34S USER'S MANUAL

Dear customer, what you hold in your hands is the result of careful customizing. This calculator is based on the HP-30b. I.e. its mechanics and its hardware are of the new HP-30b as is, while its software and user interface are newly written from the scratch to make this little device a compact scientific calculator like you have never had before.

The function set of the 34S is based on the one of the renowned HP-42S [1]. It is extended to completely include the functionality of the famous HP-16C (we even added and expanded some commands), the fraction mode of the HP-32SII, statistical distributions as featured by the HP-21S, and even more mathematical and statistical functions like

- Euler's Beta function, incomplete Beta and Gamma, the error function, Fibonacci number calculation, Lambert's W as well as
- many distributions and their inverses like Binomial and Poisson, exponential and geometric, Gaussian for arbitrary mean and standard deviation, Weibull for reliability analysis,
- extended date and time calculations based on a real time clock, and more.

The 34S features a fixed amount of 99 general purpose registers, 500 program steps, 3 programmable hotkeys, 99 user flags, a 4 level stack extended by a complex LASTx register, and a xx byte alpha register capable of holding extended Latin as well as Greek upper and lower case fonts in two sizes.

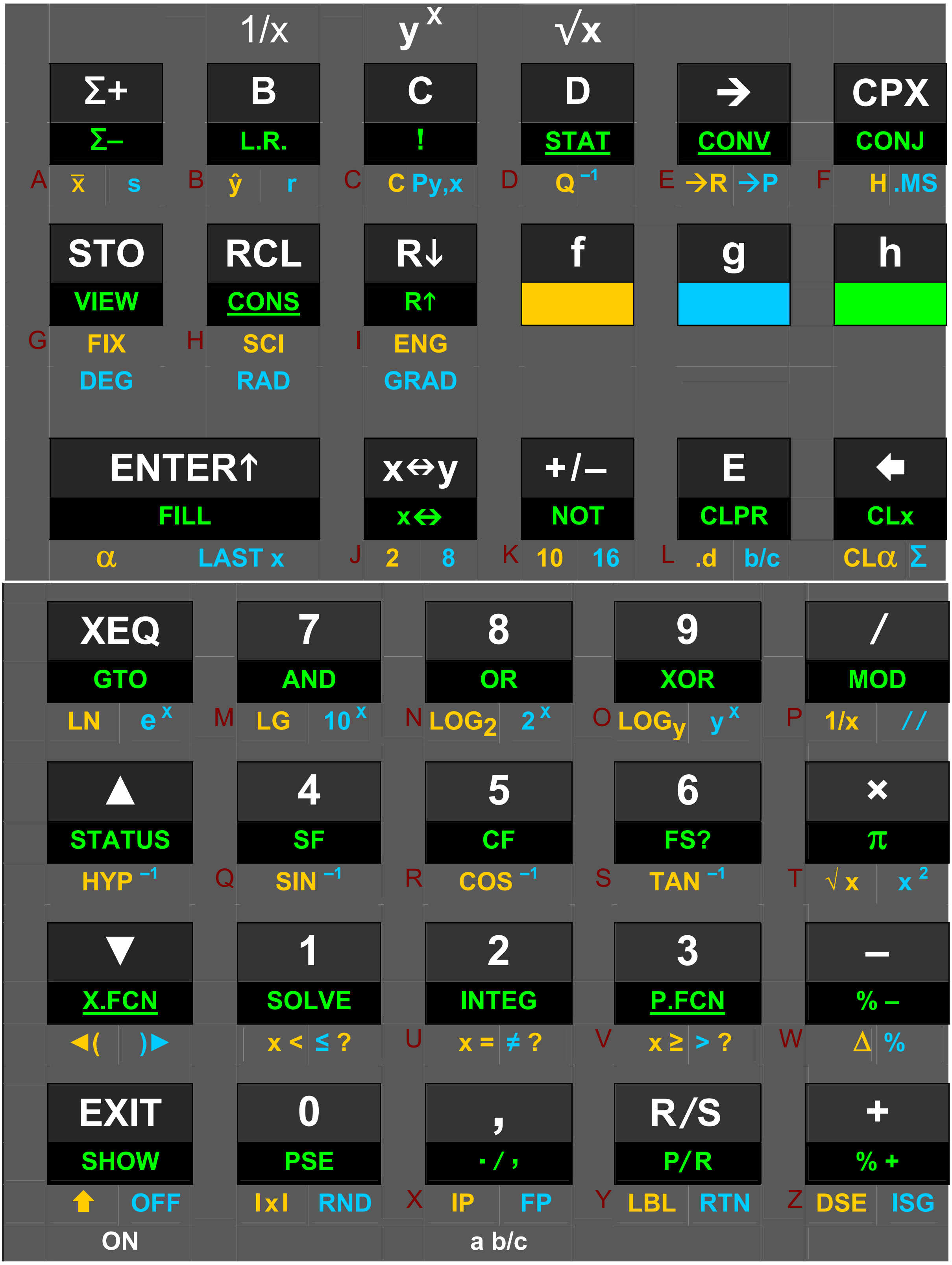
If you know how to deal with a good old HP RPN calculator, you can start with your 34S right away. To show you the features of the 34S completely, however, we wrote this little manual. It starts with an overview on the active keyboards in various modes, so you know where to find what you look for. It continues with some tables about addressing and short paragraphs explaining the display and the indicators used to tell you what's going on. The major part of this little booklet is taken by the index of operations, catalogue contents, provided constants and conversions. It closes with a list of messages this calculator will display if special input conditions prevent it from executing your command.

The 34S is the result of a serious collaboration of two individuals, one Australian and one German, though we did this in our free time, and so you may call it our hobby to some extent. We have checked everything we could think of to our best knowledge, so we hope it being free of bugs. We cannot guarantee this, however, nor can we bear any liability for any errors in calculations nor their possible consequences. Nevertheless, what we can promise is we will improve this device wherever it will turn out being necessary – so if you detect any strange results, please tell us, and if it is unveiled being an error you will get an update as soon as we have one.

Enjoy!

Paul Dale and Walter Bonin

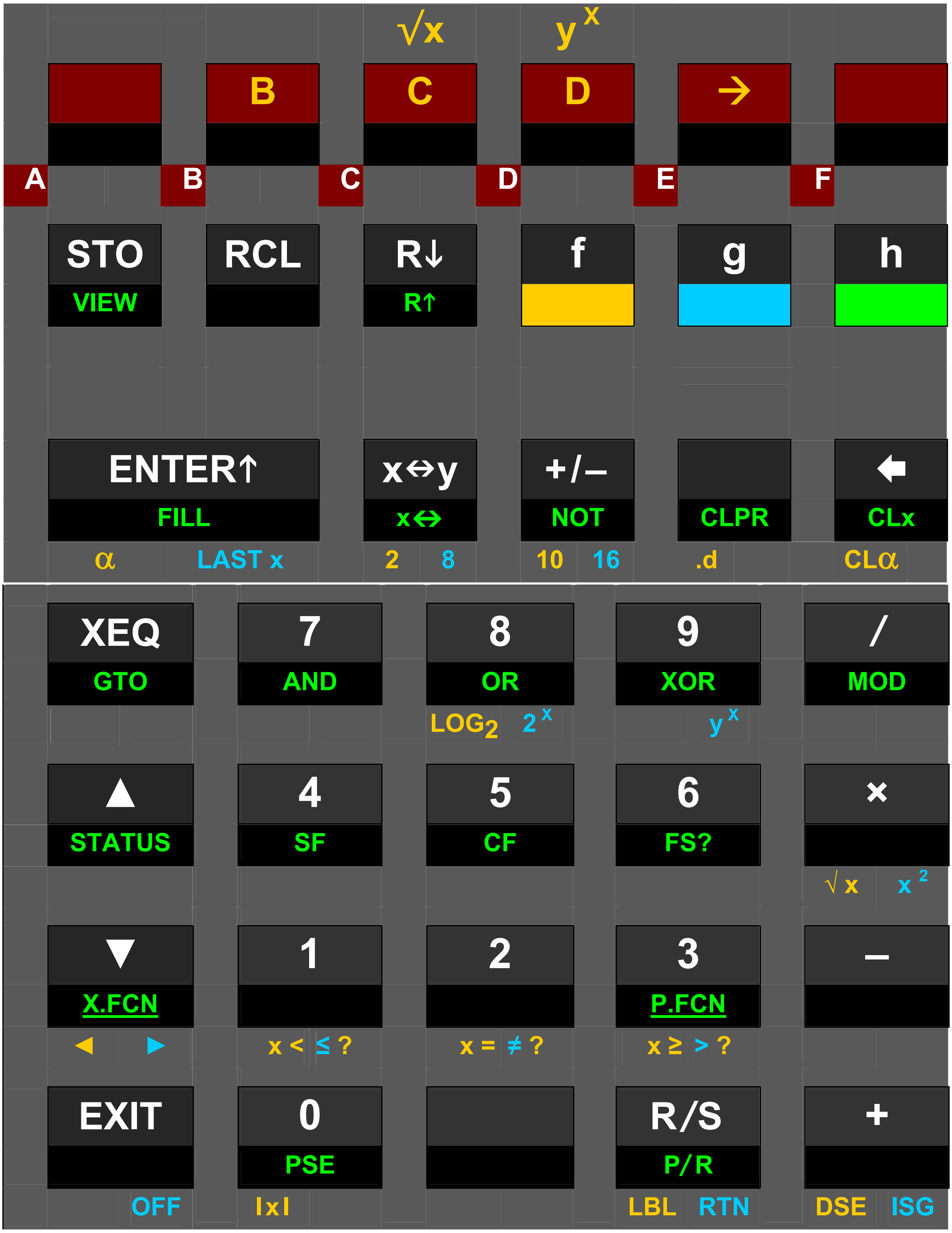
[1] Due to hardware restrictions, matrix math cannot be supported by the 34S. Sorry for this.



Keyboard layout:

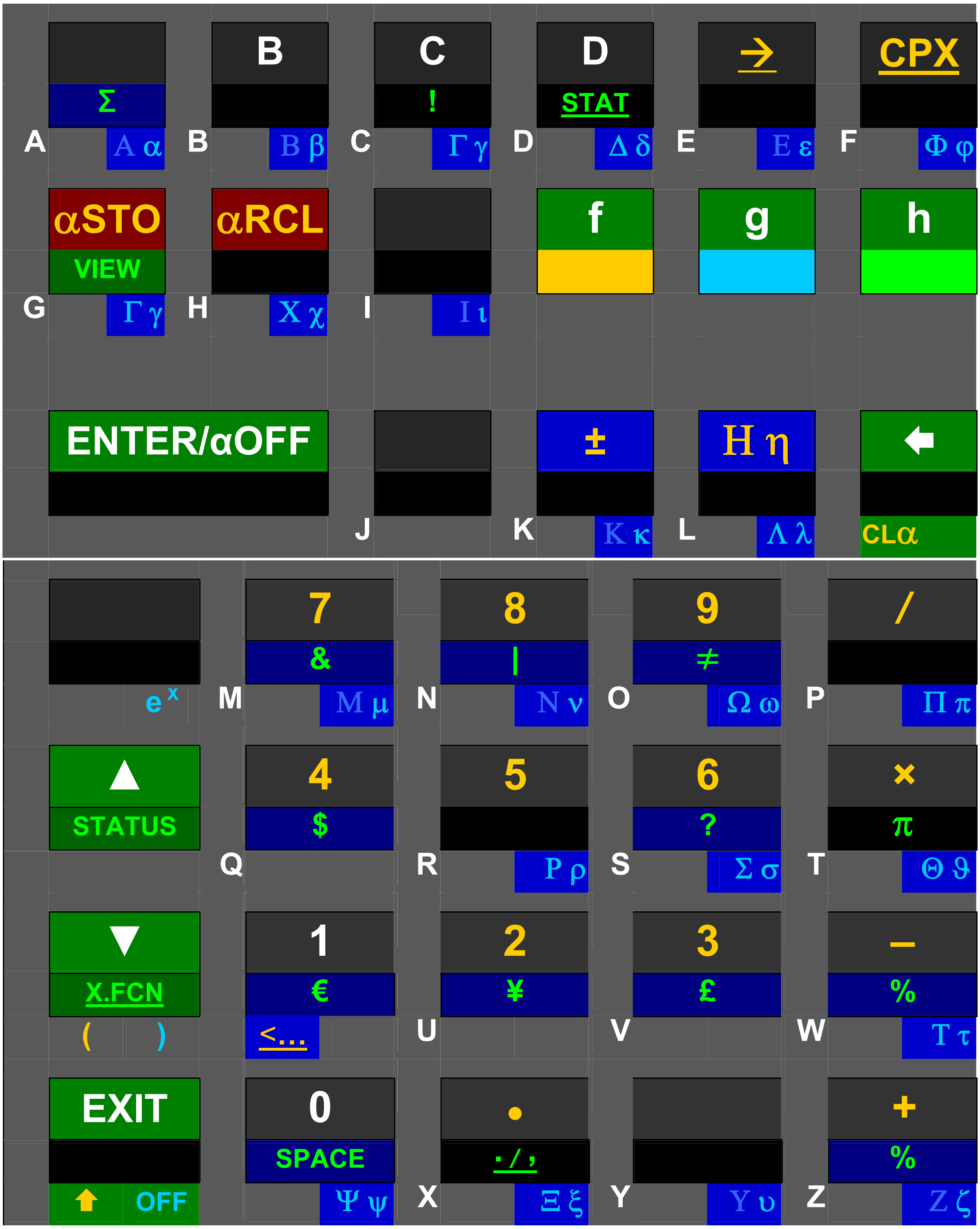
- CPX may be combined with many arithmetic and transcendental functions.
- → does conversions combined with H, H.MS, DEG, RAD, GRAD, 2, 8, 10, 16.
- The keys B, C, and D immediately call the respective user programs if defined.
- Calculator modes are as described in the paragraph about indicators below.
- Labels calling catalogues are underlined.

