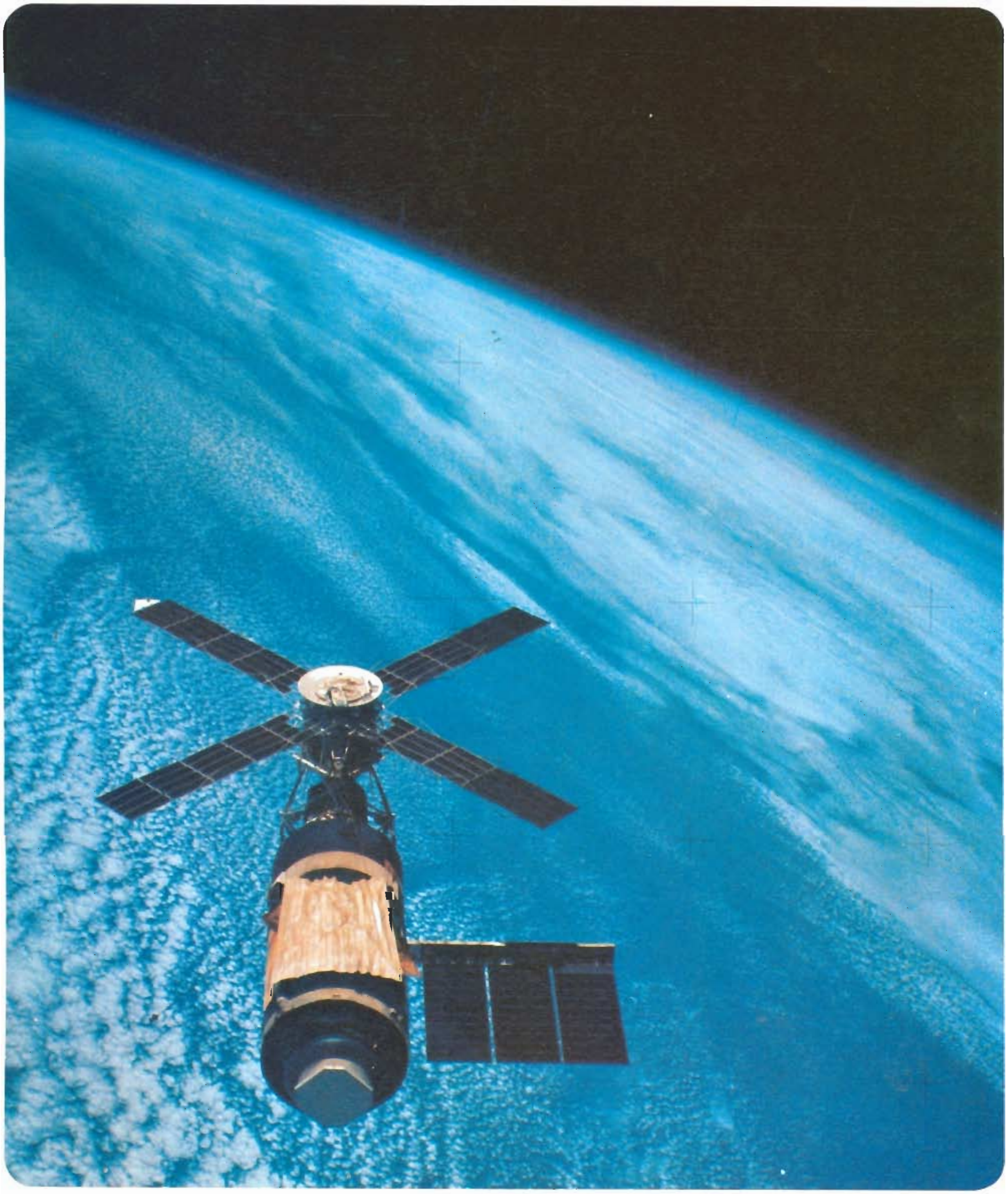
Keyboard

Jan-Feb/79 A Publication of Hewlett-Packard Desktop Computer Division



Keyboard

January-February 1979

Notice

All Keyboard subscriptions expire with this issue — but don't panic. In the issue is a subscription renewal card. Please fill it out COMPLETELY or we will not be able to process it. Just drop the completed detachable postcard into the mail, and you will continue to receive Keyboard. If you are an occasional Keyboard reader and would like to have your own subscription, or if someone else has already taken the card out of this issue, call your nearest HP sales office, or contact us: Keyboard, Hewlett-Packard, 3404 East Harmony Road, Fort Collins, CO USA 80525; or call us at 303/226-3800.

Computer Museum

Desktop aids Skylab effort

by Ray Reynolds

An object in stable orbit exhibits equilibrium between forces that would send the object away from the larger body and gravitational forces pulling the object toward the body. This is the case with the orbit of the Earth around the sun. Our orbit around Sol will be relatively stable for probably billions of years to come.

Orbits of some objects around the Earth however, are not so stable. Man-made objects placed into orbit during the past 20 years often have been low enough that collisions with molecules in the upper reaches of our atmosphere gradually have caused their orbits to deteriorate.

This is the case with *Skylab*, the experimental space station launched into space May 14, 1973, and used by NASA in a series of endurance experiments undertaken by three different manned missions extending through 1973 and ending February 8, 1974. Immediately after the third manned mission, *Skylab* was powered down and placed in a completely dormant mode. A Hewlett-Packard Desktop Computer now is being used at the Johnson Space Flight Center, Houston, Texas, to help maintain *Skylab's* attitude and preserve a stable orbit as long as possible.

Dormant For Four Years

For more than four earth-years, a silent *Skylab* skimmed the fringes of our world, touching whisps of air molecules. Without its solar panels constantly aimed at the sun, power drained from its batteries. In the absence of power to drive them, its stabilizing gyros gradually slowed to a halt. Without its gyros, *Skylab* began to cone and tumble, increasing the friction with the bits of atmosphere it encountered. Increased sun spots during this period raised the temperature of the Earth's atmosphere, extending the atmosphere outward and increasing the drag on *Skylab* beyond what had been expected, slowing the craft's speed, and giving gravity the upper hand.

As the altitude of *Skylab's* orbit began to decrease, the North American Air Defense Command, the Swiss Astronomical Observatory, and other entities informed NASA of the changes in the orbit of *Skylab*. The space agency, once it realized that *Skylab* would not stay in orbit for the originally planned 10 years without assistance, reevaluated its plans. It was when NASA decided to reactivate *Skylab* and change its orientation to reduce the drag that I became involved.

