a message, instructing the mailroom to collect it from the user's out tray and deal with it. It is also written to by the slave truck when it receives mail from a remote computer bound for the local computer, alerting the mailroom to deliver it.

TMIPCIN is the transport manager input pipe, which is written to by the mailroom to inform it about new mail bound for remote computers which it should attempt to send, and by the clock tick process, which runs as a son of the transport manager and sends it a message on this pipe every fifteen minutes to cause it to reschedule itself to take account of any change in the availability of routes out of the local computer.

MTIPCIN is the master truck input pipe. The transport manager sends instructions, such as "open a DS line to computer X," to a master truck, which attempts to perform this operation and reports back to the transport manager on its input pipe the success or failure of the operation. The transport manager can support up to eight master trucks simultaneously and each has a separate input pipe, referred to as MTIPCIN0, MTIPCIN1, etc.

**Tracing and Error Reporting.** Since HPMAIL consists of a number of intercommunicating programs and each program consists of a large number of modules, each separately compiled, it is important that status information about the success of each operation is returned to the caller. It is also important to have a trace facility built into the product so that errors can be traced and diagnosed with minimum trouble.

In HPMAIL, a single data structure called the mail common area is used to communicate shared data such as completion status, file numbers, and user information between the various procedures in a program and with the mail procedures. This is very much in line with products such as VPLUS/3000 or IMAGE/3000.

Each procedure in the product (and there are over 300) takes the mail common area as its first, and sometimes only, parameter. The convention was established that all the procedures call a mail procedure called MTRACE immediately upon entry and before exit. MTRACE registers the procedure's identity in the mail common area and can also be made to write a log message to a file if required.

Error reporting is centralized in another mail procedure, MERROR, which interprets the status words in the mail common area and can produce a customized error message from the message catalog for each error, together with as much additional information as can be deduced. Since all the trace messages are derived from the message catalog, they can, if necessary, be translated to other languages.

The tracing level of these procedures can be varied by setting a job control word (JCW) or by passing a run-time parameter (PARM) to the program. Thus, when debugging a program, the engineers can easily get full trace information and errors can be located more easily. This level of control was particularly important in testing the transport programs, where it was not practical to use the MPE debug facility. These tracing facilities remain in the released product, but cause little performance penalty since they are normally disabled.

## HPMAIL Data Bases

HPMAIL uses two IMAGE/3000 data bases to contain the user and network directory information and the users' elec-
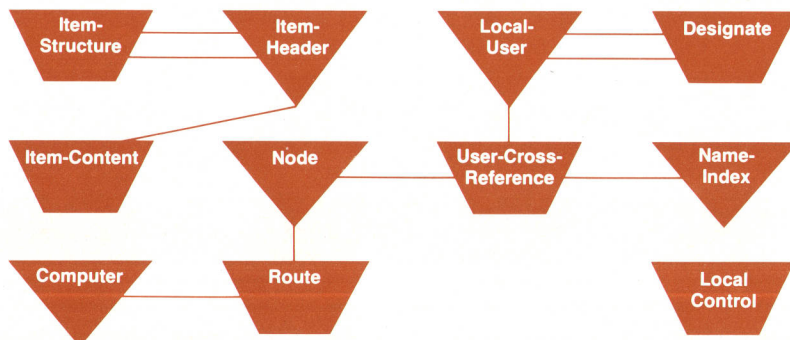


**Fig. 2.** The HPMAIL LOCAL data base contains local directory information about users, network directory information, and mail items.

tronic desks, all maintained securely and reliably.

All the HPMAIL programs on a single computer work with the same data base set. Items that are simultaneously owned by more than one user, for example a message delivered to two in trays, are held physically only once per system, with pointers to assert individual ownership.

The HPMAIL GLOBAL data base is a directory of all users known to this system. The LOCAL data base contains local directory information about the users supported there, network directory information to control intercomputer transmission, and mail items. Mail items can be in trays, messages, queues of messages, individual text items, distribution lists, or MPE files imported onto the data base—in fact, nearly anything an HP 3000 can store.

The HPMAIL data bases perform three functions: directory, storage, and item composition. The directory function enables HPMAIL to identify users when they sign on and determine those items that they can access. The directory also identifies users on mail distribution lists and enables user names as entered to be expanded to the full canonical form of name, location and sublocation. The network directory functions enable HPMAIL to determine which computers to contact to send mail to a given location, how and when to contact the computers, and what telephone number or public data network (PDN) node name to specify.

The storage functions enable HPMAIL to hold everything in the IMAGE data bases so that the information is always available. Thus HPMAIL will take an HPWORD document and import it onto the LOCAL data base so that at some later time the transport system will be able to send it to another computer without requiring access to the sending user's MPE group, or having to worry if the user deleted it.

The composition functions enable HPMAIL to handle hybrid message types, to have the same message in several in trays or filing cabinet folders at the same time, and to operate queues of messages destined for different locations in the network. This leads to the important concept in HPMAIL that the simplest message contains two parts: a distribution list and some content (usually text). The message itself is a single item that contains these two parts. This split enables the transport mechanism to refer to the message as a whole, while the mailroom can access the distribution list, its primary concern, without having to scan the text. The end users also derive the benefit that they do not have to read the message distribution list but can go straight to the text.

HPMAIL uses the IMAGE/3000 Data Base Management System for many reasons. Almost the least important is that it is a data base management system. The main reason is that it is solid and well-established and provides a firm foundation for a reliable product.

It also offers its own space management, allocating new space as required and handling direct access problems transparently. Because of the hybrid nature of most HPMAIL items, the chain management system of IMAGE is very important. It handles pointer chains very easily. IMAGE uses shared data storage and shared buffering, adding to efficiency and performance. It has a good locking mechanism to prevent unreliable operation caused by multiple updates of the same item. Finally, an IMAGE data base can be dumped for security purposes very much faster than many MPE files.

The main drawback to using IMAGE is that it is not designed to hold nonstructured items such as MPE files, and getting them in and out of the data base (importing and exporting) is a fairly lengthy process. We estimate, however, that the time required to implement the project would have nearly doubled if the facilities of a system like IMAGE had not been available.

### Data Base Descriptions

Figs. 2 and 3 show the schemas (definitions) for the two data bases. Two examples of the way HPMAIL uses the data bases in the execution of common functions are item storage and name searching.

The most important data sets to the understanding of HPMAIL's item storage are the ITEM-HEADER, ITEM-STRUCTURE and ITEM-CONTENT data sets on the LOCAL data base. Items in HPMAIL are considered to be either basic, meaning that they can be represented by a single MPE file (for example a distribution list or an HPWORD document), or composite, which have no MPE file form as such but are composed of a number of basic items. An in tray is an example of a composite item, consisting of a header linking it to the messages in that particular in tray.

Every mail item, basic or composite, has a type (e.g., user folder, message, text), a creator, and a subject. This information is kept in its item header. The ITEM-STRUCTURE data set maintains the links between items that make up composites. For example, a message consists of a message header linked by two item structure records to the headers of its distribution list and content. The content of a basic item, the MPE file content, is held in the ITEM-CONTENT data set, linked by IMAGE pointers to its item header.

In HPMAIL, every item is linked to at least one other, except for the root item, which we call item zero. If an item becomes detached from all its parents, which happens, for example, when a message is deleted from the in trays of all the users who received it, it becomes an orphan item and will be physically deleted from the data base by the maintenance program.

Fig. 4 shows the HPMAIL object hierarchy in more detail. Although superficially complex, it is in fact quite efficient and lends itself very well to processing by IMAGE via the mail procedures.

