

The HP 35s

Gene Wright



At last, the BIG ENTER key is back. It's as if HP did a LASTx and brought back the BIG ENTER key. Yes, this was a conscious choice by HP management and can be taken as a signal to all classic HP calculator users that HP has not forgotten their roots. It almost seems as if they are apologizing for their recent wanderings in the wilderness of TI Land.

What's different in the 35s?

Industrial Design. The industrial design of the new 35s is much improved over the 33s. Color choices have much better contrast than many recent machines (33s, 39g+, etc). The yellow, blue and pink colors stand out vividly from the black theme of the machine. This is truly a handsome machine. On the back of the unit, you'll find two small rubber feet at the top and an entire rubber bar across most of the bottom to help keep it in place when being used on a desk. Again, HP heard the 33s feedback. Finally, the folded keyboard look of the 33s is history and we have a normal horizontal layout of rows of keys. So, although the HP 35s is a redesigned HP33S, it has altogether a much more "HP-like" look and feel to it.

Algebraic mode. Algebraic mode might now really be called an Equation Operating System (EOS) rather than a classic AOS. You can key in a long calculation much like a formula and it is evaluated when you press the BIG ENTER key. Need to change a value in the previous calculation? Press the left arrow key and the cursor appears at the end of the previous input allowing you to change any values you wish and press the BIG ENTER again to evaluate the new expression. The algebraic mode also keeps a stack of the four previous numerical results; these are available by pressing the RDN key. In a bow to algebraic users (who are needed to help sales) and given their frequent use in EQN mode, parentheses are an unshifted key function. These are "intelligent" parentheses that put an open and close parenthesis into an expression or equation when pressed, so only one key location is used. All functions in algebraic are prefix now (except factorial/gamma which are postfix and arithmetic operations which are infix). To compute the sine of an angle, you press SIN and are prompted SIN() with the cursor between the parentheses. If a function requires two arguments, such as a permutations, the prompt is nPr(,) with the cursor placed before the comma awaiting input of the first argument. Once entered, press right arrow to move beyond the comma, enter the second value and the BIG ENTER key will compute the result.

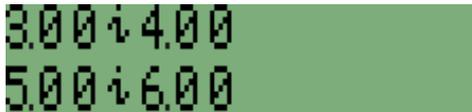
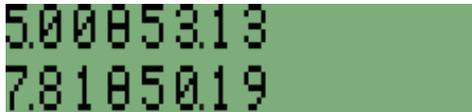
Key layout. Key layout has been reworked to group similar functions together better. Statistics functions are in the lower right corner, conversions are in the middle of the keypad, scientific functions are in a row above the BIG ENTER key, programming functions are on the top rows of keys, slightly to the left of center, etc. The EQN, SOLVE, and integrate functions are now on a dedicated key right above the shift keys to better highlight one of the basic features of the 35s. Did I mention the BIG ENTER key is back?

Function set. There are a few deletions. Cube and cube root are gone. GTO(i) and XEQ(i) are also gone. There are also no direct polar/rectangular conversions (see below).

What's new and improved in the 35s?

Complex-number support is much better on the 35s in many areas. There is a dedicated **i** key on the keyboard for entering complex numbers. To enter $5+6i$, press $5i6$ and the number is not only displayed on one line of the stack, it only uses one stack register. The 35s therefore has a 4-level complex stack.

There are three complex display modes available: XiY , $r\theta a$, and $x+yi$ (algebraic only). As a side effect, the polar/rectangular conversions are gone. You change how a complex number is displayed using the DISPLAY menu to change "modes."

XiY mode:	
$r\theta a$ mode:	
$x+yi$ mode:	

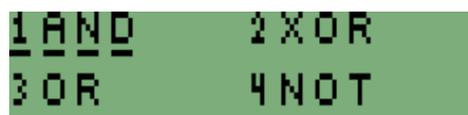
If you wish to decompose a complex number to get the angle when in polar mode, use the function ARG. To get the magnitude, use ABS. To get the X and Y coordinate of a complex number, the transformations $X = r \cos \theta$ and $Y = r \sin \theta$ must be used.

Some functions that would be expected to generate complex numbers as an answer do not. Some examples: the square root of -2 returns an error rather than a complex number, squaring a complex number must be done using Y^X since X^2 generates an error, and the natural log can be taken of complex numbers but not the common log. Operations on complex numbers that give a real result return a complex number with an imaginary part of zero.

There is no "alphabetic" data type, as there was on the HP-41 and HP42S models, but as on the 32S and 33S models, an equation can be displayed as a message.



The 35s continues the menu approach adopted with the 33s, much to the disappointment of those who liked the 32SII approach. A choice must be made using the arrow keys and ENTER or by pressing a digit key. The top row keys are not used to select a choice.

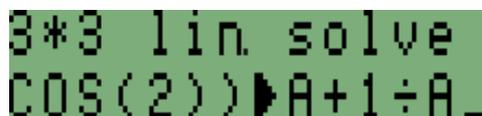


The 35s supports another new data “type”: Vectors. These are entered using the blue shift of the parentheses keys to open the [] symbols. Vectors can be one, two or three dimensional and only take up one stack register or one data register if stored. Not only does this allow for things like easy dot and cross products, but other tricks are possible. See the indirect-register store/recall routine that will appear in the next issue.