Please see the index for more.

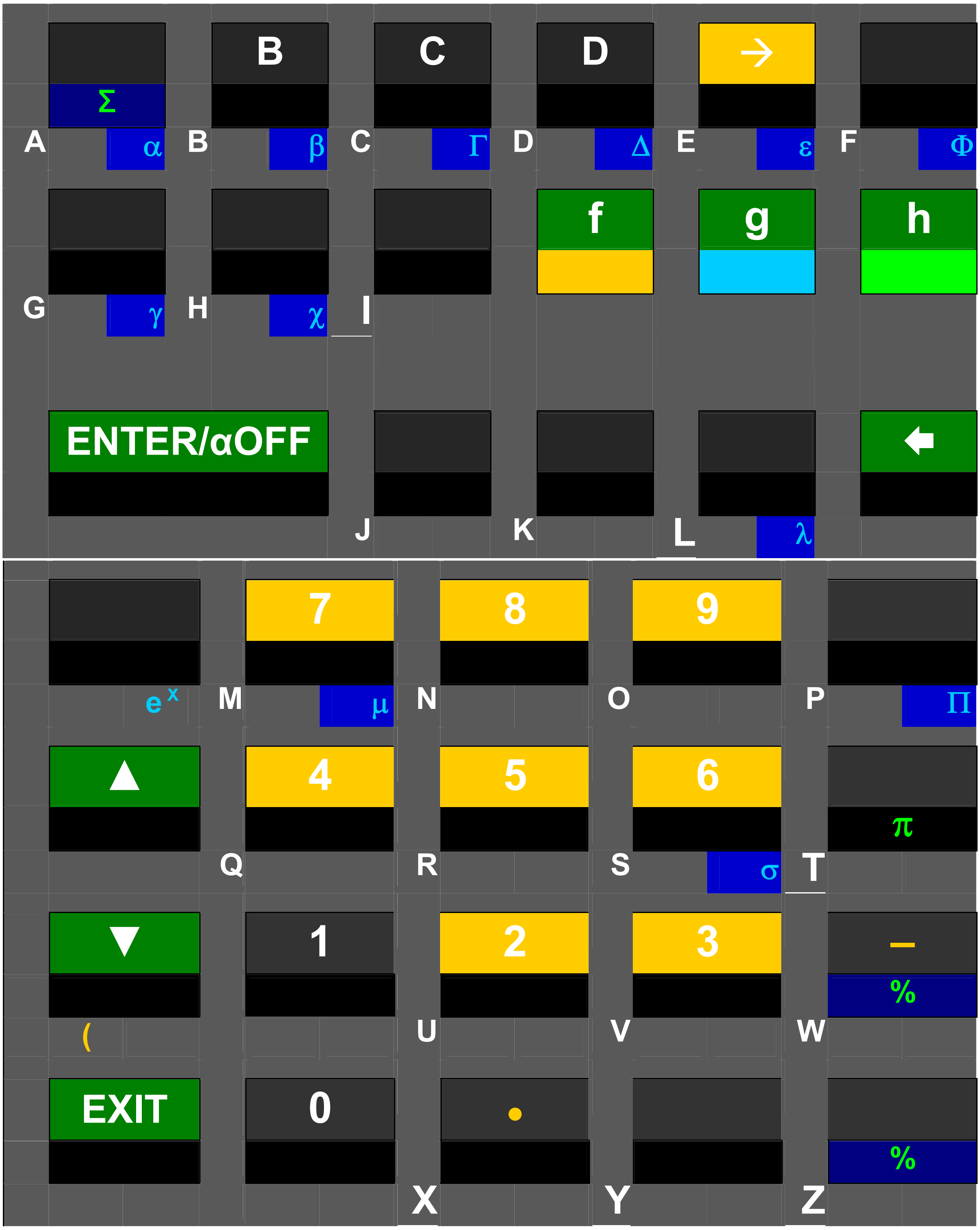


Virtual active keyboard in hexadecimal mode. → is for addressing and temporary display in other bases only (see the index below). Primary functions of the top 6 keys will be numeric input, so their default primary functions are accessed using f .

In the other integer bases, the active keyboard will look alike, but those of the 6 keys not needed for numeric input there will keep their default primary functions, except Σ+ and CPX . Attempts to enter an illegal digit will throw an error.



Virtual active keyboard in alpha mode. Therein, the 15 bytes alpha register is displayed in the dot matrix and most input is inserted directly into this register – the numeric line is accessible by commands only. All labels except those shown on green or red background call alpha catalogues or insert characters into the alpha register directly; those shown on blue background deviate from the standard printed on these locations. Primary function of most keys will be inserting the dark red letter printed bottom left of such a key. f is used for reaching the key tops there, and g leads to homonymic Greek letters, where applicable. There are 3 exceptions: **η** is accessed via f E, and **τ** via g – (one key below **T**), and **ψ** via g 0 (below PSE). Omicron is not featured since looking exactly like **O**. Generally, ⬆ toggles upper and lower case, and PSE inserts a space. Currency symbols may be accessed via h-shifted keys 1 through 4. The catalogues h STAT, f ➡, f CPX, f 1, and h ./, offer even more characters (see below).



Virtual active keyboard in temporary alpha mode, entered when calling catalogues or comparisons. Functions printed on green background allow for catalogue browsing, error recovery, and leaving. ➡ and 2 … 9 are primary in comparisons but need f-shift in catalogues. See previous page, addressing tables, and catalogues for more.

MEMORY

Stack

L	I

T	
Z	
Y	
X	Display

General purpose registers

00
01
02
…
…
86
87 $\Sigma \ln x$
88 $\Sigma \ln^2 x$
89 $\Sigma \ln y$
90 $\Sigma \ln^2 y$
91 $\Sigma (\ln x \cdot \ln y)$
92 $\Sigma (x \ln y)$
93 $\Sigma (y \ln x)$
94 n
95 Σx
96 Σx^2
97 Σy
98 Σy^2
99 Σxy

Flags

00
01
02
…
…
96
97
98
99
B Overflow
C Carry
D Danger

Alpha

Program memory

001
002
003
…
…
496
497
498
499
500

In complex arithmetic, stack registers **X** and **Y** contain the real and imaginary part of the first complex number, while **Z** and **T** carry the second. Stack register **I** takes the imaginary part of the last argument if a complex function is used. See LASTx.
As long as no complex function is used, **I** may be taken as another general purpose register.

General purpose registers 01 and 02 may be used for some statistical distributions. Registers 87 through 99 take the statistical sums indicated as soon as **Σ+** is used.

Flag D is set if “NaN” and “infinite” are allowed as results of commands.

COMPARING AND ADDRESSING REGISTERS

1	User input	x< ?, x≤ ?, x= ?, x≠ ?, x≥ ?, or x> ?				RCL, STO, αRCL, αSTO, VIEW, x≷, DSE, ISG, DSZ, ISZ, FIX, SCI, ENG, DISP, BASE, CF and the other flag commands, CB and many more bit commands		
	Dot matrix display	**OP _** (e.g. x > _) Temporary alpha mode is on.				**OP _** (e.g. RCL _)		
2	User input[2]	0 or 1	X, Y, Z, T, L, or I	ENTER↑ [3] closes alpha.	→ closes alpha.	X, Y, Z, T, L, or I [4]	*Number of register or flag or bit(s) or decimals* [5]	→
	Dot matrix display	**OP *n*** e.g. x ≤ 0 ?	**OP *x*** e.g. x ≥ y ?	**OP r_**	**OP →_**	**OP s *x*** e.g. SCI sZ	**OP *nn*** e.g. SF 15	**OP →_** (indirect addressing)
3	User input	Compares ***x*** with the number **0**.	Compares ***x*** with the number on stack level **Y**.	00 … 99	Look right for more about indirect addressing.	Sets scientific display with the number of decimals specified in stack level **Z**.	X, Y, Z, T, L, or I	00 … 99
	Dot matrix display			**OP r *nn*** e.g. x ≠ r23?			**OP →s *x*** e.g. VIEW →sL	**OP → *nn*** e.g. STO →45
				Compares ***x*** with the number in register **23**.			Shows the content of the memory where **LASTx** is pointing to.	Stores ***x*** into the location where register **45** is pointing to.

[2] For RCL and STO, any of +, −, ×, /, ▲, or ▼ may precede step 2. See the index of operations. RCL Σ+ recalls Σx and Σy.

[3] This step may be skipped for register numbers >19, since pressing a numeric key >1 will close temporary alpha mode implicitly in comparisons.

[4] Only RCL and STO require ENTER↑ Z or ENTER↑ T , respectively, for accessing one of these two stack levels here. – Some legal stack operations may be useless, e.g. **x<>sX** . It is user responsibility not to mix pairs in complex mode, since a complex operation will always affect two registers: the one specified and the one following this.

[5] Register and flag numbers may be 00 … 99, number of decimals 0 … 11, integer bases 2 … 16, bit numbers up to 63 and integer word size up to 64 bits. For numbers <10, you may key in e.g. 5 ENTER↑ instead of 0 5 . There are three additional flags addressed via B, C, and D .

ADDRESSING LABELS

1	User input	GTO, XEQ, LBL, Σ, π, SOLVE or INTEG [6]				
	Dot matrix display	**OP _** (e.g. `GTO _`)				
2	User input	B, C, or D	ENTER↑ turns alpha mode on.	→ [7]		*2-digit numeric label* 0 0 … 9 9
	Dot matrix display	**OP '*name*'** e.g. `Σ 'B'`	**OP '_**	**OP →_** (indirect addressing)		**OP *nn*** e.g. `LBL 07`
3	User input		*Label* [8]	X, Y, Z, T, L, or I [9]	0 0 … 9 9	
	Dot matrix display		**OP '*name*'** e.g. `SLV'F1µ'`	**OP →s *x*** e.g. `INT →sY`	**OP → *nn*** e.g. `XEQ →44`	
			Solve the function **F1µ** (with F1µ keyed in).	Integrate the function which's label is on stack level **Y**.	Execute the routine which's label is in register **44**.	

[6] SOLVE and INTEG will be displayed and listed as SLV and INT, respectively. The routines labelled B, C, and D may be called for execution directly via B , C , or D , respectively, without pressing XEQ before.

[7] Works with all these operations except LBL.

[8] Such a label may consist of up to 3 alphanumeric characters. The 3rd character terminates entry and closes alpha mode. For labels with less than 3 characters, a closing ENTER↑ is mandatory.

[9] There is no need for switching to alpha mode before.