By activating *Skylab's* gyros and using its thrusters, the coning and tumbling could be stopped, and the small end of the spacecraft could be pointed forward, all reducing drag. By reducing drag, the chances of being able to reach *Skylab* before it could fall into the atmosphere and burn would be increased. NASA's plan back in 1974 had been to wait for the space shuttle and either boost *Skylab* into a safe higher orbit, or pilot it down into the atmosphere so it could burn up without endangering human life with debris. So the plans did not change, but they were put into effect sooner than had been expected.

Computer Needed

It became obvious to NASA that a computer system would be needed to convert data on *Skylab's* position and velocity into a form that would be usable by the *Skylab* onboard computer. I usually work with large computers developing flight design systems for the space shuttle. I have worked with Hewlett-Packard 9825A Desktop Computers off and on for mission programming. We use them fairly extensively.

Reactivation signals were sent to *Skylab* in March of last year after NASA determined that the orbital life should be extended as long as possible. The onboard computer was turned on, and the battery system began charging itself up from March through May.

Navigational Parameters

Part of the *Skylab* reactivation process required generating parameters which are required by the *Skylab* onboard computer for initialization and maintenance of the orbital navigation program. Satellite tracking data is used to determine *Skylab* position and velocity information at some time in the past. The position and velocity is then mathematically propagated to a time in the future when the *Skylab* computer is to receive the data.

The predicted position and velocity data is then converted into a special set of parameters required by the *Skylab* computer. That set of parameters is then transmitted to *Skylab*. During the original *Skylab* missions, all these functions were performed in the Real-Time Computer Complex (RTCC) which also performed all the telemetry and trajectory processing. However, for various reasons the RTCC was not available for the *Skylab* reactivation.

We also didn't have access to the old programs that were used during the previous *Skylab* missions. Those were used on the large-scale computer system which is not available to us. When this application came up, we had some programs for the 9825 Desktop Computer that were similar to what was needed for *Skylab*, and we redesigned them to fit this application. We are using a 9872A Plotter and a 9866A Printer with the 9825 Desktop Computer.

During the *Skylab* reactivation, the satellite tracking was performed by the NORAD tracking facilities. The mathematical propagation was performed by Marshall Space Flight Center. Programs developed on the 9825 Desktop Computer at Johnson Space Center were used to convert the predicted position and velocity to the parameters compatible with the *Skylab* computer formats.

Data Conversion Programs

The first program converted the predicted position and velocity supplied by Marshall Space Flight Center to that needed by the 9825 trajectory program. Most of this capability was contained in an existing program, so minor changes were made to that program.

The second program generated the update parameters themselves. The input parameters required are the time (Greenwich Mean Time) that the parameters would be valid and

Rocky D. Duncan, aerospace engineer, assists in running the *Skylab* programs. Use of the programs is helping to maintain control over *Skylab* until the space shuttle can reach it.

information on *Skylab's* current estimate of the trajectory. The *Skylab* parameters are the current estimated time of orbital midnight (or when *Skylab* is farthest from the sun) and the orbital midnight period (time between successive midnights). There is also an option of one of four different formats in which to generate the uplink octal. A sample run is provided below.

A definition of the uplink parameters and their units are as follows:

Lamda-yo (pirads) — Right ascension of the orbital ascending node.

TA (seconds) — Time between two ascending node crossings.

Lamda-z (pirads) — Orbital inclination. Angle between the orbit plane and the equatorial plane.

```
    GMTU   156   5 30  0.00
    ATMZS  155   3 17 57.00
    TMID     1   2  5 15.00
    DMID     0   1 32 27.25

UPLINK FORMAT= A-T
```

No	Parameter	Value	Data Octal	Uplink Octal
1	Lamda-yo	9.4360040E-01	50743 50100	766206174300 766206024000
2	TA	5.5303903E 03	50531 50520	766206127100 766206126000
3	Lamda-z	2.7797209E-01	50216 50250	766206045600 766206055000
4	Rev	9.4220707E-01	50742 50320	766206174200 766206066000
5	DTM	6.0873191E 00	50000 50300	766206004000 766206064000
6	Omega-o	3.6123139E-04	50572 50620	766206137200 766206146000
7	Mo	8.3738139E-01	50654 50570	766206155400 766206137000
8	Eta-yp	-6.2244902E-01	51301 50240	766206264100 766206054000
9	Lamda-ygo	-9.2191202E-01	51047 50770	766206214700 766206177000
10	Lamda-dot	-3.3451527E-07	51230 50640	766206247000 766206154000
11	DTAUm	3.3491050E-01	50000 50010	766206004000 762206005000

Rev (no units) — Ratio of the radius of the Earth to the semi-major axis of the *Skylab* orbit.

DTM (seconds) — Desired change to make to the *Skylab* time of orbital midnight.

Omega-o (pirads/second) — Average orbital angular velocity.

Mo (pirads) — Mean anomaly of the sun.

Eta-yp (pirads) — Position of the sun measured in the orbit plane from the previous ascending node.

Lamda-ygo (pirads) — Geographic longitude of the ascending node.

Lamda-dot (pirads/second) — Regression rate of the ascending node.

DTAUM (seconds) — Desired change to make to the *Skylab* orbital midnight period.

In order to convert the engineering values to the desired format required for uplink, the engineering values were scaled and converted to octal. The data octal words contain the converted octal parameter. The uplink octal was then generated from the data octal depending on which of the four different formats were selected.

The uplink octal parameters were then manually punched on a paper tape. The parameters were sent via teletype to a remote site to be relayed to *Skylab* at the desired time.

Parameter Updates

In order for the control law, which maintained the *Skylab* altitude, to be efficient, it was desirable to keep the *Skylab* time of orbital midnight within 15 seconds of the actual time of orbital midnight. Thus, new parameters would be required as new tracking information indicated changes to the orbital parameters. During the early part of the reactivation, new parameters were sent almost daily. However, since the orbit was stabilized, new parameters have been required only approximately twice a week.

Use of the system aided the modification of *Skylab's* orientation during June 1978, but power control problems developed in July, partially due to the condition of the *Skylab* gyros. One of the gyros had ceased working during the original *Skylab* missions in 1973. One of the other gyros worked well, and the third limped along.

Late last July, further corrections were made in the orientation, and *Skylab* settled into a stable orbit and orientation designed to maximize the life of its gradually sinking orbit and buy time for a shuttle mission. Both remaining gyros began working well, and the craft has not developed any problems since.

Skylab Monitored 24 Hours Per Day

Skylab is monitored nearly 24 hours per day with the antennas of the Deep Space Tracking Network, which includes antennas around the world. A NASA team works day and night to monitor condition of the craft.

The objective of the efforts to stabilize *Skylab* is to buy time until the space shuttle can intercept *Skylab* and attach a remotely-controlled rocket motor to it. The first orbital flight of the space shuttle is not expected before September of this year. Because it will be the first flight of the shuttle, it is very unlikely that NASA would risk a mission to *Skylab* at that time. The second flight of the space shuttle into orbit is expected in February of 1980.

