WHAT HAPPENED?

We have decided to suspend High Performance Computing Magazine, the magazine for Hewlett-Packard personal computer users. We believe it would not be a prudent business decision to start this publication.

Our faith and enthusiasm in Hewlett-Packard and the HP 150 remains undiminished. In fact, the positive reviews of the HP 150 appearing in publications like Byte and Infoworld have simply confirmed our early assessment of this computer. However, this market is not maturing at the pace we expected when we announced High Performance Computing Magazine, and we feel it would be premature to spend the considerable funds necessary to lunch a slick magazine. No new launch date for the magazine has been set.

This venture has been an emotional roller-coaster that has consumed our time, energy and considerable cash over the last six months. We were approached by a West Coast publisher to launch HPC, but at the last minute this publisher had financial difficulties and couldn't raise the capital which we felt was necessary to assure the survival of the magazine during the critical first year of publication. My wife and I formed a new company to publish the magazine, and friends and business associates funded the new publication.

On March 15, 1984, we had to make a go/no go decision for the publication. Our financial commitment was in 5 figures by this date, but by starting our advertising campaign and printing our first issue, our commitment would rapidly become 6 figures. Although we love the Series 80, the plain fact is that what made a slick magazine a financial possibility was the introduction of the HP 150. With the slow sales start for this computer, we were faced with launching a publication that would lose money for several years, with no hope of profit for a considerable time. Frankly, we couldn't ask our friends and associates to put money into such a venture, and we decided to suspend publication of the magazine.

For those of you that have expressed an interest in our future plans, NEWS80S will continue to be published. Economic realities force us to return to a newsletter format. I hope that our continued focus on Series 80 material will compensate for the change in format from a folio to a newsletter. My book, "HP 150: Fingertip Computing," remains in production and is being published by the Que Corporation.

IN THIS ISSUE

ARE YOU SERIALous? by David Efron

Interfacing a peripheral device to a computer is sometimes as easy as just plugging it in, but sometimes it is no laughing matter. Often it is perplexing, and sometimes it requires the use of mechanical skills and crude language. If the device requires connection to the serial interface (part number HP 82939), don't despair; don't reach for the 10-pound hammer as a solution; an explanation of serial communications and how to use the serial interface might help in connecting it and expanding its usefulness.

A Sexual Problem

The serial interface comes in male and female genders. Which one to order depends on how it is to be mated. The "standard" cable is female, usually connecting to a peripheral device such as a printer, which normally has a male connector. (Nevertheless there are exceptions to this general rule, and they will require some "digging" through the printer's manual.) The Option 001 model is male, which normally connects to a telecommunications modem, but which can also be used to directly link one computer to another.

Let's work with the standard model and hook up a printer which performs serial data communications. If you have the Option 001 interface, you will find that it can be mated with the printer anyway. A sex change operation is possible:
Have a short cable constructed which has a female connector on each end; pass wires between pins 1,2,3,7, and 20, but cross-connect pin 2 on one side with pin 3 on the other, and vice versa; and naturally, if your printer uses other pins, make sure that those wires are also passed through the cable adaptor.

Voodoo Magic

These pins and the electronic chanting, known as protocol, enable the computer and printer to talk to each other. The printer usually "listens" on pin 3 and it talks on pin 2. Normally, that is what the computer does also, but they both cannot do that if they are to communicate.

Modems are devices which oftentimes are used to solve that problem. One would typically accept digital data from the computer on pin 2 and MODulate it with a frequency for transmission over a telephone network. Another would receive that information, DEModulate it, then send the digital data through pin 3 of the cable which links it to the printer. When the printer talks back to the computer, the modems perform in the same way, but the direction of the data path is reversed. The printer's data is still sent out on pin 2, converted, received by the other modem, and sent to the computer on pin 3.

Essentially, the Option 001 serial interface is designed to operate in a "remote telecommunications" mode as described above. In contrast, the standard model is designed to be used in situations where the modems are not used, such as attaching a printer to a computer in a "local" setting. The standard

interface is designed to send out its data on pin 3 and to receive data on pin 2. There are official terms describing both types of configurations, depending on whether the computer is being used with data communications equipment (a modem) or data terminal equipment (a printer, crt, etc.).

Hold the Phone

The computer and printer are now talking on the proper channels. If the computer sends data at a rate of 30 characters per second and the printer is capable of printing that fast, then there is little need for any talking between the two. But what if the computer sends data at four times the speed at which the printer can handle it?