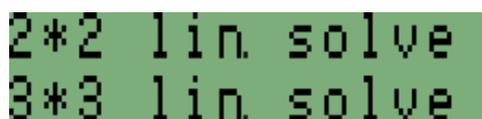
Using the Equation Mode has also been improved. First, there is no longer an arbitrary limit of 255 characters in an equation. Equations can be as long as you like as long as they will fit into memory. Second, pressing the left arrow key when an equation is on the bottom line of the screen allows you to **non-destructively** move within an equation. You can then delete characters and retype them without deleting from the end, as the 33s required. Another good change is that intermediate result can be stored within an equation and reused later. For example, if you wish to compute $B=(A+1/A)/(1-A)$, where A is equal to $SIN(e+2) / COS(2)$, the equation may be input as:

$$B=((SIN(e+2)/COS(2))STOA+1/A)/(1-A)$$

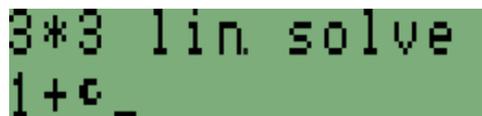
which certainly takes fewer keystrokes. STO is not displayed as STO in the equation but as the now familiar darkened store triangle. Equations can also reference stack register contents in either mode. See the separate article included in this issue for details.



There are now two built-in equations on the 35s solving 2x2 and 3x3 linear equations. Pressing Solve when one of these equations is on the bottom line of the display begins the solution for the linear equation. These use lettered variables for storage.



Using a constant in an equation now shows the constant’s symbol rather than its numerical value. For example, to add 1 centimeter per second to the speed of light, the equation would be shown as at right.



There’s also an additional constant built-in: the value of e is now the 41st constant.

Finally! 30K of memory that might really be useful!

One of the biggest complaints about the 33s (other than the keyboard and color scheme) was the inability to use the memory provided. No longer.

Line-number addressing. The 35s allows for line number GTOs and XEQs within the still limited 26 global labels. As a historical note, this is the way the never introduced HP95C was to work.

The usual issues with line number transfers (such as exist on the 12c models) do not occur on the 35s. That's because the line number GTO and XEQ statements dynamically renumber themselves if you insert or delete steps. For example, if you have a program:

Original program	and then decide to insert	Revised program
A001 LBL A	an ABS instruction	A001 LBL A
A002 x=0?	between lines A001 and	A002 ABS
A003 GTO A005	A002, the new program	A003 x=0?
A004 1/X	will be revised by the	A004 GTO A006
A005 PSE	35s to look like this,	A005 1/X
	<u>automatically:</u>	A006 PSE

When a GTO or XEQ instruction is keyed into a program, the 35s “locks” in upon the destination location. If that destination moves because of inserted or deleted steps, the transfer instructions are updated. If the actual destination step itself is deleted, the locations in any GTO or XEQ instructions will point at the new instruction occupying that location, which used to be the instruction in the step after the now deleted step. While additional global labels would still be useful, it is now possible to write long programs without using up all 26 labels.

In case you didn't notice in the program listing, steps are now listed as Label-Letter followed by a 3-digit number, rather than 4 digits on the 33s. Partly this was to make sure that lines such as A094 GTO B010 would fit in the display rather than scrolling off (some instructions DO scroll off the display to the right, but if the 4-digit addresses had been kept, ALL GTO and XEQ instructions would have scrolled off to the right). Also, since nearly every instruction takes 3 bytes, a 10,000 line program would have used all available memory anyway. If an individual global label would have more than 999 steps (which would use around 3000 bytes), it will need to be broken into two or more pieces.

To execute a program from the keyboard, you press XEQ label-letter but are now prompted for a three-digit line number. Rather than keying in 001 each time you wish to start a program from line 001, the shortcut XEQ label-letter ENTER starts execution at line 001 of the label. Fortunately, the owner's manual even uses this shortcut.

In addition, the 35s now has up to 20 subroutine levels available.

Lots of registers. The other big change on the 35s is registers.

All memory registers, including stack registers, on the 35s use 37 bytes per register, regardless of what is stored within the register. This is to allow each possible data type to be stored in a register. Each register now has 1 byte set aside to indicate what is contained within the register. Each real number uses 12 bytes –

8 for the mantissa, 1 for the sign, 2 for the exponent, and 1 for the sign of the exponent. Since a register can hold a 3-D vector containing 3 real numbers, the register must have 36+1 bytes available. Complex numbers and 2-D vectors may only require 24+1 bytes, but the register still has 37 bytes allocated to it.

The user now has up to 801 indirect registers available. Although variables A-Z, stack and statistics registers are always allocated, these indirect registers are dynamically allocated from available program memory at the rate of 37 bytes per register based upon the highest numbered non-zero register. Indirect registers are accessed using the I and J variables as indices – the dedicated i register is gone. Store your index in either I or J and use (I) or (J) to perform indirect operations.