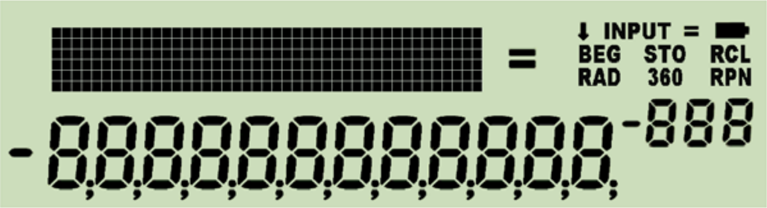
ADDRESSING ITEMS IN CATALOGUES

1	User input	X.FCN, P.FCN, STAT, CONS, CONV	CPX or STAT in alpha mode	./, <..., → in alpha mode
	Dot matrix display	**Shows 1st item in selected catalogue.** Temporary alpha mode is on. (e.g. BC? in P.FCN)	(e.g. Á in CPX)	(e.g. , in PUNCT)
2	User input	ENTER↑, EXIT, ▼, ▲, or 1st character (e.g. F)	ENTER↑, EXIT, ▼, ▲, or letter (e.g. O)	ENTER↑, EXIT, ▼, or ▲ (e.g. ▼)
	Dot matrix display	**1st item starting with this character *)** (e.g. FB)	**1st item starting with this letter *)** (e.g. Ó)	**Next item in catalogue** (e.g. ")
3	User input	ENTER↑, EXIT, ▼, ▲, or 2nd character (e.g. S)	ENTER↑, EXIT, ▼, or ▲ (e.g. ▼)	
	Dot matrix display	**1st item starting with this sequence *)** (e.g. FS?)	**Next item in catalogue** (e.g. Ò)	
4	User input	ENTER↑, EXIT, ▼, or ▲ (e.g. ▼)		
	Dot matrix display	**Next item in catalogue** (e.g. FS?C)		
		… continue browsing until the desired item is displayed		
		(e.g. FS?F).	(e.g. Ö).	(e.g. :).
n	User input	ENTER↑	ENTER↑	
	Dot matrix display	**Selected mode or operation** (e.g. FS?F) Calculator returns to the mode set before and executes or inserts the selected operation.	**Character inserted into alpha register** (e.g. Östl. Seite:) Character is inserted on the cursor, any character(s) there or trailing will be shifted to the right. Calculator leaves the catalogue returning to alpha mode.	

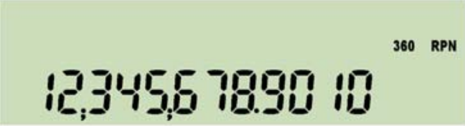
*) or first item following alphabetically in this catalogue if specified character or sequence is not found.

DISPLAY

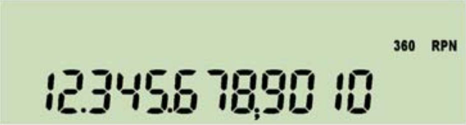
The display features 3 sections: numeric, dot matrix and fixed symbols. The numeric section features a minus sign and 12 digits for the mantissa, as well as a minus sign and 3 digits for the exponent. The dot matrix is 6 dots high and 43 dots wide, allowing for some 7 to 12 characters, depending on their widths. The fixed symbols (except the big "=") are called annunciators, and are for indicating modes (see next paragraph).



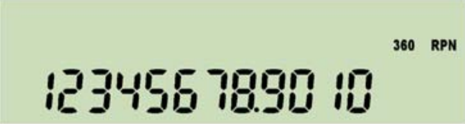
For floating point numbers, the mantissa will be displayed adjusted to the right, the exponent to the left. Within the mantissa, either points or commas may be selected for radix marks [10], and additional marks may be chosen to separate thousands. Assume the display set to FIX 4, then 12.345678901 millions may look like:



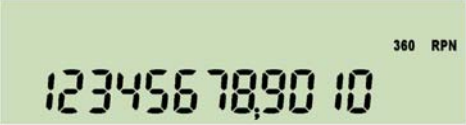
or



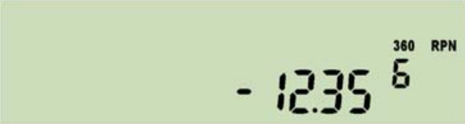
with thousands separators on, and without them like:



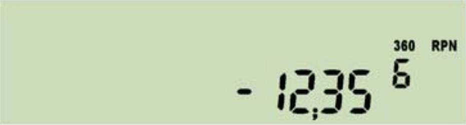
or



With ENG 2 and after changing the sign, the same number looks like this:



or



The dot matrix section is used for

1. indicating some more modes than the annunciators allow, adjusted to the right,
2. passing additional information to the user, adjusted to the left.

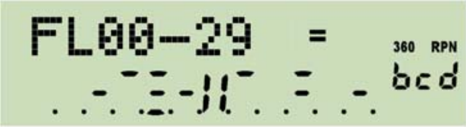
If two or more requests concur for display space there, then the items will be shown according to their priorities. Priorities are as follows

1. error messages as described in a paragraph next to the end of document,
2. special information as explained below,
3. mode information.

Some commands and modes use the display sections in a special way:

[10] From here on, decimal input is written using a point as radix mark in this manual, though significantly less visible, unless specified otherwise explicitly. By experience, the „comma people" are more capable to read radix points and interpret them correctly than vice versa.

1. STATUS displays the status of the flags in a very compact way, allowing a brief status overview after some training. For example, if the flags 2, 3, 5, 7, 11, 13, 14, 17, 19, and 23 are set, and the labels B, C, and D are defined in program memory, STATUS will display this:

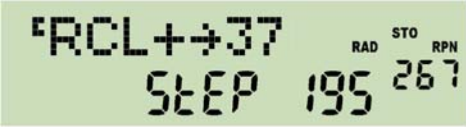
   Within the numeric section, each row of horizontal bars in the mantissa shows the status of 10 flags. When a flag is set, the respective bar turns black. So here the top row of bars indicates flags 0 and 1 being clear, flags 2 and 3 set, and flag 4 clear. Then, a vertical divider separates the first group of five flags from the next. Thereafter, flag 5 is set, 6 clear, 7 set, 8 and 9 clear. In the next row, flag 10 is clear, 11 set, etc.

   Scrolling down by ▼ will display flags 10 – 39, then 20 – 49 etc. until 80 – D. Scrolling up by ▲ reverts this. Alternatively, pressing a digit will show the flags starting with 10 times this digit. The numeric exponent displays the status of the 3 hotkeys top left on the keyboard.

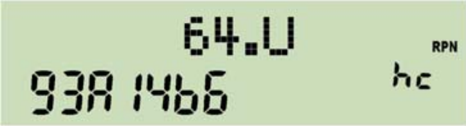
   The STATUS display will disappear when any key is pressed but ▼, ▲, or a digit < 9.

2. In addressing, the dot matrix records the progress as explained in the addressing tables above in detail. During input, the prefixes f, g, and h will be shown until they are resolved.

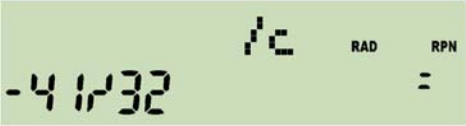
3. In programming mode, the numeric display indicates the program step (001 – 500) in the mantissa and the number of unused steps in the exponent, while the dot matrix shows the command contained in the respective step.

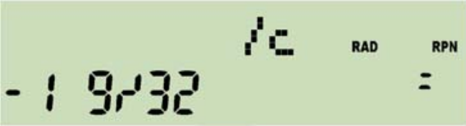
4. In integer modes, numbers are displayed adjusted to the left. Further information is passed in the dot matrix and the exponent. The example shows a display in hexadecimal mode with word size 64, unsigned, and carry set (see next paragraph for more):

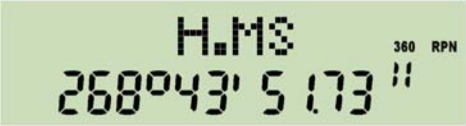
5. In fraction mode, the fraction will be shown in the mantissa section of the numeric display, adjusted to the left. “=”, “Lt”, or “Gt” is indicated in the exponent if the fraction is exactly equal, slightly less, or greater than the floating point number converted, respectively. E.g. -1.28125 will be displayed as follows, depending on the setting:

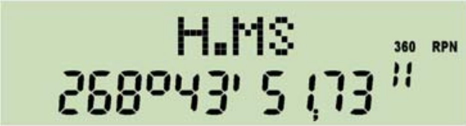
   or

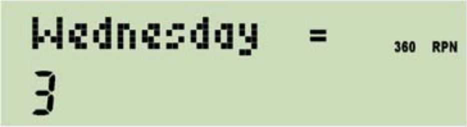
6. In H.MS mode, input format is HHHH.MMSSDD (with the number of hours or degrees limited to 9000) and output may look like this:

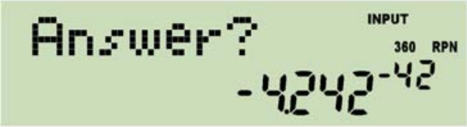
   or

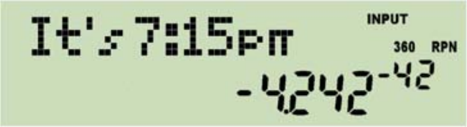
7. Output of the function DAY will look like the following for an input of 01.132010 in M.DY mode (equivalent to inputs of 13.012010 in D.MY or 2010.0113 in Y.MD). The display may look alike for a result of DAYS+.


Wednesday =
3

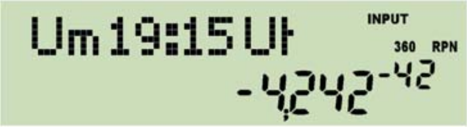
8. In alpha mode, the contents of the alpha register are displayed in the dot matrix while the numeric section keeps the result of the last numeric operation.


Answer?
-4.242^{-42}

Different information may be appended to or inserted in the alpha register. See the commands starting with the letter α in the index of operations below. E.g. αTIME allows creating texts like


It's 7:15pm
-4.242^{-42}

or


Um 19:15 U
$-4,242^{-42}$

INDICATORS

There are a number of indicators signaling the mode the calculator is running in.

Indicator	***INPUT***	**b**	**d**	**h**	**o**	***STO***
Mode name if different	α	**2**			**8**	**PRG**
Set by operation	αON	BASE 2	BASE 10	BASE 16	BASE 8	PRGON
Cleared by operation	αOFF	BASE ≠2 FLOAT FRACT	BASE ≠10 FLOAT FRACT	BASE ≠16 FLOAT FRACT	BASE ≠8 FLOAT FRACT	PRGOFF

Indicator	***360***	***RAD***	**G**	**H.MS**	**/c**
Set by operation	DEG	RAD	GRAD	H.MS TIME >H.MS	BASE 1 FRACT 2nd . in input
Cleared by operation	GRAD RAD	DEG GRAD	DEG RAD	BASE COS, SIN, TAN FLOAT FRACT >HR	BASE ≠1 FLOAT

INPUT, STO, 360, and *RAD* are annunciators (see previous paragraph). Outside integer modes, everything else is indicated in the dot matrix section. The different date modes are signaled by

D.MY or **M.DY**. Defaults Y.MD and FLOAT are not indicated. Time modes (12h/24h) are seen in the time string as shown above. RPN may be shown permanently.

Within integer modes, word size and complement setting are shown in the dot matrix in a format WW**.**C, with C being **1** or **2** for 1's or 2's complement, **U** for unsigned, or **S** for sign-and-mantissa mode. In these modes, sign and first digit of the exponent show the base, a "**c**" in the second digit signals a carry bit set, an "**o**" in the third an overflow. Integer bases are indicated as follows:

Base	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Display	b	3	4	5	6	7	o	9	d	-1	-2	-3	-4	-5	h

Some mode and display format settings may be stored and recalled collectively by STOM and RCLM. RCLM calls a 16-bit word into **X**. Therein, mode data are stored in binary and packed as follows, starting from the right end of the number (least significant bit):

Bit numbers	Meaning	Values and corresponding settings
0, 1	Display format for real numbers	0 = ALL 1 = FIX 2 = SCI 3 = ENG
2 … 5	Number of decimals	0 … 12
6, 7	Angular mode	0 = DEG 1 = RAD 2 = GRAD
8, 9	Date display format	0 = Y.MD 1 = D.MY 2 = M.DY
10 … 12	Curve fit model	0 = LinF 1 = ExpF 2 = PowerF 3 = LogF 4 = BestF
13	Time display format	0 = 24h 1 = 12h
14, 15	Integer sign mode	0 = 2COMPL 1 = 1COMPL 2 = UNSIGN 3 = SIGNMT

For example, FIX 9, DEG, Y.MD, LinF, 24h, 2COMPL is $0000000000100101_2 = 100101_2 = 37_{10}$

and SCI 2, RAD, D.MY, BestF, 24h, UNSIGN is $1001000101001010_2 = 37194_{10}$.

STOM takes such a number, interprets it as a 16-bit word and sets the calculator modes accordingly. Please see the index for the commands mentioned under settings in the table above.

All keyboard inputs are interpreted according to the modes set at input time. – A running program is signaled by a flashing *RCL* annunciator.

INDEX OF OPERATIONS

This lists all functions available on the 34S with the necessary keystrokes. Functions accessible via catalogues will show up there with their names unless specified differently explicitly. Sorting is as follows: 0 … 9, A … Z, α … ω, (,), +, -, *, /, ±, ",", ".", !, ?, ↔, ←, ↑, ↓, →, <, ≤, =, ≠, ≥, >, #, °, %, √, ∞. Super- and subscripts are handled like normal characters prefixed by ↑ or ↓, respectively. Sorting is case insensitive.