### Searching for Names

Every time a user enters a name into the user interface program, whether it is the user's own name entered during the HPMAIL sign-on, or names on a distribution list,
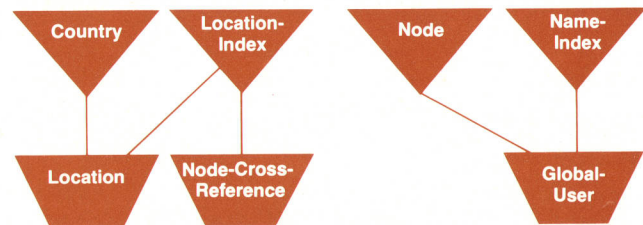


**Fig. 3.** The HPMAIL GLOBAL data base is a directory of all users known to the particular system.
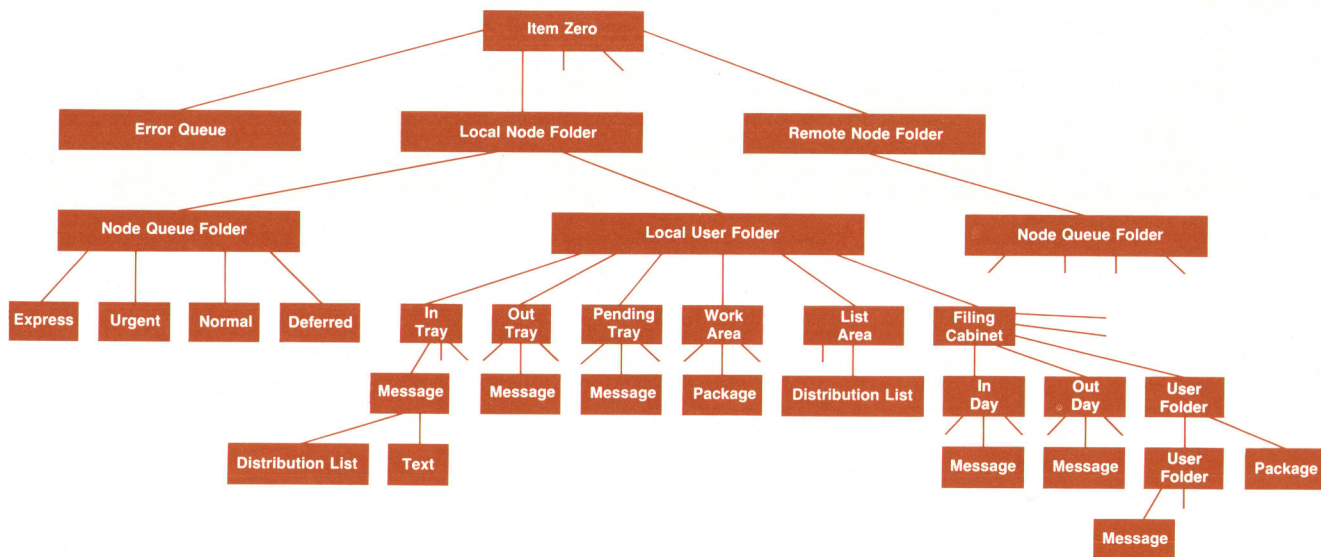
**Fig. 4.** *HPMAIL object hierarchy.*

HPMAIL searches its directory for that name or a similar match. Names in HPMAIL are held in what we call canonical form, which is:

<SURNAME>,<FIRST NAME> <MIDDLE NAME>/
<LOCATION>/<SUBLOCATION>

where the surname, first name, and middle name add up to no more than 36 characters, the location is up to six characters long, and the sublocation is two characters long. Canonical names are all uppercase, converted by means of an internal translation table to take care of national character sets. For example, the canonical form of user Ian John Fuller, location HP1600, sublocation 01 is:

FULLER,IAN JOHN/HP1600/01

HPMAIL does not insist that the user type in the full name, just enough to identify it uniquely on the data base. Thus if there are no other users with the surname FULLER defined on the data base, FULLER will be sufficient. If any doubt exists, the user interface displays those available and asks the user to choose one. Of course, user names must be unique to a mail node (location+sublocation).

When HPMAIL searches the data base for a name, it does so by first making a name probe from the surname supplied. The name probe consists of the first letter of the surname, followed by the next three nonrepeating consonants. For example, the surname FULLER yields the name probe FLR. This probe is taken as the key to a search of the NAME-INDEX master data set, which will give a chain of names that have this probe. The user interface scans down this chain comparing each with the canonical form of the input name. There may be several degrees of match: all components agree, first names differ, locations differ, etc. If a perfect, nonambiguous match occurs, then the user is not troubled further. Otherwise the user is given a choice of those available. The use of the name probe means that a slight misspell-

ing of a user's name can sometimes but not always yield a choice that reveals the correct spelling of the name the user was looking for. For example, if the user types the surname COALMAN instead of COLEMAN, then the program will be able to give a choice of the correct name since both names result in the name probe of CLMN.

In the case of the user sign-on process, the LOCAL data base is searched for the name, since it holds the name information for the users supported on that computer. For the construction of distribution lists, the GLOBAL data base is searched since it contains the directory of all the users known to that computer, but not necessarily supported on it. The user has no need to know where another is supported; this is sorted out automatically. The name searches are done in a similar way on both data bases by mail procedures.

**User Interface**

The HPMAIL user interface provides the user with an environment designed to mirror the concepts familiar to office users. As mentioned earlier, the user is provided with an in tray into which incoming messages are delivered by the mailroom, an out tray in which messages the user sends are assembled and from which they are collected by the mailroom for transmission and delivery, a pending tray which holds messages awaiting acknowledgments (the user can find the acknowledgment status of any message in the pending tray by reading its distribution list), a work area which allows the user to create items and assemble them into packages before incorporating them into messages, a distribution list area which holds lists of users with whom the user may wish to communicate, a filing cabinet in which the user can file messages for future reference, and the administration area which gives users extra commands to do such utility functions as defining or changing their passwords and naming persons who may work on their behalf.

The command set of HPMAIL is designed to reflect the

terminology used in offices. Examples of HPMAIL user interface commands are:

READ      to display an item on the user's terminal
PRINT     to print an item on the system line printer
SEND      to start to send a message
MAIL      to commit a message for transmission
FILE      to copy an item to a folder in the filing cabinet
REPLY     to send a reply to a message in the in tray.

The function keys of Hewlett-Packard terminals are used if possible to give the user an easy method of using the product. Function keys **f1** to **f7** select entry to each of the areas listed above. **f8** allows the user to leave HPMAIL.

The prompt displayed by HPMAIL is another aid to the user. This changes according to the area. The in tray prompt is IN TRAY>, the filing cabinet prompt is CABINET>, etc. When the user has OPENed a folder the prompt changes to the first word of the folder's label, so a folder with the label User Group Presentation will, when OPENed, give the prompt USER>.

Fig. 5 shows an example of HPMAIL's main menu, which is displayed as soon as the user signs on. Following that, a typical user presses function key **f1**, selecting the in tray to read any new messages. These messages are summarized in a list as shown in Fig. 6. Note also how the prompt has changed from HPMAIL> on the main menu to IN TRAY> to indicate the change of area.

Typing the command READ 1 after the IN TRAY> prompt causes the first message in the in tray to be displayed on the user's terminal. Fig. 7 shows an example of the format of a message displayed by HPMAIL.

The user interface also offers an extensive HELP facility which is organized on three levels. Typing the HELP command by itself gives general help about the area the user is in. For example, HELP in the filing cabinet gives the new user some idea of the purpose of this area and suggests some likely commands. The command HELP followed by another command name gives the user specific help about that command as it is used in that area. For example, typing HELP MAIL while composing a message explains the meaning of the command and how to use it. Finally, typing a question mark (?) in any area gives the user a brief list of the commands available.

HPMAIL was designed to be translated easily into other languages. The translator doesn't need knowledge of the internal workings of the product or access to the source code. To this end, all messages produced by the product come from a central message catalog (which is currently about 4,000 lines long) while the HELP displays come from another catalog which is even longer.

