HHC 2026

RPN ENTER 47 +

Endurance Meets Evolution

Gary J. Carter

Fairview, Texas

HP35, HP41C, HP16C, HP50g

DM41X, DM32, DM42n, C47, R47



The 47 Project Team



- Robbert van Meenen, Documentation (25, 41, 48)

- Jaco Mostert, Developer (67, 11, 28, 42)

What is an R47?

The next RPN scientific

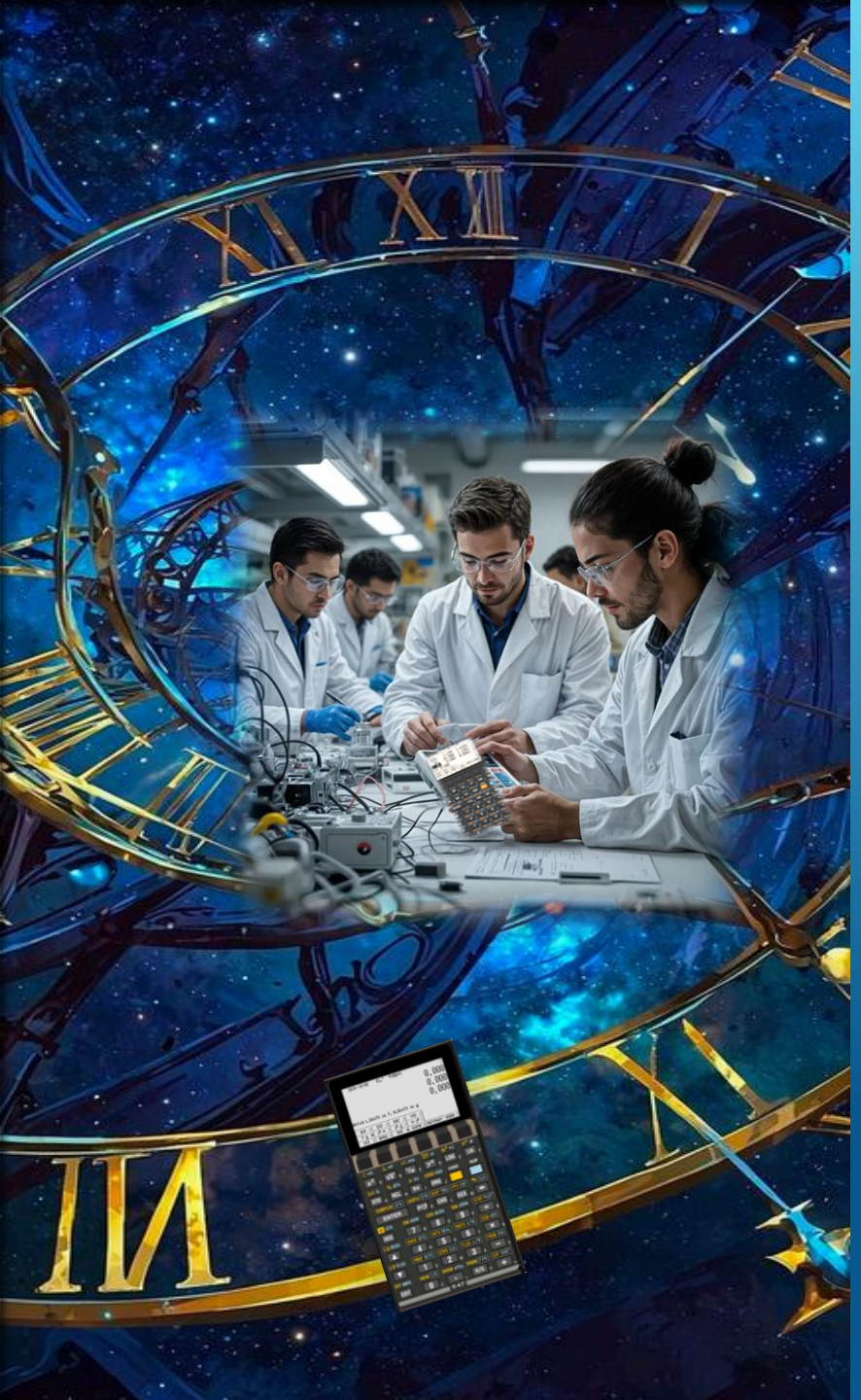
The R47 is a powerful, RPN-based programmable scientific calculator with an extensive feature set. Its software has been refined over many years, born from the vision of a *dream calculator* developed by a team of engineers and mathematicians, all dedicated calculator enthusiasts.

Background and lineage

Community-developed and manufactured by SwissMicros, the R47 runs firmware that is functionally identical to that of the C47, which offers a way to transform a DM42n into a feature-rich RPN calculator using a dedicated keypad overlay.

Inspired by the HP-41C and HP-42S, the R47 builds on this foundation with a redesigned keypad featuring two shift keys, preserving a familiar feel while being entirely new in execution.



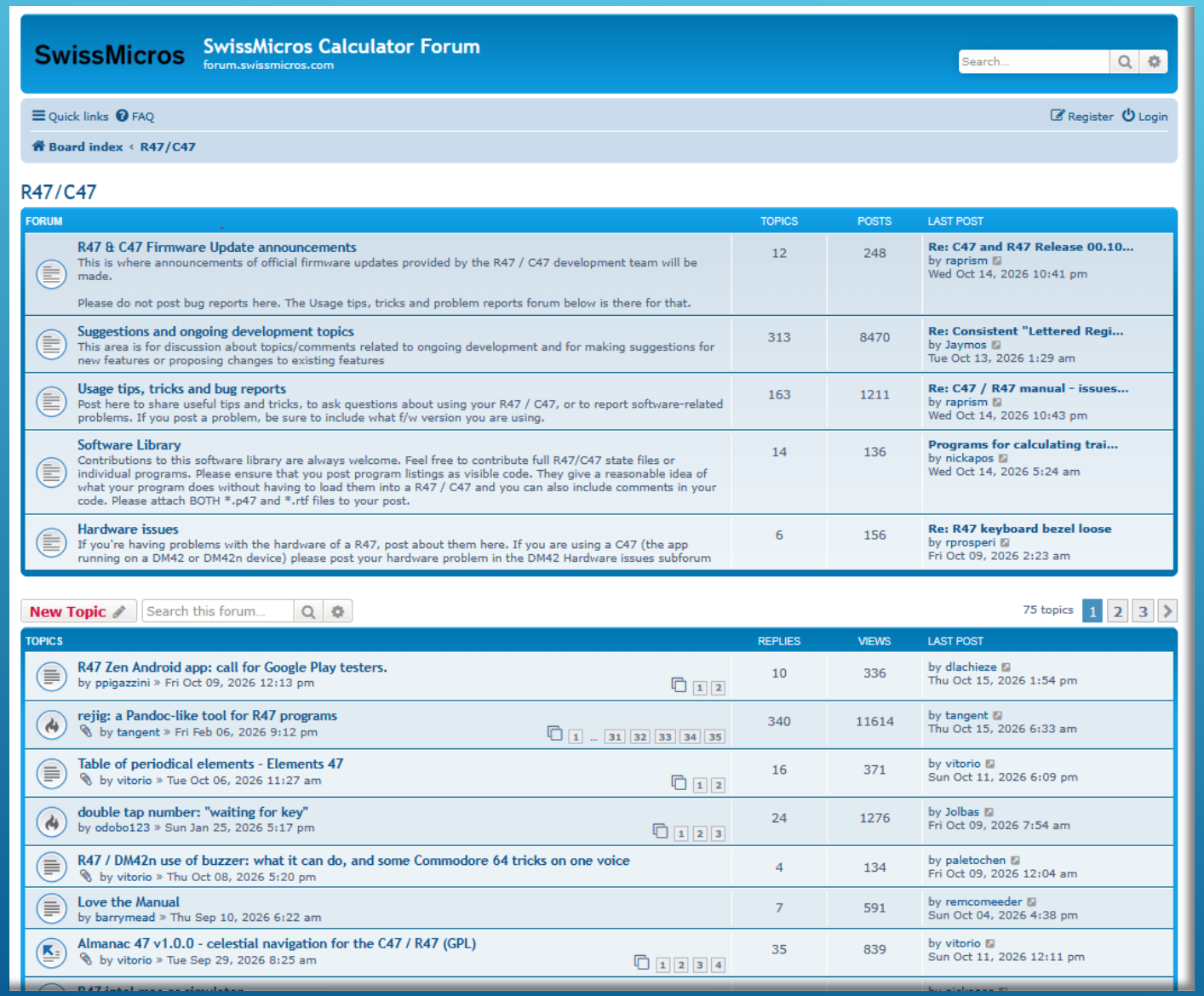


If a new, advanced RPN scientific handheld calculator were to be developed today …

… **What would HP do?**



C47/R47 Community

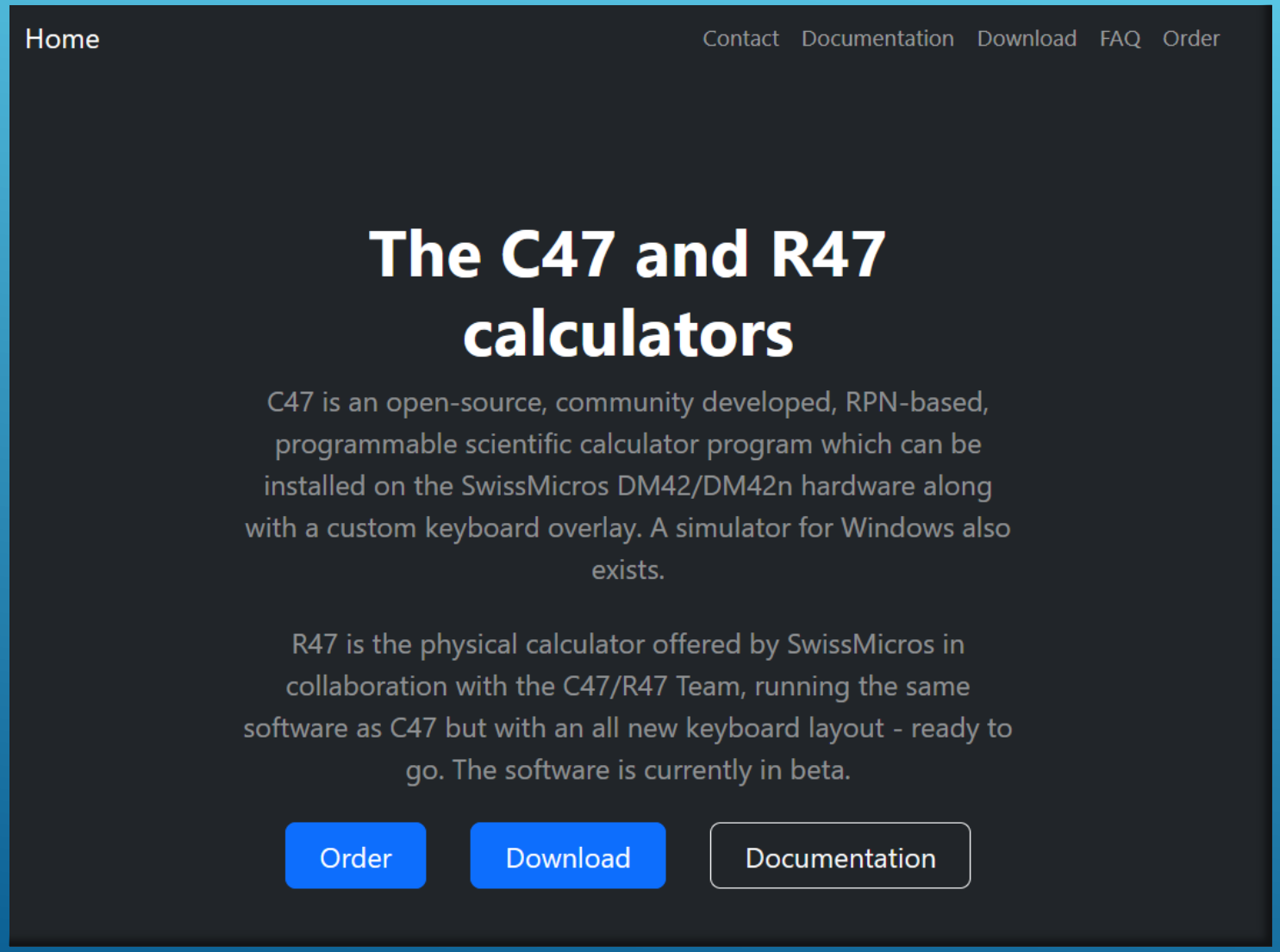


forum.swissmicros.com

R47/C47

- *R47 & C47 Firmware Update announcements*
- *Suggestions and ongoing development topics*
- *Usage tips, tricks and bug reports*
- *Software Library*
- *Hardware issues*

C47/R47 Community



https://47calc.com

- Order
- Download
- Documentation
- Contact
- Frequently Asked Questions

C47/R47 Project

- 47calc.com
  - Ordering
  - Download
  - Documentation
- R47
- C47 Bezel
- Purchase R47 from SM
- Purchase C47 Bezel
- Software C47
- Software R47
- Faceplate.pdf

00.109.05.00b2 - latest
00.109.04.00b0 - archive
00.109.03.03b0 - archive



- Reference
- Searchable Indexes
- Community
- Unofficial
- HPCC 2024
- Allschwil 2022
- Videos
- Browse the PDF docs
- R47 Index Cards
- C47 Index Cards
- Read the Wiki
- Read the Wiki

Commands and Functions
Flags and Settings
Menus

C47/R47 Searchable Indexes

R47v0 Documentation: Commands & Functions index, 2026 August 4 Rev5

Acceleration conversions	ft/s²→m/s²	ft/s² to m/s²	Menu: FLoads: p1 gF6	Function (monadic, linked)
Acceleration conversions	in/s²→m/s²	in/s² to m/s²	Menu: FLoads: p1 fF6	Function (monadic, linked)
Acceleration conversions	m/s²→ft/s²	m/s² to ft/s²	Menu: FLoads: p1 gF5	Function (monadic, linked)
Acceleration conversions	m/s²→in/s²	m/s² to in/s²	Menu: FLoads: p1 fF5	Function (monadic, linked)
Angle conversions	deg→grad	deg to grad	Menu: Angle: p1 F3	Function (monadic, linked)
Angle conversions	deg→rad	deg to rad	Menu: Angle: p1 F1	Function (monadic, linked)
Angle conversions	DRG	Degrees-radians-gradians	Menu: MyM p1 F1	Function (monadic, cyclic, nonpgm)
Angle conversions	DRG	Degrees-radians-gradians	Menu: MyM p5 F1	Function (monadic, cyclic, nonpgm)
Angle conversions	DRG	Degrees-radians-gradians	Menu: MyM p9 F1	Function (monadic, cyclic, nonpgm)
Angle conversions	DRG	Degrees-radians-gradians	Menu: TRG p1 F6	Function (monadic, cyclic, nonpgm)
Angle conversions	DRG	Degrees-radians-gradians	Keyboard row 3 col 4 [DRG]	Function (monadic, cyclic, nonpgm)
Angle conversions	DRG	Degrees-radians-gradians	Keyboard row 3 col 4 normalSIM [>]	Function (monadic, cyclic, nonpgm)
Angle conversions	grad→deg	grad to deg	Menu: Angle: p1 F4	Function (monadic, linked)
Angle conversions	grad→rad	grad to rad	Menu: Angle: p1 F5	Function (monadic, linked)
Angle conversions	rad→deg	rad to deg	Menu: Angle: p1 F2	Function (monadic, linked)
Angle conversions	rad→grad	rad to grad	Menu: Angle: p1 F6	Function (monadic, linked)
Angle tagging or conversion	⇒D.MS	Set D.MS tag or convert to D.MS	Menu: TRG p1 F4	Function (monadic)
Angle tagging or conversion	⇒D.MS	Set D.MS tag or convert to D.MS	Menu: TRG_R47 p1 fF4	Function (monadic)
Angle tagging or conversion	⇒D.MS	Set D.MS tag or convert to D.MS	Menu: Ellipt p1 gF5	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: TRG p1 F1	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: TRG... p1 F1	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: TRG_R47 p1 fF1	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: VECTOR p1 gF1	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: VECTOR p2 gF1	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: ELEC p1 fF1	Function (monadic)
Angle tagging or conversion	⇒DEG	Set DEG tag or convert to DEG	Menu: Ellipt p1 gF4	Function (monadic)
Angle tagging or conversion	⇒GRAD	Set GRAD tag or convert to GRAD	Menu: TRG p1 F3	Function (monadic)
Angle tagging or conversion	⇒GRAD	Set GRAD tag or convert to GRAD	Menu: TRG... p1 F3	Function (monadic)
Angle tagging or conversion	⇒GRAD	Set GRAD tag or convert to GRAD	Menu: TRG_R47 p1 fF3	Function (monadic)
Angle tagging or conversion	⇒MULπ	Set MULπ tag or convert to MULπ	Menu: TRG p1 F5	Function (monadic)
Angle tagging or conversion	⇒MULπ	Set MULπ tag or convert to MULπ	Menu: TRG_R47 p1 fF5	Function (monadic)
Angle tagging or conversion	⇒MULπ	Set MULπ tag or convert to MULπ	Menu: VECTOR p1 gF3	Function (monadic)
Angle tagging or conversion	⇒MULπ	Set MULπ tag or convert to MULπ	Menu: VECTOR p2 gF3	Function (monadic)
Angle tagging or conversion	⇒RAD	Set RAD tag or convert to RAD	Menu: TRG p1 F2	Function (monadic)
Angle tagging or conversion	⇒RAD	Set RAD tag or convert to RAD	Menu: TRG... p1 F2	Function (monadic)
Angle tagging or conversion	⇒RAD	Set RAD tag or convert to RAD	Menu: TRG_R47 p1 fF2	Function (monadic)