Remotely-Controlled Booster

NASA has committed itself to trying to reach the *Skylab*, probably on that second space shuttle flight. A reusable, remotely-controlled booster rocket motor called the Teleoperator Retrieval System would be released from the payload bay doors of the shuttle. From the shuttle, crew members would pilot the TRS to a docking with *Skylab*. The system then could be used either to push *Skylab* into a higher orbit to spare it at least temporarily from the atmosphere, or it could be used to push the *Skylab* into a carefully-computed burnup in the atmosphere designed to prevent any damage on earth.

NASA will decide which way to go in the next month or so. In the meantime, however, it will be the job of the tracking stations, the 9825 Desktop Computer, and NASA workers to preserve *Skylab's* orbit, so that the craft will be there when the space shuttle arrives more than a year from now.

Editor's Note: As Keyboard went to press, NASA announced that plans to rendezvous with Skylab using the space shuttle would be abandoned. The planned mission to attach the Teleoperator Retrieval System to Skylab also was shelved. At least for the present, NASA will maintain the low-drag orbit of Skylab, so the work of the desktop computer will continue. Skylab is expected to fall into the Earth's atmosphere and burn in the next six to eighteen months. NASA attributed the decision to delays in the production of the TRS, uncertainty about whether the shuttle could be made ready for the rendezvous mission, and deterioration in Skylab's systems. According to NASA, the likelihood of a successful mission had fallen to less than 1 in 10.

Ray Reynolds received a BS degree in mathematics from Arkansas Polytechnic College in 1963, and an MA in mathematics from the University of Oklahoma in 1965. He has been at the Johnson Space Center since 1966. Ray received a number of awards for his work in support of the Apollo and Skylab missions, which included development of 9820 programs for Skylab attitude and experiment pointing requirements.

Ray Reynolds
Mail Code F76
Johnson Space Center
Houston, TX 77048
USA

New Products

Results of computerized tests on circuit boards reel out of HP's new stand-alone 9876A Thermal Graphics Printer at speeds up to 480 lines per minute. The printer can be dedicated to a single purpose as shown, or can be linked with a desktop and used for many different applications.

9876A Thermal Graphics Printer

by Bill Boles, Hewlett-Packard Company, Desktop Computer Division

Introduction of the 9876A Thermal Graphics Printer brings speed and performance to Hewlett-Packard's product line in the form of a stand-alone peripheral. While the 9876 was designed primarily with the System 35 in mind, it can be used with almost any desktop computer. It provides high speed (480 lines per minute) full width printing, (80 columns) and graphics for most needs.

The 9876 is truly an international product. Seven international character sets (US ASCII, German, Spanish/Italian, French, Swedish/Finnish, Danish/Norwegian and Japanese/Katakana) are all resident in the product. Each set is easily accessible under program control. In addition, the user can define up to six separate special characters of his own choosing and design.

The 9876 also has features such as overbar and underbar for highlighting purposes as well as character sizes of 150% of the normal size. Typewriter functions such as backspace and tabbing are provided in the 9876. A choice of two interfaces (HP-IB and 8-bit parallel) is available.

Paper for the printer comes in a flat package which contains 330 sheets of pre-perforated fan-fold thermal paper. Both Metric and English-sized paper are available in either blue or a new high contrast, fade resistant black. The pre-perforated fan-fold paper means an end to the annoying curl problem encountered with roll paper. The top of form sensor stops the paper with the perforation aligned with the tear bar for clean and easy cut-offs.

High Resolution

The 9876 is more than just a thermal printer. It is a true high resolution graphics output device which can reproduce raster type CRT displays up to 560 dots wide. The monolithic print head uses an integrated circuit manufacturing process to provide 560 print resistors across the head.

Graphics data can be accepted in accordance with HP's new raster graphics standard. Graphics data is accepted in 8-bit bytes with each bit representing the presence or absence (1 or 0) of a dot on a CRT screen or printer. The implementation of this standard means that graphical data can be printed out on the 9876 from many graphic devices including desktop computers and HP's 2647 and 2648 graphics terminals.

When ordered with the support option for the System 35 (9876 Opt. 035) a binary program is supplied which allows the use of HP's high level graphics commands (PLOT, SCALE, AXIS and LABEL). In order to use this capability a minimum memory configuration for the System 35 of 128K bytes is required. This binary program sets up a 32K byte array in the System 35 memory which can be accessed by the plotting commands.

A program can be written to plot to this array similar to plotting to an invisible CRT screen. Once the plotting is completed, a command can be sent to output the plot to the printer. A typical plot can be output on the 9876 in less than 30 seconds.

The other DCD mainframe which is supported with plotting software is the 9825A. A utility cartridge containing programs performing function plotting, barcharts, and data plotting is provided with Option 025.