Imagine the dictation of a memorandum. If a tape recorder is used, the rate of dictation is of little significance ; when typing it , the play-back can be started and stopped to match the pace of the typist. If it is taken down in short-hand, some method is needed to parcel the information in units manageable and comfortable to both the speaker and the stenographer. Perhaps sentences provide bursts of dictation of the proper size. The word "period" could signal the halt of dictation until the stenographer has it all down and gives the response "go ahead" for the next sentence.

On the other hand, you call a busy switchboard. "Serial Communications Division, may I help you?", and you reply, "Yes, I'd like to order a computer manual on the . . ." Another telephone line lights up on the switchboard, and you hear "Could you please hold!". Notice that it is not really a question. A little later you hear,. "I'm sorry, please continue." Whereas you probably would re-phrase your request from the beginning, in computer communications you would have picked up from where you were interrupted.

These two examples illustrate two often-used methods of communications "hand-shaking". They are methods of using software to send signals that control the transmission of data, as opposed to "hardware hand-shaking", which requires the computer to monitor the voltage on one pin of the cable, which is set by the printer to indicate that it is ready, or not, to accept more data.

The first of the examples of software hand-shaking is known as ENQ/ACK, or ETX/ACK. The computer's interface sends an ENQuire character (ASCII 5) or an end-of-transmission character (ASCII 3) after sending a block of information. Then it waits. The length of the block depends on how the computer's interface has been programmed, but it had better not be greater than the size of the buffer of the printer. The printer then prints out the characters it has received, and when it encounters the ENQ or ETX character in the stream, it sends back to the printer an ACKnowledge character (ASCII 6). The computer has been waiting for one, and it now knows that another block of data can be sent.

The second method aids printer performance by striving toward constant operation of the printer. With the ENQ/ACK method, the printer empties its buffer,and finding the ENQ character as the last character, it sends the signal to resume data transmission. A small loss of time occurs while

waiting for the first character of the next block to be sent. With the XON/XOFF (also known as DC1/DC3) method, a printer can be programmed to send the XOFF character (ASCII 19) when its buffer becomes near-full, and to later send the XON character (ASCII 17) when it becomes near-empty. The computer, listening for the XOFF while sending data, stops its transmission before the printer's buffer overflows. Then, waiting for the XON, it sends more data in time to replenish the printer's buffer before it runs out.

Attaching an HP 260lA Printer

Let's say you have the HP 26OlA daisywheel printer. An Application Note (#5953-7816) has been published on the subject, and it is explained here. It specifies a series of SET I/O statements, to be executed prior to printing anything. We'll take them one by one.

SET I/O 10,2,5   Assuming you have not changed the default select code from 10, this statement places into control register 2 the value 5. It tells the interface that there are no modems operating between the computer and the printer; i.e. it fakes out the interface, making it believe that the modem is always ready and it is always "clear to send" data.

This could also be accomplished with a cable adaptor, such as the one specified earlier to convert an Option 001 interface cable to mate with a printer. Tie together pins 4,5,6, and 8 r at both ends of the cable. This will hold the signals high, . and eliminate the need for this SET I/O instruction.

SET I/O 10,3,8   Sets the baud rate at 1200 bits per second, or roughly 120 characters per second. The baud rate can be set "under the hood" of the interface, but in either case it must match the rate which your printer is set to operate at.

SET I/O 10,4,26   Specifies 7 data bits, even parity, and one stop bit. These, too, can be set in dip switches of the interface. A total of ten bits are to be sent: A start bit, 7 data bits (ASCII values 0-127), one parity bit (set or cleared such that there are an even number of "1" bits among bits 2-8), and one stop bit.

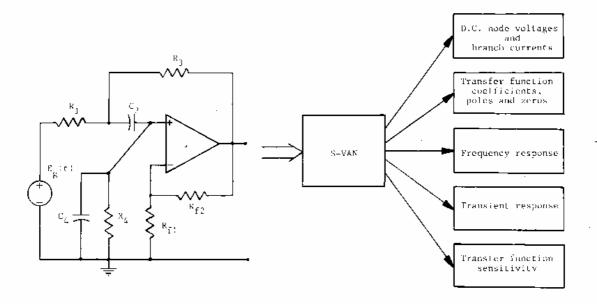If your printer can interpret character values in the ASCII range 128-255, consider using SET I/O 10,4,3 for 8 data bits, no parity bit, and one stop bit.