All user input except the content of messages is parsed by common parsing routines and commands are recognized by reference to a dictionary. The development team wrote a special utility to build this dictionary, which goes under the name of Webster. It is available to Hewlett-Packard software centers worldwide to help them adapt the product to their local language. Synonyms and common abbreviations for commands can be built into the dictionary as required for each language.

Local character sets are handled by a translation table as part of the dictionary so that users may, in their local versions, use commands containing their local characters. One limitation to the use of local character sets in HPMAIL, however, is that user names, locations and sublocations have to be in the standard USA ASCII code. This is because an HPMAIL network can span countries, not all of which may use the same conventions regarding extensions to the character set.

## User Interface Structure

The HPMAIL user interface program is a direct analog of the data structures it supports in the LOCAL data base. It is a direct result of the maxim, "Design the data structures correctly and the applications programs will almost design themselves." The program is a tree structure, as shown in Fig. 8, which follows the object structure quite closely.

The outer block of the user interface controls all the other

```
HPMAIL                                          1325 02/07/83

Ian Fuller                                Location HP1600/01

You have 3 new messages.

Select    1    IN TRAY - Read your incoming mail
          2    OUT TRAY - Send messages
          3    PENDING TRAY - Track messages you have sent
          4    WORK AREA - Create new work
          5    DISTRIBUTION LIST AREA - Create new lists
          6    FILING CABINET - Look in your folders
          7    ADMINISTRATION
          8    LEAVE HPMAIL - Sign off and end the program
          9    Sign off, and Sign on as someone else

Please type a number to indicate your choice.
Type HELP if you need help at any time.

HPMAIL  >
```

**Fig. 5.** *HPMAIL main menu display.*

```
IN TRAY of Ian Fuller                                02/07/83 1349.

4 messages.

                                                       N A P U
                                                       E C R R
                                                       C   K I
    Item   Subject              From            Received  W K I G

     1     Travel Plans         WILLIAMS,PETER  02/06/83
     2     Security             DRAPER,COLIN    02/07/83   *     *
     3     Publications         MUELLER,AMY     02/07/83   *
     4     Part Numbers         LOVETT,JULIE    02/07/83   * *

IN TRAY >
```

**Fig. 6.** *Typical in tray display for HPMAIL.*

components, which can broadly be broken down into initialization, user sign-on, environment handlers, sign-off, and shutdown. The initialization module opens the data bases, the user's input and output files, and the message catalog and reads the dictionary from a disc file onto the user's stack. It initializes the mail common area with all relevant data and generally prepares the program for execution. It is also responsible for checking that the software has not been stolen and that incompatible versions of the software are not being used. The last task it performs is to print the HPMAIL banner on the user's terminal.

The user sign-on module is the first that actually solicits input from the user (by asking for a name). The name allows the user's user record to be located in the LOCAL-USER data set of the LOCAL data base, and from that point on, the user interface can go to a specific point in the object hierarchy shown in Fig. 4.

After a successful sign-on, control is passed to the main menu processor, the first and simplest of the environment handlers. This simply displays the list of options available to the user, along with the number of newly delivered messages in that user's in tray. The user is then prompted for a choice of area (environment) to enter next. This choice can be made by pressing the appropriate function key or by typing the corresponding number.

The other environment handlers for the in tray, out tray, pending tray, work area, list area, and filing cabinet, are very similar in structure. They consist of five basic states: initialization, building a list file as a temporary MPE file of the current contents of the area (e.g., all the messages in the in tray), producing an initial LIST of the area to give the user the information about its contents, prompting the user for a command, and executing the command.

Each command (READ, FILE, etc.) has its own executor, which handles the operation of that command in all areas. Once the environment handler has recognized that a certain command has been entered by the user, it passes control to the command executor for that command. It is the responsibility of the executor to determine whether the command is correct, possibly asking the user for additional information if necessary. If the command is correctly formed, the executor passes control to the action procedure for the command. This is often just a simple mail procedure. For example, the task of the FILE command executor is to determine what the user wishes to file, and where. This is resolved internally into a pair of internal item numbers in the ITEM-HEADER data set. These are presented to the action procedure for the FILE command, which is the mail procedure MATTACH, whose task it is to perform the required attachment, linking, say, an in tray message to the incoming day folder in the user's filing cabinet.

## Command-Based User Interface

We are frequently asked why we adopted a command-based user interface for HPMAIL when many of Hewlett-Packard's Interactive Office products have standardized upon point-and-push interfaces, that is, the user positions the display cursor and pushes a function key. This question caused a considerable amount of discussion in the development and marketing teams during the design phase of the product and we finally decided in favor of an unsophisticated command-based interface for two main reasons.

First, we realized that many of the potential users of our product would be managers, working away from their desks, perhaps in hotel rooms, on hard-copy terminals connected to their office HP 3000 by a slow-speed telephone line. We thought that these users would derive a major benefit from using HPMAIL and so we designed our product to be no more demanding of a terminal than that it be teleprinter-compatible. We would not go so far as to claim that the HPMAIL user interface works well on every terminal that can be connected to an HP 3000, for we have not tried them all. However, we would be very interested to know of any upon which it does not work.

The other main reason for adopting the command-based interface is that, for regular users, such an interface is far quicker to use than one where you have to cycle through several screens just to access the menu you require. Users often comment that, as long as you are not making errors, the user interface is extremely terse. It does not usually ask for confirmation of actions, nor indulge in long dialog about exactly what to do. This agrees with our belief that an electronic mail system is of little use to an occasional user and to derive major benefit from it, it must be used regularly. Otherwise users would cease to rely on it as a timely communication medium. Regular users, we believe, do not want verbose or long-winded user interfaces, they know what they want to do and do not want the software to get in the way.

Ideally, of course, we would have written a user interface for every level of user who might encounter our product. However, we compromised with one we thought would

```
IN TRAY > READ 2

Start of Item 2.

Message.                                        Dated: 02/07/83 at 1135.

Subject:  Security

Sender:  Colin DRAPER / HP1600/01                   Contents:  2.

Part 1.

 TO: Ian FULLER / HP1600/01

Part 2.

Could you arrange for me to have a key to the building, please.
Regards, Colin.

End of Item 2.

IN TRAY >
```

**Fig. 7.** *Reading the second message of the in tray list shown in Fig. 6 gives this display.*

please the maximum number of people most of the time.

One drawback of a command-based user interface is that it is often perceived as being harder to learn initially than point-and-push interfaces. Many of our users are busy managers who have no time to go on long product training courses or to wade through long manuals just to learn how to send a simple message. I was told once while I was testing the product with some Hewlett-Packard managers in the United States that if they could not learn the essentials of the product in a maximum of thirty minutes it would not be accepted. To this end we developed an on-line training package for the product, known as the interactive training facility. This gives users the opportunity to learn about those aspects of the product they need to use at their own speed in their own time. We find this to be an extremely valuable facility in getting the product used throughout the company.

It is interesting to note while discussing user interfaces that the HPMAIL configurator program has a point-and-push user interface based entirely on VPLUS/3000. We felt that the configurator was an ideal application for such an interface and makes the potentially complex process of data base building, configuration, modification, and expansion far easier to achieve for system administrators.

### Transport System Overview

As stated previously, one of the major product objectives for HPMAIL was to provide a system in which the end user did not need to become involved with the intricacies of message routing and transmission, understanding DS/3000, or dealing with line failures, message congestion or any of the many other problems that can befall a multicomputer messaging system.

Once a user MAILs a message it is removed from the out tray by the mailroom and handed over to the transport system for sorting, transmission, and delivery. The user can track the progress of the message by setting an acknowledgement on the message and monitoring it in the pending tray while it is sent to all the recipients, wherever they may be.

Fig. 9 shows an overview of the data and control flow required to implement the store-and-forward system used in HPMAIL.

The transport system is implemented by background jobs communicating via IPC files. The jobs are known as the mailroom, transport manager, and at least one master truck communicating with a slave truck on a remote computer.