C47/R47 Searchable Indexes

R47v0 Documentation: Flags & Settings index, 2026 August 4 Rev2

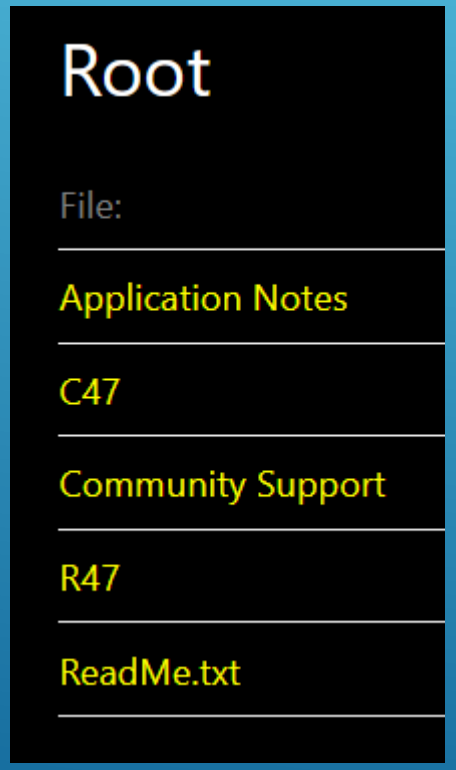
Angle tagging or conversion	D.MS ○	Set ADM to D.MS	Menu: TRG p1 gF4	Setting (pgm)
Angle tagging or conversion	D.MS ○	Set ADM to D.MS	Menu: TRG_R47 p1 F4	Setting (pgm)
Angle tagging or conversion	DEG ◉	Set ADM to DEG	Menu: MODE p1 F1	Setting (pgm)
Angle tagging or conversion	DEG ◉	Set ADM to DEG	Menu: TRG p1 gF1	Setting (pgm)
Angle tagging or conversion	DEG ◉	Set ADM to DEG	Menu: TRG_R47 p1 F1	Setting (pgm)
Angle tagging or conversion	DEG ◉	Set ADM to DEG	Menu: XFCNS p1 fF1	Setting (pgm)
Angle tagging or conversion	DEG ◉	Set ADM to DEG	Menu: ELEC p1 gF1	Setting (pgm)
Angle tagging or conversion	GRAD ○	Set ADM to GRAD	Menu: MODE p1 F3	Setting (pgm)
Angle tagging or conversion	GRAD ○	Set ADM to GRAD	Menu: TRG p1 gF3	Setting (pgm)
Angle tagging or conversion	GRAD ○	Set ADM to GRAD	Menu: TRG_R47 p1 F3	Setting (pgm)
Angle tagging or conversion	MULπ ○	Set ADM to MULπ	Menu: TRG p1 gF5	Setting (pgm)
Angle tagging or conversion	MULπ ○	Set ADM to MULπ	Menu: TRG_R47 p1 F5	Setting (pgm)
Angle tagging or conversion	RAD ○	Set ADM to RAD	Menu: MODE p1 F2	Setting (pgm)
Angle tagging or conversion	RAD ○	Set ADM to RAD	Menu: TRG p1 gF2	Setting (pgm)
Angle tagging or conversion	RAD ○	Set ADM to RAD	Menu: TRG_R47 p1 F2	Setting (pgm)
Angle tagging or conversion	RAD ○	Set ADM to RAD	Menu: XFCNS p1 fF2	Setting (pgm)
Angle tagging or conversion	RAD ○	Set ADM to RAD	Menu: ELEC p1 gF2	Setting (pgm)
Bits and bytes	10CMPL ○	BCD 10's complement	Menu: BITSET p1 gF2	Setting (pgm)
Bits and bytes	1COMPL ○	1's complement	Menu: BITSET p1 fF1	Setting (pgm)
Bits and bytes	2COMPL ◉	2's complement	Menu: BITSET p1 fF2	Setting (pgm)
Bits and bytes	9CMPL ○	BCD 9's complement	Menu: BITSET p1 gF1	Setting (pgm)
Bits and bytes	BCD □	Binary coded decimal	Menu: BITSET p1 gF4	Setting
Bits and bytes	BCD □	Binary coded decimal	Menu: SYS.FL p1 fF4	Flag
Bits and bytes	BCD □	Binary coded decimal	Menu: TamFlag p1 gF1	Flag
Bits and bytes	BCD □	Binary coded decimal	Menu: CFLG p1 fF4	Flag
Bits and bytes	BCDUNS ◉	BCD unsigned	Menu: BITSET p1 gF3	Setting (pgm)
Bits and bytes	CARRY □	Carry (bit)	Menu: BASE2 p2 fF3	Setting
Bits and bytes	CARRY □	Carry (bit)	Menu: TamFlag p1 gF5	Flag
Bits and bytes	CARRY □	Carry (bit)	Menu: SYS.FL p1 fF6	Flag
Bits and bytes	CARRY □	Carry (bit)	Menu: CFLG p1 fF6	Flag
Bits and bytes	OVERFL □	Overflow	Menu: BASE2 p2 fF2	Setting
Bits and bytes	OVERFL □	Overflow	Menu: TamFlag p1 gF4	Flag
Bits and bytes	OVERFL □	Overflow	Menu: SYS.FL p3 gF5	Flag
Bits and bytes	OVERFL □	Overflow	Menu: CFLG p3 gF5	Flag

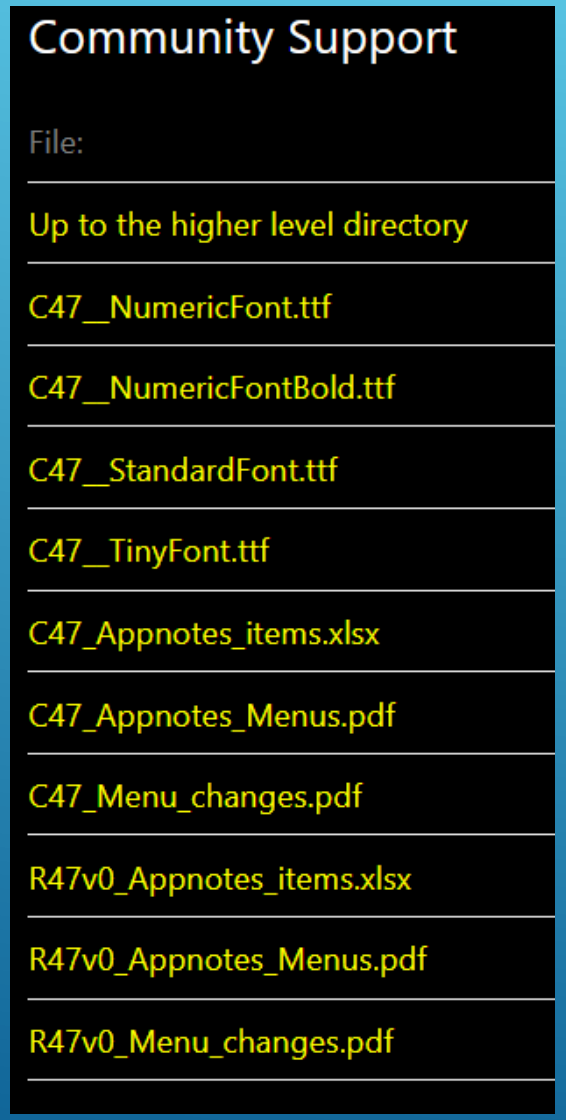
C47/R47 Searchable Indexes

R47v0 Documentation: Menus index, 2026 August 4 Rev1

Angular/Speed conversions	Angle:	Angle and angular velocity conversions	Menu: CONV p1 f F3	MENU
Area conversions	Area:	Area conversions	Menu: CONV p1 F5	MENU
Bits and bytes	BITS	Bitwise functions	Menu: BLUE47 p3 f F2	MENU
Bits and bytes	BITS	Bitwise functions	Keyboard row 6 col 2 g [4]	MENU
Bits and bytes	BITSET	Bit set functions	Menu: BASE p1 g F6	MENU
Bits and bytes	BITSET	Bit set functions	Menu: BASE p2 g F6	MENU
Bits and bytes	BITSET	Bit set functions	Menu: BITS p2 g F6	MENU
Bits and bytes	BITSET	Bit set functions	Menu: BASE2 p2 g F6	MENU
Bits and bytes	BITSET	Bit set functions	Menu: BITSHFT p1 g F6	MENU
Bits and bytes	BITSHFT (BITS↲2)	Bit shift functions	Menu: BASE2 p2 f F6	MENU (item)
Bits and bytes	BITSHFT (BITS↲2)	Bit shift functions	Menu: BASE p1 f F6	MENU (item)
Catalog	ALL (ALLVARS)	All variables	Menu: VARS p1 F6	MENU (ASM)
Catalog	ANGLES	Angle variables	Menu: VARS p1 F4	MENU
Catalog	CAT (CATALOG)	Catalog	Keyboard row 8 col 5 f [+]	MENU
Catalog	CAT (CATα)	Catalog (alpha)	Keyboard row 8 col 5 alpha f [+]	MENU
Catalog	CHARS	Characters	Menu: CAT p1 F3	MENU
Catalog	CHARS	Characters	Menu: CAT p1 F3	MENU
Catalog	CNST	Constants	Menu: CAT p1 F2	MENU (ASM)
Catalog	CNST	Constants	Menu: BLUE47 p2 f F5	MENU (ASM)
Catalog	CNST	Constants	Menu: CAT p1 F2	MENU (ASM)
Catalog	CNST	Constants	Keyboard row 8 col 5 g [+]	MENU (ASM)
Catalog	CONFIGS	Config variables	Menu: VARS p1 f F1	MENU
Catalog	CPXS	Complex variables	Menu: VARS p1 F2	MENU
Catalog	DATES	Date variables	Menu: VARS p1 f F3	MENU
Catalog	FCNS	Functions	Menu: CAT p1 F1	MENU (ASM)
Catalog	FCNS (FCNSe)	Functions (equation)	Menu: CAT p1 F1	MENU (ASM)
Catalog	L.INTS	Long integer variables	Menu: VARS p1 F5	MENU
Catalog	MATRS	Real matrix or vector variables	Menu: VARS p1 f F2	MENU
Catalog	MENUS	Menus	Menu: CAT p1 F6	MENU (ASM)
Catalog	MENUS	Menus	Menu: DELITM p1 F6	MENU (ASM)
Catalog	NUMBRS	Number variables	Menu: VARS p1 F1	MENU
Catalog	PROGS	Programs	Menu: CAT p1 F4	MENU (ASM)
Catalog	PROGS	Programs	Menu: DELITM p1 F4	MENU (ASM)
Catalog	REALS	Real variables	Menu: VARS p1 F3	MENU
Catalog	S.INTS	Short integer variables	Menu: VARS p1 f F5	MENU
Catalog	STRINGS	Text string variables	Menu: VARS p1 f F6	MENU
Catalog	TIMES	Time variables	Menu: VARS p1 f F4	MENU
Catalog	VARS	Variables	Menu: CAT p1 F5	MENU

C47/R47 Reference





Root

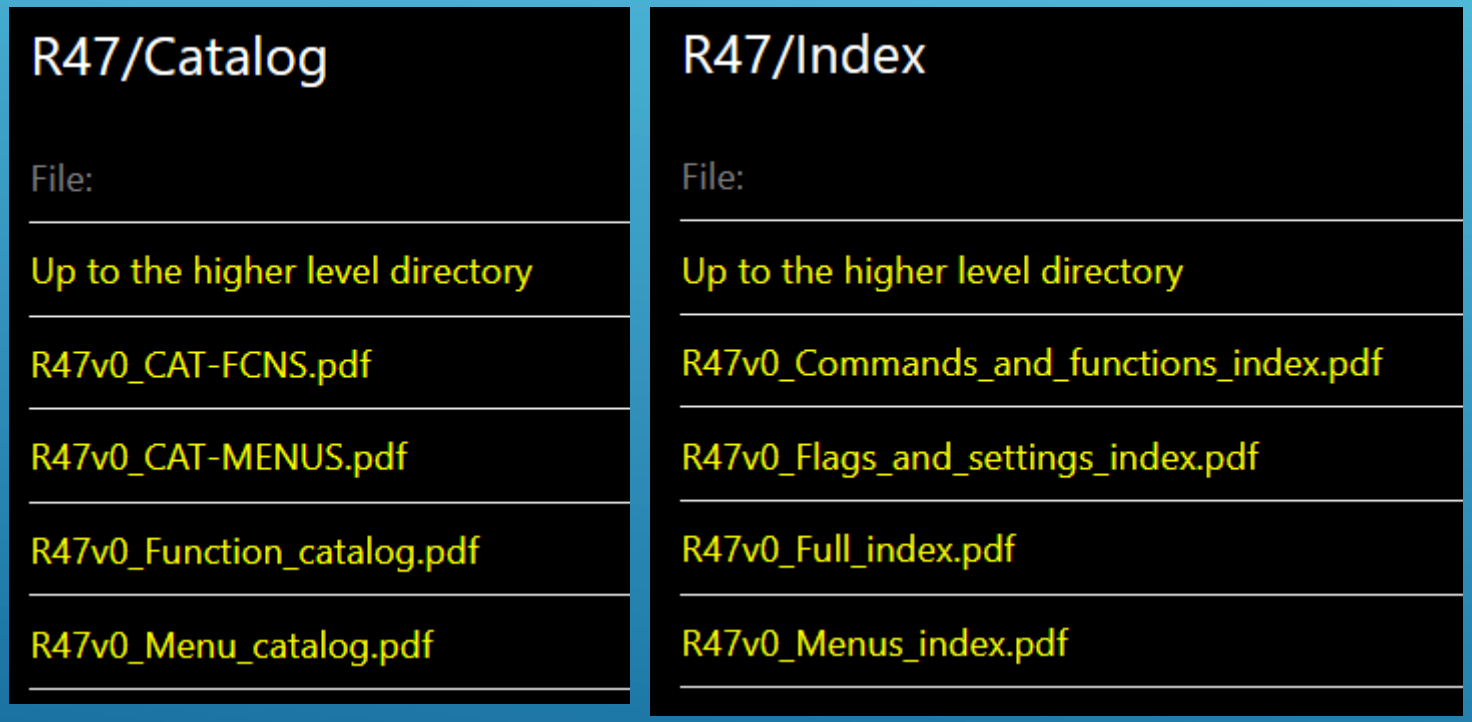
&

Community Support

C47/R47 Application Notes

AN0000 - Firmware installation
AN0001 - Complex number engineering example
AN0002 - Graphs and complex solver
AN0003 - Graphing
AN0005 - Programmed sums
AN0006 - Electrical engineering functions
AN0007 - Structured programming guide
AN0008 - Programmed sums 1
AN0009 - Programmed sums 2
AN0010 - Complex solver continued
AN0011 - EQN and ADV menus
AN0012 - Time value of money
AN0013 - Graphs
AN0014 - dms, hms and ms
AN0015 - Setting groups, write and read
AN0016 - The 1000 digit issue
AN0017 - Display formats
AN0018 - Keyboard
AN0020 - Matrix and vector
AN0021 - Printing
AN0022 - Solve, integrate and plot
AN0023 - Elliptic functions, with annex
AN0024 - Custom conversion pairs
AN0025 - The d47 file format
AN0026 - HP-42S compatibility
AN0027 - Structural conversions
AN0028 - Configurability tour
AN0029 - First steps
AN0030 - Measurement Uncertainty
AN0031 - Algebraic number identification
AN0032 - Character set
AN0033 - Polynomial roots
AN0034 - Units and Unit Algebra
AN0035 - Financial Package Guide