The parity setting specified in the Application Note matches the printer switch settings also specified in the Application Note. But you can choose odd parity or no parity, as long as the serial interface is set to match the printer setting.

AT LAST! A CIRCUIT ANALYSIS PROGRAM FOR THE HP-85 THAT DOES MORE THAN JUST FREQUENCY RESPONSE ANALYSIS!

OPAL Engineering proudly introduces <u>S-VAN</u>!

A Computer-Aided Design software package for the HP-85 desktop computer which provides the complete analysis of linear circuits.

No longer must the professional circuit designer be satisfied with just the magnitude and phase response characteristics of his circuit as found by so many desktop circuit analysis programs. Instead, with <u>S-VAN</u>, the designer has access to a complete set of analysis features which provide a thorough, accurate simulation of a given circuit. These features include -

- D.C. Analysis - node voltages and branch currents.

- Transfer Function Analysis - extraction of coefficients, and poles and zeros.

- A.C. Analysis - magnitude, phase, group delay, real and imaginary parts.

- Transient Analysis - responses include impulse, step, ramp, pulse, square-wave, piece-wise linear, and more.

- Sensitivity Analysis - transfer function coefficient, and pole and zero sensitivities.

In addition, <u>S-VAN</u> is completely user-friendly by offering a free-format interactive input structure, an interactive circuit editor, interactive analysis commands, graphics outputs, mass storage commands, and much more. For more information regarding this revolutionary new circuit design tool, contact OPAL Engineering at

OPAL Engineering & Consulting
828 Opal Dr., Suite 1
San Jose, CA 95117
(408) 246-1254

SET I/O 10,11,192   These three instructions program the interface to use the
SET I/O 10,14,19   XON/XOFF communications protocol. If your printer only can
SET I/O 10,15,17   understand ETX/ACK, use:

> SET I/O 10,11,128   These program the interface to send a return
> SET I/O 10,15,6   character (ASCII 13) and an ETX character
> SET I/O 10,16,66   (ASCII 6) after every line of data that is sent; and
> to wait for the ACK before sending the next line of
> output. If your printer needs a line-feed character in
> addition to the carriage return, use these two
> SET I/O 10,16,67   instructions to provide a 3-character end-of-
> SET I/O 10,19,3   line sequence. To program the ENQ/ACK
> method substitute the character: SET I/O 10,15,5

SET I/O 10,9,137   Not really needed -- this is the default setting upon reset or power-up

PRINTER IS 10,NNN   Naturally, you have to turn on the printer and optionally specify a page width

To attach any other serial-type printer, the same interface programming techniques are used. It requires a little knowledge of how your printer operates and perhaps some trial-and-error experimentation. Follow these general steps:

1.   Set your printer's switches. If selectable, choose the baud-rate, parity on-or-off, number of data bits, number of stop bits, and auto-line-feed-or- not (if set for line feed, make sure that your end-of-line sequence does not contain a line-feed character, unless you want double-spaced output).

2.   Make sure your cable contains all the necessary wires. Remember that you might need a cable adaptor to cross wires 2 and 3 if you have the Option 001 serial interface, or if your printer is non-standard. Also note that to implement hardware hand-shaking you have to wire your printer's "on-off" signal. Pin 20 is commonly used for this, as it is on the HP 2601A and the Diablo and Qume letter-quality printers. However, other printers have been known to use pins 8,11,12, or 19 to signal the busy-condition. If your printer uses any pin other than pin 20, your cable adaptor must wire pin 20, on the HP computer side, to the "busy" pin on the printer; and to keep it confusing, if you have the Option 001 interface, it is pin 5 on the computer interface to which the printer "busy" line must be channeled.

3.   Program the serial interface to match the printer specifications by setting dip switch settings inside the interface or by using BASIC statements to SET I/O control registers as in the examples above.

Attaching Another Computer

What else is the serial interface good for? It is a handy tool for getting data from another computer into your HP system, or vice versa. As an example, you might have another computer system on which you have a data base from which data can be extracted for graphing or other analysis, using specially developed programs on your HP system. Or you might want to transfer programs and data files from one machine to another, without having to re-keyboard everything.