The mailroom has two basic functions, first to collect messages from users' out trays, and second to deliver incoming messages to recipients on message distribution lists where the users are supported on the same computer. It also handles delivery acknowledgments, autoforward, and autoanswer. The mailroom is structured internally in two halves. The outgoing half deals with mail collection and sorting, and the incoming half deals with local mail delivery.

Outgoing mail originates from the user interface in response to the user's MAILing a message created by the SEND, FORWARD or REPLY commands. The IPC message sent to the mailroom when a message is MAILed identifies the specific message in an out tray that has been committed for trans-
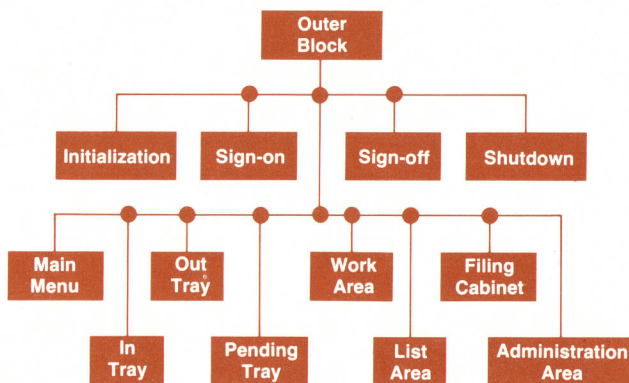


**Fig. 8.** *The HPMAIL user interface program has a tree structure that closely follows the object hierarchy.*
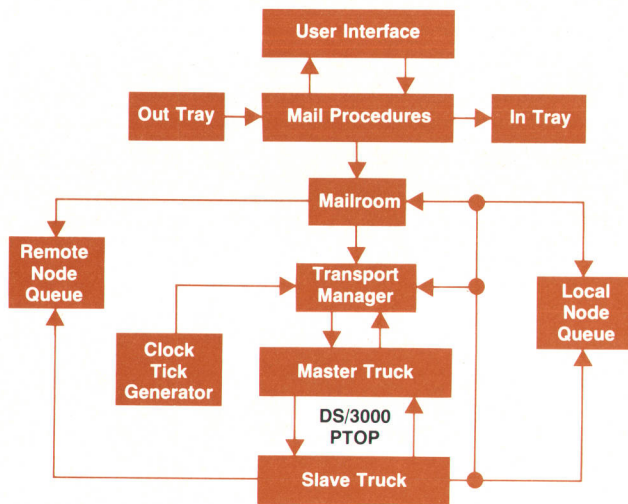
**Fig. 9.** *Overview of the data and control flow required to implement the store-and-forward system used in HPMAIL.*

mission. The mailroom responds by accessing the message and scanning its distribution list. It builds an internal list of the number of unique mail nodes (local or remote) that the message is bound for and then attaches the messages to the node queues for the correct destination nodes. For locally bound messages the outgoing mailroom sends an IPC message to its incoming counterpart to deal with the delivery. For remote messages it sends a message to the transport manager alerting it of the new mail. If an outgoing message requires any acknowledgment, the mailroom attaches the message to the sending user's pending tray, ready to receive the acknowledgments as they come in. The last action it performs is to detach the message from the sending user's out tray.

The incoming half of the mailroom deals with mail originating from two sources. First are the local messages sorted by the outgoing half. Second, the incoming mailroom receives IPC messages from a slave truck operating on its computer to handle messages received from a remote computer and to be delivered locally.

In each case, the incoming mailroom examines the message distribution list to determine to whom the message is to be delivered. It then locates the required users' in trays on the LOCAL data base and attaches the message to them. It also has to handle the various types of acknowledgment that can be received, amending the user's pending tray copy of the message with the fact that the acknowledgment has been received. Its other main tasks are to generate delivery acknowledgments for the message if necessary and to perform any autoforward or autoreply that a receiver has set.

### Transport Manager

The transport manager is another background job. Its function is to control and schedule the operation of HPMAIL in sending mail to remote computers. It maintains tables of mail nodes for which mail is waiting and the possible routes that it can use to move the mail. This information is derived from the HPMAIL system configuration specified using the configurator program. Every fifteen minutes the transport manager receives a clock tick from its

son process, the clock tick generator. This causes it to reexamine its priorities and reschedule the master trucks under its control if necessary.

The transport manager is constantly striving to ensure that the remote mail nodes with the highest-priority mail to send are being serviced by the available master trucks. Once it selects an available mail node, together with an available route to another computer, it sends an IPC message to a free master truck instructing it to establish a DS/3000 connection with another HPMAIL computer. Once the master truck has reported that it has done this successfully, the transport manager instructs the truck to clear a certain remote node queue. Once this has been done, the master truck reports back, ready to receive further instructions.

The transport manager is capable of controlling up to eight master trucks simultaneously, although it would be a very ambitious network that required this level of resources.

The master truck receives IPC instructions from the transport manager and handles all the interaction with DS/3000 and PTOP to move mail physically between computers.

The first instruction received by an idle master truck is to establish connection with a certain remote computer (identified by name) using a specified DS/3000 line, and if necessary, a specified telephone number or public data network node name. It uses an MPE DSLINE command to open a line to the remote computer, followed by an MPE REMOTE HELLO command to establish a remote session. It then attempts to create and activate a slave truck on the remote computer, using the DS/3000 intrinsic POPEN. If the slave truck is successfully awakened the master truck determines that it is, in fact, communicating with the correct remote computer by interrogating the slave for its computer name. Once this protocol has been observed, the master truck reports back to the transport manager "truck idle with DS line open" and awaits further instructions.

The next command received by the master truck in the message transmission operation is the move mail command. This instructs the master truck to transmit mail from a given remote node queue on the LOCAL data base. The master truck is responsible for serializing messages from the data base and packing them into transmission buffers for sending to the remote computer via the PWRITE PTOP intrinsic.

At the other end of the communication link the slave truck accepts these transmissions, if possible, and recreates the message on its computer. After the message has been completely transmitted, the slave truck deserializes the message onto its data base. Depending on the ultimate destination of the newly received message, the slave truck either alerts the mailroom for locally bound messages or its own transport manager for messages that are destined for onward transmission. Once the master truck has sent a complete message successfully, it detaches that message from the remote node queue, since responsibility for it has now passed to the next computer in the chain.

This process is repeated for all the messages that the master truck has to send. Once the master truck has complied with its instructions to move mail, it reports back to the transport manager "truck idle with DS line open." The

transport manager either responds with an instruction to begin transmitting messages bound for another mail node, taking advantage of the open DS line, or instructs the master truck to terminate the link if there is no more mail to be sent on that route.

The main challenge in designing the transport mechanism was to provide for all the failures that can take place during transmissions over communications lines. Examples of the type of problems that have to be handled are:

- Inadvertent connection to the wrong computer, perhaps because of an incorrectly dialed telephone number
- Finding that the remote data base cannot accommodate the message to be sent, either temporarily because of congestion, or semipermanently because the data bases are not sufficiently large
- Configuration errors resulting in transmissions being refused or an infinite loop being set up
- Line failure during any stage of the process.

Since detachment from a node queue is the last operation that the master truck performs, and this happens only after it has been determined that the message was transmitted successfully, it is unlikely that messages will be lost because of line failure. However, it is possible that HPMAIL will send messages twice in some circumstances.

## Why Use PTOP?

We chose the program-to-program intrinsics provided by DS/3000 in preference to the other methods of transmitting files across DS (distributed systems) connections for four reasons. First, PTOP ensures that there is a communicating process at the other end of the DS line to receive the mail. Without it, we would have had to implement yet another level of IPC communication to ensure that the remote computer was aware of the mail being delivered to it. Second, the performance of PTOP is superior to, for example, remote data base access (RDBA), since in good circumstances, the master truck can be assembling transmission buffers for the next transmission at the same time DS is taking care of the transmission of the previous buffer over the line. This makes optimum use of the communications line.