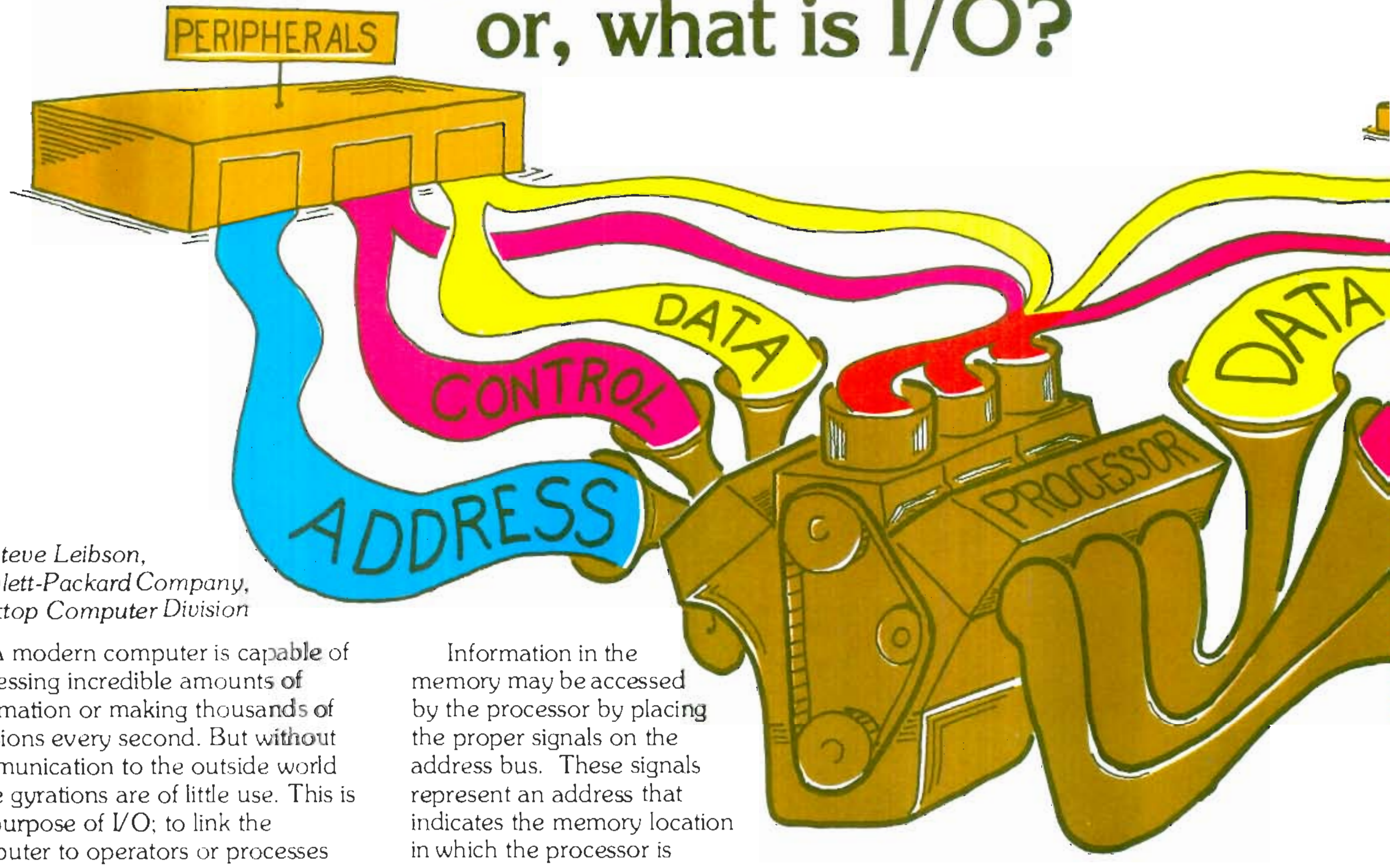
Reliability

Reliability has always been the hallmark of HP products and the 9876 is no exception. Over temperature protection for both the thermal print head and the power supply ensures that the print head and electronics cannot be damaged by circuit or component failures. The microprocessor is utilized to perform self checking.

A user self test can be initiated from the control panel. This test will print out all of the characters and highlight features of the product so that the user can determine if the printer or some other part of the system is causing the problem. Indicator lights can isolate the problem to a specific module of the printer. This is especially helpful to the service technician so that the problem can quickly be determined and the repairs easily accomplished.

The new 9876 printer offers great versatility as an output device. High speed, high resolution printing as well as graphics output make the 9876 a valuable part of any desktop computer system, graphics system or scientific computation system.

How do computers communicate, or, what is I/O?



by Steve Leibson,
Hewlett-Packard Company,
Desktop Computer Division

A modern computer is capable of processing incredible amounts of information or making thousands of decisions every second. But without communication to the outside world these gyrations are of little use. This is the purpose of I/O; to link the computer to operators or processes that require the problem-solving power provided by data processing equipment.

I/O is an abbreviation, it stands for input/output and represents communications between a computer and the world surrounding it. In order to understand the various means used to effect these communications, we are going to start at the core of the system, the computer itself, and work our way out to the rest of the world.

A general purpose computer is composed of two main components: a processor and memory. The processor is the engine of the system, following sequences of instructions which cause it to process data. Instructions and data are stored in the memory for the processor's use. The processor and the memory are linked together by three sets of lines called busses; the address bus, the data bus and the control bus. The computer memory is organized into thousands of locations, each having its own unique address and capable of storing one piece of data or one instruction in a sequence. It is the processor's job to differentiate between instructions and data.

Information in the memory may be accessed by the processor by placing the proper signals on the address bus. These signals represent an address that indicates the memory location in which the processor is interested. The processor also must signify whether it wishes to extract information into this location. This signaling is performed on the control bus. The control bus also contains signal lines to synchronize the processor and memory. In either case, the information passes between memory and the processor over the data bus, which is capable of transmitting information in either direction.

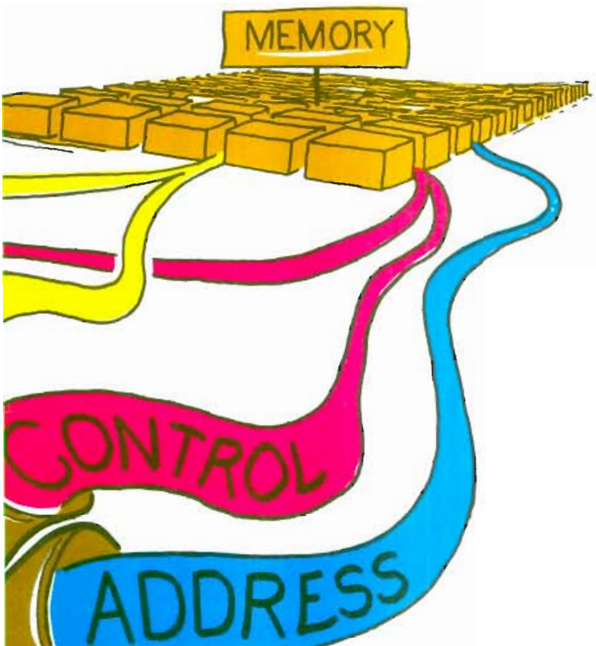
Since both data and instructions pass over the data bus, the processor must interpret the information correctly. This is achieved through timing cycles internal to the processor. In order to obtain its next instruction, the processor performs an *instruction fetch.* It then performs the operations necessary to execute the instruction.

The current location being accessed for instructions is held in a register within the processor called the *program counter.* The instruction thus obtained may cause the processor to again access memory, this time to obtain or to place data in the memory. Such operations are caused by executing *memory reference instructions.*

The computer is now able to perform the operations involved in running a program: it can obtain instructions from the memory, it can access the memory for data, process the data, and place this processed data back into the memory. A problem now arises; how do the program and the data get into the memory and how does the operator obtain the results of the processing? It is precisely this problem which is addressed by I/O.

A complete computer system, such as an HP desktop computer, is not merely composed of a processor and a memory. Peripheral devices such as a keyboard, display, printer and tape storage device are also included. These peripheral devices connect the computer to the outside world. The keyboard, display and printer allow communications with a human operator while the tape storage device provides for storing and retrieving programs.

How are these devices connected to the processor-memory combination residing inside of the computer? There are currently two methods in use. The



first is to place these devices on the memory bus already discussed. Thus the peripheral devices "appear" to the processor as memory locations. Data can be sent to or obtained from the peripherals using memory reference instructions. This configuration is called *memory-mapped I/O* because some portion of the computer memory has been allocated to peripheral devices.