Gaia Communications is pleased to offer two new products in the NETWORKING series:

NET66 [tm] $450
for the HP 9816 and HP 9836

and

NETWORKING II [tm] $295
for the HP-86 and HP-87

We are also pleased to extend a new low price of

$150
for the original
NETWORKING [tm]
for the HP-83 and HP-85A
and tape/Edisc version for the HP-85B

Serious modem communications packages for those who require high performance. The software's uncompromising design gives your machine a whole new feel. Makes your micro an advanced telecommunications center and adds information management for fast-track performance.

- Programmable for customized, automatic operation with information utilities, HP Mail, and other personal computers.

- Complete electronic mail (automatic answer mode).

- Edit and File information for future reference.

Moneyback guarantee and no-nonsense support. Quantity prices and custom services. Call for more information or to order.

Gaia Communications	User's Group HP Bulletin-Board System
(503) 929 5412	**(503) 929 5433**
Rt. 2 Box 321C Corvallis, OR 97333	7 bit/odd parity 300/1200 baud

Programming the two computers to talk to each other is more difficult than attaching a printer to a computer. Both devices are designed to be the "host" device, and now one must be taught to be subservient.

First, the cables have to be matched. Unless you have the Option 00l model of the serial interface, both computer interfaces are probably inclined to send information on pin 3 and receive it on pin 2, in which case you will need an adaptor cable to cross-wire pins 2 and 3.

Next, the data transmission characteristics have to be matched. Baud rate, parity, data bits and stop bits. This may require some experimentation.

Finally, you will need two programs, one to send the data and another to receive it. A simple approach is to select "no protocol" in both interfaces, and to program a delay 1 on the sending computer after each transmitted line (or block of data), to give the receiving program enough time to edit it and store it in memory or in a disk file.

If you want to experiment with data transfers using a protocol to control the data transfer, keep in mind that the ability to program a protocol on the HP serial interface, and on microcomputer interfaces in general, only works when sending data. As a receiver of data, the serial interface cannot cooperate with the XON/XOFF or ENQ/ACK as it does when sending data to a printer. Your HP program which processes received data must handle it.

In this regard, the flexibility of the I/O ROM makes this, and much more, possible. Concluding this article, a sample program is provided to demonstrate the transfer of data from an Apple II computer to the HP serial interface, using the ETX/ACK protocol used by the Apple II serial interface.

```
10 ! APPLE II to HP Transfer Program
15 !
20      DIM A$ [256],F$ [256]
25      CREATE "FILE", 50 !         For storage of the incoming data
30      CONTROL 10,4 ; 6 !          7 data bits, 2 stop bits
35      !               ( the Apple interface sets the high bit on all characters sent, so
40      !               specifying 7 data bits essentially strips off the high bit and
45      !               gives us the proper character values in the range 0-127)
50      S$=CHR$(3)&CHR$ (10)&CHR$ (13)&CHR$ (12)&CHR$ (10)&cHR$ (12) ! Convert
55      !       incoming characters by substituting ASCII 10 for 3, and 12 for both
60      !       13 and 10. ETX (ASCII 3) ends each block of received data, but
65      !       the ENTER statement wants a line-feed (10) character to signal
70      !       the end. set up that conversion. But what if the Apple
75      !       sends a line-feed? We don't want it to cause the ENTER to be
80      !       ended prematurely, so we convert it to a CHR$(12), as we do for all
                carriage returns . CHR$ (12) is r the line-end-marker ; this will
                avoid the situation where CR is the last character in the received
                block , because the interface wou1d see the sequence CRLF and throw
                both of then away . Now it sees CHR$ (12) &LF , keeping the CHR (12).
85 CONVERT IN 10 PAIRS ; S$ ! Invokes the pair-wise substitutions
```

```
90 F$= "" !                   Initial value. F$ is the carry-over from one
95 !                          Incoming block to another
100 ENTER 10 USING "#,1A" ; A$@ X=NUM (A$) @ DISP X @ IF X<> 10 THEN 100
105 !                         Wait for an ETX (converted to LF), which the apple II serial
110 !                         interface sends to inquire "are you there?"
115 OUTPUT 10 ;CHR$ (6) !     Send back the PCK, "Yes, I'm ready"
120 ENTER 10 ; A$@ A$=F$&A$ ! Get a block; it may, or may not, end with a CR
125 X=POS (A$,CHR$ (0)) @ IF X>0 THEN A$=A$[1,X-1]&A$[X+1,LEN (A$)] @ GOTO 125
130 !                   Reserve any unwanted control characters which you might know to
135 !                   exist in the incoming data
140 X=LEN (A$)
145 Y=POS (A$,CHR$ (12)) !  Parse the block into "lines" which are marked by
150 !                       carriage return characters (ASCII 13)
155 IF Y=0 THEN 210 !        IF none (or no more of them), save this segment
160 !                        of the block to be attached to the next one
165 F$=A$[1,Y-1] @ DISP F$ @ PRINT# 1 ;  F$
170 !                        Echo the line to the screen and store on disk
175 IF Y+1>LEN (A$) THEN A$= "" @ GOTO 210
180 !                        CR was the last character in the block
185 A$=A$ (A$)[1, LEN(A$)] @ IF NUM (A$)<> 127 THEN 140 ELSE 230
190!                         Drop off the line we just parsed. If the next character is
195 !                        ASCII 127, that's the signal to end this data transfer. The
200 !                        sending program must PRINT CHR$ (127) ;CHR$ (3) to effect this
205 !                        signal after the data transfer is complete
210 F$=A$ @ OUTPUT 10 ;CHR$ (6) @ GOTO 120
215 !                        Save the remainder of the block for concatenation in line 150.
220 !                        Seal the CK character to the sending computer to start the
225 !                        transmission of the next block of data
230 ASSIGN# 1 TO * !        Close the file. All done.
235 END
240!
245 !
500 ! Program to SEND the data from the Apple II
505 !
510 ! D$CHR$ (4) :PRINT D$; "OPEN XFER FILE" :PRINT D$; "READ XFER FILE"
515 ! ONERR GOTO 600
520 ! PRINT D$; "PR#l" :      PRINT CHR$(0); R)KE 2041,0 BEN !lb direct the
525 !                         outputs to the interface and disable the interfaces
530 !                         line feeds after carriage returns. The CHR$(0) is
535 !                         needed to "initialize" the interface,
540 ! INPUT L$ : PRINT L$ GOTO 540
545 ! REM                     Read lines from a sequential file and print them
550 ! REM                     through the interface. At end-of-file the program
560 ! REM                     will branch to line 600.
600 ! PRINT CHR$ (127) ;CHR$ (3) :   REM Send the end-of-transfer signal
605 ! PRINT D$; "PR#O" : END :       REM and turn off the interface
```