Third, PTOP provides very good application program control over the communication activity with the provision of a tag field in the PTOP intrinsics that can be used to send status, control, and block-level acknowledgments at the same time as the buffer transmissions. Lastly, the PTOP intrinsics are implemented on other Hewlett-Packard computers, for example the HP 1000 Technical Computers, giving us the option of using these machines as mail nodes in a mixed DS network.

By removing the transport mechanism to a completely separate layer of the product we will be able to take advantage of enhancements to Hewlett-Packard's data communications products as they become available with minimal disruption to the product's users. It is notable that the HPMAIL transport mechanism uses only supported features of DS/3000 and works entirely in user mode. This, we feel, contributes greatly to the robustness of the product. This remark applies to the product as a whole. In fact the only line of privileged code in the product is one where the system cold loa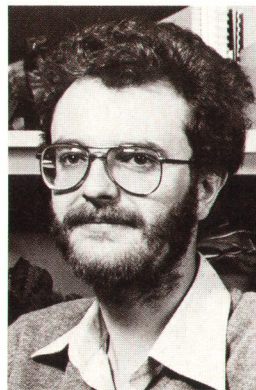d identity is obtained. This is required to maintain system integrity, since we have to know if there has been a system restart since the data bases were last accessed.

**Ian J. Fuller**

Ian Fuller received the BSc degree in mathematics, computer studies, and physical sciences from Oxford Polytechnic, Oxford, England in 1978. After two years working on software for message switching and PBX telephone systems, he joined HP in 1980. At HP, he has been involved with the HPMAIL project as a development engineer and a project leader and currently is the HPMAIL project manager. Born in Gosport, Hampshire, Ian now lives in Little Sandhurst, Camberley, Surrey, four miles from his office at HP's Commercial Systems Pinewood software development center. His interests include photography and travel.

# Integrated Tools Improve Programmer Productivity

*This software subsystem for the HP 3000 Computer System saves program development time by giving the programmer access to several utilities through a single command interpreter.*

**by Anil K. Shenoy and Carolyn M. Bircher**

HPTOOLSET IS A SOFTWARE SUBSYSTEM that provides an integrated program development environment to improve programmer productivity. Through a single, friendly user interface, it offers programmers access to several programming tools needed for software development. It lets programmers perform multiple functions simultaneously and access information about other program development functions being performed.

HPToolset runs under the MPE operating system on the HP 3000 Computer System. The first release is geared towards improving COBOL II programmer productivity but HPToolset is structured such that many of the features do not depend on any specific language.

The phases of implementing a software program include source code creation, modification, compilation, execution, and testing. These steps are executed repeatedly during the life of a program and can be very time consuming. HPToolset relieves the programmer of the responsibility of manually directing program flow through each of these steps and minimizes development time by allowing the programmer to move between the different phases of program development without leaving the HPToolset environment. Since program development is often a team effort, HPToolset promotes sharing of information between programmers, which in turn increases the effectiveness of the programming effort.

HPToolset offers the following aids towards improving productivity:[1]

- Menu and labeled function key interface
- Ability to perform multiple functions without exiting and reentering subsystems
- Sharing of files between team members through version management
- Full-screen editing with a point-and-push philosophy (i.e., the user positions the cursor and pushes a function key)
- Background compilations and on-line listings
- Point-and-push method to find compilation errors
- A GO key to eliminate the manual steps between compilation and execution
- Dynamic symbolic debugging
- Ability to go directly from symbolic debugging to the related compilation or source listing
- Context-sensitive HELP facility.

## Basic Concepts

The following concepts are fundamental to an understanding of HPToolset.

A workspace is a directory of the files that are used in the development of a program. It represents the programmer's development environment. The programmer is required to create or reference a previously created workspace before executing most HPToolset functions.

The HPToolset access method, TSAM, is a customized file access mechanism that is built on top of MPE's file system. It provides, among other features, the ability to define and access versions of changes made to a source file. HPToolset provides interface procedures for compilers and other procedures that wish to access TSAM files. The COBOL II compiler uses these procedures to read TSAM source files. A conversion utility is also provided to convert MPE ASCII files or KSAM/3000 (keyed sequence access method) files to TSAM format. Only TSAM files can be edited via the HPToolset editor.

When a source file is compiled it is translated from a higher-level language, like COBOL II, to a lower-level language of machine instructions. Source files are compiled into a user subprogram library (USL) file. MPE's segmenter is called by HPToolset when the USL file is to be prepared into a program file. The segmenter binds all the needed routines to the program so they can be found by the system when the program is executed. Each workspace has
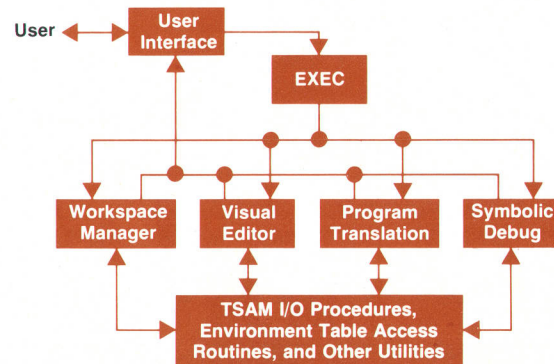


**Fig. 1.** *HPToolset program structure. All activities are managed by the EXEC. TSAM is the HPToolset file access method, which allows multiple versions of a file to reside within the same file.*

one USL and one program file associated with it. Multiple source files can be compiled into a single USL file.

Message files are used for communication between processes. They act as a first-in-first-out queue of records with one process writing and the other reading. If the queue is empty, the reader process is put into a wait state until a new record is written. When a record becomes available, the read is satisfied and the record is deleted from the queue.

**The HPToolset EXEC**

Fig. 1 shows the basic components of HPToolset. All activities that occur within HPToolset are managed by the EXEC. It interprets the user input, determines if it is valid for the current context, and calls the appropriate processor to act upon it. By having all of the HPToolset utilities operate under the umbrella of a single command interpreter, a consistency is achieved which has not existed between the unrelated tools previously provided on the HP 3000. The user need learn only one interface and time is saved by not having to exit from one utility before entering another.

The EXEC also makes it possible for the programming tools to work together. It allows several functions to be active at one time. For instance, a compiler listing can be displayed at a debug breakpoint, or one file can be read while another is being edited. This relationship between the tools is maintained by means of an environment table. This table includes information about all current HPToolset activities. It is used by the EXEC to ensure that commands are not executed in an unsuitable environment. It also makes it possible for the user to see a list of all activities related to the current workspace by issuing a single command.

Because the EXEC can determine what other functions are currently being performed, it can also provide context-sensitive defaults. For instance, if the Edit function key is pressed while a program is being debugged, the user will not be prompted for an edit file name. Instead, the source file related to the program being debugged will automatically be displayed such that the first source line on the screen corresponds to the current breakpoint location. Also, when the HELP key is pressed, it will display the HELP screen most appropriate for the current context. If the error has just occurred, it will give further explanation of the message and/or a corrective action. Otherwise, an overview of the most recent function is provided.

Even though the environment table resides in the global variable area, it is not accessed directly by any of the HPToolset command processors. Special-purpose routines were written to retrieve and update information in the table. By requiring all processors to access the table through these routines, control is centralized and a mechanism is provided to ease future debugging of the product.

**TSAM**

The HPToolset access method, TSAM, provides a mechanism for multiple versions of a file to reside within the same file instead of having to define a different file for each set of changes.

During the program development cycle each source file goes through many changes. These changes are typically cycled through compile, prepare, and execution iteratively until the programmer is satisfied that the changes are stable. At this point the programmer may wish to freeze the source so that others can share it. Before HPToolset, to make further code modifications the programmer would have to copy the contents of the frozen source to another file so modifications could be made to it while the team members accessed the frozen source file. With HPToolset, this function can be done by establishing a new version of the file by an HPToolset command and designating the frozen source as the reference version, that is, the version that others will access by default.