The operations will work like on the HP-42S, special bit and integer functions like on the HP-16C, unless stated otherwise under remarks. Functions available on the 34S for the first time on an RPN calculator are highlighted under remarks, as are functions deviating from known ones carrying the same name. If no *parameters* are specified though required, they will be taken from the stack. *Parameters underlined* may be specified using indirect addressing, too, as shown in the tables above. Modes are abbreviated by their indicators. In this column an "&" represents logical AND, and a backslash stands for "all but", so e.g. 2^X works in all modes but alpha.

Name in listings	Keys to press	Works in modes	Remarks
^{c}…	CPX …	FLOAT	Indicates complex operations [11]. CPX may be combined with any function which's name is printed in *italics* in this table. The name in the listing will me merged, e.g. CPX f COS will be listed as ^{c}COS.
10^x	g 10^x	FLOAT	
12h	h X.FCN 12h	FLOAT	Sets 12h time display.
1COMPL	h X.FCN 1COMPL	Integer	Sets 1's complement mode like in HP-16C.
1/x	f 1/x	FLOAT	
	B	FLOAT	Works as long as label B is not defined yet.
24h	h X.FCN 24h	FLOAT	Sets 24h time display.
2COMPL	h X.FCN 2COMPL	Integer	Sets 2's complement mode like in HP-16C.

[11] Such operations work with pairs of adjacent registers. In each such pair, the first register contains the real and the second the imaginary part of the respective complex number. Unless stated otherwise explicitly, where a real function works with $\boldsymbol{x}$ (and maybe $\boldsymbol{y}$), its complex sibling works with $\boldsymbol{x} + i\,\boldsymbol{y}$ (and maybe $\boldsymbol{z} + i\,\boldsymbol{t}$). Where a real function works with a register at address $\boldsymbol{a}$, the respective complex function works with the registers at $\boldsymbol{a}$ and $\boldsymbol{a} + 1$.

Name in listings	Keys to press	Works in modes	Remarks
2^x	g 2^x	\α	
ABS	f |x|	\α	
	CPX f |x|	FLOAT	Returns the magnitude $x = \sqrt{x^2 + y^2}$ and clears **Y**.
ACOS	g COS^{-1}	FLOAT, H.MS	
ACOSH	g HYP^{-1} COS	FLOAT	
ALL	h X.FCN ALL	FLOAT	Selects “all” display format.
AND	h AND	Integer	
		FLOAT	Works like AND in HP-28S, i.e. ***x*** and ***y*** are interpreted before executing this operation. Any real number except 0 is “true”.
ANGLE	h X.FCN ANGLE	FLOAT	Calculates the angle between the positive x-axis and the straight line connecting the origin with the point (***x, y***).
ASIN	g SIN^{-1}	FLOAT, H.MS	
ASINH	g HYP^{-1} SIN	FLOAT	
ASR	h X.FCN ASR ***n***	Integer	Works like *n* (1 … 64) consecutive ASRs on HP-16C. In indirect addressing *n* may be 0.
ATAN	g TAN^{-1}	FLOAT, H.MS	
ATANH	g HYP^{-1} TAN	FLOAT	
BASE	h X.FCN BASE ***n***	\α	Sets the base for integer calculations, with $2 \le n \le 16$. Popular bases are directly accessible on the keyboard. Furthermore, BASE 0 calls FLOAT, and BASE 1 calls FRACT. Actual base setting is indicated as explained above.
BASE2	f 2		
BASE8	g 8		
BASE10	f 10		
BASE16	g 16		
BC?	h P.FCN BC? ***n***	PRG & integer	Tests the *n*-th bit in ***x*** and executes the next program line if this bit is clear, else skips the next program line.

Name in listings	Keys to press	Works in modes	Remarks
BestF	h STAT BestF	FLOAT	Selects the best curve fit model, maximizing the correlation like BEST in HP-42S.
BS?	h P.FCN BS? ***n***	PRG & integer	Tests the *n*-th bit in ***x*** and executes the next program line if this bit is set, else skips the next program line.
B(k)	h STAT B(k)	FLOAT	= BINOMDIST(***x***; ***r01***; ***r02***; 1) in MS Excel, with the sample size in *r01* and the gross error probability in *r02*. B^{-1} returns the number of successes *k* for a given probability *p* .
B^{-1}(p)	h STAT B^{-1}(p)		
CB	h X.FCN CB ***n***	Integer	Clears the *n*-th bit in ***x*** .
CEIL	h X.FCN CEIL	FLOAT	Returns the smallest integer ≥ ***x*** .
CF	h CF ***n***	\α	Clears one flag, working in analogy to CB.
CLALL	h X.FCN CLALL	\PRG	Clears all registers and all programs after confirmation.
CLFLAG	h P.FCN CLFLAG	\PRG	Clears all user flags.
CLPR	h CLPR	PRG	Clears current program after confirmation (not programmable).
		\PRG, \α	Clears active program after confirmation.
CLREG	h X.FCN CLREG	\α	Clears all general purpose registers.
CLSTK	0 h FILL	\α	Clears stack registers **X, Y, Z**, and **T**.
CLx	h CLX	\α	Clears **X**.
	←	\α	Clears **X** if no input is pending.
	CPX h CLx	FLOAT	Clears both **X** and **Y**.
CLα	f CLα	All	Clears the alpha register like CLA in HP-42S.
CLΣ	g CLΣ	FLOAT	Clears registers 87 through 99 to reset the statistical sums.
COMB	f Cy,x	FLOAT	Returns the number of possible sets of ***y*** items taken ***x*** at a time. No item occurs more than once in a set, and different orders of the same ***x*** items are not counted separately. Formula: $C_{y,x} = \binom{y}{x} = \frac{y!}{x!\cdot(y-x)!}$

Name in listings	Keys to press	Works in modes	Remarks
CONJ	h CONJ	FLOAT	Changes the sign of ***y*** .
CORR	g r	FLOAT	Returns the correlation coefficient for the current statistical data and curve fitting model
COS	f COS	FLOAT, H.MS	
COSH	f HYP COS	FLOAT	
DATE	h X.FCN DATE	FLOAT	Recalls the date from the real time clock and displays it in the numeric section in the format selected. See D.MY, M.DY, and Y.MD. The function DATE in HP-12C corresponds to DAYS+ here (see below).
DAY	h X.FCN DAY	FLOAT	Takes ***x*** as a date and returns the day of week in the dot matrix and a corresponding integer in the numeric display (Sunday = 7).
DAYS+	h X.FCN DAYS+	FLOAT	Works like DATE in HP-12C, adding a number of days in **X** on a date in **Y** and displaying the resulting date including the day of week in the same format as DAY does.
DBLR	h X.FCN DBLR	Integer	Double precision commands like in HP-16C, but here for up to 128 bits.
DBL*	h X.FCN DBL*		
DBL/	h X.FCN DBL/		
DEG	g DEG	FLOAT	Sets angular mode to degrees.
DECOMP	h X.FCN DECOMP	/c	Decomposes the fraction in **X**, i.e. puts its numerator in **Y** and its denominator in **X**.
DENANY	h X.FCN DENANY	FLOAT	Sets default fraction format like in HP-35S, allowing maximum precision. Any denominator up to the maximum given by DENMAX is possible.
DENFAC	h X.FCN DENFAC	FLOAT	Sets "factors of the maximum denominator" format like in HP-35S. With e.g. 60 for the max. denominator, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, and 60 are possible denominators.
DENFIX	h X.FCN DENFIX	FLOAT	Sets fixed denominator format like in HP-35S. The denominator will then be the max. denominator always.

Name in listings	Keys to press	Works in modes	Remarks
DENMAX	h X.FCN DENMAX	FLOAT	Works like /c in HP-35S, but maximum value settable is 9999. The max. denominator will be set to 9999 if **X** contains 0 or a number >9999 at execution time. If **X** contains 1 then the current setting is recalled.
DISP	h X.FCN DISP ***n***	FLOAT	Changes the number of decimals while keeping the display format.
DROP	h P.FCN DROP	\α	Drops ***x***, changing stack contents from [***x, y, z, t***] to [***y, z, t, t***].
DROPY	h P.FCN DROPY	\α	Drops ***y***, changing stack contents from [***x, y, z, t***] to [***x, z, t, t***].
DSE	f DSE ***r***	PRG	Given `cccccc.fffii` in *r*, this function decrements `ccccccc` by `ii` and skips the next program line if `ccccccc` is now ≤`fff` for DSE, or = 0 for DSZ.
DSZ	h P.FCN DSZ ***r***		
D.MY	h X.FCN D.MY	FLOAT	Sets the format for date calculations.
D→J	h X.FCN D→J	FLOAT	Assumes ***x*** is a date and converts it to a Julian day number.
D→R	h X.FCN D→R	FLOAT	Assumes **X** containing degrees and converts them to radians. Angular mode is kept.
E3OFF	h X.FCN E3OFF	FLOAT	Toggle the thousands separator, being either a comma or a point depending on the radix setting.
E3ON	h X.FCN E3ON		
ENG	f ENG ***n***	FLOAT	Selects engineering display format.
ENTER↑	ENTER↑	\α	Copies ***x*** in **Y**, changing stack contents from [***x, y, z, t***] to [***x, x, y, z***].
	CPX ENTER↑	FLOAT	Copies ***x*** in **Z** and ***y*** in **T**, changing stack contents from [***x, y, z, t***] to [***x, y, x, y***].
ERF	h STAT ERF	FLOAT	Calculates the error function erf(***x***).
ExpF	h STAT ExpF	FLOAT	Selects the exponential curve fit model.
Ex(t)	h STAT Ex(t)	FLOAT	= EXPONDIST(***x***; ***r01***; 1) in MS Excel, with *r01* containing the rate λ. Ex^{-1} returns the survival time *t* for a given probability *p*.
Ex $^{-1}$(p)	h STAT Ex $^{-1}$(p)		
$\boldsymbol{e^x}$	g e^x	FLOAT	
e^x -1	h X.FCN e^x-1		

Name in listings	Keys to press	Works in modes	Remarks
FB	h X.FCN FB ***n***	Integer	Inverts (“flips”) the *n*-th bit in ***x*** .
FC?	h P.FCN FC? ***n***	PRG	Tests the flag specified and executes the next program line if this flag is clear, else skips the next program line. Clears, flips, or sets this flag after testing, if applicable.
FC?C	h P.FCN FC?C ***n***		
FC?F	h P.FCN FC?F ***n***		
FC?S	h P.FCN FC?S ***n***		
FF	h X.FCN FF ***n***	\α	Flips the flag specified.
FIB	h X.FCN FIB	\α	Calculates the Fibonacci number F_x .
FILL	h FILL	\α	Copies ***x*** in **Y**, **Z**, and **T**.
FIX	f FIX ***n***	FLOAT	Selects fixed point display format.
FLOAT	f .d	\α	Works like DECM in HP-42S. Additionally, converts possible H.MS data in **X** to decimal.
	g H	H.MS	
FLOOR	h X.FCN FLOOR	FLOAT	Returns the largest integer ≤ ***x*** .
FP	g FP	FLOAT	Returns the fractional part of ***x*** .
FRACT	g b/c	FLOAT	Sets fraction mode like in HP-35S. Maximum denominator is 9999. Absolute values must be > 10E-5 and < 10E5.
FS?	h FS? ***n***	PRG	Tests the flag specified and executes the next program line if this flag is set, else skips the next program line. Clears, flips, or sets this flag after testing, if applicable.
FS?C	h P.FCN FS?C ***n***		
FS?F	h P.FCN FS?F ***n***		
FS?S	h P.FCN FS?S ***n***		
F(x)	h STAT F(x)	FLOAT	F works like Q(F) , F^{-1} like F_P in HP-21S. The degrees of freedom are in *r01* and *r02*.
F^{-1}(p)	h STAT F^{-1}(p)		
GCD	h X.FCN GCD	\α	Returns the Greatest Common Divisor of ***x*** and ***y*** .
Ge(k)	h STAT Ge(k)	FLOAT	Geometric distribution, returns $1-(1-p_0)^k$. The gross error probability p_0 must be stored in *r01*. Ge^{-1} returns the number of failures *k* before the 1st success for a given probability *p*.
Ge^{-1}(p)	h STAT Ge^{-1}(p)		

Name in listings	Keys to press	Works in modes	Remarks
GRAD	g GRAD	FLOAT	Sets angular mode to gon or grads.
GTO	h GTO ***label***	PRG	Inserts an unconditional branch to the label specified.
		\PRG, \α	Moves program pointer to this label.
	h GTO . .	\α	Sets program pointer to PRGM TOP (not programmable).
H.MS	g H.MS	FLOAT	Sets H.MS mode for time calculations. See the paragraph about display above.
H.MS+	+	H.MS	Assumes ***x*** and ***y*** containing times in the format HHHH.MMSSDD, and adds or subtracts them.
H.MS–	−		
IMPFRC	h X.FCN IMPFRC	FLOAT	Allows improper fractions in fraction mode. Default are proper fractions.
INT	h INTEG ***label***	FLOAT	Integrates the function given in the routine specified. Lower and upper integration limits must be supplied in **Y** and **X**, respectively. Otherwise, the user interface is as in HP-15C.
IP	f IP	FLOAT	Returns the integer part of ***x*** .
ISG	g ISG ***r***	PRG	Given `cccccc.fffii` in *r* , this function increments `ccccccc` by `ii` and skips the next program line if `ccccccc` is now `>fff` for ISG, or = 0 for ISZ.
ISZ	h P.FCN ISZ ***r***		
I β	h X.FCN I β	FLOAT	Calculates the regularized incomplete beta function β(***x***, ***a***, ***b***) with ***a*** taken from **Z** and ***b*** from **Y**.
I Γ	h X.FCN I Γ	FLOAT	Calculates the regularized incomplete gamma function $\gamma(x,y)/\Gamma(x)$.
J→D	h X.FCN J→D	FLOAT	Assumes ***x*** is a Julian day number and converts it to a date in the format selected.
LASTx	g LASTx	\α	Recalls ***l*** into **X**.
	CPX g LASTx	FLOAT	Recalls ***l*** and ***i*** into **X** and **Y**, respectively.
LBL	f LBL ***label***	PRG	Identifies programs and routines for execution and branching. See opportunities for *label* in the table above.