COMMENTS ON UCSD p-SYSTEM FORTRAN ON THE HP-86 by Kirt Miller, PE.
Miller Consultants, 2719 Mountain View Drive Boise, Idaho 83704

UCSD p-SYSTEM is an operating system that supports Fortran and Pascal and is adapted for use on the HP-86. I am currently using the p-SYSTEM to run aircraft performance modeling programs, written in FORTRAN, on my HP-86. After using an IBM mainframe for many years, my first impression of pS-SYSTEM was that it is SLOW!! A program that executes in less than 5 seconds on a mainframe can require more than 15 minutes on the HP-86. The same relative time is required for the compilation of the program. However, the 86 is not a mainframe and if you've got the time, p does get the job done very nicely.

In implementing and using p-SYSTEM FORTRAN on my Micro, I had to shift from use of FORTRAN IV, which is what I had been using, and educate myself in the ways of FORTRAN 77, used by p-SYSTEM. During this process, I noticed some things which I felt were shortcomings of either the system or its instruction manuals. I've listed some of these below under the heading of "Miller's Frustrations and Aggravations with the p-SYSTEM".

o The p-SYSTEM FORTRAN reference manual contains numerous valueless comments such as "The FORTRAN compiler is invoked as the Pascal compiler would be . . . ." Since I know nothing of Pascal, this type of comparison didn't help me one bit in understanding the function they were trying to describe.

o Both the p-SYSTEM introductory manual and the operating system reference manuals contain indexes a definite help to users. The FORTRAN manual has no index (What fun!). In addition, the manual is very short on examples -- especially those that would clearly illustrate the various types of FORTRAN READ AND FORMAT statements. And just to make things even more interesting, after page 15 the page numbers no longer correspond to the page numbers specified in the table of contents. Definitely a just-plain-careless feature.

o The p-SYSTEM FORTRAN manual carries no information regarding the command to skip to a new page on the printer. Hewlett-Packard customer service helped me with this one. The p-SYSTEM will not handle special characters, thus the use of the "CHAR(nn)" form is necessary. I have an IDS Prism 132 printer which uses control code "12" for advancing to the next top-of-form. (Other printers may have other character code requirements).
- The command to skip to a new page on my printer turned out to be:

WRITE(IPRNT, '(A/)') CHAR(12)

Once you learn this technique, the door is open for using all the special control features of your printer.

o The p-SYSTEM FORTRAN compiler offers the option of displaying errors on screen or writing them to a file you can later edit or print out. When using the error display option, you can continue to compile after an error is displayed, or you can enter the edit mode, on the source listing being compiled, to correct the error.