For example, let us assume that a file goes through the following changes:

| Version 1 | Version 2 | Version 3 |
|-----------|-----------|-----------|
| a | a′ | a′ |
| b | - | b′ |
| c | c | c |
| - | d | d′ |

Here a, b, c, and d represent COBOL II source lines. Line a was changed in version 2 but not in version 3. Line b was
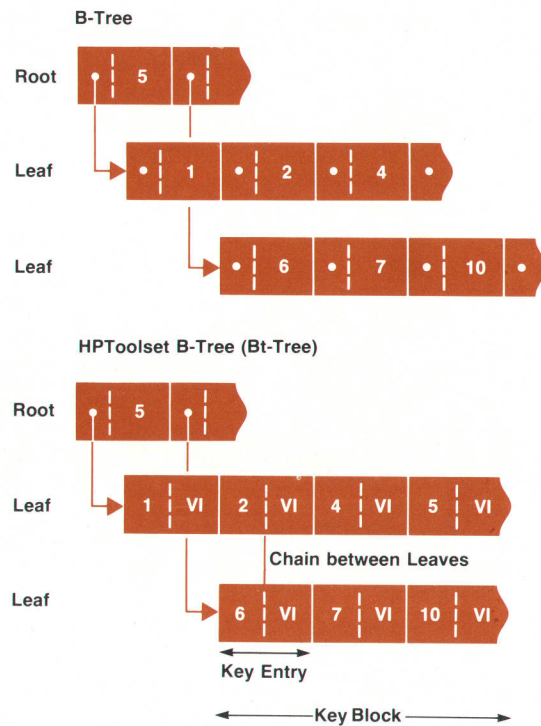


**Fig. 2.** *HPToolset supports both random (keyed) and sequential file access. 12-byte alphanumeric keys are organized in a modified B-tree structure called a Bt-tree. The trees shown here have only two levels, but in general there are many intermediate levels between the root and leaf levels. At each level are one or more key blocks containing several key entries that include pointers to lower levels. Blocks at higher levels are considered parent blocks of the lower-level blocks they point to. In the HPToolset Bt-tree all keys are represented at the leaf level to aid sequential access, and each leaf key entry contains variable-length version information (VI), including pointers to the data associated with the key value.*

deleted in version 2 and added again in version 3. Line c was left unchanged. Line d was first introduced in version 2 and modified in version 3.

Before HPToolset, these versions had to be saved in three separate files. HPToolset saves the changes associated with each version of the file and at retrieval time returns all the records relevant to the version being accessed. Logically, therefore, the user sees each version as a separate file, but physically they are all in the same file. This saves disc space and eliminates the bookkeeping of associating changes with file names.

When a TSAM file is created it has one version—version 1. This is also the latest version. After the development cycle referred to earlier, the programmer freezes version 1. The latest version is automatically updated to version 2. Only the latest version, version 2 in this case, can be modified. The programmer can set version 1 as the reference version so that other users will access version 1 by default. After making further changes to version 2, the programmer may wish to freeze it. Version 3 is now the latest and is the only one that can be modified. The programmer may choose to have version 1 continue to be the reference version or move the reference to version 2. While the latest version is being modified other versions of the file may be read concurrently. HPToolset also provides commands to purge one or more versions of a file, to label the versions with comment strings, and to list the changes between versions.

TSAM supports both random (keyed) and sequential access. Data records are stripped of trailing blanks before storage. The keys are 12-byte ASCII alphanumeric keys and are organized in a modified B-tree structure.[2] For purposes of this discussion we will refer to this modified structure as a Bt-tree. The B-tree structure was modified primarily to:

- Optimize sequential access to the file without sacrificing random access performance. This is because the majority of the accesses made by the HPToolset editor are sequential. This is especially true in visual edit mode where blocks of data are read for screen display. The Bt-tree structure attempts to minimize tree traversals, thereby minimizing key block swapping. This is important when constraints do not allow multiple key blocks to be memory-resident concurrently.
- Keep version-related information with the key indexes.

Fig. 2 shows the B-tree modifications. In Fig. 2, the numbers symbolize key values and VI represents variable-length

version information, including pointers to the data associated with the key values. The major differences between the B-tree and Bt-tree structures are:

- Each leaf entry in the Bt structure consists of the key and version information, which is of variable length. In the B-tree each key entry has a fixed length.
- Leaf key blocks are chained forward and backward in the Bt-tree.
- In a B-tree both leaf and nonleaf entries contain pointers to the next lower level. Even though these are unused in the leaf (since the leaf is the lowest level), space is reserved for them. This is avoided in the Bt-tree, saving wasted space.
- In the Bt-tree all keys are represented at the leaf level. This is to provide for sequential traversal without having to read in the parent blocks. Although this introduces data redundancy, it improves sequential access time, especially if multiple key blocks cannot be held concurrently in memory.

**Workspace**

A workspace is a TSAM file with one version. Each data record is 80 bytes long. The workspace is a directory of the files that pertain to the programmer's development environment.

A file is considered owned by a workspace if it is created via the HPToolset editor, converted from ASCII or KSAM, or copied from another workspace while the programmer is associated with this workspace.

A file is considered used if the programmer wishes to share it without making a copy of it. In using a file the programmer will access the reference version of that file by default, but can explicity override the default and access other versions.

A workspace contains information about owned and used files, such as the total number of active versions in the file, the reference version of the file, the source language (currently COBOL II or "other"), and whether it is a main program or a subprogram. In addition, the names of the USL file, the program file, and on-line listing files are maintained in the workspace. The programmer can display this information via the Show Files and Program commands. Fig. 3 and 6 show these displays.

Information in the workspace is partitioned by key ranges for easy access. Workspace control information is in a range

```
 REFRESH   NEXTPAGE  PREVPAGE    CMDWIN   ---->>    Edit              Read
  HELP                           ALTSET   ---->>    SETV            END MENU  ALTSET
```

```
                                                            Used Files    1
                                                          * File Eqns      1
              WORKSPACE: PROJECT1.CAROLYN.TOOLS             Owned Files    3

 =========================================================================
       Filename          Tot Active   Latest    Reference   Owner
                           Vrsns       Vrsn       Vrsn       Wksp
 [ ] ZENDSRC.CAROLYN.TOOLS                                   ZEND      USE
 [ ] ZENDSUB.CAROLYN.TOOLS#2                                 ZEND      USE  *
 [ ] PROJMAIN              2            3          2                   OWN
 [ ] PROJSUB1              4            4          3                   OWN
 [ ] PROJSUB2              3            3          3                   OWN
 [ ] PROJECTU                                                         USL
 [ ] PROJECTP                                                         PROG
 [ ]
 [ ]
 [ ]
```

**Fig. 3.** *The* Show Files *command displays information about the files owned and used by a workspace.*

from 000001000000 to 000002000000. Used file information lies between 000002000000 and 000003000000, owned file information lies between 000003000000 and 000004000000 and so on.

When HPToolset executes a command that opens a source file, it reads the workspace to determine if the file is used or owned. If it is owned, the latest version of the file is accessed by default. If it is used, the reference version is accessed by default. During compilation and execution, the workspace is read to obtain the names of the USL and program files.

### User Interface

The HPToolset user interface is designed to be friendly and yet provide the flexibility required to support multiple utilities. An effort was made to make the commands natural and to prompt the user for omitted parameters rather than giving a syntax error and requiring that the entire command be retyped.

The terminal's function keys (also called softkeys) are programmed to correspond with most of the commands. The function keys are designed so that there are always two sets available, with eight keys in each set. One set, known as the permanent set, always remains the same. It contains functions that are relevant in most HPToolset environments. The key functions in the other set change as the environment changes so that the most useful set is present. One key in every set is reserved for switching back and forth between the permanent and context-sensitive softkey sets.