The advantage of this system is that the existing processor instructions now serve the dual purpose of interfacing to memory and to I/O devices. The disadvantage is that the full range of the memory is not available for program and data storage. The maximum memory size of the computer has been reduced.

A second method of implementing I/O in a computer is to create a new bus, the I/O bus. The I/O bus is very similar to the memory bus. There is an address bus, called the peripheral address bus to differentiate it from the memory address bus, there is a second set of data lines and there is a peripheral control bus. The signals on these I/O busses may be similar to those of the memory bus or they may be very different. This system has the advantage of full memory capability at the expense of creating a new set of instructions for the processor called *I/O instructions.*

Let us briefly discuss instructions before going on. The memory reference and I/O instructions belong to a class of instructions called processor or machine instructions. This class of instructions is used for controlling the operation of the computer at the very lowest level because the instructions cause the processor to perform very simple tasks such as obtaining one piece of information from memory or dispatching one character to a peripheral device.

The typical operator of a computer would have a tremendous programming task if all problems had to be solved by writing programs at this level of complexity. Therefore the computer supplier generally provides a systems program or operating system which in effect implements a new set of instructions with far greater capability. This new set of instructions is called a *high-level language* because the instructions, now referred to as statements, allow programming on a much higher level of complexity.

Digital Signals

We have discussed briefly the sets of lines called busses and stated that the processor and other systems components send signals along these busses. That implies that these busses are metallic carriers upon which voltages may be impressed and currents caused to flow, which is correct.

The simplest signal which might be sent along such a conductor is the presence or absence of voltage or current flow. This would then be a binary signal because it may only assume two states: present or absent. In the case of a voltage related signal, the voltage is either there or it isn't: the voltage is either X volts or zero volts. Voltages are measured with reference to a zero point, usually called ground. The ground is often a heavy conductor interconnecting all components of the computer system.

Binary signals are the primary means of communications in computer systems because the circuitry required to generate and detect mere signal presence or absence is much simpler to construct than circuits concerned with "how much" signal is present. This circuit simplification allows the construction of highly complex processors because simple binary circuits require less room than other types and therefore large numbers of them may be constructed in small spaces. This is the key to the construction of large scale integrated circuits which incorporate thousands of circuits on a small chip of silicon.

Busses are simply sets of parallel conductors upon which binary signals are impressed. The most common binary signal at present is the "TTL" level set. "TTL," which stands for Transistor-Transistor Logic, is the name of a family of integrated circuits that are used as the building blocks of computer systems. These digital circuits not only define the presence or absence of voltage as proper binary levels but define regions of voltage as proper levels. These regions are:

High region = 2 volts to 5 volts
Undefined = .8 to 2 volts
Low region = 0 to .8 volts

Thus we have a hardware system for transmitting signals as long as the circuits that send and receive the signals agree on the levels to be used. As we shall see later in this series, one of the tasks of I/O is to convert levels used by one portion of a system to those used in another portion. Unfortunately, not all peripheral devices use "TTL" levels. The computer busses that we discuss will all use these levels, however.

Data Representations

Now that the signal levels have been established, an agreement must be made on what they represent. For instance what is the digital representation of an "A" or how is the number 123 represented? The alphabet from which the "A" was obtained can assume any of 26 values: "A" through "Z". Numbers may assume an infinite number of values. How can all of these values be represented with only two levels: on and off?

The answer is to use more than one signal line, to create a bus. If we were to use eight lines with each line able to assume one of two levels, then two to the eighth power or 256 values could be represented. This is sufficient to represent all of the characters in the alphabet, both upper and lower case, plus the other printed characters and punctuation marks on the typewriter, along with a few other characters.

With eight lines, values need only be assigned to each symbol to be represented, and as long as both the sender and receiver agree on what each value represents, communication can occur. Thus the second task of I/O is to assure agreement between sender and receiver, or at least to convert from one set of values to the other.

In addition, not all devices communicate on the same number of lines. Some use a single wire (plus ground) and send one binary "bit" of information at a time. The receiver assembles these sequential bits of information back into parallel representation. Some devices need only to send numerals, which may be represented with ten values requiring only four digital signal wires. Other forms of representation may require 16, 24, 32 or 64 lines making universal interconnection very difficult. The interfacing among these devices must somehow adapt one system of representation to another for communication to be accomplished.

Summary

It would simplify matters greatly if all devices could agree on data representation, format, signal levels, timings or even the number of wires to be used for interconnection. Attempts at such standardization have been made but due to the swift pace of technological development some standards are obsoleted before they are published. In addition, compromises always need to be made and different systems require different compromises. Older equipment also needs to be interconnected since the replacement of a computer should not mandate a complete system replacement.

Fortunately, present technology can reach backward as well as forward. The representational adaptations may be made by the computer itself; computers excel at changing one value into another. The hardware incompatibility can be overcome with interface circuitry which links the computer's memory or I/O bus to the I/O of the peripheral device. The techniques for accomplishing this interconnection are the topics to be covered in future I/O articles.

Steve Leibson, lead engineer, received his BSEE from Case Western Reserve University in 1975. He has been with the Desktop Computer Division of HP for 3.5 years. His efforts include work on the 9878A I/O Expander, 98036A Serial Interface Card, 98224A Systems Programming ROM, and the System 45.

Powerful and precise x-ray therapy has become more affordable

by William Saylor

Radiation therapy is that medical specialty which employs high energy x-rays, gamma-rays, and electrons to sterilize human cancers. Multiple four million to twenty-five million volt radiation beams, produced by linear accelerators and betatrons, are focused on the patient's tumor from several directions.

In order to completely eradicate the disease, the radiation dose is usually very great, barely sublethal to the patient in many cases. Therefore, precise estimates of the amount and distribution of the radiation dose within the patient are essential, and it is here that the computerized radiotherapy treatment planning system assumes its important role.

Most radiation therapy for cancer even today takes place in clinics which do not have a dedicated in-house treatment planning system. In this day and age, one would think that the cancer patient deserves the benefit of this technology. Unquestionably, the availability of affordable planning computers constitutes a real increased potential for the control of cancer.

Hewlett-Packard's most powerful desktop computer, the System 45, is now being used in a new system designed to provide an accurate and cost-effective means of determining the best treatment plan for each individual patient.

The first known application of an HP System 45 to radiation treatment planning is at the Ocala Radiation Therapy Center in Ocala, Florida, under the direction of Dr. John F. Marshall. The system consists of the System 45 Desktop Computer with 62K bytes of memory, an integral high-speed thermal printer, two high density tape drives, a 12" CRT display, graphics package, the 9872A four-color plotter, and a 20 x 20 inch digitizer.

The linear accelerator being used to treat this patient is one of the most up-to-date devices of its kind. Planning radiation treatments with the speed and precision of a desktop computer helps treatment teams to make the most economical use of their time and their patients' money.