R47 Specific Reference

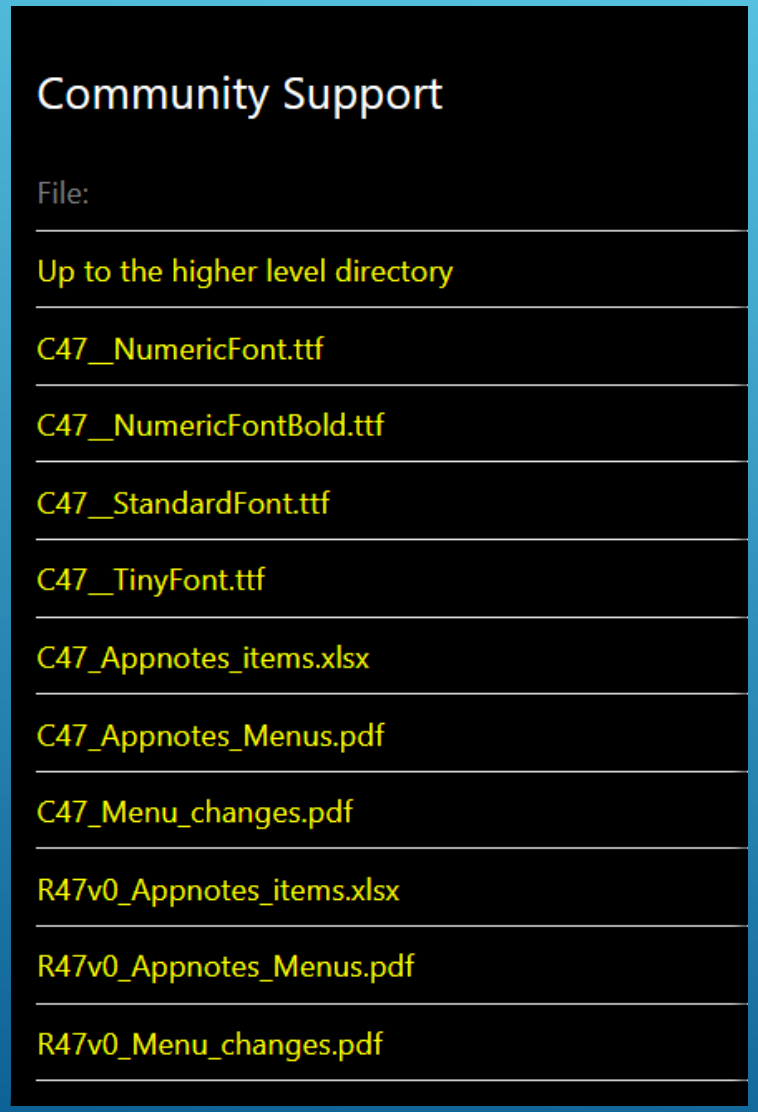


Catalogs

&

Indexes

C47/R47 Community Support



Authoring Support:

Fonts, Appnote Items, Appnote Menus, Menu Changes

R47 Specific Reference

R47/Keyboard

File:

- Up to the higher level directory
- Layers
- Rows
- R47v0_Delta_keyboard.pdf
- R47v0_Keyboard_categories.pdf
- R47v0_Keyboard_Keys_USERMODE_XFN.pdf
- R47v0_Keyboard_layers.pdf
- R47v0_Keyboards_side_by_side.pdf

R47/Keyboard/Layers

File:

- Up to the higher level directory
- R47v0_Keyboard_Keys.pdf
- R47v0_Keyboard_Keys_alpha.pdf
- R47v0_Keyboard_Keys_alphaSIM.pdf
- R47v0_Keyboard_Keys_doublepress.pdf
- R47v0_Keyboard_Keys_hashTAM.pdf
- R47v0_Keyboard_Keys_keyIndex.pdf
- R47v0_Keyboard_Keys_keyType.pdf
- R47v0_Keyboard_Keys_longerpress.pdf
- R47v0_Keyboard_Keys_longestpress.pdf
- R47v0_Keyboard_Keys_longpress.pdf
- R47v0_Keyboard_Keys_SIM.pdf
- R47v0_Keyboard_Keys_TAM.pdf
- R47v0_Keyboard_Keys_triplepress.pdf

R47/Keyboard/Rows

File:

- Up to the higher level directory
- R47v0_Keyboard_row_1.pdf
- R47v0_Keyboard_row_2.pdf
- R47v0_Keyboard_row_3.pdf
- R47v0_Keyboard_row_4.pdf
- R47v0_Keyboard_row_5.pdf
- R47v0_Keyboard_row_6.pdf
- R47v0_Keyboard_row_7.pdf
- R47v0_Keyboard_row_8.pdf

Keyboard Layers, Rows, & Categories

R47 Specific Reference

R47/Menus

File:

Up to the higher level directory

R47v0_Menu_42_page_1.pdf

R47v0_Menu_42_page_2.pdf

R47v0_Menu_ADV_page_1.pdf

R47v0_Menu_ALLVARS_page_1.pdf

R47v0_Menu_ALLVARS_page_2.pdf

R47v0_Menu_ALPHA__page_1.pdf

R47v0_Menu_alpha_page_1.pdf

R47v0_Menu_alpha-FN_page_1.pdf

R47v0_Menu_alphaINTL_page_1.pdf

R47v0_Menu_alphaINTL_page_2.pdf

R47v0_Menu_alphaINTL_page_3.pdf

R47v0_Menu_alphaINTL_page_4.pdf

R47v0_Menu_alphaintl-lower_page_1.pdf

R47v0_Menu_alphaintl-lower_page_2.pdf

R47v0_Menu_alphaintl-lower_page_3.pdf

R47v0_Menu_alphaintl-lower_page_4.pdf

R47v0_Menu_alphaMATH_page_1.pdf

R47v0_Menu_alphaMATH_page_2.pdf

R47v0_Menu_alphaMATH_page_3.pdf

R47v0_Menu_alphaMATH_page_4.pdf

R47v0_Menu_alphaMATH_page_5.pdf

R47v0_Menu_alphaMATH_page_6.pdf

R47v0_Menu_alphaMATH_page_7.pdf

R47v0_Menu_alphaMisc_page_1.pdf

R47v0_Menu_alphaMisc_page_2.pdf

R47v0_Menu_alphaMisc_page_3.pdf

R47v0_Menu_alphaMisc_page_4.pdf

R47v0_Menu_alphaMisc_page_5.pdf

R47v0_Menu_Alpha-Omega_page_1.pdf

R47v0_Menu_Alpha-Omega_page_2.pdf

R47v0_Menu_Alpha-Omega_page_3.pdf

R47v0_Menu_alpha-omega-lower_page_1.pdf

R47v0_Menu_alpha-omega-lower_page_2.pdf

R47v0_Menu_alpha-omega-lower_page_3.pdf

R47v0_Menu_AMORT_page_1.pdf

R47v0_Menu_Angle_page_1.pdf

R47v0_Menu_ANGLES_page_1.pdf

R47v0_Menu_Area_page_1.pdf

R47v0_Menu_Arrow_page_1.pdf

R47v0_Menu_ASSESS_page_1.pdf

R47v0_Menu_AUDIO_page_1.pdf

R47v0_Menu_BASE_page_1.pdf

R47v0_Menu_BASE_page_2.pdf

R47v0_Menu_BASE2_page_1.pdf

R47v0_Menu_BASE2_page_2.pdf

R47v0_Menu_Binom_page_1.pdf

R47v0_Menu_BITS_page_1.pdf

R47v0_Menu_BITS_page_2.pdf

R47v0_Menu_BITSET_page_1.pdf

R47v0_Menu_BITSHFT_page_1.pdf

R47v0_Menu_BLUE47_page_1.pdf

R47v0_Menu_BLUE47_page_2.pdf

R47v0_Menu_BLUE47_page_3.pdf

R47v0_Menu_CASHFL_page_1.pdf

R47v0_Menu_CAT_ALPHA_page_1.pdf

R47v0_Menu_CAT_EQN_page_1.pdf

R47v0_Menu_Uniform_page_1.pdf

R47v0_Menu_VAR_page_1.pdf

R47v0_Menu_VAR_page_2.pdf

R47v0_Menu_VarMNU_page_1.pdf

R47v0_Menu_VARS_page_1.pdf

R47v0_Menu_VECTOR_page_1.pdf

R47v0_Menu_VECTOR_page_2.pdf

R47v0_Menu_VectorConv_page_1.pdf

R47v0_Menu_Volume_page_1.pdf

R47v0_Menu_Volume_page_2.pdf

R47v0_Menu_Weibl_page_1.pdf

R47v0_Menu_XFCNS_page_1.pdf

R47v0_Menu_XFCNS_page_2.pdf

R47v0_Menu_XFCNS_page_3.pdf

R47v0_Menu_X-FN_page_1.pdf

R47v0_Menu_X-FN_page_2.pdf

R47v0_Menu_X-FN_page_3.pdf

R47v0_Menu_YESNO_page_1.pdf

R47v0_Menu_Ymmv_page_1.pdf

Menus

R47 Specific Reference

R47/Reference

File:

Up to the higher level directory

R47v0_Abbreviations.pdf

R47v0_AKA_menus.pdf

R47v0_Annunciators.pdf

R47v0_Assignment.pdf

R47v0_Base_digits.pdf

R47v0_BestF_param.pdf

R47v0_CAT-FCNS.pdf

R47v0_CAT-MENUS.pdf

R47v0_Configuration.pdf

R47v0_Constants.pdf

R47v0_CONVG_param.pdf

R47v0_Datatypes.pdf

R47v0_Date_formats.pdf

R47v0_Delta_keyboard.pdf

R47v0_Distr_param.pdf

R47v0_DMCP_menu.pdf

R47v0_Equation_editor.pdf

R47v0_Errors_and_messages.pdf

R47v0_Flags_named.pdf

R47v0_Flags_p1.pdf

R47v0_Flags_p2.pdf

R47v0_Font_browser.pdf

R47v0_Formulas.pdf

R47v0_Fractions.pdf

R47v0_HP35.pdf

R47v0_Keyboard_layers_schema.pdf

R47v0_Keyboards_side_by_side.pdf

R47v0_KEYMAP.pdf

R47v0_KeyType.pdf

R47v0_Legend_colours.pdf

R47v0_Prime_numbers.pdf

R47v0_Regional.pdf

R47v0_Registers.pdf

R47v0_Shortcuts.pdf

R47v0_SIM_Switches.pdf

R47v0_SIM+AIM_function_table.pdf

R47v0_Skipping_program_steps.pdf

R47v0_Solver_Result_Codes.pdf

R47v0_Status_bar.pdf

R47v0_Stopwatch.pdf

R47v0_Unit_conversions.pdf

R47v0_UNIT_prefix.pdf

R47v0_Variable_naming.pdf

R47v0_Vector_display.pdf

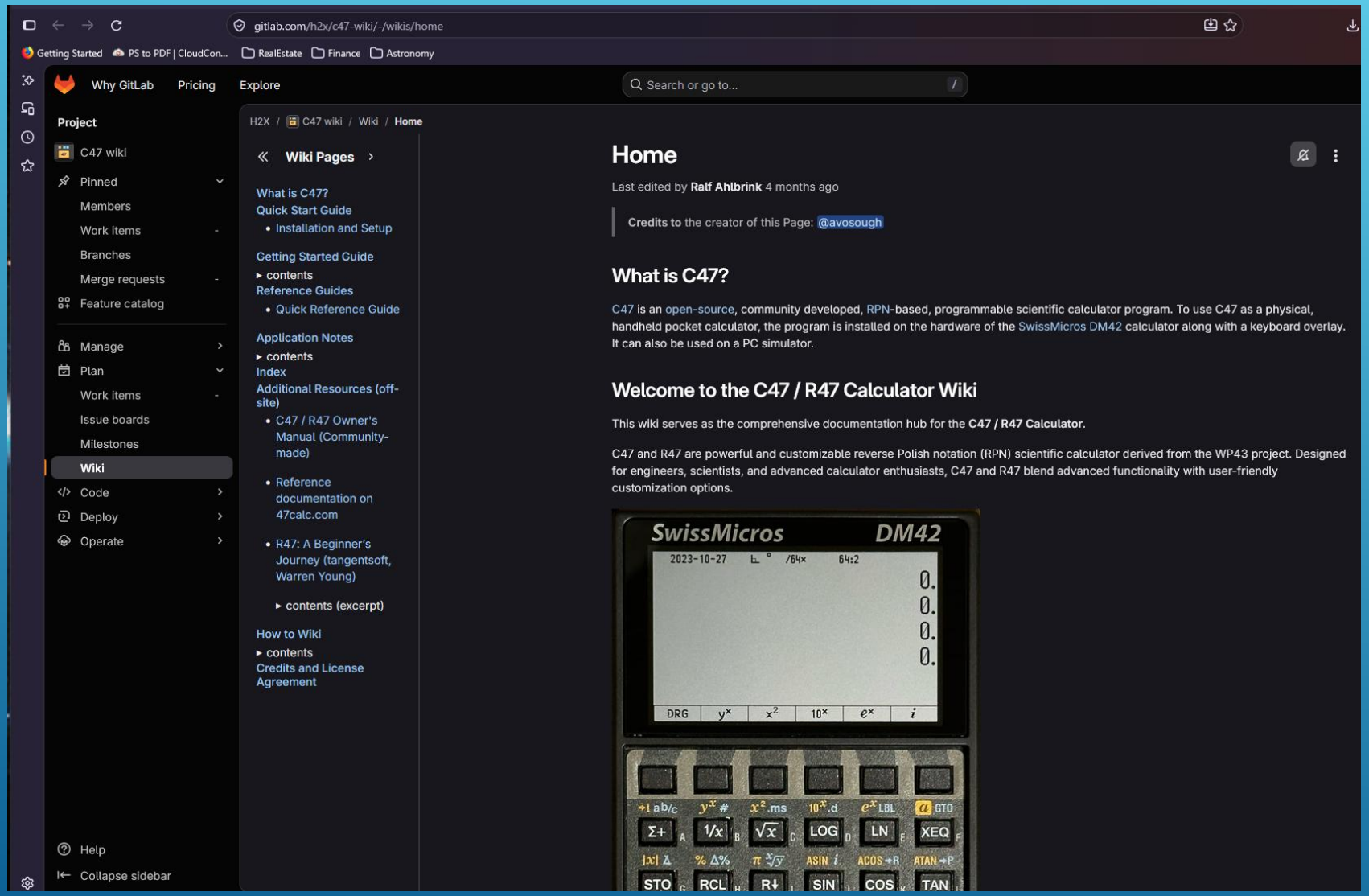
Other Reference Documents

R47 Simulator Reference



Simulator Keyboard Layouts

C47/R47 Community Wiki/GitLab



The GitLab Repository for the C47/R47 Project

C47/R47 Unofficial Wiki

c47.miraheze.org/wiki/Main_Page

Getting Started | PS to PDF | CloudCon... | RealEstate | Finance | Astronomy

C47 Unofficial Documentation

Search C47 Unofficial Documentation | Search

Create account | Log in

Main Page

Contents hide

Beginning

- The C47 Project
- Sources Of Information About The C47 Calculator
- About This Wiki
- INDEX page of this wiki
- START page
- The Hausmeisterbude (restricted area)

Main Page | Discussion | Read | Edit | View history | Tools

Appearance hide

Text: Small / Standard / Large

Width: Standard / Wide

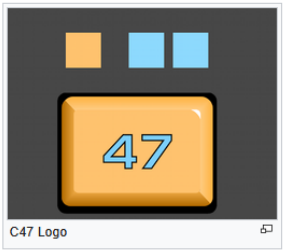
Color (beta): Automatic / Light / Dark

The C47 Project [edit]

Preamble:

This whole Wiki dates back some time when there was no Swissmicros DM42n, but only the DM42. And there was no R47, either! So, everything you can find in here, is related to the DM42 as the hardware basis. Moreover, the C47 software is under such a rapid development that things in this Wiki might apply or not.