Name in listings	Keys to press	Works in modes	Remarks
LCM	h X.FCN LCM	\α	Returns the Least Common Multiple of ***x*** and ***y***.
LEAP?	h P.FCN LEAP?	PRG & FLOAT	Takes ***x*** as a date in the format selected, extracts the year, and tests for a leap year. Executes the next program line if true, else skips this line.
LinF	h STAT LinF	FLOAT	Selects the linear curve fit model.
LJ	h X.FCN LJ	Integer	
LN	f LN	FLOAT	
LN1+X	h X.FCN LN1+X	FLOAT	
LN β	h X.FCN LN β	FLOAT	Calculate the natural logarithm of *β* or Γ, respectively. See these functions.
LN Γ	h X.FCN LN Γ		
LOG$_{10}$	f LG	FLOAT	
LOG$_{2}$	f LOG2	\α	Calculates the logarithm for base 2.
LogF	h STAT LogF	FLOAT	Selects the logarithmic curve fit model.
LOGy	f LOGy	FLOAT	Calculates the logarithm of ***x*** for base ***y***.
	CPX f LOGy	FLOAT	Calculates the logarithm of the complex number ***x*** + *i* ***y*** for base ***z*** + *i* ***t*** .
LR	h L.R.	FLOAT	Calculates the parameters of the fit curve through the data points accumulated, according to the model selected. Returns *A0* in **X** and *A1* in **Y**. In the linear fit model, *A0* is the intercept and *A1* the slope of the regression line.
MASKL	h X.FCN MASKL ***n***	Integer	Work like MASKL and MASKR on HP-16C, but with the parameter following the command instead of taken from **X**.
MASKR	h X.FCN MASKR ***n***		
MAX	h X.FCN MAX	\α	Returns the maximum of ***x*** and ***y*** .
MIN	h X.FCN MIN	\α	Returns the minimum of ***x*** and ***y*** .
MIRROR	h X.FCN MIRROR	Integer	Reflects the bit pattern in ***x*** (e.g. 000101 → 101000 for word size 6).
MOD	h MOD	\α	MOD of HP-42S equals RMD of HP-16C.
M.DY	h X.FCN M.DY	FLOAT	Sets the format for date calculations.

Name in listings	Keys to press	Works in modes	Remarks
NAND	h X.FCN NAND	Integer	
		FLOAT	Works in analogy to AND.
NaN?	h P.FCN NaN?	PRG	Tests ***x*** for “not a number” and executes the next program line if true, else skips it.
nBITS	h X.FCN nBITS	Integer	Counts bits set in ***x*** like #B on HP-16C.
NOP	h P.FCN NOP	PRG	
NOR	h X.FCN NOR	Integer	
		FLOAT	Works in analogy to AND.
NOT	h NOT	Integer	
		FLOAT	Works in analogy to AND.
nΣ	h STAT nΣ	FLOAT	Recalls the number of accumulated data points. Necessary for basic statistics.
N(x)	h STAT N(x)	FLOAT	= NORMDIST(***x***; ***r01***; ***r02***; 1) in MS Excel, with the mean value in *r01* and the standard deviation in *r02*.
N $^{-1}$(p)	h STAT N $^{-1}$(p)	FLOAT	= NORMINV(***x***; ***r01***; ***r02***) in MS Excel.
OFF	h P.FCN OFF	PRG	
ON	h P.FCN ON		
OR	h OR	Integer	
		FLOAT	Works in analogy to AND.
PAUSE	h PSE	PRG	Pauses program execution for about 1 s.
PERM	g Py,x	FLOAT	Returns the number of possible arrangements of ***y*** items taken ***x*** at a time. No item occurs more than once in an arrangement, and different orders of the same ***x*** items are counted separately. – Formula: $P_{y,x} = x! \cdot C_{y,x}$
PowerF	h STAT PowerF	FLOAT	Selects the power curve fit model.
PROFRC	h X.FCN PROFRC	FLOAT	Allows only proper fractions in fraction mode.
PROMPT	h X.PCN PROMPT	PRG	Displays the alpha register and stops program execution

Name in listings	Keys to press	Works in modes	Remarks
P(k)	h STAT P(k)	FLOAT	= POISSON(***x***; ***r01*r02***; 1) in MS Excel, with the gross error probability in *r01* and the sample size in *r02*. Alternatively, the Poisson parameter λ may be in *r01*, if *r02* contains 1. P^{-1} returns the number of successes *k* for a given probability *p* .
$P^{-1}(p)$	h STAT $P^{-1}(p)$		
Q(x)	f Q	FLOAT	Works like Q in HP-32E and Q(z) in HP-21S.
Q^{-1}(p)	g Q^{-1}	FLOAT	Works like Q^{-1} in HP-32E and z_P in HP-21S.
RAD	g RAD	FLOAT	Sets angular mode to radians.
RAND#	h STAT RAND#	\α	Returns a random number between 0 and 1 like RAN in HP-42S.
RCL	RCL ***s***	\α	Recalls *s* into **X**. See the note on the first page of this index for complex RCL.
RCLM	h X.FCN RCLM	PRG	Recalls selected mode settings into **X**. See the paragraph about indicators above.
RCLWS	h X.FCN RCLWS	Integer	Recalls the word size set. See WSIZE.
RCL+	RCL + ***s***	\α	Recalls the content of address *s* , executes the specified operation on it and places the result in **X**. E.g. RCL– 12 recalls the contents of register 12, subtracts ***x*** from it and voilá. RCL▲ (▼) takes the maximum (minimum) of the value in *s* and **X**. Complex RCL affects ***x*** and ***y*** as well as two source registers as explained on the first page of this index.
RCL–	RCL – ***s***		
RCL×	RCL × ***s***		
RCL/	RCL / ***s***		
RCL▲	RCL ▲ ***s***		
RCL▼	RCL ▼ ***s***		
RDX,	h ./,	FLOAT	Toggle the radix mark.
RDX.			
RJ	h X.FCN RJ	Integer	Works in analogy to LJ.
RL	h X.FCN RL ***n***	Integer	Works like *n* consecutive RLs / RLCs on HP-16C. For RL, 1 ≤ *n* ≤ 63. For RLC, 1 ≤ *n* ≤ 64. In indirect addressing *n* may be 0.
RLC	h X.FCN RLC ***n***		
RNDINT	h X.FCN RNDINT	FLOAT	Rounds ***x*** to next integer. ½ rounds to 1.

Name in listings	Keys to press	Works in modes	Remarks
ROUND	g RND	FLOAT	Rounds ***x*** using the current display format, like RND in HP-42S.
		/c	Rounds ***x*** using the current denominator, like RND in HP-35S.
RR	h X.FCN RR ***n***	Integer	Works like *n* consecutive RRs / RRCs on HP-16C. See RL / RLC for more.
RRC	h X.FCN RRC ***n***		
RTN	g RTN	PRG	Moves the program pointer to first line of current routine.
		\PRG	In a running program, moves the program pointer back to the line following the most recent XEQ instruction. If there is no matching XEQ, program execution halts.
RTN+1	n/a	PRG	Internal support routine.
R-CLR	h X.FCN R-CLR	FLOAT	Interprets ***x*** in the form `ss.nn` . Clears the contents of *nn* registers starting with number *ss* . If **X** contains e.g. 34.56, R-CLR will clear registers 34 through 89.
R-COPY	h X.FCN R-COPY	FLOAT	Interprets ***x*** in the form `ss.nndd` . Takes *nn* registers starting with number *ss* and copies their contents to *dd* . For example, if **X** contains 7.0345678, contents of registers 07 – 09 will be moved to registers 45 – 47, overwriting their old contents.
R-SORT	h X.FCN R-SORT	FLOAT	Interprets ***x*** in the form `ss.nn` . Sorts the contents of *nn* registers starting with number *ss* . Assume **X** contains 49.026 and registers 49 and 50 contain 1.2 and –3.4, respectively; then R-SORT will end with the contents of these 2 registers swapped.
R-SWAP	h X.FCN R-SWAP	FLOAT	Works like R-COPY but swaps the register contents of source and destination.
R/S	R/S	\PRG, \α	Runs a program (beginning with the current program line) or stops a running program.
R↑	h R↑	\α	Rotates the stack contents one step up.
R↓	R↓	\α	Rotates the stack contents one step down.
R→D	h X.FCN R→D	FLOAT	Assumes **X** containing radians and converts them to degrees. Angular mode is kept.

Name in listings	Keys to press	Works in modes	Remarks
s	g s	FLOAT	Calculates s_x and s_y using the current statistical data, and returns them in **X** and **Y**.
SB	h X.FCN SB ***n***	Integer	Sets the *n*-th bit in ***x*** .
SCI	f SCI ***n***	FLOAT	Selects scientific display format.
SEED	h STAT SEED	FLOAT	Stores a seed for random number generation.
SERR	h STAT SERR	FLOAT	Calculates $s/\sqrt{n}$ and returns the respective values in **X** and **Y**.
SETDAT	h X.FCN SETDAT	FLOAT, H.MS	Sets the date or time, respectively, for the real time clock.
SETTIM	h X.FCN SETTIM		
SF	h SF ***n***	\α	Sets the flag specified.
SIGN	h X.FCN SIGN	\α	Returns 1 for ***x*** > 0, –1 for ***x*** < 0, and 0 for ***x*** = 0 or non-numbers.
	CPX h ...	FLOAT	Returns the unit vector of ***x*** + *i* ***y*** in **X** and **Y**.
SIGNMT	h X.FCN SIGNMT	Integer	Sets sign-and-mantissa mode for integers.
SIN	f SIN	FLOAT, H.MS	
SINC	h X.FCN SINC	FLOAT	Calculates $\frac{\sin(x)}{x}$.
SINH	f HYP SIN	FLOAT	
SL	h X.FCN SL ***n***	Integer	Works like *n* (1 … 64) consecutive SLs on HP-16C. In indirect addressing *n* may be 0.
SLV	h SOLVE ***label***	FLOAT	Solves the equation f(x) = 0, with f(x) calculated by the routine specified. Two initial estimates of the root must be supplied in **X** and **Y** when calling SLV. Otherwise, the user interface is as in HP-15C.
SR	h X.FCN SR ***n***	Integer	Works like *n* consecutive SRs on HP-16C. See SL for more.
STO	STO ***d***	\α	Stores ***x*** into destination *d* . See the note on the first page of this index for complex STO.