Treatment Planning Software

The treatment planning software was originally written for the 9830 by Walter Furr, Margaret Dillard and myself. The 9830 is comparatively slow; therefore in order to speed the calculations, we decided to use a grid spacing of 1 cm. This in turn presented a major problem of interpolation (interpolation is the process used within the desktop to determine exposure levels at points between the known data points). Since the actual dose values between grid spaces could range from 10% to 150%, we could not use a simple linear interpolation or

other conventional routine. Margaret Dillard solved the problem by using a modified Everett inverse collocation polynomial. This allowed us to interpolate accurately to within a fraction of a millimeter over the centimeter grid space.

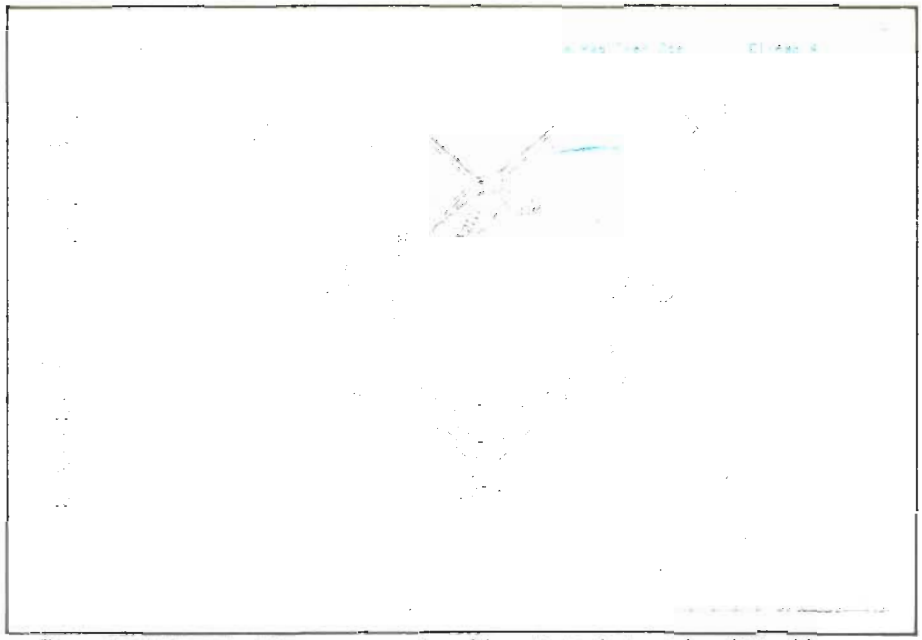
Conversion from the 9830 program to the System 45 was accomplished with an HP translator package, which converted 90% of the "9830 BASIC" program into the System 45's HP enhanced BASIC. It excluded routines involving I/O commands, which had to be converted manually. The total conversion process required about four man-weeks.

Conversion to System 45

By making the change between desktops, we were able to make use of the System 45's speed, which is ten times greater than the 9830. The increased speed of the 45 coupled with its graphics capabilities allows us to do real-time optimization of multiple-beam plans. This wasn't possible on the 9830 because it was just too slow.

Our efforts have been to develop programs which are fully interactive. This characteristic, combined with the "friendliness" of the 9845, has resulted in a software package which can be easily and safely used by radiotherapy personnel lacking any formal computer training. We also endeavored to write programs for use on all patients, not just those requiring multiple beam plans, for better utilization of the computer and reduced cost per patient.

The software available at this time includes programs for multiple and rotational beam plans, irregular field plans (i.e., mantle fields), plans involving compensating filters, implants (needles and seeds), gynecological applicators, machine settings for unilateral and parallel opposed treatments, optimization of central axis plans, NSD calculations, gaps between fields, skin dose and tolerance and cell survival curves. Additional programs are under development.



Four-color plotter output shows a cross section of the patient to be treated, as digitized from various x-rays. Green contour shows the outline of the patient's body, the lungs are outlined in blue, red indicates the area of the spine (lower), and tumor. Black lines indicate the direction of the 4-MV x-ray beams 1 and 2, and outlines areas of various exposure levels.

The plot above is an example of the output of the MULTI-BEAM program for the case of a patient being treated by two wedged 4MV x-ray beams. The tumor and vertebral body are shown in red, the patient's outline in green and the lungs in blue. Isodose lines and labeling are in black. The patient's contour is first entered into the computer using the digitizer. Lungs, tumor volume and other structures are also digitized as appropriate.

Radiation Dose Calculation

The contours are then displayed on the CRT and the entry point and direction of each beam is entered directly on the CRT using the DIGITIZE mode. The radiation dose rate is calculated for each beam at each point on a centimeter grid, corrections being made for wedges, and oblique incidence on the skin surface. The beam matrices are then added and passed to the interpolation and plotting routine. The calculation time for this plan is 1 minute.

For the first pass the program calculates the beam sizes and directions. If the plan is not satisfactory the operator may then enter the beam sizes and directions himself, and this continues until the most satisfactory dose distribution is achieved. Since certain radiation sensitive normal tissues often require that the dose be constrained to low levels, the planning objective is to achieve complete coverage of the tumor-bearing volume with a sterilizing dose, while keeping the dose to normal structures within safe limits.

The final plan is output on the 9872A plotter. For this program the LIMIT and LOCATE statements

Dr. John F. Marshall, director of the Ocala Radiation Therapy Center, Florida, tailors a radiation treatment plan to meet the needs of a patient.

provide a convenient method to establish protected areas for the necessary labeling of the plots. Finally the central axis program is linked to provide the machine settings necessary to deliver the prescribed dose.

Since their original writing in 1974, the programs have been extensively modified and tested against measurements in water and anatomical phantoms by Patricia Heffron, Thomas Ames and myself. In spite of this testing it is my own feeling that the installation of a computer in radiation therapy, with the attendant conversion from manual calculations to computer methods, is a period of danger to the radiotherapy patient. I, therefore, believe that an experienced clinical physicist should perform, or closely supervise, this conversion in order to ensure that the radiation therapists' dosage reference points are not altered. My own practice is to spend two to four weeks at the site customizing and testing the software, interpreting the programs to the staff of the clinic, and ensuring safe implementation of the system.

Compared to most other computerized treatment planning systems, the desktop computer offers a significant cost advantage. Amortized over a five year period, the per plan cost would be less than $8.00, about half the cost of using minicomputer systems.

William Saylor, a diplomate of the American Board of Radiology and former Associate Professor of Radiological Physics at the University of North Carolina at Chapel Hill, now writes and consults from his home in Sacramento, CA.

William L. Saylor
5036 Sky Parkway
Scaramento, CA 95823
USA

Programming Tips

9825A Memory Files

by George B. Bosco, Jr., Bosco Engineering, Whittier, California, U.S.A.