Basically, the C47 project is an attempt to create a more powerful calculator than the Swissmicros DM42 running Thomas Okken's famous Free42. Free42 is engineered to be a full-fledged clone of HP's 42S and thus has quite logically also its limitations. The C47 has been created as an advancement over the HP 42S and wants to offer more features, more possibilities and an enhanced user interface.



C47 Logo

Hence, the C47 is a software whose main purpose is to make full use of the DM42's hardware basis.[1] On the other hand, it is restricted to this hardware, esp. regarding the keyboard. Stickers on keys are widely considered ugly and impractical, and that's my opinion, too. So the development team chose to restrict hardware changes to a change of the bezel. Such a bezel can be obtained via the C47 homepage:

C47 Homepage, named 47calc.com, not C47calc.com!

Here you find the link to order a bezel which is of fine quality. It's made of polycarbonate, looks very good, feels quite nice and gives your new calculator an appealing look!

[1]) As some of you might know already, there are two different hardware versions for the portrait-format calculators like the DM41X, the DM32 etc. About the end of 2024 Swissmicros has published a second version of the DM42, based on the newer and more capable hardware version, it is called DM42n. To make things a bit more complicated: The older hardware works with the firmware version DMCP (the underlying OS if you will), whereas the newer hardware (which has been introduced with the DM32) needs the so-called DMCP5. (To make things even one step more complicated: The internal version number of the old firmware is 3, and the new one is 3.5 or greater. But mostly you can ignore this and refer to DMCP and DMCP5.) **And now the good news: The C47 software comes in two flavors, and thus can be used on the old DM42's as well as on the new DM42n's! (But you must use the correct version which matches your hardware basis!)**

Sources Of Information About The C47 Calculator [edit]

1. C47 Homepage (1)
2. C47 Community Wiki (1)
3. Calculator Culture (3)
4. Recommended: Calculator Culture's view on the C47! (3)
5. Swissmicros Forum on 3rd party software for their calculators (2)

Historical Wiki

Enter RPN – *R47: A Beginner's Journey*

tangentsoft.com/rpn/wiki?name=R47:+A+Beginner's+Journey#index

Getting Started | PS to PDF | CloudCon... | RealEstate | Finance | Astronomy

Enter RPN

Home | Timeline | Files | Forum | Wiki | Login

R47: A Beginner's Journey

Help | History

Motivation



When I began this project, there was no first-party user manual for the C47/R47, nor even a third-party one; the closest it had was its reference manual. I therefore set out to document my own journey toward enlightenment with the R47.

First-Party User Documentation

As of Spring 2026, we now have the LLM-assisted compilation led by `@raprism`, which is slowly becoming increasingly human-written through community contributions, including some from yours truly. It started as another third-party effort, following my own, but it was pulled into the calculator's main tree later that summer, with PDFs published here. It subsumes much of the previously available community wiki material.

The Application Notes are also worth your time. Their deep-dive nature means they are likely to give details that are neither in the community manual nor here. Their primary downside is that they do not collectively cover the entire calculator, leaving large gaps where a topic you might wish to know about is entirely missed.

Reference Tables

What the C47/R47 project calls a "reference manual" is actually a huge list of loosely connected PDFs generated from a single master Excel spreadsheet, which is not itself published. The entry points least likely to leave you lost within that maze are:

- Index Cards (HTML) — when you know what to call a thing but need a pointer to more information about it
  - Commands & Functions
  - Flags & Settings
  - Menus
- Reference Tables (PDF) — when you are searching by concept, before you know the exact name
  - Full Names — the index cards, merged, with more detail included besides
  - Function Catalog — verbose version of the *Commands & Functions* index
  - Flags and Settings — verbose version of the *Settings* index
  - Menus — verbose version of the *Menus* index

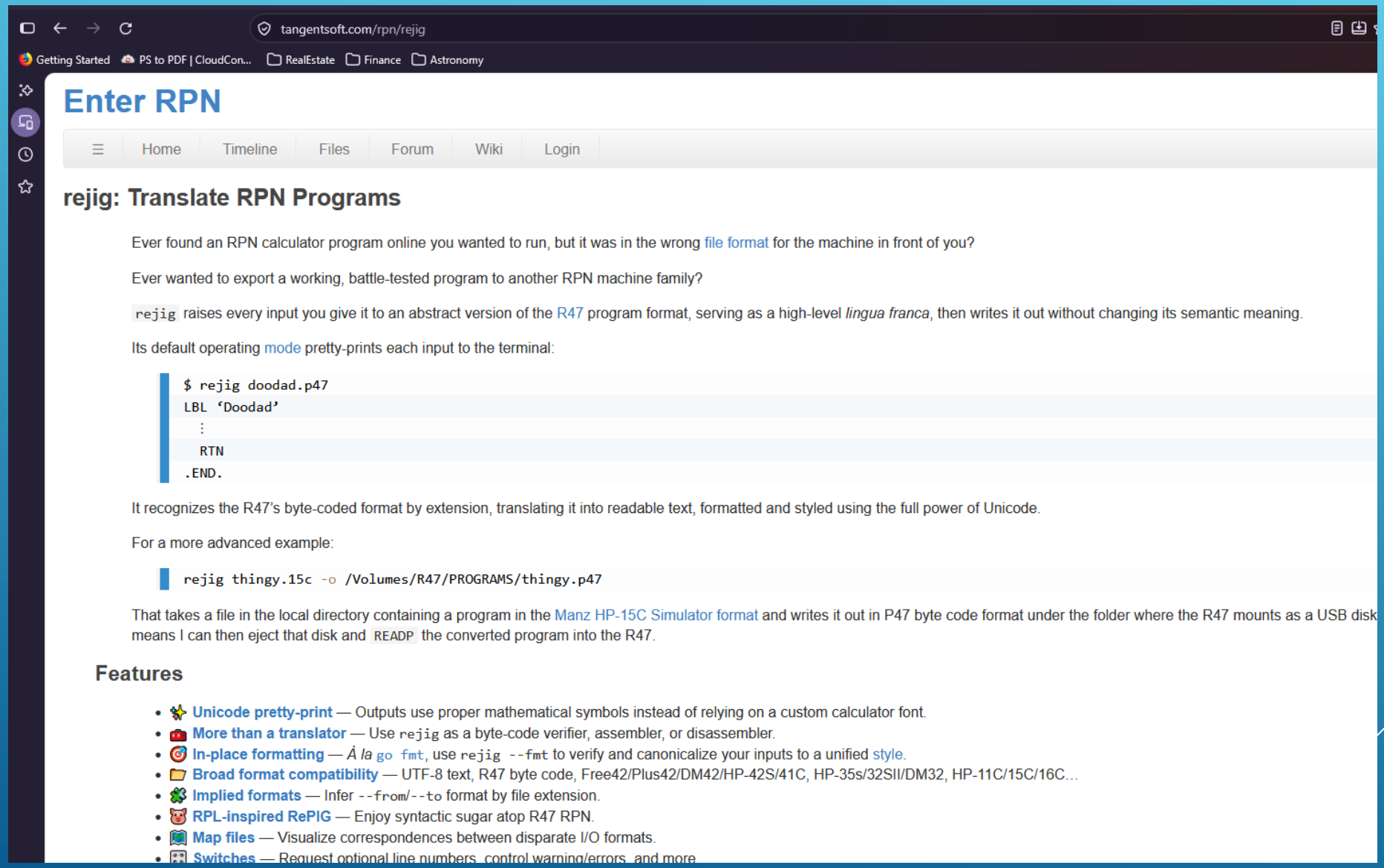
An Outsider's Humble Offerings

That brings me to my own writings on this topic, largely meant to fill gaps left by the above:

1. A Beginner's Journey — you are here!
2. Unboxing — a vignette
3. R47 vs DM32 — a comparative review
4. Basic User Functionality

Other Reference Materials

Enter RPN – *R47: A Beginner's Journey*



Translate Calculator Programs to R47 Executables

C47/R47 Community

SwissMicros **SwissMicros Calculator Forum**
forum.swissmicros.com

Search...

Quick links FAQ Register Login

Board index ‹ R47/C47 ‹ Usage tips, tricks and bug reports

C47 / R47 manual - issues and suggestions

Post Reply | Search this topic... | 46 posts 1 2 3 4 5

C47 / R47 manual - issues and suggestions

by **raprism** » Thu Sep 10, 2026 3:37 pm

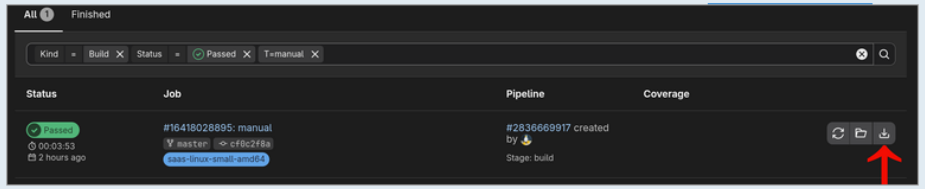
raprism
Posts: 669
Joined: Sun Dec 17, 2023 9:53 pm
Location: Berlin
Contact:

Latest (pre-)release is: Documents 2026-10-13-7e9b8fe7, tag: docs-00.109.05.00b2.rel1

See docs/manual/README.md for up-to-date information.

CI (continuous integration) build results are not available to everyone. Project members can reach them at rpncalculators/c43 manual builds.

As a short-term *workaround*, I will occasionally update my fork of the C47 repository so that everyone can reach raprism/c47 manual builds.



raprism--c47---manual-job(s).png (63.58 KiB) Viewed 714 times

Click the download symbol to get an artifacts.zip containing docs/manual/C47_R47_calc.pdf.

The best way to contribute is to fork the C47 repository, change the manual sources and open a merge request. That is not for everyone, so the next best is an issue report labelled 'documentation'. If you have no GitLab account, or would rather not use one, post your bug or feature report for the manual here. I will check this thread from time to time and pick things up.

---

make-pdf.sh.zip
(1.94 KiB) Downloaded 6 times

make-pdf.sh builds the manual on a LaTeX kernel older than 2025/06/01, where make pdf stops. It turns the command index off for that one build, which is the one defect known today, and puts the one file it edits back afterwards, so your checkout is as you left it. What an older kernel gets wrong can grow as the manual does.

It is POSIX sh, sed and make, with no mktemp, no bashism and no GNU extension. Tested on Linux, and it should therefore work on macOS and on MSYS2, neither of which I have tried.

PS: Head line added to show latest (pre-)release.
PPS: Added make-pdf.sh with short description

Last edited by raprism on Wed Oct 14, 2026 10:46 pm, edited 6 times in total.

C47 & R47 Calculator

Owner's Manual

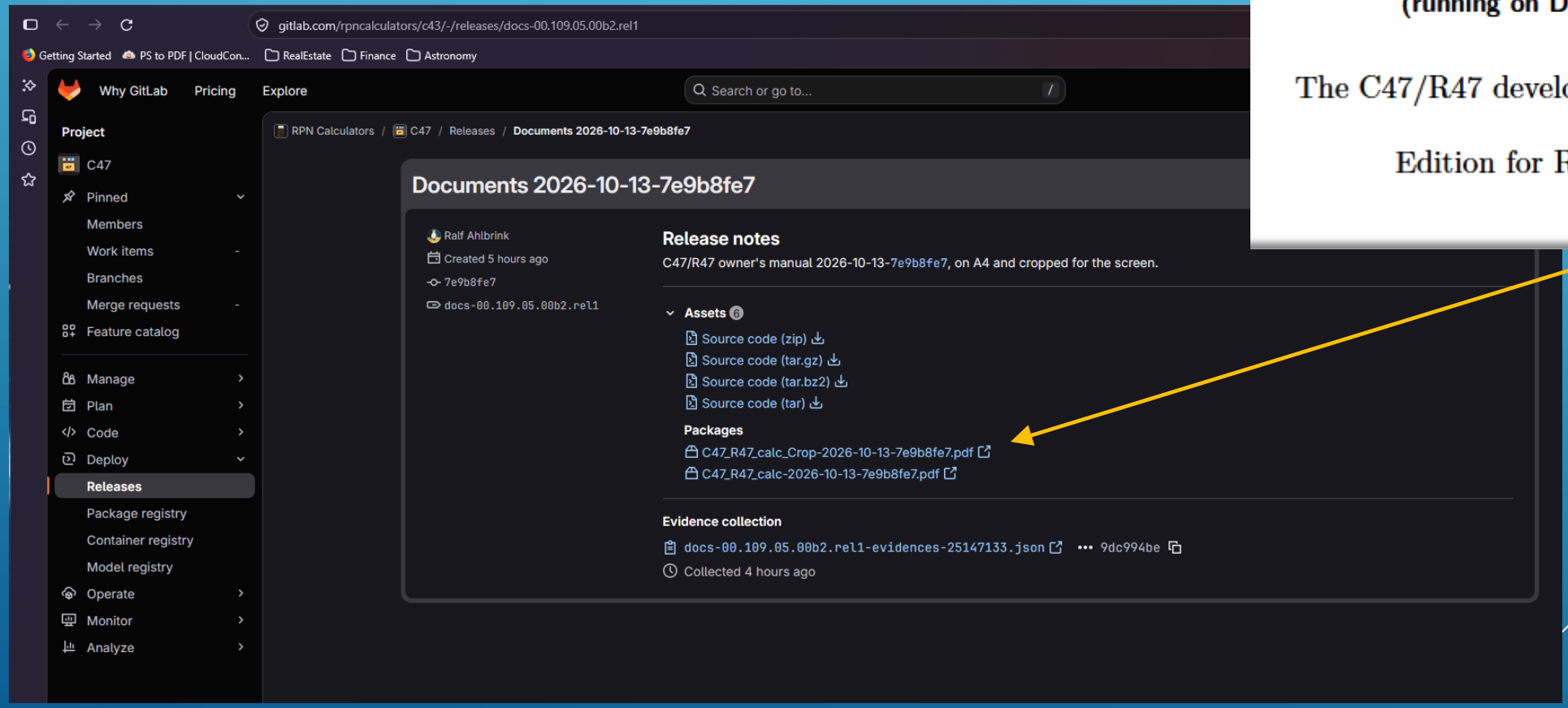
For the SwissMicros R47 calculator and the C47 software (running on DM42 / DM42n hardware)

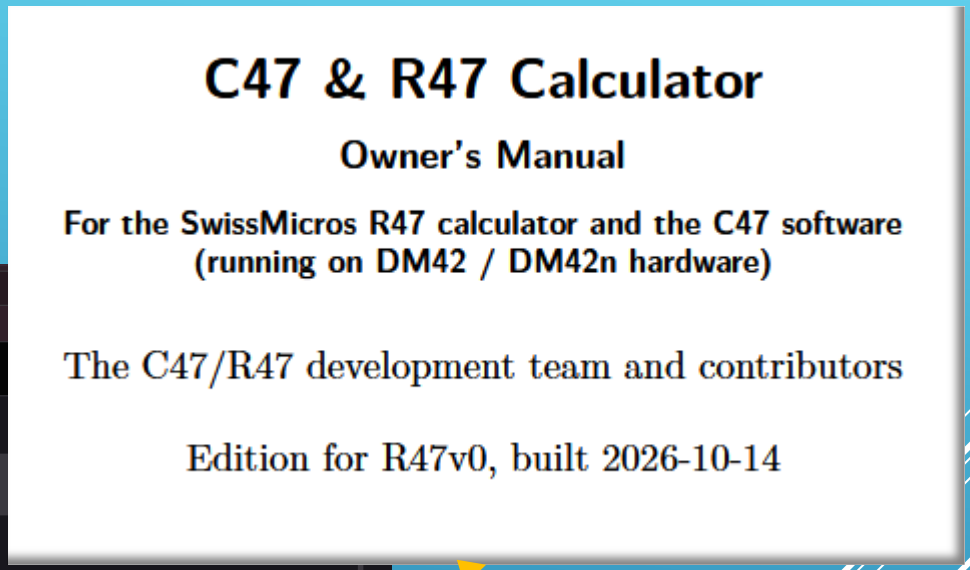
The C47/R47 development team and contributors

Edition for R47v0, built 2026-10-14

GitLab location for New Owner's Manual for C47/R47

C47/R47 Community





C47 & R47 Calculator

Owner's Manual

For the SwissMicros R47 calculator and the C47 software (running on DM42 / DM42n hardware)