Name in listings	Keys to press	Works in modes	Remarks
STOM	h X.FCN STOM	PRG	Sets selected modes as encoded in **X**. See the paragraph about indicators above.
STOP	R/S	PRG	Stops program execution.
STO+	STO + ***d***	\α	Executes the specified operation on the content of address *d* and stores the result into said address. E.g. STO– 12 subtracts ***x*** from the contents of register 12, and stores the result there again. STO▲ (▼) takes the maximum (minimum) of the values in *d* and **X** and stores the result. Complex STO affects ***x*** and ***y*** as well as two destination registers as explained on the first page of this index.
STO–	STO − ***d***		
STO×	STO × ***d***		
STO/	STO / ***d***		
STO▲	STO ▲ ***d***		
STO▼	STO ▼ ***d***		
SUM	RCL Σ+	FLOAT	Recalls ***Σx*** into **X** and ***Σy*** into **Y**.
TAN	f TAN	FLOAT, H.MS	
TANH	f HYP TAN	FLOAT	
TIME	h X.FCN TIME	FLOAT, H.MS	Recalls the time from the real time clock.
t(x)	h STAT t(x)	FLOAT	t works like Q(t) , t^{-1} like tp in HP-21S. The degree of freedom is stored in *r01*.
t^{-1}(p)	h STAT t^{-1}(p)		
UNSIGN	h X.FCN UNSIGN	Integer	Sets unsigned mode for integers.
VERS	h X.FCN VERS	All	Displays the firmware version.
VIEW	h VIEW ***s***	All	Views the contents of *s* .
W	h X.FCN W	FLOAT	W returns Lambert's W for given ***x*** ≥ -1/e , while W^{-1} returns *x* for given W (≥ -1).
W^{-1}	h X.FCN W^{-1}		
Wb(t)	h STAT Wb(t)	FLOAT	= WEIBULL(***x***; ***r01***; ***r02***; 1) in Excel, with *r01* containing the *shape parameter*, and *r02* the *characteristic lifetime*. Wb^{-1} returns the survival time *t* for given probability *p* .
Wb^{-1}(p)	h STAT Wb^{-1}(p)		
WSIZE	h X.FCN WSIZE ***n***	Integer	Works like WSIZE on HP-16C, but with the parameter following the command instead of taken from **X**. WSIZE 0 will set the word size to maximum, i.e. 64 bits.

Name in listings	Keys to press	Works in modes	Remarks
XEQ	XEQ ***label***	PRG	Calls the respective subroutine.
		\PRG, \α	Executes the respective program.
	B , C , or D (f may be necessary for accessing these hotkeys for integer bases >10.)	PRG	Calls the respective subroutine, so e.g. XEQ C will be inserted when C is pressed.
		\PRG, \α	Executes the respective program if defined.
XNOR	h X.FCN XNOR	Integer	
		FLOAT	Works in analogy to AND.
XOR	h XOR	Integer	
		FLOAT	Works in analogy to AND.
x !	h !	FLOAT	
***x*↔**	h x≷ ***r***	\α	Swaps the contents of **X** and *r* .
	CPX h x≷ ***r***	FLOAT	Exchanges ***x*** and ***y*** with the contents of two registers as explained at the top of this index.
x*↔*y	x≷y	\α	Swaps ***x*** and ***y*** , performing Re↔Im if a complex operation was executed immediately before.
	CPX x≷y	FLOAT	Swaps ***x*** with ***z*** and ***y*** with ***t*** , , changing stack contents from [***x, y, z, t***] to [***z, t, x, y***].
x ^{2}	g x^2	\α	
x → α	h X.FCN X→α	All	Interprets ***x*** as a code of up to 6 characters. Appends these characters to the alpha register, similar to XTOA in HP-42S.

Name in listings	Keys to press	Works in modes	Remarks
x < … ?	f x < ? ***a***	\α	Compares ***x*** with *a* . The three dots will be replaced in the listing by *a* according to the examples given in the addressing table above. CPX f x = ? ***a*** and CPX g x ≠ ? ***a*** compare ***x*** and ***y*** with *a* . Here, if *a* = … • 0 then ***x***+*i****y*** will be compared with 0; • 1 then ***x***+*i****y*** will be compared with 1; • Z then ***x***+*i****y*** will be compared with ***z***+*i****t*** ; • L then ***x***+*i****y*** will be compared with ***l***+*i****i*** ; no other stack levels may be specified; • ENTER↑ ***nn*** then ***x*** will be compared with the contents of register *nn* and ***y*** with those of register *nn*+1. • → ***nn*** then *nn* points to the first register to be used for comparison. See the table above for more.
x ≤ … ?	g x ≤ ? ***a***		
x = … ?	f x = ? ***a***		
x ≠ … ?	g x ≠ ? ***a***		
x ≥ … ?	f x ≥ ? ***a***		
x > … ?	g x > ? ***a***		
x̄, ȳ	f x̄	FLOAT	Recalls $\frac{1}{n}\sum x$ into **X** and $\frac{1}{n}\sum y$ into **Y**.
x̄w	h STAT x̄w	FLOAT	Returns the weighted mean $\sum xy \big/ \sum y$.
x̂	h STAT x̂	FLOAT	Returns a forecast ***x*** for a given *y* according to the fit model chosen. See L.R. for more.
Y.MD	h X.FCN Y.MD	FLOAT	Sets the format for date calculations.
y^x	g y^x	\α	In integer modes ***x*** must be ≥ 0.
	C or f C	\α	Works as long as label C is not defined yet. f is needed for integer bases >12 only.
ŷ	f ŷ	FLOAT	Returns a forecast *y* for a given ***x*** according to the fit model chosen. See L.R. for more.
αAPP	h X.FCN αAPP ***char***	\α	Enters alpha mode for the input of 1 character, appends this to the alpha register, and returns to the mode set before. Eventually, this equals the sequence αON *char* αOFF.
αBEG	h X.FCN αBEG	\α	Positions the cursor to the beginning of the alpha register, i.e. below the first character.
αDATE	h X.FCN αDATE	FLOAT, α	Takes ***x*** as a date and inserts it into the alpha register in the format selected, directly left of the cursor position.

Name in listings	Keys to press	Works in modes	Remarks
αDAY	h X.FCN αDAY	FLOAT, α	Takes ***x*** as a date, recalls the name of the respective day and inserts its first 3 letters into the alpha register, directly left of the cursor position.
αEND	h X.FCN αBEG	\α	Positions the cursor to the end of the alpha register, i.e. immediately behind the last character.
αIP	h X.FCN αIP	All	Inserts the integer part of ***x*** into the alpha register, directly left of the cursor position, similar to AIP in HP-42S.
αLENG	h X.FCN αLENG	All	Returns in the numeric display the number of characters found in the alpha register, like ALENG in HP-42S.
αMONTH	h X.FCN αMONTH	FLOAT, α	Takes ***x*** as a date, recalls the name of the month and inserts its first 3 letters into the alpha register, directly left of the cursor position.
αRCL	f RCL ***s***	α	Interprets the contents of source *s* as characters and inserts them into the alpha register, directly left of the cursor position.
	h X.FCN αRCL ***s***	\α	
αRC#	h X.FCN αRC# ***s***	All	As αRCL, but αRC# interprets the contents of *s* as a number and inserts this in current format into the alpha register.
αRL	h X.FCN αRL ***n***	All	Rotates the alpha register by *n* characters like AROT in HP-42S, but with a positive parameter trailing the command instead of taken from **X**. In indirect addressing *n* may be 0. αRR works like αRL but rotates to the right.
αRR	h X.FCN αRR ***n***		
αSL	h X.FCN αSL ***n***	All	Shifts the *n* left-most characters out of the alpha register, similar to ASHF in HP-42S. In indirect addressing *n* may be zero. – αSR works like αSL but takes the *n* right-most characters.
αSR	h X.FCN αSR ***n***		
αSTO	f STO ***d***	α	Stores the first 6 characters in the alpha register into destination *d* .
	h X.FCN αSTO ***d***	\α	
αTIME	h X.FCN αTIME	FLOAT, α	Takes ***x*** as a time HH.MM, converts it to a string in the format selected, and inserts this into the alpha register, directly left of the cursor position.
αVIEW	h X.FCN αVIEW	\α	Displays the alpha register.

Name in listings	Keys to press	Works in modes	Remarks
α → x	h X.FCN α→X	All	Returns the character code of the left-most character in the alpha register and deletes this character, like ATOX in HP-42S.
β	h X.FCN β	FLOAT	Calculates Euler's Beta function B(***x***, ***y***).
Γ	h STAT Γ	FLOAT	
	h X.FCN Γ		
ΔDAYS	h X.FCN ΔDAYS	FLOAT	Takes ***x*** and ***y*** as dates in the format chosen and calculates the number of days between them. Works like in HP-12C.
Δ%	f Δ%	FLOAT	Calculates $100 \cdot \frac{x-y}{y}$ like %CH in HP-42S.
π	h π	FLOAT	
	CPX h π	FLOAT	Copies π in **X** and clears **Y**.
Π	h P.FCN Π ***label***	PRG	Generates a product or a sum, respectively, based on the routine starting with the label specified. The loop control number is taken from **X** in the standard format `cccccc.fffii` like e.g. for DSE.
Σ	h P.FCN Σ ***label***		
σ	h STAT σ	FLOAT	Calculates $s \cdot \sqrt{\frac{n}{n-1}}$ and returns the respective values in **X** and **Y**.
$\Sigma \ln^2 x$	h STAT $\Sigma \ln^2 x$	FLOAT	Recall the respective statistical sums. These sums are necessary for curve fitting models beyond pure linear. Calling them by name enhances readability of programs significantly.
$\Sigma \ln^2 y$	h STAT $\Sigma \ln^2 y$		
Σlnx	h STAT Σlnx		
Σlnxy	h STAT Σlnxy		
Σlny	h STAT Σlny		
Σxlny	h STAT Σxlny		
Σylnx	h STAT Σylnx		

Name in listings	Keys to press	Works in modes	Remarks
Σx	h STAT Σx	FLOAT	Recall the respective statistical sums. These sums are necessary for basic statistics and linear curve fitting. Calling them by name enhances readability of programs significantly.
Σx^2	h STAT Σx^2		
Σxy	h STAT Σxy		
Σy	h STAT Σy		
Σy^2	h STAT Σy^2		
Σ+	Σ+	FLOAT	
Σ–	h Σ–		
$\chi^2(x)$	h STAT $\chi^2(x)$	FLOAT	χ^2 works like $Q(\chi^2)$, the inverse like $\chi^2{}_p$ in HP-21S. The degree of freedom is in *r01*.
χ^2INV	h STAT χ^2INV		
+, –, ×, /	+, –, ×, /	\α	
+/–	+/–		
//	g //	FLOAT	Calculates $\left(\frac{1}{x}+\frac{1}{y}\right)^{-1}$.
%	g %	FLOAT	Calculates $\frac{x \cdot y}{100}$.
%T	h X.FCN h % T	FLOAT	Calculates $100 \cdot \frac{x}{y}$.
%Σ	h STAT h % Σ	FLOAT	Calculates $100 \cdot \frac{x}{\sum x}$.
%+	h %+	FLOAT	Adds a markup of ***x*** % to ***y***, calculating $x = y \cdot \left(1+\frac{x}{100}\right)$.
%–	h %–	FLOAT	Subtracts a discount of ***x*** % from ***y***, calculating $x = y \cdot \left(1-\frac{x}{100}\right)$.
∞?	h P.FCN ∞?	PRG	Tests ***x*** for infinity and executes the next program line if true, else skips this line.

Name in listings	Keys to press	Works in modes	Remarks
√‾	f √x	\α	
	D or f D	\α	Works as long as label D is not defined yet. f is needed for integer bases >13 only.
→DEG	→ g DEG	FLOAT	Assumes **X** containing an angle in current angular mode and converts it to degrees. Angular mode is kept.
→GRAD	→ g GRAD	FLOAT	Works like →DEG, but converts to grads.
→HR	→ f H	H.MS	Takes the hours or degrees in **X** and converts them into decimal numbers.
→H.MS	→ g H.MS	FLOAT	Assumes **X** containing *decimal* hours or degrees and converts them into the format HHHH**.**MMSSDD.
→POL	g ▶P	FLOAT	Assumes **X** and **Y** containing the coordinates (***x, y***) and converts them to (*r* , *ϑ*) .
→RAD	→ g RAD	FLOAT, H.MS	Works like →DEG, but converts to radians.
→REC	f ▶R	FLOAT	Assumes **X** and **Y** containing the coordinates (*r* , *ϑ*) and converts them to (***x, y***) .

Catalogues (not programmable):

Calling a catalogue will set temporary alpha mode to allow for keying in the first 2 characters of the item wanted. In general, ▲ and ▼ browse the catalogue, ENTER↑ selects the item displayed and exits, and EXIT leaves the catalogue without executing anything, returning to the mode as set before. See the table above for addressing catalogued items, and the next paragraph for detailed item lists.

Name in listings	Keys to press	Works in modes	Contents
ARROWS	f →	α	Arrows and mathematical symbols.
COMPAR	f x <	α	Comparison symbols and brackets. Parentheses are called by f (and g) , respectively.
CONST	h CONS	FLOAT	Constants like in HP35s. See them listed in a separate table below. CPX h CONS will clear **Y** in recalling the constant selected.
CONV	h CONV	FLOAT	Conversions as listed in a separate table below.
CPX	f CPX	α	"Complex" letters mandatory for languages beyond English. Upper or lower case will be displayed according to setting (see f ↑ below).
PUNCT	h ./,	α	Punctuation marks and text symbols.
P.FCN	h P.FCN	\α	Extra programming functions.
STAT	h STAT	FLOAT	Extra statistical functions.
		α	Some special letters for statistics.
X.FCN	h X.FCN	FLOAT	Extra real functions.
		Integer	Extra integer functions.
		α	Extra alpha functions.
	CPX h X.FCN	FLOAT	Extra complex functions.