You can fool your HP 9825A Opt. 001 or Opt. 002 desktop computer into thinking it has no optional memory.

1. Find an HP 9825 without optional memory. Install the ROMs you plan to use. Enter a one-line dummy program (such as 0: "OPT000":). Create a 7990 byte memory file on tape, or a 7478 byte memory file on a diskette.
2. Put your ROMs in your Opt. 001 or Opt. 002 machine. Turn the 9825A on and load the memory file of step 1. Press RESET.
3. Your 9825 now has no optional memory. Load small programs and or data. Run programs. Record memory programs and later recall them. In other words; use as a calculator without optional memory. Do not execute "erase a" (erase all).
4. To restore 9825A to normal full memory size: execute "erase a" or turn machine off and then on.

You save file space and time by using this technique if your programs are small enough to run in this size memory. Similarly, use a 9825 with Opt. 001 memory to create an "Opt. 001" memory file for use in an Opt. 002 machine. To create the "Opt. 000" & "Opt. 001" memory files using your Opt. 002 calculator ask your HP Service Representative about memory switches.

Cataloging Files (9825A)

Submitted by Eldon Brown, Hewlett-Packard Company, Bellefield Office Park, 1203 114th S.E., Bellevue, WA 98004

The following program will produce a catalog of the files on the 9825A tape cartridge. Each file number will be printed along with the file type, current size and absolute file size. The program requires the use of the Strings, General I/O and Extended I/O ROMs.

If the file to be cataloged is a program and line zero of that program is a label (possibly the name or a description of the program), that label will be printed instead of the file type.

The files will be cataloged starting with track zero, file zero. When the null file (the last file) on track zero is encountered, this program will automatically switch to track one and catalog the files on that track.

You can prematurely switch tracks by setting flag zero. This can be done in live keyboard while the program is running.

If you want to modify this program, make certain that variable N equals the next available program line number (in this program, 34) as it is assigned in line three. Also, ensure that the third parameter of the ldf statement (found in line 21) has the value of the line number that follows the ldf statement (in this program, 22).

```
0: "9825A Cart Catalog
    Program":
1: dim I$[106]
2: buf "lstb",I$,1
3: fxd 0;34→N
4: dsp "Insert Tape, Press:
    CONTINUE";stp
5: 0→T;cfg 0,1
6: trk T
7: spc 2;prt "Track"&str(T)
8: prt "----------------"
9: 0→F
10: if flg0#T;flg0→T;spc
    ;gto 6
11: fdf F
12: idf A,B,C,D,E
13: if B=0 and D>0; prt
    str(F)&"--Empty File"
14: if B=0 and D=0;prt
    str(F)&"--Null File"
15: if B=1;prt
    str(F)&"--Binary"
16: if B=2;prt
    str(F)&"--Numeric Data"
17: if B=3;prt
    str(F)&"--String Data"
18: if B=4;prt
    str(F)&"--Memory File"
19: if B=5;prt str(F)&"--Key
    File"
20: if B#6;gto 26
21: ldf F, N, 22
22: buf "lstb";list
    #'lst',N,N
23: red "lstb",I$
24: pos(I$,":")→X;X+2→X;if
    I$[X,X]#char(34);prt
    str(F)&"--Program";gto
    26
25: prt str(F)&"-"&I$[X+1,
    X-1+pos(I$[X+1],char(34))]
26: prt" "&str(C)&str(D);
    spc
27: if D=0 and T=1;spc 3; end
28: if D=0;cmf 0;flg0→T;gto 6
29: F+1→F
30: gto 10
31:
32: "lst":ret "lstb.1"
33: end
*4170
```

Instrument Approach and Landing Game (9825A)

Submitted by Chris Mills, 23 Dwyer Street, Cook ACT 2614, Australia

Frequently during program execution it is valuable to be able to modify a variable.

Extended I/O ROM

Enter this program:

0: rdi (4) → A; dsp A; jmp 0

Now press (RUN) and you will see a free running display. I think interface 4 is the register which holds the result of the machine scan of the keyboard. This can be used to advantage in 'real time' simulations. The following game program simulates an aircraft making an instrument approach and landing. As the program continually loops, rdi (4) is used to scan the keyboard for control inputs which modify variables and hence the power setting, pitch and azimuth angles. Lines 29, 32 and 35 are the control inputs. Note that the rdi (4) statement is used twice to check to see if a key is being held down. Although this simulation was written for fun, the rdi (4) statement could be used in many "serious" applications, e.g. in control applications you could vary the value of variables to increase a temperature limit or change a motor speed.