The C47/R47 development team and contributors

Edition for R47v0, built 2026-10-14

Pre-generated .pdf

C47/R47 Owner's Manual

Contents

I Tutorial 15

1 Introduction 16
1.1 What is the R47? 16
1.1.1 Capabilities at a glance 16
1.1.2 What this manual covers 17
1.1.3 What this manual does *not* cover 17
1.2 R47 versus C47 versus DM42n 18
1.3 The lineage: where R47 came from 20
1.4 Conventions used in this manual 20
1.5 What you should know about RPN before reading further 22
1.6 Getting help while you read 22

2 Quick Start 23
2.1 Switching on, and getting back 23
2.2 Your first calculation 24
2.2.1 A second calculation: chained operations 24
2.2.2 A third calculation: percent 25
2.3 Save your state *first* 26
2.3.1 Naming a state file 26
2.4 Reading the display 26
2.4.1 Temporary information area 27
2.5 Be lenient: get out of trouble fast 27
2.5.1 Undo 28
2.6 Softkey menus, shift states, and pages 28
2.6.1 Multi-row menus and arrow navigation 29
2.6.2 Sub-menus 29
2.7 Settings worth making first 29
2.8 The catalog 30
2.8.1 Two ways to call a function 31
2.9 Long-press, double-press, and the longpress timer 31
2.10 The DMCP system menu and the USB disk 31
2.11 A short tour 32
2.12 Program-entry mode, in passing 33
2.13 Making it your own 33
2.14 What to do next 33

3 RPN and the Stack 34
3.1 A brief introduction to RPN 34
3.1.1 Three ways to write an expression 34
3.1.2 A formula with real nesting 34
3.1.3 The same formula in RPN 35
3.1.4 The stack model 36
3.1.5 Why RPN, and why algebraic survives 36
3.1.6 A few more conventions 36
3.1.7 Where to read more 36
3.2 Hello, World! 37
3.3 Keying a polynomial: Horner's scheme 37
3.4 Stack size: 4 or 8 38
3.4.1 Why might you want 4 instead of 8? 39
3.5 eRPN versus classical RPN 40
3.5.1 When does the difference bite? 40
3.5.2 Automatic stack-lift 41
3.6 Roll, swap, and other rearrangements 41
3.6.1 LASTx as a safety net 42
3.7 Injecting and inspecting stack values 42
3.7.1 Save and restore the entire stack as a unit 42
3.8 The hidden 9th register: *L* for last *X* 43
3.9 Stack manipulation in programs 43
3.9.1 Stack queries 43
3.10 Stack examples 44
3.10.1 Heron's formula for triangle area 44
3.10.2 Mean of three numbers 44
3.10.3 Compound interest 44
3.11 Multistack helpers 45

4 Number Entry and Display 46
4.1 Numeric input mode (NIM) 46
4.1.1 Open vs. closed numbers 46
4.2 Default input type 47
4.2.1 Why exact integers matter 47
4.3 Display formats 47
4.3.1 Choosing a display mode 48
4.3.2 Rounding the display: `DRM` 48
4.3.3 Showing all the digits: `SHOW` 49
4.3.4 The `SCIOVR` / `ENGOVR` settings 49
4.4 Group separators and the regional presets 49
4.4.1 Separator settings 50
4.5 Angles, dates, and times: a first look 50
4.5.1 Times: hours/minutes/seconds vs. decimal hours 51
4.6 The `DSP` setting and the digit count 52
4.7 The exponent range: `RNG` 52
4.7.1 Hiding very small numbers: `HIDE` 52
4.8 Examples of display formats in action 52

5 Real, Complex, Integer, and Other Data Types 54
5.1 The full type table 54
5.2 Long integers 55
5.2.1 When integer arithmetic falls back to real 55
5.2.2 Working with very large integers 55
5.3 Real numbers 55
5.3.1 Special results: $\pm\infty$ and NaN 56
5.4 Complex numbers 56
5.4.1 The complex multiplier 57
5.4.2 Imaginary unit $\boldsymbol{i}$ and $\boldsymbol{j}$ 57
5.4.3 Polar versus rectangular display 57
5.5 Short integers and `#BASE` mode 57
5.6 Matrices, vectors, and strings 58
5.7 A vector is also a polynomial 58
5.8 Configuration and the 'virtual triplet' 59
5.8.1 Multiple configurations 59
5.9 Type-aware dispatch 59
5.10 Type queries inside programs 60

6 Tags, constants, and unit conversions 61
6.1 Tagging 61
6.1.1 Angle tags on reals 61
6.1.2 Polar/rectangular tags on complex numbers and 2D/3D vectors 62
6.1.3 Date and time tags 63
6.1.4 Tag interactions 63
6.1.5 Tags inside programs 64
6.1.6 Tags and storage 64
6.1.7 Untagging shortcuts 64
6.1.8 Why tags are worth the cognitive overhead 64
6.2 Constants 64
6.2.1 Selected physical constants 65
6.3 Unit conversions 65
6.3.1 The CONV menu 65
6.3.2 Direction and the `HPCONV` flag 66
6.3.3 Sexagesimal and time conversions 66
6.3.4 Custom conversion pairs 66

7 Working with Complex Numbers: a Worked Tour 68
7.1 The example 68
7.2 Method 1: via the mode settings in the `[TRG]` and `[CPX]` menus 68
7.3 Method 2: via the `[TRG]` menu 69
7.4 Method 3: direct entry with $\boldsymbol{i}$ and $\boldsymbol{i}_\odot$ 69
7.5 Method 4: a custom `MyMenu` 69
7.6 Common complex operations 70
7.6.1 The `CROSS` and `DOT` idioms 70
7.7 Complex roots: more than one 70
7.8 Common pitfalls 71
7.9 Seeing it all on the stack 71
7.10 Worked engineering example: cable sizing under fault 71

C 2026

C47/R47 Owner's Manual

8 Fractions, in Detail 73
8.1 Reading a fraction display . . . 73
8.2 Setting the maximum denominator . . . 74
8.3 The DENFIX / DENANY interactions . . . 74
8.4 Entering fractions . . . 74
8.5 Cycling display modes . . . 75
8.6 One-touch toggle via an unused key . . . 75
8.7 Example: irrational fractions . . . 76
8.8 Approximation tolerance FDIGS$_n$ . . . 76
8.9 Fractions in programs . . . 76
8.10 Approximations of π, e, φ . . . 77

II Reference 78

9 Memory: Registers, Variables, Flags 79
9.1 Numbered, lettered, and local registers . . . 79
9.2 Default contents and the demo registers . . . 79
9.3 Local registers and flags . . . 80
9.4 Storage variants of STO and RCL . . . 80
9.4.1 The full table of STO variants . . . 81
9.5 Flags 00–99 and the lettered flags . . . 81
9.5.1 Flag categories . . . 81
9.6 Variable naming rules . . . 82
9.6.1 Names the machine has already spoken for . . . 82
9.6.2 Names inside an equation . . . 82
9.6.3 Naming conventions in user programs . . . 83
9.7 Memory housekeeping . . . 83
9.8 The [CLR] menu . . . 84
9.9 The [DELETE] menu . . . 84

10 Mathematical Functions 85
10.1 Standard functions . . . 85
10.1.1 Trigonometric and hyperbolic . . . 85
10.1.2 Logarithms and exponentials . . . 85
10.1.3 Rounding, integer parts, and floor/ceiling . . . 86
10.2 Three-phase power engineering: the [ELEC] menu . . . 86
10.2.1 Triple registers . . . 86
10.2.2 Triple-storage and stack operations . . . 86
10.2.3 Ohm's-law operators . . . 87
10.2.4 Star–delta and balanced-set transforms . . . 87
10.2.5 Sequence components . . . 87
10.2.6 Symmetrical-component matrices . . . 88
10.2.7 Example: average current of an unbalanced motor load . . . 88
10.2.8 Example: symmetrical components . . . 88
10.2.9 ELEC menu shortcuts . . . 89
10.3 Sexagesimal: hours, minutes, seconds and degrees . . . 89
10.3.1 The HP-compatible aliases: HR and HMS . . . 89
10.3.2 The .ms polymorphic key . . . 89
10.3.3 The .d and .ms^{-1} extractors . . . 90
10.3.4 The [CLK] menu: explicit named conversions . . . 90
10.3.5 Example: HP-15C arctic-sunrise problem . . . 90
10.3.6 Pre-2026 C47 conversion habits (legacy) . . . 91
10.4 Special functions . . . 91
10.4.1 Hyperbolic and trigonometric on complex arguments . . . 93
10.4.2 Elliptic parameter convention: m vs k . . . 93
10.5 Extended-precision XFCNS (the 1000-digit subsystem) . . . 94
10.5.1 The composite $(X \cdot Y + Z)$ data type . . . 94
10.5.2 XFN function list . . . 95
10.5.3 Example: π to 1000 digits, four ways . . . 95
10.5.4 Why this matters: large-integer trigonometry . . . 96
10.5.5 Storage and stack handling . . . 96
10.6 Constants . . . 96
10.7 Solver and integrator . . . 97
10.7.1 The MVAR declaration . . . 97
10.7.2 Solver tolerance and convergence . . . 98
10.8 Example: the Lambert W function . . . 98
10.9 Bessel functions . . . 98
10.10 Prime numbers and factorisation . . . 99

11 Statistics, Regression, and Distributions 100
11.1 Entering data . . . 100
11.2 Statistics, regression, best fit . . . 100
11.2.1 The summary statistics functions . . . 101
11.3 Regression and best-fit selection . . . 101
11.3.1 Example: linear regression . . . 102
11.4 Distributions . . . 102
11.4.1 Example: z-score from Φ . . . 103
11.4.2 Example: binomial tail . . . 103
11.4.3 Example: Poisson event count . . . 103
11.4.4 Example: chi-squared critical value . . . 104
11.5 Histogram and plotting . . . 104
11.6 Random numbers . . . 104
11.7 Statistics housekeeping . . . 104

12 Time Value of Money 106
12.1 The TVM menu . . . 108
12.1.1 Clearing: CLRTVM vs. RSTTVM . . . 108
12.1.2 End vs. begin payments . . . 108
12.2 Example: a car loan . . . 108
12.3 Effective rate vs. nominal rate . . . 109
12.4 Amortisation . . . 109
12.5 Mean rate of return . . . 111
12.6 Multiple-solution problems and solver bias . . . 111
12.7 Bi-weekly savings (PPER/a $\neq$ 12) . . . 111
12.8 Cash flow problems . . . 112
12.9 TVM clearing . . . 112
12.10 Curious about the internals? . . . 112

13 Base Modes and Bit Manipulation 113
13.1 Entering and leaving #BASE mode . . . 113
13.1.1 The TOPHEX flag . . . 113
13.2 Word size . . . 113
13.2.1 Choosing a sign mode . . . 113
13.2.2 HP-style stack-wide base mode . . . 114
13.3 Bit operations . . . 114
13.4 Other bases . . . 114
13.5 The two display lines . . . 115
13.6 Common bit-manipulation idioms . . . 115
13.7 Leaving the world of bits cleanly . . . 115

14 Matrices, Vectors, and Polynomials 116
14.1 Creating a matrix . . . 116
14.2 The matrix editor . . . 116
14.3 Viewing a matrix full-width: SHOW . . . 117
14.4 Matrix arithmetic . . . 117
14.5 Vectors . . . 118
14.6 Polynomials: the [POLY] menu . . . 118
14.6.1 Building the coefficient vector . . . 119
14.6.2 The same program without the formula . . . 120
14.6.3 From vector to equation: V→EQ . . . 120
14.7 Example: solve a 3×3 system . . . 121
14.8 Conditioning: how far an answer can be trusted . . . 121
14.9 Element-wise access . . . 123
14.9.1 Column extremes . . . 123
14.10 Tagged 3D vectors . . . 123
14.11 Getting a matrix off the calculator . . . 124
14.12 Vector geometry: parallelogram area . . . 124
14.13 Coordinate transformations . . . 125
14.14 Simultaneous equations and SIM_EQ . . . 125
14.15 Using the solver with matrices . . . 126
14.16 Other useful matrix verbs . . . 126
14.17 See also . . . 126

15 The Alpha Subsystem 127
15.1 Entering and leaving AIM . . . 127
15.2 Case lock and shift . . . 127
15.3 The character set . . . 127
15.4 Common alpha-mode menus . . . 128
15.5 The alpha entry buffer is not a register . . . 128

2026

C47/R47 Owner's Manual

15.6 The Font Browser and [Mya] . . . 128
15.7 Alpha string functions . . . 129
15.8 The alpha register of the [42] commands . . . 129
15.9 Example: prompt for input in a program . . . 129
15.10 Editing alpha strings on the stack . . . 130
15.11 Alpha as an annunciator . . . 130
15.12 The C47 font: characters, encoding, and 'not quite Unicode' . . . 131

16 Programming . . . **133**
16.1 Program-entry mode (PEM) . . . 133
16.2 Program input and output . . . 133
16.3 Labels and flow control . . . 135
16.3.1 Local *named* labels . . . 136
16.3.2 The BACK verb . . . 137
16.3.3 Structured blocks . . . 137
16.4 The do-if-true rule . . . 140
16.4.1 The one exception: IF, WHILE and UNTIL . . . 140
16.4.2 Choosing with IF, ELSE and ENDIF . . . 140
16.4.3 The 62 tests . . . 141
16.5 Loops . . . 142
16.5.1 DO, WHILE, ENDDO: the test at the top . . . 143
16.5.2 REPEAT, UNTIL: the test at the bottom . . . 143
16.5.3 FOR, FOR_y^x, NEXT: counting . . . 144
16.5.4 Which structure to use . . . 145
16.5.5 Programmable sums and products . . . 145
16.6 Specifying function parameters: TAM . . . 149
16.6.1 Indirect register addressing . . . 149
16.7 Subroutine calling and returning . . . 149
16.7.1 VARMNU as a standalone input UI . . . 150
16.7.2 The MVAR convention for solver-friendly programs . . . 151
16.8 Catalog of programmable commands . . . 154
16.9 Saving and exporting programs . . . 154
16.10 Example: a Fibonacci program . . . 154
16.10.1 The same algorithm as a structure . . . 155
16.11 Example: zero-intercept regression 'LRO' . . . 156
16.12 Debugging programs . . . 158
16.13 Example: Σ+ column swap with error handling 'SwapXY' . . . 158
16.14 Further reading: 'Lazy Programming' . . . 159
16.15 The [P.FN] programmer's menu . . . 160
16.16 Compatibility with HP-42S programs . . . 160
16.17 The rejig translator and copy-n-paste workflow . . . 163
16.17.1 Copy-n-paste from the PDF . . . 163
16.17.2 About the PDF indent . . . 163
16.17.3 What the listings look like . . . 164
16.18 Verifiable program listings (prototype) . . . 164

17 User Mode and Key Assignment . . . **166**
17.1 The ASN workflow . . . 166
17.2 What can be assigned . . . 166
17.3 User menus, ribbons, and rivets . . . 167
17.3.1 The [HOME], [MyM], and [P.FN] menus . . . 167
17.3.2 The ten built-in MyMenu ribbons . . . 167
17.3.3 Creating, deleting, and editing menus . . . 168
17.4 Special key assignments in normal mode . . . 169
17.5 The KEYMAP browser . . . 169
17.6 Example: a custom unit-conversion menu . . . 169
17.7 Example: unit-bumping with a user menu . . . 170
17.8 Customising the [Σ+] key (C47) or Blank key (R47) . . . 171
17.9 Resetting USER mode . . . 171

18 Equations and the Equation Editor . . . **173**
18.1 Why equations? . . . 173
18.2 Equation syntax . . . 173
18.2.1 Naming an equation . . . 174
18.3 The function alias table . . . 174
18.4 Operating on equations . . . 174
18.4.1 Solver versus solver-by-program . . . 175
18.4.2 Solver result codes . . . 175
18.4.3 Naming rules for variables . . . 175
18.5 Example: solver via equation . . . 176
18.6 Plotting an equation . . . 176
18.7 Differentiation and integration . . . 177
18.7.1 Example: derivative of $\sin x$ . . . 177
18.8 The ADV menu: solving and integrating RPN programs . . . 177
18.9 Finding complex roots . . . 179
18.10 Every root at once: SLVP . . . 180
18.10.1 What it refuses . . . 181
18.10.2 How far to trust the answer . . . 181
18.11 Exhaustive root finding by polynomial division . . . 182