Pure input commands:

Name in listings	Keys to press	Works in modes	Remarks
0 … 9	0 … 9	\α	Numeric input. The top row of keys on the keyboard is used to enter digits >10 in the respective integer bases. For bases <16, their defaults are switched as applicable. For bases <10, input of illegal digits throw an error message.
A … F	A …F (printed on key plate)	-1, -2, -3, -4, -5, h	
A … Z	A …Z (printed on key plate)	α	Alphabetic input. See page 3 for more information. Find alpha catalogues below.
	f ⇧	α	Toggles upper and lower case.
	⬅	Input pending	Deletes last digit or character put in (not programmable)..
		PRG	Deletes current step if no input is pending (not programmable).
E	E (the key)	FLOAT	Like EEX in vintage calculators.
αOFF	ENTER↑	α	Works like AOFF in HP-42S.
αON	f α	\α	Works like AON in HP-42S.
[ ] or **[/]**	,	/c	First , is taken as a space, second as a fraction mark, e.g. 2 , 3 , 4 results in 2 ¾ in the dot matrix display.
[.] or **[,]**	,	FLOAT	Inserts the radix mark as selected.
		α	Inserts a point.
[.]	,	Y.MD, D.MY, M.DY	Separates the leading unit in date modes. It is left to the user to decide if a number displayed represents a date or not.
[°]	,	H.MS	Separates degrees (or hours) from minutes and seconds, so input format is HHHH.MMSSDD.

Pure navigation, mode switching and information commands (not programmable):

Name in listings	Keys to press	Works in modes	Remarks
	g OFF	All	
	ON	Calc. off	
	▲ / ▼	Status open	Go to previous / next set of flags.
		Cat. open	Go to previous / next item in this catalogue.
		α	Move the cursor 1 character to the left / right in alpha register. Movement will stop at first or behind last character. Shift the display window if necessary.
		PRG	Like BST / SST in HP-42S.
		\PRG, \α	
	f ◀ and g ▶	Integer	Shift the display window like in HP-16C. Useful for numbers with small bases.
EXIT	EXIT	All	Exits catalogues and other operations with pending input, canceling the execution of this operation.
PRGOFF	h P/R	PRG	Toggle programming mode.
PRGON	h P/R	\PRG, \α	
SHOW	h SHOW	FLOAT, \PRG	Shows the full mantissa until this key is released.
		PRG	Displays a CRC-32 checksum of program memory's contents (8 hex digits), allowing to validate program integrity.
STATUS	h STATUS	All	Shows the status of user flags, similar to STATUS on HP-16C. See the paragraph about display above.
→BIN	➡ f 2	\α, \h, \-5	Show *x* in target integer representation until the next key is pressed. Mode is kept. In modes -5 and h, an f must precede the key ➡ .
→DEC	➡ f 10		
→HEX	➡ g 16		
→OCT	➡ g 8		

DETAILED CATALOGUE CONTENTS

Here the contents of the catalogues X.FCN, P.FCN and STAT are listed. A single operation, e.g. BASE, may be contained in more than one catalogue. The characters necessary to access a specific function in the catalogue are printed bold in this table – each red character must be replaced by a ▼ – if even the last letter of a function name is red, one may need more strokes of ▼ to access this function. The alpha catalogues ARROWS, COMPARE, CPX, and PUNCT are found further below. See also the catalogues CONST and CONV in separate paragraphs.

Content of X.FCN			Content of CPX X.FCN	Content of P.FCN	Content of STAT
… in FLOAT	… in integer modes	… in alpha mode			
12h	**1**COMPL	**C**LALL	**e**^{x} -1	**B**C?	**B**estF
24h	**2**COMPL	**V**ERS	**F**IB	**BS**?	**B(**k)
ALL	**A**SR	**x**→ α	**L**N1+x	**C**B	**B** $^{-1}$(p)
ANGLE	**B**ASE	αDATE	**LN** β	**CL**FLAG	**E**RF
BASE	**C**B	α**D**AY	**LN Γ**	**D**ROP	**Ex**pF
CEIL	**CL**ALL	α**I**P	**S**IGN	**DR**OPY	**Ex(**x)
CLALL	**CLR**EG	α**L**ENG	**SI**NC	**DS**Z	**Ex** $^{-1}$**(**p)
CLREG	**D**BLR	α**M**ONTH	**W**	**F**B	**F**(x)
DATE	**DB**L*	α**R**C#	**W** $^{-1}$	**FC**?	**F** $^{-1}$(p)
DAY	**DBL**/	α**RL**	β	**FC?**C	**G**e(k)
DAYS+	**F**B	α**RR**	**Γ**	**FC?F**	**Ge** $^{-1}$(p)
DECOMP	**FF**	α**S**L		**FC?S**	**L**inF
DENANY	**FI**B	α**SR**		**FF**	**Lo**gF
DENFAC	**G**CD	α→ x		**FS**?C	**n**Σ
DENFIX	**L**CM			**FS?**F	**N(**x)
DENMAX	**M**ASKL			**FS?S**	**N** $^{-1}$(p)
DISP	**MA**SKR			**I**SZ	**P**owerF
D.MY	**MAX**			**L**EAP?	**P(**k)
D→J	**MI**N			**N**aN?	**P** $^{-1}$(p)
D→R	**MIR**ROR			**NO**P	**R**AND#
E3OFF	**N**AND			**O**FF	**S**EED
E3ON	**NB**ITS			**ON**	**SE**RR
e^{x} -1	**NO**R			**P**ROMPT	**t**(x)
FF	**R**AND#			**R**CLM	**t** $^{-1}$(p)
FIB	**RC**LWS			**R-**CLR	**W**b(t)

Content of X.FCN			Content of CPX X.FCN	Content of P.FCN	Content of STAT
… in FLOAT	… in integer modes	… in alpha mode			
FLOOR	RJ			R-COPY	Wb $^{-1}$(p)
GCD	RL			R-SORT	$\bar{x}$w
IMPFRC	RLC			R-SWAP	$\hat{x}$
I β	RR			SB	Γ
I Γ	RRC			STOM	σ
J→D	SB			Π	Σln^{2}x
LCM	SIGN			Σ	Σln^{2}y
LN1+x	SIGNMT			∞?	Σlnx
LN β	SL				Σlnxy
LN Γ	SR				Σlny
MAX	UNSIGN				Σx
MIN	VERS				Σx^2
M.DY	WSIZE				Σxlny
NAND	XNOR				Σxy
NOR	x→ α				Σy
PROFRC	αAPP				Σy^2
RNDINT	αIP				Σylnx
R→D	αLENG				χ^2(x)
SETDAT	αON				χ^{2}INV
SETTIM	αRCL				%Σ
SIGN	αRC#				
SINC	αRL				
TIME	αRR				
VERS	αSL				
W	αSR				
W $^{-1}$	αSTO				
XNOR	αVIEW				
x→ α	α→ x				
Y.MD					
αAPP					

Content of X.FCN			Content of CPX X.FCN	Content of P.FCN	Content of STAT
… in FLOAT	… in integer modes	… in alpha mode			
α**D**ATE					
α**DA**Y					
α**I**P					
α**L**ENG					
α**M**ONTH					
α**O**N					
α**R**CL					
α**RC**#					
α**RL**					
α**RR**					
α**S**L					
α**SR**					
α**STO**					
α**T**IME					
α**V**IEW					
α→ x					
β					
Γ					
ΔDAYS					
%T					

Here are the contents of the alpha catalogues:

STAT
x̂
x̄
ŷ
ȳ

ARROWS
→
←
↑
↓
↔
√
∫
°
▯$^{-1}$
▯^{2}
ħ
▯^{x}
^
∞

CPX	
À	à
Á	á
Â Ã Ā Ă	â ã ā ă
Ä	ä
Å	å
Ć	ć
Č	č
Ç	ç
È	è
É	é
Ê Ē Ĕ Ě	ê ē ĕ ě
Ë	ë
Ì	ì
Í	í
Î Ĩ Ī Ĭ	î ĩ ī ĭ
Ï	ï
Ñ	ñ
Ò	ò
Ó	ó
Ô Õ Ō Ŏ	ô õ ō ŏ
Ö	ö
Ř	ř
Š	š
	ß
Ù	ù
Ú	ú
Û Ũ Ū Ŭ	û ũ ū ŭ
Ü	ü
Ů	ů
Ý	ý
Ÿ	ÿ
Ž	ž

COMPAR
<
≤
=
≠
≥
>
[
]
{
}

PUNCT
,
“
#
&
‘
*
:
;
@
\
_
|
~

TABLE OF CONSTANTS

This lists all constants contained in the catalogue CONST. Names printed in a golden field represent fundamental or measured constants, while the other ones may be derived from them. The constants π and **e** are also found on the keyboard directly. The characters necessary to get to a specific function in the catalogue are printed bold in this index – each red character must be replaced by a ▼ .

Name	Numeric value	Dimension	Remarks
a	365.2425	*d*	Gregorian year (per definition)
a$_0$	5.291772083E-11	*m*	Bohr radius $= \alpha / 4\pi R_\infty$
atm	1.01325E5	Pa / atm	Standard pressure *p₀* (per definition)
c	2.99792458E8	m / s	Vacuum speed of light (per definition)
c$_1$	3.7417712E-16	$m^2 \cdot W$	First radiation constant $= 2\pi \cdot h \cdot c^2$
c$_2$	0.014387752	$m \cdot K$	Second radiation constant $= hc / k$
e	2.718281828459045	1	Euler's e. Please note the character *e* is used for the electron charge elsewhere in this table.
eV	1.602176462E-19	*J*	= Electron charge * 1V. Remember *J* = *V A s* .
F	96485.3415	$\frac{A \cdot s}{mol}$	Faraday's constant = *e N*$_A$
g	9.80665	m / s^2	Standard earth acceleration (per definition)
G	6.6742867E-11	$\frac{m^3}{kg \cdot s^2}$	Newton's gravitation constant
g$_e$	2.002319304362	1	Landé's g-factor
G$_o$	7.748091696E-5	$1 / \Omega$	Conductance quantum $= 2e^2 / h$
h	6.62606876E-34	*J s*	Planck constant
ħ	1.054571596E-34		$= h / 2\pi$
k	1.3806503E-23	J / K	Boltzmann constant $= R / N_A$

Name	Numeric value	Dimension	Remarks
m_e	9.10938188E-31	kg	Electron mass
m_n	1.67492716E-27		Neutron mass
m_p	1.67262158E-27		Proton mass
m_u	1.66053873E-27		Atomic unit mass = $10^{-3} kg / N_A$
m_μ	1.88353109E-28		Muon mass
N_A	6.02214199E23	$1/mol$	Avogadro's number
R	8.314472	$\frac{J}{mol \cdot K}$	Molar gas constant
r_e	2.817940285E-15	m	Classical electron radius $= \alpha^2 \cdot a_0$
R_K	25812.80756	Ω	Von Klitzing constant $= h/e^2$
R_∞	10973731.5685	$1/m$	Rydberg constant $= \alpha^2 m_e c / 2h$
T_o	273.15	K	= 0°C, standard temperature (per definition)
t_p	5.39124E-44	s	Planck time $= \sqrt{\hbar G / c^5}$
V_m	0.022413996	m^3/mol	Molar volume of ideal gas at standard conditions $= RT_0 / p_0$
Z_o	376.730313461	Ω	Characteristic impedance of vacuum $= \sqrt{\mu_0 / \varepsilon_0} = \mu_0 c$
α	7.297352533E-3	1	Fine-structure constant $= e^2 / 4\pi\varepsilon_0 \hbar c$
γ_{EM}	0.57721566490153286	1	Euler-Mascheroni constant
γ_p	267522212	$\frac{1}{s \cdot T}$	Proton gyromagnetic ratio $= 2\mu_P / \hbar$
ε_o	8.854187817E-12	$\frac{A \cdot s}{V \cdot m}$	Electric constant, vacuum permittivity $= 1 / \mu_0 c^2$

