

Appnote 014: h.ms, d.ms and .ms: C47 & R47**2026-08-20**

This appnote summarizes the usage of the sexagesimal conversions in C47/R47, with reference to historical usage including HP's old ways, the old C47 way and other ways. Unless stated, the functions described are from C47 / R47 version 00.109.02.06:

Reference pages exist on <https://47calc.com>, specifically:

[The CLK menu with Time and Date functions](#)

[The TRG menu with DMS related functions](#)

[Community Wiki, QRG: TIME](#)

Version table

Original issue:

h.ms, d.ms and .ms: C47 & R47

2024-12-15

Rev 01

Fixed typo 1.3 -> 1.5; update menu

2026-08-20

TIME usage (HMS etc.)

The HP names HR & HMS, and H & H.MS are used in C47 and provide a measure of backward compatibility.

In all the examples the easily identifiable problem of changing 1 ½ hour to the two representations 1.5 decimal, and 1.3000 h.ms notation.

CONV Misc: menu

The most HP-compatible way is the direct conversions using HR $\leftrightarrow$ HMS:

s $\leftrightarrow$ year		dB $\leftrightarrow$ pwr
HR $\leftrightarrow$ HMS	ftlbf $\leftrightarrow$ Nm	dB $\leftrightarrow$ fld

(layout depending on CONV_{HP} (blue: SET, red: CLEAR))

1 hour 30 minutes (1.3), to/from 1.5 hours (1.5):

HP42S: (h.ms) 1.3 $\rightarrow$ HR results in 1.5 (h)
(h) 1.5 $\rightarrow$ HMS results in 1.3 (h.ms)

C47: CONV_{HP} (blue: SET) 1.3 [HR $\leftrightarrow$], results in 1.5
1.5 [HMS $\leftrightarrow$], results in 1.3

All input modes: NIM, X, Program

C47: CONV_{HP} (red: CLR) 1.3 [HMS $\rightarrow$], results in 1.5
1.5 [$\leftarrow$ HR], results in 1.3

All input modes: NIM, X, Program

Works in all input modes: NIM, X, Program

1 hour 30 minutes (1.3), to/from 1.5 hours (1.5):

CLK menu

The CLK menu time commands have both source and destination texts in each name to fully disambiguate uncertainties. Conversions are via the Time (T) data type are using H $\rightarrow$ T, H.MS $\rightarrow$ T, .ms, .ms⁻¹ & .d:

zyx $\rightarrow$ D	x $\rightarrow$ D	zyx $\rightarrow$ T	H.MS $\rightarrow$ T	H $\rightarrow$ T	YY ₀
D $\rightarrow$ zyx	D $\rightarrow$ x	T $\rightarrow$ zyx	.ms ⁻¹	.d	LEAP?
Date $\rightarrow$ D	D $\rightarrow$ J	Time $\rightarrow$ T	DT $\rightarrow$ J	J $\rightarrow$ DT	STOPW

CLK menu, via T:

1.3 [HMS $\rightarrow$ T] [.d], results in 1.5
1.5 [H $\rightarrow$ T] [ms⁻¹], results in 1.3

Works in all input modes: NIM, X, Program

Calculator keys, via T:

1.3 [.ms] [.d], results in 1.5

Works in NIM only

Calculator keys & CLK menu, via T:

1.5 [.ms] [ms⁻¹], results in 1.3

Works in X and Program only

ANGLE usage (DMS & .ms⁻¹):

C47 has a sexagesimal **dd:mm:ss** D.MS tagged angle, as part of the ANGLE type. D.MS is included in the sub-types DEG/RAD/GRAD/D.MS/MULT π angles, and these can bidirectionally be converted to standard decimal D.MS **dd.mmss** (D.MS_{decimal}), see below:

1. [$\Rightarrow$ D.MS]:

From sexagesimal coded decimal D.MS_{decimal} in X or in NIM, to a tagged angle:

Example: 1.3 [$\Rightarrow$ D.MS] X: is an ANGLE_{D.MS} 1°30'00"
Or 1.3 ENTER [$\Rightarrow$ D.MS]: X: is an ANGLE_{D.MS} 1°30'00"

From DEG_{decimal} in X or in NIM

1.5 [$\Rightarrow$ DEG] [$\Rightarrow$ D.MS] X: ANGLE_{D.MS} 1°30'00"
1.5 ENTER [$\Rightarrow$ DEG] [$\Rightarrow$ D.MS] X: ANGLE_{D.MS} 1°30'00"

[.ms] : is multi-functional, generally meant to input sexagesimal coded decimal dd.mmss or hh.mmss from keyboard as cyclic input, in direct Normal mode:

- From NIM only, **[.ms] [.ms]** does: D.MS_{decimal} (dd.mmss) to ANGLE_{D.MS};
- From X, and PRGM, **[.ms] [.ms]** does: D_{decimal} (dd.dddd) to ANGLE_{D.MS};
- From a TIME or ANGLE tag in X: **[.ms]** does TIME $\rightleftharpoons$ ANGLE;

H_{decimal} (hh.hhhh) to TIME (π): 1.5 ENTER **[.ms]** X: 1:30:00
H_{dms} (hh.mmss) to TIME (π): 1.3 **[.ms]** X: 1:30:00
TIME (π) to D.MS (Angle tag) X: 1°30'00"
D_{decimal} (dd.dddd) to D.MS (tag): 1.5 ENTER **[.ms] [.ms]** X: 1°30'00"
D.MS_{decimal} (dd.mmss) to D.MS: 1.3 **[.ms] [.ms]** X: 1°30'00"

2. [.d]: From any tagged angle sub-type in X, [.d] converts to decimal:

If the angle tag is in D.MS, [.d] will create decimal degrees, untagged

1°30'00 [.d] X: 1.5

If the angle tag is in DEG, [.d] will create decimal degrees, untagged

1.5° [.d] X: 1.5

If the angle tag is in RAD, [.d] will create decimal rads, untagged

0.0262^r [.d] X: 0.0262

If decimal degrees is needed, and the angle tag is not in degrees, first change to DEG or DMS using [$\Rightarrow$ D.MS] or [$\Rightarrow$ DEG] then [.d], will create decimal degrees, untagged.

1.667^g [$\Rightarrow$ D.MS] [.d] X: 1.5
1.667^g [$\Rightarrow$ DEG] [.d] X: 1.5

3. [.ms⁻¹]: Provides sexagesimal coded decimal from all tagged angles:

From any tagged angle sub-type in X, DIRECTLY to DMS_{decimal}:

1°30'00" [.ms⁻¹] X: 1.3
1.5° [.ms⁻¹] X: 1.3
0.0262^r [.ms⁻¹] X: 1.3
1.6667^g [.ms⁻¹] X: 1.3

Example from HP15C Advanced Functions Handbook, p13:

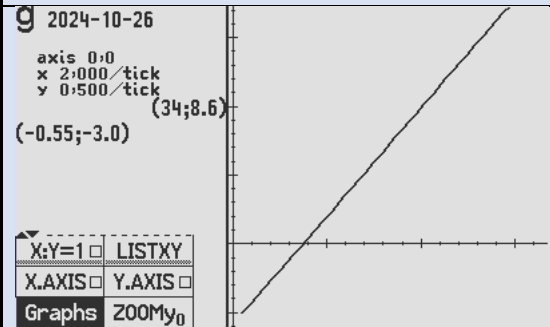
This HP example was chosen as example of solving using two different methods, both fully worked out, in the context where Time, Date and Angles are all being used and converted. The following are demonstrated below: **[.ms]**, **[D.MS]**, **[D.MS]**, **[>>D.M]**, **[H→T]**.

Example: During the winter of '78, Arctic explorer Jean-Claude Coulerre, isolated at his frozen camp in the far north, began scanning the southern horizon in anticipation of the sun's reappearance. Coulerre knew that the sun would not be visible to him until early March, when it reached a declination of 5° 18'S. On what day and time in March was the chilly explorer's vigil rewarded?