19 Plotting . . . **183**
19.1 Plotting from data: PLSTAT . . . 183
19.1.1 The ASSESS cycle . . . 183
19.1.2 Histograms . . . 183
19.2 Function graphs . . . 183
19.2.1 Marker style and line drawing . . . 184
19.2.2 Vector plots . . . 184
19.3 Plot interaction . . . 184
19.4 Pixel graphics: AGRAPH, ATEXT and GRMOD . . . 185
19.5 Saving plots . . . 185
19.6 Plotting from EQN: Draw vs. Draw_y^x . . . 186
19.7 Plotting complex-valued equations . . . 187
19.8 Plotting an arbitrary STAT-fed graph . . . 187

19.9 See also . . . 189

20 Communication and Files . . . **190**
20.1 USB mass-storage . . . 190
20.1.1 What the FAT holds . . . 190
20.1.2 Power source and CPU clock . . . 191
20.2 Folder layout on the flash drive . . . 191
20.3 Screenshots: SNAP . . . 191
20.4 Programs: import / export . . . 192
20.5 Data interchange . . . 192
20.6 State files . . . 192
20.7 Serial I/O . . . 193
20.8 IR printer . . . 193
20.8.1 Supported printers . . . 193
20.8.2 The [PRINT] menu . . . 193
20.8.3 Print mode and the four control flags . . . 194
20.8.4 Print commands . . . 194
20.8.5 Print-to-file fallback . . . 196
20.8.6 Output format by data type . . . 196
20.8.7 Simulator printing . . . 197
20.9 Time and date . . . 197

21 Errors and Recovery . . . **198**
21.1 Anatomy of an error . . . 198
21.2 The error catalogue . . . 198
21.3 Suppressing one expected error . . . 200
21.4 Special results: SPCRES . . . 200
21.5 Solver and integrator non-error returns . . . 201
21.6 Recovery checklist . . . 201
21.7 Boot failures and DM42 recovery . . . 201

III Installation, Customisation, and Appendices . . . **204**

22 Installation and Updates . . . **205**
22.1 Where to get the firmware . . . 205
22.2 Updating the DM42 DMCP firmware . . . 205
22.3 Converting a DM42 into a C47 . . . 205
22.3.1 Installing the QSPI binary . . . 206
22.4 Updating C47/R47 to the latest version . . . 206
22.5 Reverting a C47 back to a DM42 . . . 206
22.6 Resetting to factory defaults . . . 206
22.7 Building from source . . . 206
22.7.1 Get the source . . . 207
22.7.2 OS-specific preparation . . . 207
22.7.3 Build targets . . . 208
22.7.4 Code coverage and documentation . . . 208

2026

C47/R47 Owner's Manual

22.7.5 Keeping a clone up to date . . . 208
22.7.6 Variant: SSH access for contributors . . . 209
22.8 A technical overview of the source . . . 209
22.8.1 Lineage and naming . . . 209
22.8.2 Repository layout . . . 209
22.8.3 Core math library: decNumber (IBM) . . . 210
22.8.4 Platform: SwissMicros DMCP . . . 211
22.8.5 Build system . . . 211
22.8.6 Feature packages and `defines.h` . . . 211
22.8.7 Display fonts . . . 212
22.8.8 Licences . . . 212

23 Customisation Recipes **214**
23.1 Setting groups as named programs . . . 216

24 Long-Press and Multi-Press Shortcuts **219**
24.1 Long-press **[EXIT]**: a context-sensitive escape ladder . . . 219
24.2 Long-press **[←]**: clear ladder . . . 220
24.3 Long-press **[ENTER]** in AIM/EIM . . . 220
24.4 White-letter trick in EIM . . . 220
24.5 Long-press behaviour, customised . . . 221
24.6 Reaching shifted functions without the shift key . . . 221
24.7 Double-press shortcuts . . . 222
24.8 Triple-press . . . 222

25 The Stopwatch **223**
25.1 The stopwatch keymap . . . 223
25.2 Anatomy of the display . . . 224
25.3 Use cases . . . 224
25.4 The stopwatch as a benchmarking tool . . . 225
25.5 The `LASTT#` alternative . . . 225
25.6 Resetting and re-entering . . . 225

26 Status Bar Annunciators **226**
26.1 The annunciator zones, left to right . . . 226
26.2 The full annunciator table . . . 227
26.3 Showing the battery voltage . . . 227
26.4 Setting the date and time . . . 228
26.5 Temporary information . . . 228
26.6 Customising what gets shown . . . 228
26.7 Reading the icon area . . . 228
26.8 Putting it together . . . 229

27 Named Flag Reference **230**
27.1 Display and rendering . . . 230
27.2 Mode flags . . . 231
27.3 System / lifecycle flags . . . 232

27.4 Status-bar visibility flags . . . 232
27.5 Plot flags . . . 233
27.6 Programmer-helper flags . . . 233

28 Function Catalog Excerpt **234**
28.1 Stack and entry . . . 234
28.2 Storage and recall . . . 234
28.3 Real arithmetic and rounding . . . 235
28.4 Trigonometric, hyperbolic, and inverse . . . 235
28.5 Special functions . . . 236
28.6 Complex helpers . . . 236
28.7 Matrix and vector . . . 237
28.8 Statistics . . . 237
28.9 Programming primitives . . . 238
28.10 TVM . . . 238
28.11 Bit manipulation (#BASE) . . . 239
28.12 Date and time . . . 239
28.13 Alpha and characters . . . 239
28.14 I/O and state . . . 240

29 Menu Index **241**
29.1 Top-level menus reachable from the keyboard . . . 242
29.2 Sub-menus of `DISP` . . . 242
29.3 Contents of `CPX` . . . 243
29.4 Sub-menus of `STAT` . . . 243
29.5 Sub-menus of `PROB` . . . 243
29.6 Sub-menus of `PLOT` . . . 243
29.7 Sub-menus of `BASE` . . . 243
29.8 Sub-menus of `MATX` / `MATRS` . . . 244
29.9 Sub-menus of `[CONV]` (unit conversions) . . . 244
29.10 Sub-menus of `KEYS` . . . 245
29.11 Sub-menus of `CLR` . . . 245
29.12 Configuration / system flags menu . . . 245
29.13 Sub-menus of `FIN` . . . 245
29.14 Sub-menus of `ADV` (programming tools) . . . 246
29.15 Sub-menus of `ALPHA` and friends . . . 246
29.16 Sub-menus of `[I/O]` . . . 246
29.17 Other catalog and type menus . . . 247

30 Keyboard Layout Sketch **248**
30.1 The 43-key count . . . 248
30.2 The C47 layout, for comparison . . . 248
30.3 Reading the legend conventions . . . 249
30.4 Authoritative source . . . 249
30.5 The DM42 reset pinhole . . . 249
30.6 Connectors . . . 249

31 Application Notes **252**

32 Resources, Credits, and Licence **254**
32.1 Where to read more . . . 254
32.2 Credits . . . 255
32.2.1 Reference-data provenance . . . 256
32.3 Licence . . . 256
32.4 Final note . . . 256

Command and Menu Index **257**

2026

C47/R47 Owner's Manual

1.2 R47 versus C47 versus DM42n

- **R47**: Cosmetically, this is a variant on the DM42n hardware, with different legends printed on the faceplate and keys, plus other details like the brand (SwissMicros) and the model name (R47) on the black border of the display and the machine-specific details printed on the back. Aside from allowing all the functions to be laid out to better match the many new features in the firmware relative to the HP-42S, this also allowed adding a second shift key (blue, g) to ease access to the R47's three-level menus. The R47 is therefore a reskinned DM42n with the community R47 firmware preloaded by SwissMicros, and sold directly by them.
- **C47**: The R47 firmware was initially developed on the older DM42 platform, with the intent that only a custom bezel (keyboard overlay) be applied to the faceplate. The key caps were left as in the HP-42S (including its single shift key) to avoid the need for a small, easily-damaged sticker on each key cap. These constraints forced several compromises that were later lifted in the R47 redesign.

C47 difference

Compared to the two shift keys on the R47, the single shift inherited from the DM42 in the C47 produces what is arguably the single biggest difference between the two machines. In order to allow access to as much of the R47 functionality as possible, the C47's second shift state (g, blue) can only be reached by pressing the DM42's single physical shift key twice in succession; a third tap cycles back to unshifted. This also impacts the way a C47 accesses the shifted functions *above* these two shift keys on the R47, HOME and MyM; they become long presses of f and g, governed by KEY.LP. The full set of shift-layout options (including the 'blank key' variants that free up one shift slot) is described in Chapter 23.

R47 firmware variants continue to be built this way for the benefit of those continuing on with their C47s, plus those who wish to convert a preexisting DM42 or DM42n rather than buy a new R47.

Because the DM42 has half the program flash space of the DM42n, and the complete R47 firmware has expanded past that halfway point, the C47 firmware is presently shipped in 4 separate DM42 builds, each with a different subset of features crammed into that half-sized space, relative to the DM42n-based R47.

(There is a fifth C47 firmware build for the DM42n, with very nearly the same feature set as on the R47, differing in small details to match the cosmetic hardware differences.)

Unless you are comparing a C47 side-by-side with an R47, these differences are not readily apparent at the keyboard level, where the constraints of the 43 available keys, including the shift keys, allow direct access to roughly 100 of the 1500+ operations in the firmware. The primary differences are at the softmenu and programming levels, where the remaining operations are exposed.

- **DM42n**: SwissMicros designed this hardware to host a Free42-based HP-42S revival, but they left the design sufficiently open that alternate third-party firmware could replace it. This same hardware (ARM Cortex-M33, 64 Mbit flash, 400×240 transflective memory LCD,

10.2.6 Symmetrical-component matrices

For workflows that prefer explicit matrices, the [ELEC] menu's softkeys [A] and [A⁻¹] push the standard transformation matrices (positive-sequence convention, middle cell $A_{22} = a$):

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{pmatrix}, \qquad A^{-1}\,V_{012} = V_{ABC}, \qquad V_{012} = \tfrac{1}{3} A\,V_{ABC}.$$

zyx→M / M→zyx, two pages further into the same [ELEC] menu, convert between a ZYX triplet on the stack and a 3×1 matrix.

C47 difference

Prior firmware versions used a historical A-matrix with middle cell $A_{22} = a^2$. Since firmware 00.109.3.01b the matrix is the standard ABC convention with $A_{22} = a$. Programs that reference the old A will see their sequence-rotation direction reversed; swap the references to A and A^{-1} to restore the original behaviour.

10.2.7 Example: average current of an unbalanced motor load

A motor is fed via a delta-connected impedance set $[2, 1, 3]\,\Omega$ (real) from an unbalanced star-connected voltage source $[189, 203, 232]$ V (real). Find the average phase current.

Example

(Listing convention: text after ; is annotation, not keystrokes, see Chapter 16.)

```
2 ENTER 1 ENTER 3 ; delta impedances onto stack ZYX
Δ→Y ; convert to equivalent star: [0.333, 0.5, 1]
3STOZ ; store as the impedance triplet
189 ENTER 203 ENTER 232 ; star voltages onto stack ZYX
3STOV ; store as the voltage triplet
V÷Z ; phase-by-phase Ohm's law: I = V/Z
```

The current triplet is $[567, 406, 232]$ A; average: ++3÷→ 401.78 A.

The whole calculation runs with no matrix arithmetic at all: the triplet machinery keeps phase relationships consistent through the chain.

10.2.8 Example: symmetrical components

A three-phase voltage measures $V_A = 1.0\angle 0°$, $V_B = 1.2\angle -110°$, $V_C = 0.9\angle 120°$. Find the symmetrical components.

Example

Display setup: ALL 3 DEG POLAR. Then with SSIZE8 on:

```
1 ENTER  1.2 ENTER 110 CHS COMPLEX  0.9 ENTER 120 COMPLEX
```


Chapter 23

Customisation Recipes

A grab-bag of small tweaks that most users want sooner or later. All assume you will SAVEAUT when you are done.

Decimal comma instead of dot. f [DISP] RADIX. For full European-style number display in one shot, go to the third page of f [DISP] and press EUROP, the softkey of S.EUR.

Pick a regional preset. Page 3 of the f [DISP] menu exposes six pre-canned locale settings, each adjusting time format, date format, radix separator, and the grouping rules for the integer and fractional parts of a number:

Preset	Time	Date	Radix	Integer grp.	Fraction grp.
S.CHN (China)	24-hour	YYYY-MM-DD	PER.	4, COM,	4, COM,
S.EUR (Europe)	24-hour	DD.MM.YYYY	COM,	3, SPC	3, SPC
S.IND (India)	24-hour	DD.MM.YYYY	PER.	mixed 3+2, COM,	2, COM,
S.JPN (Japan)	24-hour	YYYY-MM-DD	PER.	3, SPC	3, SPC
S.UK (UK)	12-hour	DD.MM.YYYY	PER.	3, SPC	3, SPC
S.USA (USA)	12-hour	MM/DD/YYYY	PER.	3, COM,	NONE
S.DFLT	–	ship default	–	–	–

Separator names (SPC, COM,, PER., NONE) match Table 4.1. Each preset is a macro that sets a coordinated bundle of flags (TDM24, YMD/DMY/MDY, …), the radix separator the [RADIX] menu offers, the grouping widths IPGRP and FPGRP take, plus the associated regional settings such as week-start day and the date of Gregorian calendar adoption. After applying a preset you can still tweak individual flags: the preset is a starting point, not a lock-in.

Imaginary unit j instead of i. f [DISP] CPX*j* (toggles CPX*j*).

Show a continuous battery icon. [SYS.FL.4] g [] [F5] toggle SBbatV. The icon fills proportionally to voltage between 2.05 and 3.05 V.

Open the [HOME] menu when all menus are closed, instead of the bare stack. Set HOMEb; clear MyMb. The two flags are mutually exclusive.

C47/R47 Firmware Release Dates

- 2.07b13 Firmware Released November 2025
- 3.00b0 Firmware Released November 2025
- 3.00b1 Firmware Released December 2025
- 3.00b2 Firmware Released February 2026
- 3.01b1 Firmware Released March 2026
- 3.02a1 Firmware Released March 2026
- 3.02b0 Firmware Released May 2026
- 3.03b0 Firmware Released June 2026
- 4.00b0 Firmware Released August 2026
- 5.00a0 Firmware Released October 2026
- 5.00b2 Firmware Released October 2026

C47/R47 key releases

➢ 3.00b0 Firmware Released November 2025

- Mathematics and Algorithms
- Statistics and Probability
- Input Methods and Data Entry
  - PEM Angle Entry, EDIT functionality
  - Editing of Strings and Numbers
  - Cursor Key Change
- Display and Visualization
- Menu and Ribbon System
  - TRG
  - STK
  - RIBBONS
  - XFNCS
  - KEYS & LAYOUT
- System Architecture and Storage
- Matrix Management
- Date and Time Functions
- Stability and Error Handling
- R47/C47 State Files
- R47 Simulator
- Key Shortcut Added
- EQN CAT and α CAT Introduced
- Naming Rules Relaxed
- C47 on DM42
  - DM42: QSPI
  - DM42: Reduced functionality on DM42
  - DM42: Temporary reduced functionality
- Program Labels
- Program Mode and execution
- General

C47/R47 key releases

➢ 3.00b1 Firmware Released December 2025

- Package separation for c47 on DM42
- Fixed FRACT/IRFRAC interaction bug
- Added programmability of items from IPART menus
- Added CLRMOD to CLR menu since it needs not be a hidden command
- Fixed DRG/USER/ASN simulator key control bug
- Fine tuned state files to include calcModel storage in state files, to enable storage of R47v0 – v3 shift layouts
- Implemented simulator startup of demo modes (bugfix): c47 –e47 works again, etc.
- Allow assignment to USER, UP, DOWN, and BCKSPC keys using the way of typing the name
- Code optimization in softkey drawing and removed side endframe lines of softkeys
- Fixed auto digit adjustment for overflow conditions with PFX 1024
- Fixed initialization within softkey display of in-text display string (behind the scenes fixes)

C47/R47 key releases

➢ 3.00b2 Firmware Released February 2026

- Display and User Interface
- Keyboard and Input Handling
- HOME and CUST Menu Customization
- XFN (Extended Precision) Mathematical Functions
- Modulo 2π Related Improvements
- Complex Number Solver
- ELEC Menu and Electrical Engineering Functions
- Number Theory and Mathematical Functions
- Formula, Calculation, and Data Entry
- TVM improvements
- Unit Conversions
- Statistical Distributions
- Platform Updates
- Internal Item Maintenance
- Menu Organization and Restructuring
- Sample Programs and Documentation

C47/R47 key releases

➢ 3.01b1 Firmware Released March 2026

- Bug Fixes
- Package & Development “stuff”
- New Features
- Documentation

C47/R47 key releases

➢ 3.02a0 Firmware Released March 2026

- Matrix column/row norms
- MATX/VECTOR menu rework
- Added FLAG_3DPHYS and FLAG_3DXYZ
- 2D/3D vector conversions
- Multiple system-wide reductions and rectifications
- Fix long-press during TAM input
  - to not time out incorrectly & clash with printer TRACE output.
- Fix RMODE bug in testSuite.c, and dtToJ-related tests
- Fix compare.c 'almost equal' comparison function
- Fixes from RJVM in sortingOrder.xlsx.
- Fix simulator losing window control when focus is lost
- Fix unnecessary ADM updates
  - during the Spiral graph, i.e. when converting angles.
- Fix sign bug in TVM formulas; add test cases
- Fix Julian day to date conversion and add more unit tests relating to it
- Fix the stored last state name used in sim
- Font updates
  - specifically the rho needed for the vectors
- Font changes:
  - add rho/sub_alpha/sub_delta,
  - move UNDO glyph to correct place,
  - add multiple custom combined glyphs in the new sacrificial block from U+0450,
  - update ASCII conversion for new combined glyphs;
  - tinyFont: move UNDO glyph to correct place.