Conclusion

I have only used the rdi (4) statement in games but can see that it could be a great help in control applications.

```
0: "Landing Program":
1: prt "LANDING";spc 2
2: fxd 0;dsp "press 'CONT' to go";gto "ent"
3:
4: "dec";str(p1)→B$;B$[2]→A$[p2,p2+len(B$)-2];ret
5:
6: "rnd":
7: rnd(1)→Q;rnd(1)→P;if P>.5;-Q→Q;ret
8: ret
9:
10: "vec";sin(O)→W;√(1-WW)→X;sin(S)→Y;√(1-YY)→Z
11: H+J(VW+F)→H
12: I-J(VZX+E)→I
13: G+J((VY+r1)+L)→G
14: V+J((T-VV)/M-50W)→V
15: if H<0;gto "con"
16: if V<150;beep;if V<145;dsp "you stalled";wait 2000;gto "end"
17:
18: "tur":
19: if H<500;.5R→R
20: cll 'rnd';RQ+F→F;cll 'rnd';RQ+E→E;cll 'rnd';RQ+L→L
21:
22: "ang":
23: asn(H/I)→B
24: asn(G/I)→C
25: asn(F/V)→O
26: asn(r1/V)→S
27:
28: "cont":
29: rdi 4→X;rdi 4→Y;if X#Y;gto +3
30: if X=80;O+1→O;T-.1T→T
31: if X=88;O-1→O;T+.1T→T
32: rdi 4→X;rdi 4→Y;if X#Y;gto +3
33: if X=78;S+1→S
34: if X=89;S-1→S
35: rdi 4→X;rdi 4→Y;if X#Y;gto +5
36: if X=81;T+.2T→T
37: if X=79;T-.2T→T
38: min(T,1e6)→T
39: max(400,T)→T
40:
41: "dsp":
42: A+B→B
43: if B>1;char(10)→A$[16,16]→A$[17,17]→A$[18,18];gto +8
44: if B>.5 and B<=1;char(10)→A$[16,16]→A$[18,18];" "→A$[17,17];gto +7
45: if B>.2 and B<=.5;char(10)→A$[17,17];" "→A$[16,16]→A$[18,18];gto +6
46: if B<=.2 and B>=-.2;"***"→A$[16,18];gto +5
47: if B<-1.5;char(222)→A$[16,16]→A$[17,17]→A$[18,18];gto +4
48: if B>-1.5 and B<=-1;"↑↑↑"→A$[16,18];gto +3
49: if B>-1 and B<=-.5;"↑ ↑"→A$[16,18];gto +2
50: " ↑ "→A$[16,18]
51: "   "→A$[13,15]→A$[19,21]
52: if C>1.5;"→→→"→A$[13,15];gto +7
53: if C>1 and C<=1.5;" →→"→A$[13,15];gto +6
54: if C>.5 and C<=1;"  →"→A$[13,15];gto +5
55: if C<=.5 and C>=-.5;"+"→A$[15,15]→A$[19,19];gto +4
56: if C<-.5 and C>-1;char(13)→A$[19,19];"  "→A$[20,21];gto +3
57: if C<-1 and C>=-1.5;char(13)→A$[19,19]→A$[20,20];" "→A$[21,21];gto +2
58: char(13)→A$[19,19]→A$[20,20]→A$[21,21]
59: -S→r2;if r2<0;360+r2→r2
60: if C>.2;cll 'dec'(r2,19)
61: if C<-.2;cll 'dec'(r2,13)
62: "K   "→A$[1,4];cll 'dec'(V/1.69,2)
63: " "→A$[5,11];if F<=0;"D"→A$[6,6];"-"→A$[11,11];gto +2
64: "C"→A$[6,6];"+"→A$[11,11]
65: cll 'dec'(60int(abs(F)),7)
66: "A   "→A$[23,27];cll 'dec'(int(H),24)
67: " R   "→A$[28,32];cll 'dec'(.00164I,30)
68: "|"→A$[12,12]→A$[22,22]
69: dsp A$;gto "vec"
70:
71: "ent":
72: 0→X;ent "Instructions? 1=yes,CONTINUE=no",X;if X;cll 'ins'
73: dim A$[32],B$[20];-3→A→B→O;.19→J;1e4→M;2.25e4→T;60000→I
74: cll 'rnd';3000+10000→H;cll 'rnd';30000→G;cll 'rnd';250+1000→V
75: enp "AIR TURBULENCE ? (1 to 25)",R;R/10→R
76: enp "X WIND KTS?(left neg,right pos)",L;1.69L→L;L/10→L
77: sfg 14;" "→A$[1,32];lkd ;gto "vec"
78:
79: "con":
80: abs(G)→G;1200-I→I
81: prt "You landed";prt G;prt "feet off center-";prt "line at";prt V/1.69
82: prt "Kts.";spc 2;prt "Landing was";prt I;prt "feet from the"
83: prt "aim point with a ";prt "vertical speed =";prt F,"feet/sec"
84: prt "Fighter Pilots";prt "   Do It";prt "     Better!";spc 5
85: if abs(G)>300;dsp "You landed off the runway!";wait 2000;gto "end"
86: if I<0;dsp "You landed short!";wait 2000;gto "end"
87: if I>4000;dsp "You ran off the runway!";wait 2000;gto "end"
88: if F>-10;dsp "Nice landing";stp
89: if F<=-10 and F>-20;dsp "a bit heavy!";stp
90: if F<=-20 and F>-50;dsp "CRUNCH!!!!!!";stp
91: "end":dsp "wreckage is now burning";stp
92:
93: "ins":
94: prt "instructions";prt "You are landing";prt "on a runway on a"
95: prt "heading of north";prt "(000-360).";prt "to control the "
96: prt "aircraft use:";prt "'0'=LEFT";prt "','=RIGHT"
97: prt "'.'=DOWN";prt "'2'=UP";prt "'1'=POWER OFF"
98: prt "'3'=POWER ON";prt "aircraft stalls";prt "at 85 Kts.";spc 2
99: prt "K=Speed(knots) ";prt "DorC=Vertical vel";prt "A=altitude(ft)"
100: prt "R=range in naut-";prt "ical miles *10"
101: prt "→→ means fly";prt "right";prt "↑↑ means fly up"
102: prt "and so on.";prt "heading is dis-";prt "played when off"
103: prt "centerline.  ";prt "happy landings";spc 5;ret
*14535
```

Update

9862 and 9864 Support

The rapid acceptance in the marketplace of two new peripherals products this year has eclipsed two older products.

The Desktop Computer Division in Fort Collins met with great success in the introduction of its 9874A Digitizer. The new digitizer features an adjustable glass platen, a key pad with special function, numeric entry and control keys, as well as a new cursor vacuum system and a self-test capability. The 9874 supersedes the 9864A Digitizer, which has been declared obsolete as of February 19.

The 9872A Plotter produced by the San Diego Division brought to the market four-color capability, exceptional line and character quality, and 38 different built-in instructions, among other features. Its capabilities and the cost effectiveness of the new 7225A Plotter has been so well received that their predecessor product, the 9862A Plotter, has been declared obsolete as of March 1.

HP will continue supporting both products for one year after they go out of production. Technical support and consumables will be available for 10 years after the products are discontinued. Full service will be available on site for five years after obsolescence and at HP service facilities for five additional years. After that, service will be provided on a "best effort" basis — the standard HP policy for obsolete products.

9845 Waveform Analysis, 09845-12500

This package allows performing forward and inverse Fourier transforms on either single or double waveforms. Fourier coefficients for equally and unequally spaced data, the Hanning function, power spectrum, auto and cross correlation and cross power can be computed. Results can be output in video or hard copy formats.

9845 Text Processing, 09845-10520

This software package allows the user to compose and store a document on the computer, then change or update any of the material. A key feature of the system is the ability to automatically search for and replace user-specified words. As many as 45 pages of single-spaced text can be entered into the computer and permanently stored on a single magnetic tape cartridge.

9845 Statistical Graphics, 09845-15020

This package contains nine routines for plotting statistical data: time plot, histogram, probability plots, scatter plots and Andrews plot; contains Basic Statistics and Data Manipulation.

9845 Payroll, 09845-10750

This software package allows producing paychecks, payroll register, check register, employee hours, tax distribution and employee summary reports, and the 941-A quarterly and W-2 year-end forms. Results are available on the CRT screen or in a printed hard-copy format.

9845 Inventory Control, 09845-10760

This package handles finished goods inventory, provides interactive parts inquiry on the CRT display and produces printed management reports such as turnover and reorder reports.

Keyboard

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

Publications Manager - Ed Bride
Editor - Bill Sharp
Editorial Assistant - Chris Stumbough
Art Director - Hal Andersen
Graphics - Wendy Hart
Typography - Paula Dennee

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

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address. Send changes to Hewlett-Packard Keyboard, 3404 E. Harmony Road, Fort Collins, Colorado 80525, U.S.A.

Jan 2, 1979