The time in March when the sun reached 5° 18'S declination can be computed by solving the following equation for t:

$D = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$ where D is the declination in degrees, t is the time in days from the beginning of the month, and $a_4 = 4.2725 \times 10^{-8}$ $a_3 = -1.9931 \times 10^{-5}$ $a_2 = 1.0229 \times 10^{-3}$ $a_1 = 3.7680 \times 10^{-1}$ $a_0 = -8.1806$. This equation is valid for $1 \leq t < 32$, representing March, 1978.

Method 1	Method 2								
Using the direct method in the 15C Advanced Functions book, copied <i>directly</i> and applied/changed minimally to C47	Using the 15C method, but using the most elegant solution on C47								
The solution to Coulerre's problems is the value of t satisfying $-5.3000 = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$. Expressed in the form required is $-5.3 = a_4t^4 + a_3t^3 + a_2t^2 + a_1t - 8.1806$ where the last, constant term would now incorporate the value of the declination: $0 = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_{00} - a_{01}$ Using Horner's method, the function to be set equal to zero is (also changing the a_0 constant to a_{00} and a_{01}): $f(t) = 0 = (((a_4t + a_3)t + a_2)t + a_1)t + a_{00} - a_{01}$ To shorten the subroutine, store and recall the constants using the registers corresponding to the exponent of t.	Enter the expression in <i>red</i> on the left which is to be equal to 0, into the EQN editor, directly, not using Horner's method: Type directly using the EQN menu, Type f [EQN], NEW, $a_4 \cdot x^4 + \dots$ and press ENTER. $a_4 \cdot x^4 + a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_{00} - a_{01}$<table><tr><td>a1</td><td>a00</td><td>a01</td><td></td></tr><tr><td>a4</td><td>x*</td><td>a3</td><td>a2</td></tr></table> ToolS Calc f Note: "D" cannot be a variable as D refers to a stack register and is invalid for an expression, hence "a00" and "a01" were chosen instead. The default C47 'poly4' can be used as formula shortcut with said changes (type f[EQN], NEW, Dn, Dn, f[Poly4], change a0 to a00 and add a term -a01).	a1	a00	a01		a4	x*	a3	a2
a1	a00	a01							
a4	x*	a3	a2						
0001: LBL 'ARCTIC' 0002: MVAR 'x'	0001: LBL 'ARCTIC' 0002: 4.272 5.10^-8 Press f Solve and enter the constants into b4-b0 by pressing 4.2725E-8 [b4], etc.								

0003: RCL 'a4'	0003: STO 'a4'	Enter the value for D -5.18 [.ms] [.ms]
0004: RCL 'x'	0004: -1.993 1.10^-5	(or -5.18 [D.MS]) into a01 by [a01].
0005: *	0005: STO 'a3'	Solve for x by guessing x values 1 and 31
0006: RCL 'a3'	0006: 1.022 9.10^-3	by pressing [1] [x] [31] [x] and solve
0007: +	0007: STO 'a2'	for x by pressing [x].
0008: RCL 'x'	0008: 3.768 0.10^-1	x is computed to be 7.5137 days from
0009: *	0009: STO 'a1'	the beginning of the month.
0010: RCL 'a2'	0010: -8.180 6	Do fractional part using shortcut g#F (or
0011: +	0011: STO 'a00'	FP) (or simply subtract 7), then multiply
0012: RCL 'x'	0012: 5.18	by 24 to get hours, and press [H→T] to
0013: *	0013: >>D.MS	get 7 days + 12:19:45 after the beginning
0014: RCL 'a1'	0014: CHS	of the month.
0015: +	0015: STO 'a01'	
0016: RCL 'x'	0016: PGMSLV	
0017: *	'ARCTICf'	
0018: RCL 'a00'	0017: 1	
0019: +	0018: STO 'x'	
0020: RCL 'a01'	0019: SOLVE 'x'	
0021: -	0020: END	
0022: .END.		
	Run 'ARCTIC' and the	
	result is 7.5137 days.	
	Multiply by 24 and do	
	[H→T] .	
		Check if other roots are present, press
		TOOLS Graphs [1] [↓X] 31 [↑X] [Draw]
		to confirm that no other roots are
		present on the x-range applicable 1-31.