C47/R47 key releases

➢ 3.02b0 Firmware Released May 2026

- Identity Matrix and Matrix Square Root
- Matrix Norms and Operations
- TVM and AMORT
- PAUSE Rework
- Printing
  - A new PRINT menu is introduced
  - In addition to HP 82240A/B, some Martel printers are also supported
- Elliptic Functions
- Improved Numerical Accuracy
- Improved XFN Functions
- Conversions
  - New Temperature menu under CONV
- SOLVE / PGMSLV
- TYPE? and Type Conversions
- Refresh and UI Improvements
- Display and R/S
- Alpha and Shift Input
- Date and Time
- Simulator
- Fonts and Glyphs
- C47/DM42 Packages
- Demo Programs Updated
- Internal Code Housekeeping

C47/R47 key releases

➢ 3.03b0 Firmware Released June 2026

- **T47 Scripting Interface (Tcl scripting)**
- **HP-42S Compatibility**
- **Unit Conversions (CONV)**
- **ELEC Functions**
- **Printing**
- **Menus and Other Items**
- **Reserved Variables**
- **Display and SHOW**
- **Matrix Operations**
- **Date and Time**
- **Vectors 2D/3D (C47/DM42n & R47 only)**
- **Stability and Bug Fixes**
- **Build and Packaging**
- **Test Programs and Internal Housekeeping**

- **C47 on DM42 alternative packaging:**
  - **Package 1: all DIST, stripped X.FN menu, no ELEC, slow FIN, no VECTOR.**
  - **Package 2: limited DIST, full X.FN menu, no ELEC, slow FIN, no VECTOR.**
  - **Package 3: half DIST, stripped X.FN menu, ELEC, slow FIN, no VECTOR.**
  - **Package 4: minimal, no DIST, stripped X.FN, no ELEC, slow FIN, no VECTOR.**

Limited DIST: Only Normal, StdNormal & LogNormal.
Half DIST: Only Normal, StdNormal, LogNormal, cauchy, chi, expo, logis, t & weibull.
Stripped X.FN: Without Elliptic, Bessel, Orthogonal functions.
No ELEC: No Star/Delta, Impedance, phase sequence, parallel functions.
Slow FIN: All FIN functions are available, slightly lower and slower precision.
No VECTOR: All matrix functions available. No 2D and 3D conversions available.

C47/R47 key releases

➢ 4.00b0 Firmware Released August 2026

- Graphing and Plotting
- Manual plotting
- Derivatives
- Register and Variable Export/Import
- Alpha String Functions
- Programming
- Fonts and Bold Mode
- IR Printing
- T47 Scripting Improvements
- Improved Unit Conversions (CONV)
- Time Values
- Real digit display: SIG change
- Status Bar
- Matrix Operations
- HP-42S Compatibility
- C47 on DM42
- Speed improvement
- Renames and Catalogue Changes
- Stability and General Bug Fixes
- Build and Toolchain
- Internal Housekeeping
- Application Notes
- General
- Items added
  - Mechanical / structural unit conversions (24 new pairs)
- Items Removed

C47/R47 key releases

➢ 5.00a0 Firmware Released October 2026

- **Extracted space in the DM42 version of C47**
  - Package 1: Full functionality less Elliptical
  - Package 2: Full functionality less 50% if DIST (no F BINOM HYPER POISS GEOM GEV PARETO UNIFORM DISUNIFORM)
  - Package 3: Full functionality less IR printing and with slow TVM solves
  - In addition, note that 1000-digits XFN, 159-digit Eigen, 159-digit Cubic solver, Vectors, x->POLY, and large matrix SHOW are not included
- **Numerous Changes**
  - Bug Fixes
  - Enhanced functionality

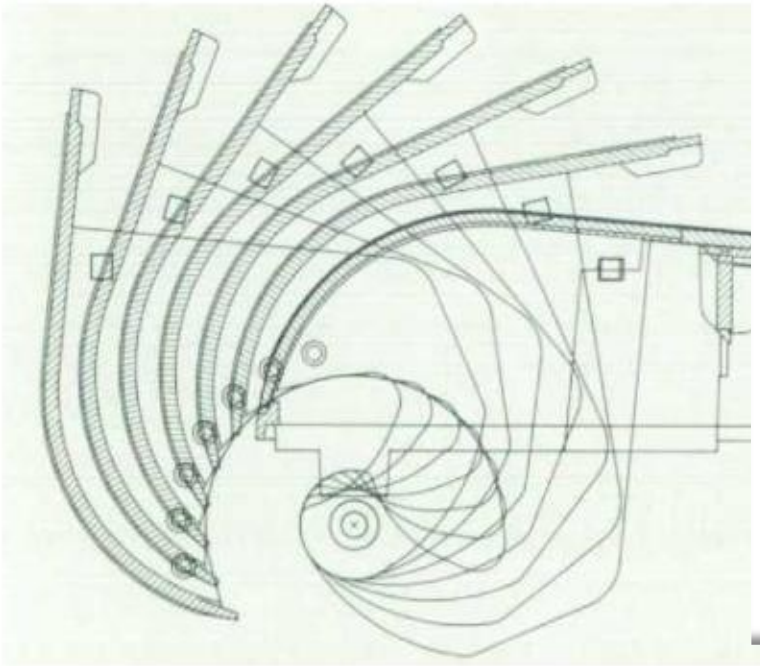
C47/R47 key releases

➢ 5.00b0 Firmware Released October 2026

- Structured Programming
- Display Rounding
- Screen Text and Fonts
- Printing
- Polynomials and Vectors
- Matrix SHOW and the Matrix Editor
- Eigenvalues and Polynomial Roots
- Roots, Powers and Elliptic Functions
- Time and Date
- Finance
- Unit Conversions (CONV)
- Keyboard
- HP-42S Compatibility
- Registers and Memory
- Display and Screen
- Status Bar
- Plotting
- Programming
- Sample Programs
  - New method for managing source
- Speed
- Data Files
- T47 Scripting Improvements
- Build and Toolchain
- C47 on DM42
  - The four original DM42 packages were balanced again
- Documentation
- Application Notes
- Items added
- Items renamed

C47/R47 IR Printers

Application note 021: IR Printing



HP-82240A Simulation of paper door rotation used to determine the location of its mounting pin (HP Journal – October 1987)

Section 2: Menus

The print commands and settings are accessible through the **PRINT** menu.

82240		MAN ●	NORM ○	TRACE ○	STRACE ○
ON ○	OFF ●	#		CHAR	LIST
ADV	STK	x	r	α	PROG

82240		MAN ●	NORM ○	TRACE ○	STRACE ○
ON ○	OFF ●	XFN	ALLr	WIDTH	LCD
ADV	Σ	xy	REGS	TAB	USER

The additional menu allows the selection of the printer being used a printer used. The printer selected is displayed within the menu item.

82240 ●	Martel ○	MODE	DLAY 18

Appendix D: Printers used to develop & test the C47/R47 printing functions



Figure 3: From left to right: HP-82240B, Martel MCP8850B, Martel MCP7850B and Midtronics 182-003B (a custom Martel MCP1830)

Function	Description	Program -mable	Print to file	Status
82240	Set the printer type to HP-82240	No	-	Done
Martel	Set the printer type to Martel	No	-	Done
MAN	Set Manual mode	Yes	-	Done
NORM	Set Normal mode	Yes	-	Done
TRACE	Set Trace mode	Yes	-	Done
STRACE	Set Trace with stack mode	Yes	-	Done
ADV	Advance printer paper	Yes	No	Done
ALLr	Print All Registers	Yes	Yes	Done
CHAR	Print Character	Yes	No	Done
DLAY	Set Delay time between lines	Yes	No	Done
LCD	Print LCD Display	Yes	Yes	Done for Martel printer, pending for HP 82240
LIST	Print a portion of a program listing	No	No	Done
MODE	Set Print Mode	Yes	-	Pending (only mode 0 implemented)
OFF	Print Off (clears PRTACT & PRTEN flags)	Yes	-	Done
ON	Print On (sets PRTACT & PRTEN flags)	Yes	-	Done
PROG	Print Program	No	Yes	Done
REGS	Print Register range	Yes	Yes	Done
STK	Print Stack	Yes	Yes	Done
TAB	Positions the print head to print column n	Yes	No	Done for HP 82240 printer, pending for Martel
USER	Print user (variables and programs)	Yes	No	Done
WIDTH	Print columns required for an alpha string	Yes	-	Pending
α	Print a Register as Alpha	Yes	Yes	Done
r	Print a Register	Yes	Yes	Done
x	Print X-register	Yes	Yes	Done
xy	Print X- and Y-registers as a table	Yes	Yes	Done
XFN	Print X, Y and Z registers as a XFN number	Yes	Yes	Pending
Σ	Print Summation registers	Yes	No	Done
#	Print Byte	Yes	No	Done
	Print annunciator	-	-	Done

C47/R47 Future Feature Support

C47/R47 APPLICATION NOTE 030 2026-10-03

Measurement Uncertainty

GFDL-1.3-or-later

This document describes the measurement uncertainty engine of the R47/C47 calculator. The engine implements the law of propagation of uncertainty of the *Guide to the Expression of Uncertainty in Measurement*, JCGM 100:2008 [1], universally called the GUM.

The idea is simple to state. You already have a way of writing down a measurement model on this calculator: a formula in the equation list, or a keystroke program with **MVAR** declarations. The solver takes those variables and finds a root. The uncertainty engine takes the same variables and answers a different question. Given a standard uncertainty attached to each input, what is the standard uncertainty of the result, and which input is responsible for most of it?

The second half of that question is the one that pays. A combined uncertainty on its own is a number to put on a certificate. A budget that ranks the inputs shows which instrument to send for calibration, which reading to average more times, and which term you can stop worrying about.

If any of this is new to you, start with Section 1: one measurement worked end to end at the calculator, every keystroke shown, in about ten minutes. The remainder of the note is the reasoning behind each of its steps and the full capability of the engine.

Seven commands are involved: **U.CALC**, **U.BUDG**, **U.EXP**, **U.EDIT**, **U.CORR**, **U.Y** and **U.P**. Two named variables contain the data they work on: **U.TAB** and **U.COR**. Section 6 states where the commands are, Section 7 describes the two variables, and Section 10 is the reference for all seven commands.

Numerical results in this document are of three kinds. Results quoted from the GUM are marked as such. Results computed by hand are given with their working, so you can check them. Results that the C47 test suite asserts against the engine are identified by the case that asserts them, in `src/testSuite/tests/uncertainty_cov.txt`. Anything that has to be read off real hardware, a screenshot or a wall-clock measurement, is marked in red.

C47/R47 APPLICATION NOTE 035 2026-08-27

FIN12C Financial Package

GFDL-1.3-or-later

The financial package adds twenty (*1) menus to the calculator and changes two that were already there. **FIN** keeps its name and gets new contents. **TVM** keeps its keys, though three of them move. This note shows each menu as the firmware displays it, names the variable behind every key that has one, and describes the preferences. Three of the eleven preferences are in use. **CRCENT** rounds the depreciation schedule. **CRCENT**, **FFIX** and **FSIG** set the display format inside the financial menus, §5.5. The rest are not used until their modules are written.

C47/R47 APPLICATION NOTE 034 2026-09-28

Units and Unit Algebra

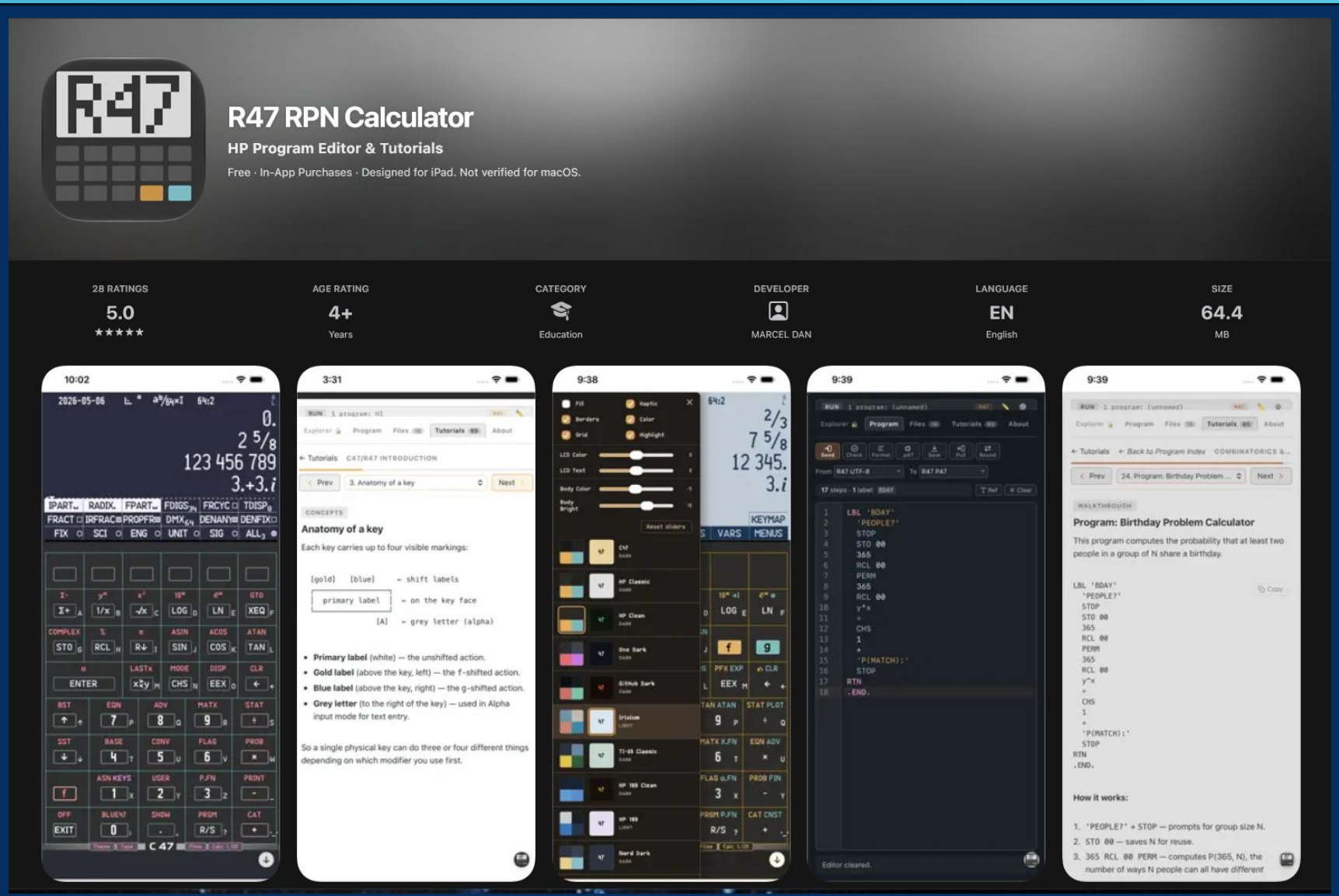
GFDL-1.3-or-later

The C47 converts units with three hundred and more fixed pairs, arranged in menus by dimension. This note adds two things. Any unit converts to any other unit of the same kind, not only to its printed partner. And a number can *take* its unit: `735.75 N` is a value the stack contains, arithmetic takes, and the state file preserves.