Indirect registers start at register 0 (a zero in I or J). To set aside 100 indirect registers, you would store a non-zero value into indirect register 99 (for 0 through 99). If you wish to keep 100 registers allocated even if register 99 might later contain a zero value, you need to put a non-zero value into register 100. Storing a non-zero value in register number “top+1” will ensure that the registers below are not released.

Since each indirect register can hold a 3-D vector, there are some interesting data-packing opportunities. See the indirect register store and recall program in the next issue to see how to store nearly 2400 real numbers in a 35s fairly easily.

The screen image at right indicates that 101 indirect registers have been allocated (registers 0 through 100). Program memory has been reduced by 3700 bytes (37 x 100 registers). An empty 35s with all 801 indirect registers allocated would still show 555 bytes of available program memory.

<u>I</u> VAR	2 PGM
101	26,455

<u>I</u> VAR	2 PGM
801	555

A common complaint about looping indirectly on the 33s was the presence of the index register i in the middle of the 33-register address space. A couple of changes have eliminated that problem.

The lettered variables A through Z are now referenced indirectly using -1 through -26. The statistics registers are now -27 through -32. True, the index registers I and J are in the middle of *that* address space, but routines can now be written using the indirect registers, which should eliminate the previous problem. If you wish, you can now write a program to loop through 801 consecutive registers.

Bugs fixed. The major bugs in the 33s are fixed in the 35s, including the combinations bug for large values of n and r, the HMS bug, and the 0 SEED bug. The COS bug remains for values very near 90 degrees, however.

Another good fix is the correction of the fraction symbol below the decimal point. Over the past few years, this symbol has increasingly looked as if it meant “a to the b divided by c power” rather than a mixed fraction.

Quirks. The 35s does have a few things to get used to. The command line is more similar to that found on the graphing models, as you can key in all sorts of strange entries which are not processed for validity until ENTER or a function key is pressed. Probably because of this, numbers entered for BASE operations are always assumed to be in decimal mode regardless of the BASE setting unless the base suffix is added. For example, if you are in HEX mode, you must append the “h” suffix to numbers keyed in or the command line will assume DEC mode and give a syntax error. Another quirk is that a common shortcut to enter a fraction of 2/3 by pressing 2 . . 3 no longer does so – the sequence 0 . 2 . 3 must be pressed now.



Memory Usage

The chart below indicates how memory is used on the 35s compared to the 33s that it is replacing. My early unit shows unexpected MEM readings at times. For example, 35 bytes are used up by placing a “1” into a program line. But no additional memory is used to place a R/S instruction right after the “1”. The chart therefore shows what usually happens. Perhaps there really is a free lunch?

Data	HP 33s	HP 35s
Variables and statistics data	No bytes. Variables (including i and the statistics registers) take up no user memory. Always allocated.	No bytes. Variables (including the statistics registers) take up no user memory. Always allocated.
Numbers in program lines	15 bytes. No short form for frequently used values, but constants in program lines take 3 bytes unless they are in an equation.	35 bytes. No short form. The number 1 takes just as many bytes as -1.23456789E-55.
Instructions in program lines	3 bytes	3 bytes
Equations in program lines	3 bytes + 1 byte for each character	3 bytes + 1 byte for each character
Numbers and operations in equations	Each entry in the equation list takes 6 bytes + 1 byte for each character (255 maximum)	Each entry in the equation list takes 6 bytes + 1 byte for each character (NO maximum)

Execution Speed and Battery Life

The new HP 35s runs at about the same speed as the HP33S, but it is a mixed set of results. For example, using the classic looping test as shown below, the HP33S completes the test in 29 seconds, while the 35s completes the test in 32 seconds.

HP33S

```
LBL A
1000
STO A
LBL B
DSE A
GTO B
RTN
```

HP35s RPN

```
LBL A
1000
STO A
DSE A
GTO A004
RTN
```

Another speed test is the classic “Add 1” test. In 60 seconds, the 33S counts to 10,042, while the 35s counts to 3931 in RPN mode and only 989 in ALG mode using the programs below. This is of course much slower.

HP33S

```
LBL A
+
GTO A
```

HP35s RPN

```
LBL A
+
GTO A002
```

HP35s ALG

```
LBL A
LASTx+1
GTO A002
```

The HP 35s is powered by two CR2032 batteries. HP estimates battery life at nine months if used one hour a day. Only time will tell.

Documentation

HP provides 55 learning modules on the HP 35s calculator webpage, covering a wide variety of topics. In addition to the usual modules, there are specific modules for working with the indirect registers, the increased complex number functionality, using line addressing rather than labels in programs, accessing the stack registers within a program, and a larger number of modules dealing with sample application problems. There is even a module that refers to Datafile and HPCC.org.

The User’s Guide contains 386 pages. Nice and complete.

Conclusion

While still an evolutionary development of the platform begun with the 32s nineteen years ago, the 35s offers much to a user wanting a low-cost RPN calculator and fixes many of the shortcomings of the 33s. Is it the perfect calculator? No, but it certainly is a big improvement and an indication that HP listens to its users. Welcome back, HP! Now, what’s next?