The function key labels for at least one set are always displayed on the terminal screen. On HP 262x-series terminals, the labels for the active set are displayed in the built-in template at the bottom of the screen. Both the permanent and context-sensitive softkey sets are displayed side by side at the top of the screen on HP 264x terminals, as shown in Fig. 4. Arrows between the sets indicate which is active. The labels are displayed in a highlighted template, which is protected by memory lock. To change the labels, memory lock is turned off, the cursor is moved to the top of the screen to write the new labels, memory lock is turned back on, and the cursor is returned to its original position so that no other contents of the screen are changed.

VPLUS/3000 was used to implement the menus for HPToolset. Instead of having the user press the **ENTER** key to transmit input, the VPLUS/3000 autoread option is used. Appropriately labeled function keys are provided with all menus, and when one of them is pressed, HPToolset does the autoread to pick up the terminal input before performing the desired function. This provides a menu interface consistent with the user interface for the rest of the product.

### Editor

HPToolset includes a visual editor. It displays a portion of the edit file on the screen. This can be edited directly with the terminal's cursor movement and editing keys. The terminal is strapped so that all cursor actions are transmitted to HPToolset. The movements are recorded into a buffer which is evaluated when the return key or any function key is pressed, or when it is full. All changes recorded in the buffer are written to the edit file with each buffer evaluation to ensure data integrity if a system crash should occur during editing. An UNDO command is provided to allow the user to cancel the last changes written to the file.

Function keys are also used in the visual editor to implement movement from one location to another in the edit file and to facilitate editing functions, such as MOVE and COPY, that cannot be done with the terminal editing keys. One of the keys, labeled MARK, is used to select lines to be moved or copied and to select the destination. Once the lines have been selected, the operation can be completed by simply pressing the MOVE or COPY function key.

Since a language is associated with all TSAM source files, the editor is able to set editing options to the values most appropriate for each file. The HPToolset editor sets tabs and margins differently, depending on the language associated with the file. Of course, any of the options set by HPToolset can be overridden by the user.

Fig. 5 shows a typical visual edit screen.

### Program Translation and Execution

As mentioned earlier, after a source file is created, a number of translation steps are needed before it reaches a stable state. The steps are compiling, preparing, executing, and debugging.

When a workspace is entered, HPToolset provides default names for USL and program files. The programmer can specify different names and the options to be used while preparing the USL file. By issuing the Program command or pressing the function key, the programmer can display compile-related information on the source files in the workspace. This is shown in Fig. 6. The programmer can now mark the file(s) to be compiled and press the Compile function key. HPToolset maps this into one or more calls to the COBOL II compiler.

The compile process is created in background mode so that HPToolset is active while the compiler is executing. HPToolset sets up file equations with the formal names that COBOL II expects for the source, USL, listing, and workspace files. Three message files are used to communicate information during compilation: let's call them MF1, MF2, and MF3. The compiler directs its listing to MF1 and compilation statistics such as number of errors and compilation time to MF2.

If the file being compiled is owned by the workspace, an on-line listing is generated. If not, the listing is sent to the printer. HPToolset reads MF1 and transfers data to a TSAM

| REFRESH | NEXTPAGE | PREVPAGE | CMDWIN | ---->> | Edit | Wrkspc | Read | ShowFils |
| HELP | | | ALTSET | ---->> | Program | | EXIT | ALTSET |

HPToolset HP32350A.00.00 - SL A.00.00 (C) Hewlett-Packard Co. 1982

>>

**Fig. 4.** *HPToolset provides two eight-key groups of function keys (softkeys), a permanent set (left) and a variable set that changes with the environment. On terminals capable of displaying both sets, arrows indicate which set is active.*

```
 1     $CONTROL SUBPROGRAM, SYMDEBUG
 2       IDENTIFICATION DIVISION.
 3       PROGRAM-ID.
 4           COMPUTE-SUB.
 5       AUTHOR.
 6           JOE PROGRAMMER.
 7       DATE-WRITTEN.
 8           5-27-81.
 8.01
 8.1     ENVIRONMENT DIVISION.
 8.2
 9       DATA DIVISION.
10       LINKAGE SECTION.
12       77  WEEKLY-SALARY      PIC  999.
13       77  AMOUNT-OF-SALES    PIC  9999.
14       77  TOTAL-EARNINGS     PIC  9999V99.
14.1
16       PROCEDURE DIVISION
17           USING WEEKLY-SALARY AMOUNT-OF-SALES TOTAL-EARNINGS.
23.2     SUB-PARAGRAPH.
```

**Fig. 5.** A typical HPToolset visual edit screen.

file until there is no data left to read and there is no writer associated with MF1. This signals completion of the compiler's listing activity. The programmer can scroll through an on-line listing interactively, mark a line or an error if any, and have HPToolset display the corresponding line in the source file. Pressing the PRINT key produces a hard copy of the listing.

Completion of compilation is signaled by MF2's having data in it and no write activity. HPToolset informs the programmer that the compilation is completed, via the user interface routines. It now checks to see if another file is to be compiled (i.e., more than one file was marked when the user pressed Compile). If so, information on this compile is passed via MF3. Compiler creation and the subsequent sequence of events is repeated.

After compilation the next step is to prepare the USL file into a program file. This is done by pressing the PREP key. Any prepare options specified in the workspace are now honored. Alternatively the user can type in the PREP command and specify the options. HPToolset now creates a temporary file and writes the MPE segmenter commands necessary to perform the PREP function to it. It then invokes the segmenter and directs it to get its input from the file.

The final step is to execute the program file. This is done by pressing the RUN key. The program executes in foreground mode. If symbolic debug was not requested, HFToolset is inactive until program execution is completed. If symbolic debug was requested, the program pauses at the first executable statement and control is returned to the programmer so that execution can be monitored.

The programmer can combine the compile, prepare, and execute steps by pressing the GO function key. To do this the programmer first issues the Program command to display the source files in the current workspace, then marks the files and presses the GO key. The file(s) are compiled, prepared into the workspace-defined program file, and executed.

## Symbolic Debug

The debugger in HPToolset allows programmers to set breakpoints, trace program flow, display variable values, and change the values of variables by using the symbolic names used in the program source. Debugging time is saved by eliminating the need to determine the octal addresses of program locations or data items before accessing them in the debugger.

To implement this, it was necessary to save additional information in the USL (object) and program files. The COBOL II compiler was modified to store the symbol table information into the USL file. Three new USL header types were added to hold this information, two to hold the PMAP information* and one for all other symbolic information such as the addresses associated with data items and program statement numbers.

The compiler also inserts a PCAL (procedure call) instruction at the beginning and end of the main program and each subprogram, and at the beginning of each section and paragraph. These instructions pass control to SDB'SPY,

*PMAP = program map, a list of procedures within a program.

```
SOURCE FILES for Workspace PROJECT1.CAROLYN.TOOLS

     PROGRAM ID                 LANG    TYPE    SOURCE FILE      LISTFILE

[ ]  SALARY-PROG                COBOL   MAIN    PROJMAIN         PROJ1687
[ ]  COMPUTE-SUB1               COBOL   SUBP    PROJSUB1         PROJ1612
[ ]  COMPUTE-SUB2               COBOL   SUBP    PROJSUB2         PROJ1632
[ ]
[ ]
[ ]
```

**Fig. 6.** The Program command displays compile-related information about a workspace's source files. To compile a program file, the programmer simply types any character in the box next to the program ID and presses the Compile key.

which is a procedure that acts as the communication link between the HPToolset debugger and the user program. The PCALs are used by SDB'SPY for tracing program flow.

During the linking process, the symbol table information is passed from the USL file to the program file. The segmenter creates a son process, called SEGSYM, which takes the information from the USL file, sorts and reformats it, and adds it to the program file. Several new MPE intrinsics (routines) can be called by HPToolset to access this information for converting symbolic names to the corresponding octal addresses.