When editing is elected, the cursor is supposed to point to the line containing the error, but unfortunately, it consistently points to the wrong line, and the line number it lists is of little help in trying to investigate

the problem - especially in large programs. I now find it preferable to use the write-to-file option almost exclusively since this option provides a good summary of errors and their locations.

I also discovered that the compiler does not detect the lack of a closing parentheses in a FORMAT statement. An error message does correctly report it at run time.

o When executing your programs under p-SYSTEM, it is necessary to have all your personal program libraries listed in a file named "*USERLIB TEXT". This point is not mentioned in the FORTRAN manual, and it's possible to waste many hours before realizing that the explanation is in the p-SYSTEM Operating System Reference Manual. It would help if this information were in the FORTRAN user's manual.

o In executing a program, if an attempt is made to write (to a file) a REAL variable that has not yet had a value assigned to it, then a "Floating Point Error" message appears. This statement is not especially helpful in diagnosing the problem.

o p-SYSTEM has a RAMDISC feature which I use for storing data files and libraries to be utilized by an executing program. This reduces execution time significantly. On one occasion my program attempted to read from a data file not in the system. There was no "END. . ." parameter in the READ statement and the error message (#698) "End of file encountered on read with no "END= . . . option" appeared. I added the "END= . . ." parameter. The program then ran with no error messages, but, of course, it also produced nothing of value because there was no input. When the missing file was added to RAMDISC, the program ran fine. It would have helped if the system had simply reported that a file was missing.

There are a few other problems and irritations I've encountered along the way with the p-SYSTEM, but I've found I can live with them. If, however, I was to identify the one feature of the system that is most disappointing, it would be the slowness of program execution. I understand this slowness stems in part from the fact that the compiler produces a p-code that must be interpreted at run time, while other FORTRAN compilers produce machine language (which runs much faster). I also realize that p-SYSTEM was designed for optimum portability between computers, but I would gladly settle for the relatively good portability of FORTRAN 77 source code if it would mean reduced run times.

My overall attitude toward p-SYSTEM is that, although it does have its shortcomings, I'm mighty glad to have the system to work with. I would like, however, to be able to look forward to improvements on a future release. If they can speed up the compilation and execution times, they will have a real winner.

BREAKER FOR BREAKPOINT BY DAVID EFRON

In case you looked carefully, the Assembler Breakpoint Subroutine published in issue (#9) contains an error. Using it as shown would clobber the registers numbered 70-77, and the breakpoint table would repeat the 20's where the 70's should be. I was going to supply the lines to correct this when a phone call made me forget the whole thing.

There is a better way to do a breakpoint. A telephone call from Corvallis provided a nifty

alternative: The routine that performs the breakpoint status display for the optional System Monitor plug-in module is already in your Assembler ROM. To use it, simply JSB =104011 which is the address of the SPAR0 hook. Voila! The breakpoint display, as depicted in the System Monitor section of the Assembler ROM manual, appears on the screen. Examine memory, change register values, do anything the manual says you can do, and then press the RUN key to continue the execution of the binary routine.

You still will not be able to set breakpoint memory addresses, or perform the STEP and TRACE through your program - that is what the System Monitor module does for you.

Thanks, H for the call and the tip.

SYNTHETIC PROGRAMMING by W. Byrd Source Number TCT759

Synthetic programming is a term originated by users of HP's 41C programmable calculator to describe non-standard instructions which cannot be produced by normal keystrokes. The term applies in this sense to the following quirk of the HP86/87.

If you have an 86/87 plotter ROM there is a BASIC language command which is available but not described in HP literature.

This command is the power interrupt reset (p.i. reset). Its effect is identical to turning the mainframe power off, then on again. This command is not programmable in a normal way since its translation results in execution and loss of the program being entered (and anything else in memory). The p.1. reset may be Invoked in the following way.