HISTORY on H, H.MS, HR and HMS:

1. What did HP do decades ago using sexagesimal-formatted-decimal, HMS and HR?

The chronology in selected HP models, from **→H** and **→H.MS**, to **→HR** and **→HMS**: The common fact across the prior HP RPN machines was that they all only had decimal digits available, and could only deal with time and angle when the available decimal digits were coded in digit pairs to form 'sexagesimal formatted decimal hours and degrees', example:

HP67₁₉₇₆, HP29₁₉₇₇, HP34C₁₉₇₉, HP15C₁₉₈₂:

These models used **→H** and **→H.MS**, with two variants on the arrow directions:

- 1.3 **→H** results in 1.5
- 1.5 **→H.MS** results in 1.3

H ↔ H.MS (67) **→H.MS→H** (34C)

HP41C₁₉₇₉, HP32S₁₉₈₈, HP42S₁₉₈₈, HP32SII₁₉₉₁:

These models used **→HR** and **→HMS** in exactly the same way as the above:

- 1.3 **→HR** results in 1.5
- 1.5 **→HMS** results in 1.3

→HR (41C OM)
→HMS (41C OM)

The HP41C OM and device were different and the 41C device omitted the arrows:

- 1.3 **HR** results in 1.5
- 1.5 **HMS** results in 1.3

HR (41C device)
HMS (41C device)

2. What did WP43S do regarding d.ms and h.ms located on the bezel?

This section was summarized some years ago, hence the reference to WP43S. It is listed for historical benefit and does not necessarily reflect on the latest WP43 specifications and in fact may be wrong. I did not check WP43.

f[h.ms] was a two-fold function, different in NIM and in closed X:

With open input from NIM:

- **1.3 f[h.ms][.d] results in 1.5**
1.3 (01.30 00 hh.mmss) could never be from X, it must be in NIM (input line) which prevents it from being programmed.

With closed input from X:

- **1.5 EXIT f[h.ms] results in a TIME type 1:30:00,**
An output of 1.3 (01.30 00 hh.mmss) is not possible.

More examples of f[h.ms]:

- 1.5 f[h.ms] results in 1:50:00 (from open input NIM)
- 1.5 EXIT f[h.ms] results in 1:30:00 (From closed input X in RUM)
- If X contains type TIME 1:30:00, then [.d] results in 1.5

f[d.ms] is a two-fold function, different in NIM and in closed X:

- With closed input it ONLY changes the ADM (mode) to $\angle$ in the status bar.

f[d.ms], when not acting on an open input in NIM, sets the ADM to d.ms, not altering the X-Register, which is similar in action to the softmenu commands DEG, RAD, etc. in [MODE].

- With open input in NIM, f[d.ms] closes the input and takes the digits to mean the following: D.MMSSffffffffff, where D is whole degrees, MM minutes, SS seconds and fff... fractional seconds.

With open input from NIM:

- **1.3 f[d.ms][.d] results in 1.5**
1.3 can never be in X, it must be in NIM which prevents it from being programmed. Additionally f[d.ms] on closed X sets the ADM to $\angle$.
1.3 g[D.MS→D][.d] results in 1.5
More similar options using ADM to influence input are available as above, the implied "ADM" superimposed in red for understanding:

D→R	R→D	Mπ→R	R→Mπ	D→D.MS	D.MS→D
DEG→ ^{ADT}	RAD→ ^{ADT}	MULπ→ ^{ADT}	MIL→ ^{ADT}	GRAD→ ^{ADT}	D.MS→ ^{ADT}
^{ADT} →DEG	^{ADT} →RAD	^{ADT} →MULπ	^{ADT} →MIL	^{ADT} →GRAD	^{ADT} →D.MS

With open input from NIM:

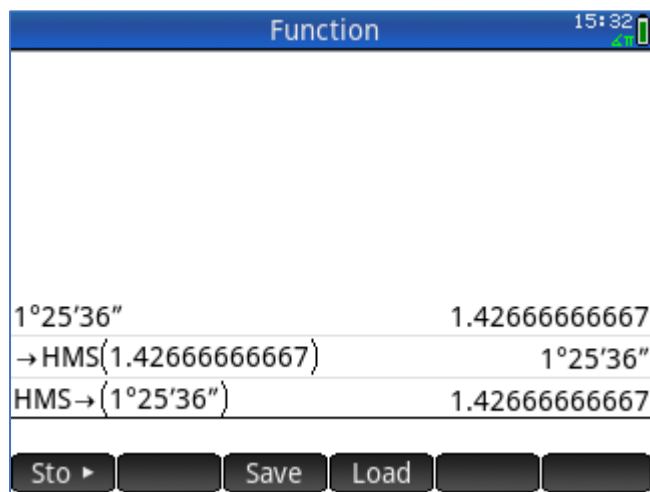
- **1.5 g[D→D.MS]** gets to an angle Tag 1°30'00''
- An output of 1.3 (for 01°30'00'' dd.mmss) is not possible.

More examples of f[d.ms]:

- 1.5 f[d.ms] results in 1°50' 0.00'' (always in degrees, regardless of ADM)
- 1.5 ENTER f[d.ms] does not change X and sets ADM.
- 1.5 g[↵][→D.MS] gives 1°30' 0.00'' (always in degrees, regardless of ADM)

3. What does Prime do regarding DMS and HMS?

Sexagesimally formatted decimal is not used in Prime, it appears.



The screenshot shows the HP Prime Function application interface. At the top, there is a blue header bar with the word "Function" on the left and the time "15:32" on the right. Below the header is a large white area for the function editor. At the bottom of the screen is a row of buttons: "Sto ►", "Save", "Load", and two empty buttons. The main display area shows the following text:

1°25'36"	1.42666666667
→ HMS(1.42666666667)	1°25'36"
HMS→(1°25'36")	1.42666666667

The HP Prime has an actual DMS mode. It uses ->HMS and HMS-> to convert with reals.

4. How did C47 previously (pre-00.109.02.06) achieve the above HP conversions?

Convert 1.3000 or 1h30 to 1.5 hours:

- `1.3 [.ms][.d]` results in 1.5
Input '1.3' must not mean 1h30 if input is closed and/or in X.
Input '1.3' only means 1h30 if input is open, i.e. in NIM.
Input '1.3' meaning 1h30 cannot be entered into a program (in PEM).
- Using stack entry: `1 ENTER 30 ENTER 0 [→TIME][.d]` results in 1.5
Stack entry of h, m & s can be used with the last input in X or from NIM.
This is the only way to enter 1h30 in PEM as program.

Convert 1.5 hours to 1.3000 for 1h30:

- `1.5 ENTER [.ms][.ms-1]` results in 1.3,
Input '1.5 ENTER' implies X input, and '1.5' must not mean 1h50.
Input cannot be done in NIM, it must be closed in X.
- `1.5 ENTER [→h.ms][.ms-1]` results in 1.3
[→h.ms] and [.ms] are identical, see previous bullet.
- Stack output: `1.5 ENTER [.ms] [TIME→]` results in 1, 30, 0
Stack output of h, m & s can be used with the last input in X or from NIM.

The problems with the pre-00.109.02.06 version:

- H→H.MS and H.MS→H via the TIME type, was inefficient & un-intuitive.
- Inconsistent NIM and X input methods for [.ms] and [→h.ms].
- No program entry for [.ms] and [→h.ms] with hh.mmss input.
- No reference to HP naming.
- The existing function →TIME (h, m, s →TIME) does not have an equivalent to convert from decimal hours (e.g. h.hh →TIME).

Author Jaco Mostert

Copyright (C) 2024 The C47 Development Team. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be downloaded from gnu.org.