Name	Numeric value	Dimension	Remarks
λ_c	2.426310215E-12	m	Compton wavelengths of electron $=\frac{h}{m_e c}$, neutron $=\frac{h}{m_n c}$, and proton $=\frac{h}{m_p c}$, respectively.
λ_{cn}	1.319590898E-15		
λ_{cp}	1.321409847E-15		
μ_B	9.27400899E-24	J/T	Bohr's magneton $= e\hbar/2m_e$
μ_e	-9.28476362E-24		Electron magnetic moment
μ_u	5.05078317E-27		Nuclear magneton $= e\hbar/2m_p$
μ_n	-9.662364E-27		Neutron magnetic moment
μ_p	1.410606633E-26		Proton magnetic moment
μ_o	1.2566370614E-6	N/A^2	Magnetic constant, vacuum permeability $=4\pi\cdot 10^{-7}$ (per definition)
μ_μ	-4.49044813E-26	J/T	Muon magnetic moment
π	3.141592653589793	1	
σ_B	5.6704E-8	$\frac{W}{m^2\cdot K^4}$	Stefan Boltzmann constant $=\frac{2\pi^5 k^4}{15h^3c^2}$
Φ	1.61803398874989485	1	Golden ratio $=\frac{1+\sqrt{5}}{2}$
Φ_o	2.067833636E-15	$V\,s$	Magnetic flux quantum $= h/2e$
∞		1	Infinity

TABLE OF CONVERSIONS

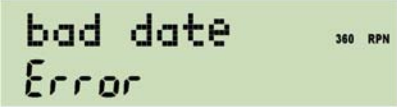
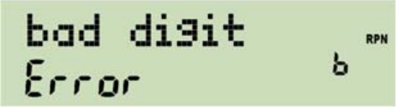
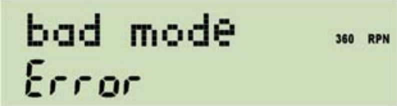
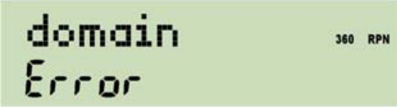
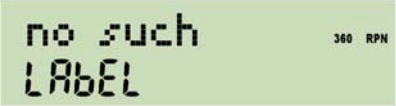
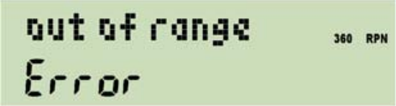
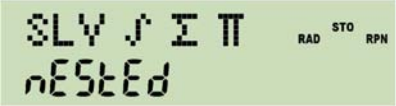
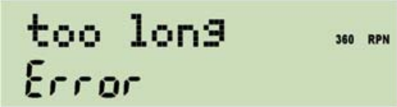
These are the conversions contained in the new catalogue CONV. The characters necessary to get to a specific conversion therein are printed bold in this index – each red character must be replaced by a ▼ . The constant $\mathbf{T_o}$ may be useful for conversions, too; it is found in the catalogue CONST. The conversion factors or divisors listed in this table will not be seen when executing a conversion.

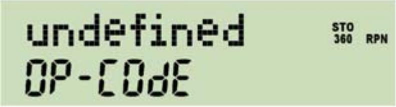
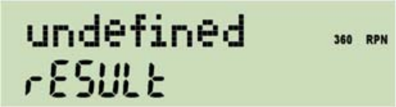
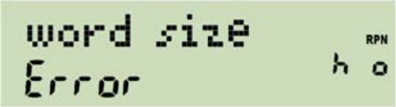
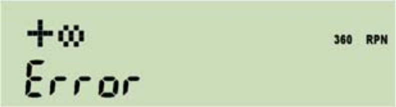
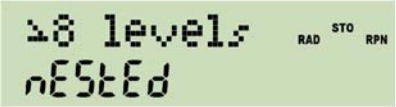
Conversion		Remarks	Class
acres→ha	* 0.4046873	Remember 1 ha = 10^4 m²	Area
atm→Pa	* 1.01325E5	Exactly	Pressure
au→km	* 1.495979E8	Astronomic units	Length
bar→Pa	* 1E5	Exactly	Pressure
bhp→W	* 745.6999	British horse power	Power
Btu→J	* 1055.056		Energy
cal→J	* 4.1868	Exactly	Energy
cm→inch	/ 2.54	Exactly	Length
feet→m	* 0.3048	Exactly	Length
flozUK→*ml*	* 28.41306	Remember 1 m^3 = 10^3 *l*	Volume
flozUS→*ml*	* 29.57353		Volume
g→oz	/ 28.34952		Mass
g→tr oz	/ 31.10348		Mass
galUK→ *l*	* 4.54609		Volume
galUS→ *l*	* 3.785418		Volume
ha→acres	/ 0.4046873		Area
HP$_e$→W	* 746	Exactly	Power
inch→cm	* 2.54	Exactly	Length
J→Btu	/ 1055.056		Energy
J→cal	/ 4.1868	Exactly	Energy
J→k Wh	/ 3.6E6	Exactly, since 1 h = 3600 s	Energy
kg→lbm	/ 0.4535924		Mass
km→au	/ 1.495979E8		Length
km→*l.y.*	/ 9.460730E12	Light years	Length
km→mi	/ 1.609344	Exactly	Length
km→nmi	/ 1.852	Nautical miles, exactly	Length

Conversion		Remarks	Class
km→pc	/ 3.085678E16	Parsec	Length
kWh→J	* 3.6E6	Exactly	Energy
l →galUK	/ 4.54609		Volume
l →galUS	/ 3.785418		Volume
lbf→N	* 4.448222		Force
lbm→kg	* 0.4535924		Mass
l.y.→km	* 9.460730E12		Length
m→feet	/ 0.3048	Exactly	Length
mi→km	* 1.609344	Exactly	Length
ml→flozUK	/ 28.41306		Volume
ml→flozUS	/ 29.57353		Volume
mmHg→Pa	* 133.3224	1 mmHg = 1 torr	Pressure
N→lbf	/ 4.448222		Force
nmi→km	* 1.852	Exactly	Length
oz→g	* 28.34952		Mass
Pa→atm	/ 1.01325E5	Exactly	Pressure
Pa→bar	/ 1E5	Exactly	Pressure
Pa→mmHg	/ 133.3224		Pressure
pc→km	* 3.085678E16		Length
PS(hp)→W	* 735.4988		Power
sh ton→t	* 0.9071847		Mass
t→sh ton	/ 0,9071847		Mass
t→ton	/ 1.016047		Mass
ton→t	* 1.016047		Mass
tr oz→g	* 31.10348		Mass
W→bhp	/ 745.6999		Power
W→HP$_{e}$	/ 746	Exactly	Power
W→PS(hp)	* 735.4988		Power
°C→°F			Temperature
°F→°C			Temperature

ERROR MESSAGES

Depending on error conditions, the following messages will be displayed:

Message	May happen in modes	Explanation and Examples
 bad date Error 	FLOAT	Invalid date format or incorrect date in input, e.g. month >12, day >31 etc.
 bad digit Error 	Integer	Invalid digit in integer input, e.g. 9 in octal or +/- in unsigned mode.
 bad mode Error 	All	Caused by calling an operation in a mode where it is not defined, e.g. SIN in hexadecimal.
 domain Error 	\α	An argument exceeds the domain of this mathematical function. May be caused by roots or logs of negative numbers (if not preceded by CPX), by LN(0), $\Gamma(0)$, ATANH(x) for $\lvert\mathrm{Re}(x)\rvert \geq 1$, or ACOSH(x) for $\mathrm{Re}(x) < 1$, etc.
 no such LAbEL 	All	Attempt to address an undefined label.
 out of range Error 	All	- A number exceeds the valid range. May be caused e.g. by specifying decimals >11, word size >64, negative flag numbers, integers ≥2^{64}, hours or degrees >9000, denominators ≥9999 etc. - A register address exceeds the valid range. May also happen in indirect addressing. - A block register operation (e.g. R-COPY) attempts to go out of valid register numbers (0 … 99).
 SLV ∫ Σ Π nEStEd 	PRG	Nested use of solve, integrate, sum or product is illegal.
 too long Error 	α	Input exceeds the length of alpha register.

Message	May happen in modes	Explanation and Examples
 undefined OP-COdE	All	An instruction with an undefined op-code occurred (should never happen, but who knows).
 undefined rESULt	\α, \PRG	• 0 / 0 • tan(90°) and equivalents.
 word size Error	Integer, \PRG	Stack or register content is too big for the word size set.
 +∞ Error (or − ∞)	\α, \PRG	• Division of a number > 0 (or < 0) by zero. • Divergent sum or product or integral. • Positive (or negative) overflow in FLOAT.
 ≥8 levels nEStEd	PRG	Subroutine nesting exceeds 8 levels.

Any key will wipe out the error message displayed and execute with the stack contents present.

Edition	Date	Release notes
1	9.12.08	Start
1.1	15.12.08	Added the table of indicators; added NAND, NOR, XNOR, RCLWS, STOWS, //, N, SERR, SIGMA, < and >; deleted HR, INPUT, 2 flag commands, and 2 conversions; extended explanations for addressing and COMPLEX & …; put XOR on the keyboard; corrected errors.
1.2	4.1.09	Added ASRN, CBC?, CBS?, CCB, SCB, FLOAT, MIRROR, SLN, SRN, >BIN, >DEC, >HEX, >OCT, BETA, D>R, DATE, DDAYS, D.MY, M.DY, Y.MD, CEIL, FLOOR, DSZ, ISZ, D>R, R>D, EMGAM, GSB, LNBETA, LNGAMMA, MAX, MIN, NOP, REAL, RJ, W and WINV, ZETA, %+ and %-; renamed the top left keys B, C, and D, and bottom left EXIT.
1.3	17.1.09	Added AIP, ALENG, ARCL, AROT, ASHF, ASTO, ATOX, XTOA, AVIEW, CLA, PROMPT (all taken from 42S), CAPP, FC?C, FS?C, SGMNT, and the …# commands; renamed NBITS to BITS and STOWS to WSIZE; specified the bit commands closer; deleted the 4 carry bit operations.
1.4	10.2.09	Added CONST and a table of constants provided, D>J and J>D, LEAP?, %T, RCL and STO ▲ and ▼, and 2 forgotten statistics registers; deleted CHS, EMGAM, GSB, REAL and ZETA; purged and renamed the bit operations; renamed many commands.
1.5	5.3.09	Added RNDINT, CONV and its table, a memory table, the description of XEQ B, C, D to the operation index, and a and g_e to the table of constants; put CLSTK on a key, moved CLΣ and FILL, changed the % and log labels on the keyboard, put CLALL in X.FCN; checked and cleaned alpha mode keyboard and added a temporary alpha keyboard; rearranged the alphabet to put Greek after Latin, symbols after Greek consistently; separated the input and non programmable commands; cleaned the addressing tables.
1.6	12.8.09	Added BASE, DAYS+, DROP, DROPY, E3OFF, E3ON, FC?F, FC?S, FIB, FS?F, FS?S, GCD, LCM, SETDAT, SETTIM, SET24, SINC, TIME, VERS, αDAY, αMONTH, αRC#; %Σ, as well as F-, t-, and χ^2-distributions and their inverses; reassigned DATE, modified DENMAX, FLOAT, αROT, and αSHIFT; deleted BASE arithmetic, BIN, DEC, HEX, and OCT; updated the alpha keyboards; added flags in the memory table; included indirect addressing for comparisons; added a paragraph about the display; updated the table of indicators; corrected errors.
1.7	9.9.09	Added P.FCN and STAT catalogues, 4 more conversions, 3 more flags, Greek character access, CLFLAG, DECOMP, DENANY, DENFAC, DENFIX, Iβ, IΓ, αDATE, αRL, αRR, αSL, αSR, αTIME, 12h, 24h, fraction mode limits, normal distribution and its inverse for arbitrary μ and σ, and Boolean operations working within FLOAT; deleted αROT and αSHIFT, the timer, and forced radians after inverse hyperbolics; renamed WINV to W^{-1}, and beta and gamma commands to Greek; added tables of catalogue contents; modified label addressing; relabeled PRGM to P/R and PAUSE to PSE; swapped SHOW and PSE as well as Δ% and % on the keyboard; relabeled Q; corrected CEIL and FLOOR; updated X.FCN and alpha commands; updated the virtual alpha keyboard.
1.8	29.10.09	Added R-CLR, R-COPY, R-SORT, R-SWAP, RCLM, STOM, alpha catalogues, 1 more constant and some more conversions, a table of error messages, as well as the binomial, Poisson, geometric, Weibull and exponential distributions and their inverses; renamed some commands; put SQRT instead of π on hotkey D.
1.9	14.12.09	Added two complex comparisons; swapped and changed labels in the top three rows of keys, dropped CLST; completed function descriptions in the index.
1.10	19.1.10	Added IMPFRC, PROFRC, complex ENTER, αBEG, αEND, and an addressing table for items in catalogues; updated temporary alpha mode, display and indicators, RCLM and STOM, alpha-commands and the error message table; renamed the exponential distribution; deleted GTO. keeping the other GTOs; wrote the introduction.