Enter and run this short program: (requires the AP ROM):

```
10 DISP "Press [END LINE] for demonstration of p.i.reset."
20 AWRIT CHR$(0);
30 END
```

Alternatively the command Is available in calculator mode:

Press control-shift 2 and' 1 [LINE].

The command is a handy way to invoke an Autost program from the keyboard in two keystrokes.

There may be more quirks like this one. Has anyone found them?

CLASSIFIED ADS

FLASH! FOR SALE: HP85A, $1450. 85 ROMS I/O $150; Plotter/Printer, $90; Mass, $90; Adv.Prog., $150; Matrix, $90; Assembler, $150. SERIES 80: 82902M 5-L/4" drive, $850; HP $215; HP $215; HP-MODEM, $2L5; ROM Drawer, $25. Walter J. Eager, 3500 NW Glenridge, Corvallis, OR 97330. (503) 753-5837.

HP85A, 32K, HPIB, ROM's (P/P, MS, AP), 82902M 5.25 single floppy, 7225B plotter, VISICALC and more. Best offer., Don Orazem - 3803 NE 151st, Seattle, WA 98155 - (206) 362-0477)

I'm adding a 15MB Bering Drive and a Hayes 300/1200 Modem to my HP 86. I will sell one of my 9130 drives (which has seen very little use) for $600. I will also sell my 82950A modem for $275 w/doc + 5-1/4 software. Bill Alderson (408) 749-1268.

HP Series 80 CP/M <> Function and Cursor Control Keys are now useable!!
KEY80 enables you to redefine your keyboard to function with CP/M software.
Keyboard definition sets for WORDSTAR, MULTIPLAN and dBASE II are available.
KEY80 with 1 set is $100. $10/additional set. Ward Brook Systems, (603) 483-8045.

FOR SALE: Have converted- will sell HP-83A, HP-85A, 16K memory, HPIB, ROM Drawer, ROMS, Application Pacs, 82949A Parallel Printer Interface, and 9lllA Graphics Tablet. All in mint condition at half price, much of the above never used. (313) 528-1571 or P.O. Box 1034, Troy, MI 48099.

FOR SALE: HP87A complete w/lm HPIB & manual EX COND $1395; or consider trade for HP 85, 87 Adv Prog ROM add $95.00 EPSON 1EEE488 Board $65 (adapts EPSON printer to HPIB) 82832a Basic Training Pak -$50 82830A Visicalc+ $95 on 5.25" disk. Cathedral Sound INC 1575 5th St. Rennselaer NY Don Person (518) 465-5689

SERIES . 80 Independent consulting: Hardware, software patches & custom solutions. Reasonable rates, Guaranteed results. * Don Person (518) 482-9023 l0a/6p EST *

FEEDBACK by the Readers

ALBERT E. HAYES, JR. (FULLERTON, CA) :
      I received one of the first HP87 Advanced Programming (AP) ROMs that my dealer received. It is a joy to be able to MERGE and use the various other features I had on my old HP85 AP ROM. But there is another problem here. The AP ROM doesn't like to reside in the same machine with the UTIL/1 binary program I use for a lot of formatting. One of the big problems is that they have an extended LINPUT command in the AP ROM which is quite different from the LINPUT in UTIL/1. The entry of a step using the extended LINPUT command causes an instant dump of the entire program if I am using UTIL/1 in the same program. I brought this to the attention of the HP rep, and have received neither help nor sympathy, and

no hope of ever getting things straightened out.

Probably related to the previous problem is the fact that my keyboard will "lock up" from time to time when I want to rerun a long program and am sitting at an END step a few thousand program steps from the beginning. All that happens when I press RUN is that the little "power" light starts blinking and the keyboard locks up. No amount of reset or anything else will get me out of the loop. If I do a re-load from the disc it is happy to run again. With the AP ROM out of the machine everything runs fine!

Apparently the word PAGE has been reserved by the AP ROM, but is not discussed in the manual. It's my great luck that I have about 20 programs in which I use PAGE as a numerical variable. Of course I can't use the RENAME VARIABLE command in the AP ROM to change it, because it won't accept the instruction. More hand work, and no help!

(NOTE: Our information is that a "B" series HP 86/87 Advanced Programming ROM is available to early buyers of the AP ROM, at little or no cost. Contact HP or your dealer for more information if you're having problems caused by the HP 86/87 AP ROM).