Of course, it is not necessary to include this extra information in program files that are not going to be run with symbolic debug. By default it will not be added to the USL file by the compiler. A $CONTROL SYMDEBUG statement must be included at the beginning of the source files for which the symbolic debug information is desired. The segmenter will automatically pass it from the USL file to the program file unless the NOSYM parameter is included in the PREP command. Including the NOSYM parameter will also cause all of the inserted PCALs to be converted to NOP instructions.

As soon as HPToolset creates a user process to be run with symbolic debug, it does a handshake with SDB'SPY. Communication between HPToolset and SDB'SPY is executed via three message files. HPToolset sends the names of the message files in which it will write and receive messages (MF2 and MF3) to SDB'SPY in MF1. It then initiates a read from MF3 and waits for a reply. Because of the inserted PCAL instruction, SDB'SPY gets control as soon as the user program is created. SDB'SPY returns the handshake by writing a message to MF3 to inform HPToolset that it is ready to receive a debug command.
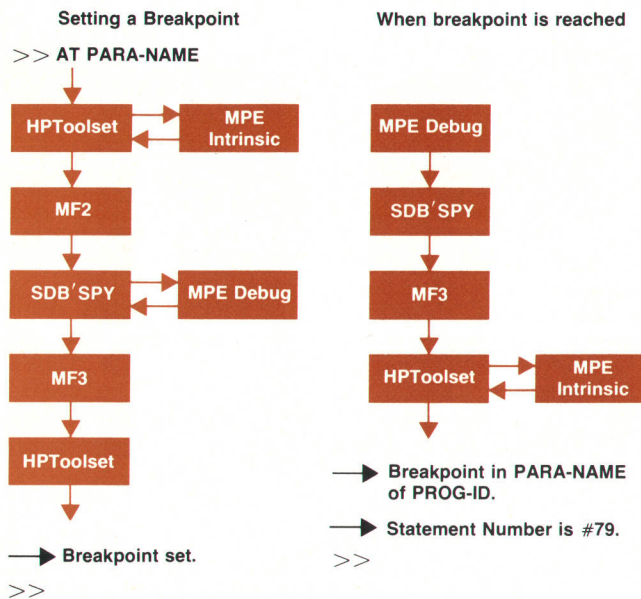
When the user enters a symbolic debug command to set or clear a breakpoint, HPToolset determines the PMAP name and offset for the symbolic location or statement number and passes them to one of the new MPE intrinsics. The intrinsic converts them to the corresponding octal address. HPToolset then passes the address to SDB'SPY through MF2. SDB'SPY calls the MPE debug utility programmatically to set or clear the breakpoint. This capability was added to MPE debug by the addition of two intrinsics, SET'BREAKPNT and CLEAR'BREAKPNT. When the operation has been done, SDB'SPY returns the status to HPToolset in MF3. Fig. 7 diagrams this flow of events.

When the user resumes program execution, HPToolset sends a message to SDB'SPY and then initiates a read from MF3. This puts HPToolset in a wait state until SDB'SPY satisfies the read. One of the parameters passed to the SET'BREAKPNT intrinsic is an address in SDB'SPY where control will be passed when the breakpoint is reached. When SDB'SPY gets control at a breakpoint, it writes the program location to MF3 to satisfy the read which was issued by HPToolset. HPToolset then converts the octal breakpoint location to the symbolic equivalent, displays it, and prompts the user for a command.

When a DISPLAY or MOVE command is entered to display or change the value of a symbolic data item, the symbolic name is converted to an address in the data stack and a length (in words). The request is then passed to SDB'SPY through MF2. SDB'SPY can directly access the program data stack since it is called by the user program. It returns the contents of the requested locations to HPToolset where it is formatted and displayed.

### Context-Sensitive HELP

As was mentioned earlier, HPToolset has a context sensitive HELP facility. This facility provides an overview of each of the utilities, specific text about each of the commands, and further explanation of error messages. The command descriptions are contained in a TSAM file, keyed on the internal command numbers which are used by the command interpreter. By using the same command numbers as are used by the command interpreter, the HELP facility can share the procedures for converting command names to numbers, and can easily find the correct syntax for the command just entered by picking up the current command number from the command interpreter. This is used if HELP is requested immediately after receiving a syntax error.

When a nonsyntax error is displayed, the error number is saved. The HELP facility uses that number to locate the error explanation and corrective action. This information is obtained from the error catalog where it has been included as comment lines immediately after the error text. After MPE's GENMESSAGE intrinsic has been called to locate the error message in the catalog, the file pointer is left in a position that makes it easy for the HELP facility to find the error and read the comment lines following it. The result might look like:

```
>>ADD 10
***No file currently open for editing. (85)
>>HELP
```



**Fig. 7.** *Intrinsics (routines) were added to the HP 3000 operating system, MPE, to facilitate converting between symbolic names and octal addresses. Intrinsics were also added to call the MPE debug facility programmatically for setting and clearing breakpoints. MF2 and MF3 are message files used by HPToolset.*

Cause: A command was entered which is only valid while an Edit file is open, and there was not one open.

Action: Do an Edit command before reentering the command.

>>

## Technical Summary

The HPToolset subsystem is written entirely in SPL, the HP 3000 systems programming language. It is currently made up of 260 source files which contain a total of 130,000 lines of code. HPToolset is contained in two program files named TOOLSET and TSETUTIL. TOOLSET is the primary program file and contains 37 segments. TSETUTIL contains one segment and is created as a separate process by TOOLSET to implement background compiles and other special functions.

In addition, two new segments were added to the system segmented library to support HPToolset. They contain the I/O procedures for accessing TSAM files and SDB'SPY.

Several existing products were enhanced to support HPToolset. The COBOL II compiler, the MPE segmenter, and the MPE program loader were all changed to handle the symbol table and PMAP information in the USL and program files. The COBOL II compiler was given the ability to read TSAM source files and workspace files. Also, intrinsics were added to MPE for programmatically calling the MPE debugger to set and clear breakpoints and to facilitate the conversion of symbolic locations to the corresponding octal addresses.

## Acknowledgments

HPToolset is the result of the hard work and cooperation of a dedicated team of engineers. The EXEC and editor were designed and implemented by Linda Lawson. Pat Miyamoto developed the user interface. The program translation interfaces were developed by Lynn Smith. Barbara Packard designed the symbolic debugger which was implemented by her and Alan Padula. Important contributions were also made during the final stages of the project by Paul Chan and Greg Gloss.

## References

1. HPToolset Reference Manual (Hewlett-Packard Company, 1982).
2. R. Bayer and C. McCreight, "Organization and Maintenance of Large Ordered Indexes," Acta Informatica, Springer Verlag, 1972, pp. 173-189.

### Carolyn M. Bircher

Carolyn Bircher joined HP in 1979 after receiving her BS degree in computer science from California Polytechnic State University, San Luis Obispo. She's done current product engineering for EDIT/3000 and FCOPY, two HP 3000 software subsystems, and she developed and is doing current product engineering for HPToolset. Born in Fresno, California, she is married and now lives in Sunnyvale, California. Her interests include square dancing and church activities.

### Anil K. Shenoy
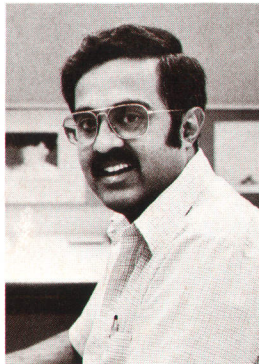
Anil Shenoy holds an MS degree in Computer Science from the University of California at Berkeley. With HP since 1979, he helped develop HPToolset and is now a software project manager. Before coming to HP he was involved with data base software development and project management for Control Data Corporation and operating system development for Xerox Corporation. Born in Mangalore, India, Anil is married, has a daughter, lives in Los Altos, California and enjoys swimming and badminton.