ROSS WHISTLER (DOVER, MA)

With regard to the 'THANKS FOR THE MEMORY' item on page 11 of NEWS80S Number 5, there can be other problems in addition to memory loss when using unessential ROMs with the 85.

For example, I have found that adding a Plotter Printer ROM to my ROM drawer disrupts a program that runs fine without it. Specifically, I have a program that uses LABEL instructions in the graphics mode to print a number of contiguous frames of text side by side along the long axis of my printer paper, thus providing up to 16 horizontal lines of text of unlimited length.

The addition of a P/P ROM apparently truncates the final character of a 32- label, which unfortunately leaves vertical blank strips between adjacent frames in the assembled text. This gratuitous glitch is not mentioned in the P/P ROM literature.

To generalize this phenomenon, the ROM apparently discards all label characters that come too close to an edge. Try running the sample program on page 226 of the HP-85 Owner's Manual and Programming Guide with a P/P ROM in place and you'll see what I mean: all you get is a single printing of ****WRITES ON TOP OF ITSELF**** across the bottom. (Note that the fifth * is discarded). It no longer wraps around on top of itself as described at the - bottom of page 225.

Unfortunately, if you have written a program with the P/P ROM in place, there is no UNTRANSLATE command available that will modify the program so that it will run satisfactorily with the ROM removed; you just have to rewrite it all over again.

FRANK HOPEWELL (MEDFORD, OR):

The reason for this is to alert you to a faults in the HP "File Manager" program. Should a person attempt to expand the number of records in a file he will get a 15 system error and furthermore be locked out of his data. Calls to HP confirm that this is a fault of the program and there seems to be no way around it. The "File Manager" cannot be run with the I/O chip installed, and the upgrade which might have offered a solution has a I/O on the board!

HP BINARY UTILITIES by Hewlett-Packard

Several popular binary programs have been compiled into one economical package. The Binary Utilities package contains 13 programs written for the HP-86 and HP-87, and either a 3 or 5-l/4" recorded flexible disc. The $100.00 price represents a savings of 40% over the individual price of the programs and media. Included in the package are the following programs:

- Logical Interchange Format
- Storing and retrieving Programs as Data
- Sorting Groups of Elements of a String
- Enhanced String Manipulation
- Mathematical Functions
- Numeric Array Organization & Initialization
- Programmer Control of the Internal CRT & Keyboard
- Enhanced CRT, Cursor & Keyboard Control
- Forms/80 Binary Utility Programs
- Graphics Cursor
- Enhanced String Input & Cursor Control
- Immediate Execute Keys
- Graphics Tablet Display

The Binary Utilities Package for the HP-86 and HP-87 should be ordered by name from the Series 80 Users' Library. A similar package containing 22 Binary programs and one BASIC program is available for the HP for only $150.00 Consult the Series 80 Software Catalog or a 1ist of the programs included in the HP-85 Binary Enhancements Package.

(Advertisement)
HP-85A "GARAGE SALE"

We have an HP-'85A and assorted other material that we wish to sell. This particular machine has been used for back-up and light program development. Although we don't keep logs on our computers, we believe this machine has had less than 150 hours of operating time and no more than 2 1/2 rolls of printer paper run through it. It has 32K of memory, a ROM drawer, Assembler ROM, I/O ROM, Printer/Plotter ROM, Mass Storage ROM and Advanced Programming RON. It comes with complete documentation, cover and several tapes of software, including user library software, text editor and Visicalc. An outstanding value at $1800, or we'll take the best offer received by July 15, 1984.

NEW HP DATA TAPES: We have 15 unused Hewlett-Packard brand HP-85 data tapes for sale. Price $11 each.

HP BLACK IMAGE THERMAL PAPER: We have 5 rolls of HP brand black image thermal paper for sale. Price $12 each.

USED DATA TAPES: Why not? Especially since these tapes were used just once to record back-ups for storage or archives. 18 tapes, 3M and HP brand: $5 each.

Prices include postage. Contact NEWS80S, P.O. Box 1329, Redondo Beach, CA 90278.

news80s

The Microcomputer Journal For HP Series 80

News80s ran for 12 issues between 1982 and 1984
(#1, #2, Special Issue and #3 thru #11).

It was an independent newsletter edited by Dale Flanagan for:
HP-83, HP-85, HP86 and HP87 Personal Computer users.

Scanned and converted by M. A. Cragg