Section 1 is the any-to-any conversion, through the unit menus and **CONVERT**; everything from Section 2 to Section 5 is the unit algebra. Both need the larger flash of the DM42n, the R47's DM32 or the simulator: on the original DM42 the unit commands and menus are struck through, the marking for a function a build does not include. The appendix lists every unit the algebra defines.

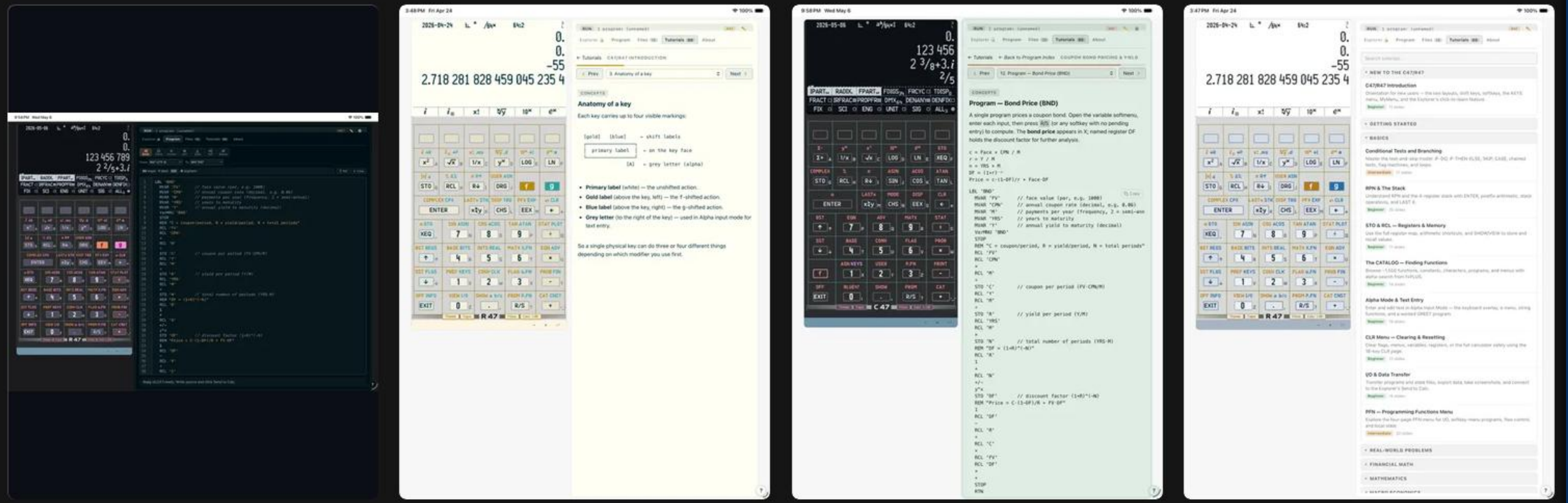
Worked examples are given with the arithmetic behind them, so the reader can verify them on paper. The conversion factors and the algebra rules are what the build verifies: `unitsCheck.py` audits every conversion against the master unit table, and the generated `unitAlgebra.txt` and `unitAssign_cov.txt` exercise the algebra and the assignment machinery on every test build.

C47/R47 Apple iOS Simulators



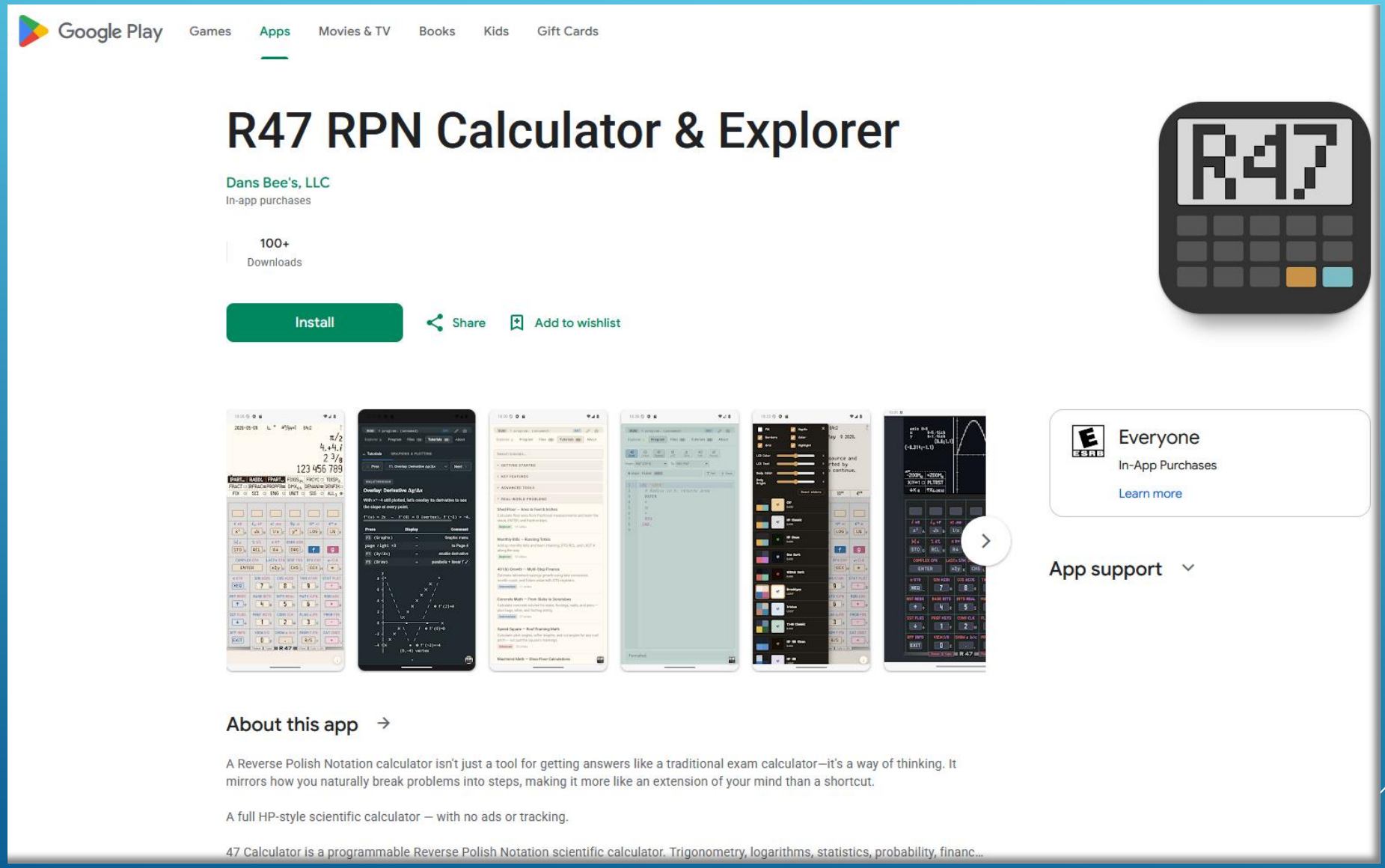
Apple iPhone

C47/R47 Apple iOS Simulators



Apple iPAD

C47/R47 Android Simulator



Android Phone

C47/R47 Web Resources

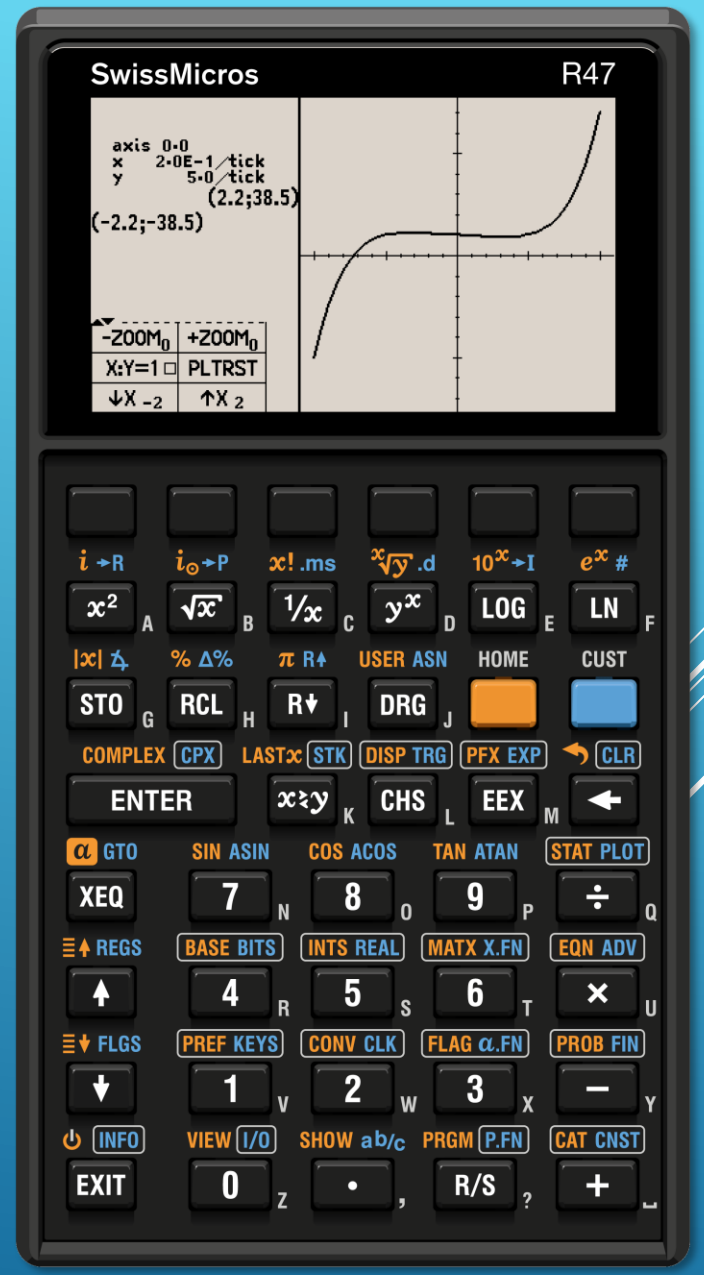
- **Project:** https://47calc.com/index.html
- **Firmware:** https://47calc.com/download.html
- **Documentation:** https://47calc.com/documentation.html
- **Community Wiki:** https://gitlab.com/h2x/c47-wiki/-/wikis/home
- **R47: A Beginner's Journey:** https://tangentsoft.com/rpn/wiki?name=R47:+A+Beginner%27s+Journey#index
- **User Manual:** https://forum.swissmicros.com/viewtopic.php?f=47&t=4949&sid=093e2c773f8d069ee618e17ee7fdd6d3
- **SwissMicros Forum:** https://forum.swissmicros.com/viewforum.php?f=41
- **YouTube:**
  - https://www.youtube.com/@TeamTricky3.14
  - https://www.youtube.com/@N.A._McBee
  - https://www.youtube.com/@hpcalc
- **User Communities:**
- https://hhuc.us/
- https://www.hpcc.org/
- https://www.hpmuseum.org/
- https://www.hpcalc.org/
- **Literature:** https://literature.hpcalc.org/

THANK YOU!

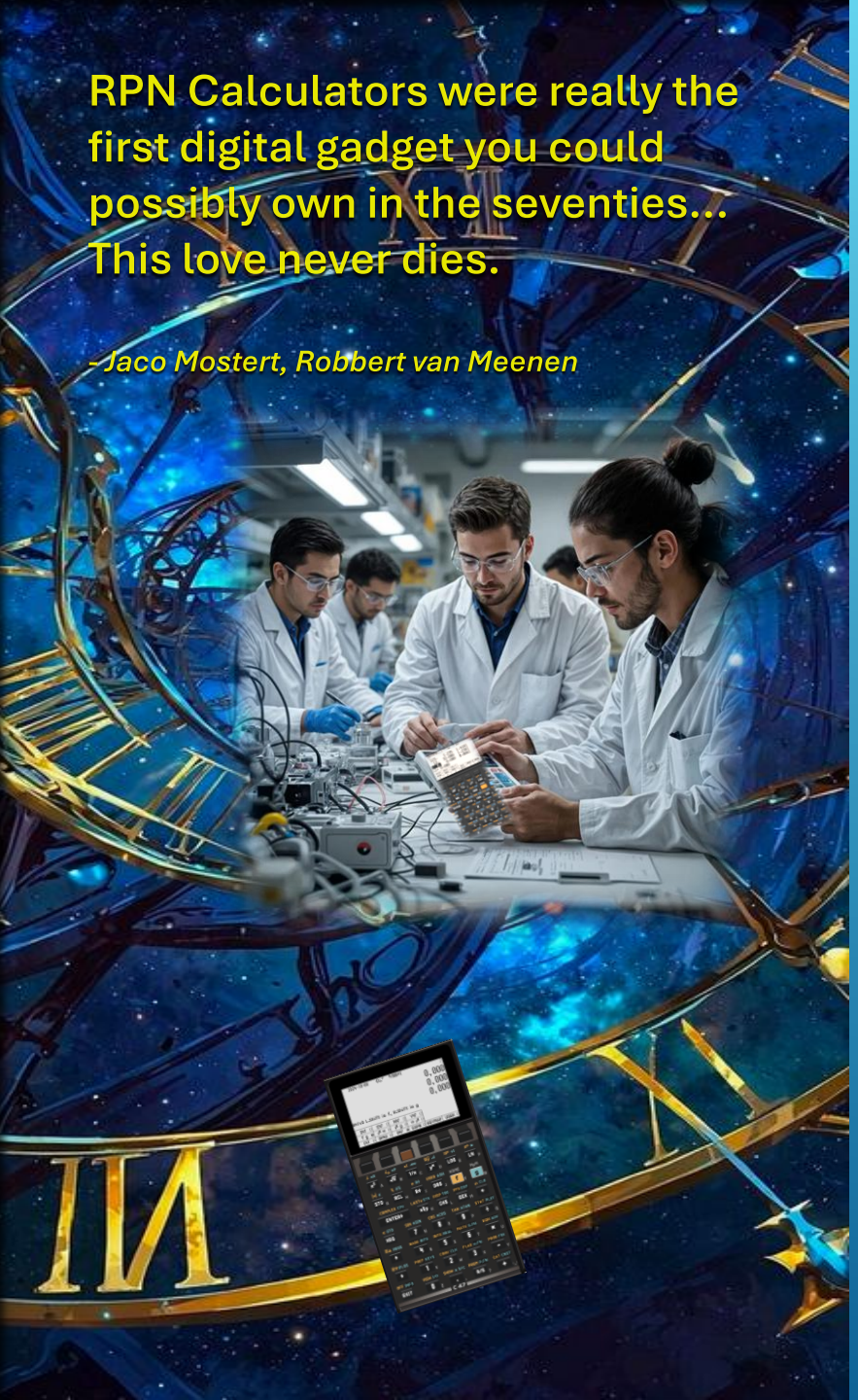
- **Jaco Mostert** – C47 Calculator Project, Prototype R47 loan, App Notes, C47 Overlay
- **Robbert J. van Meenen** – 47 Documentation, Preview of new features
- **Michael Steinmann, SwissMicros** – DM42n, R47 Hardware Support
- **Ralph Ahlbrink** – C47 Wiki
- **Warren Young** – Enter RPN, R47: A Beginner's Journey
- **Marcel Dan** – R47 RPN Calculator on Apple iOS and Android
- **Didier** – R47 printing Application Note 21
- – *and the rest of the C47/R47 project team for their community driven efforts.*
- **HPCC** – Calculator Camaraderie, YouTube Videos, Datafile, RCL 20, RCL 40
- **HHC** – Calculator Camaraderie, YouTube Videos, Extensive HP Library
- **HP** – For all the Handheld Calculators and a lifelong journey down a well traveled road

Questions and Comments?





HHC 2025



Where did R47 come from?



Evolutionary Timeline

1972 HP-35 Released
First RPN Scientific Handheld Calculator

1979 HP-41C Released
First Alphanumeric Programmable Scientific Handheld Computer

1982 HP-12C, HP-15C, HP-16C Released
Voyager Series Calculators

1987 HP-28C Released first Graphing, first CAS

1988 HP-42S Released Most Advanced Pioneer Series

HEWLETT PACKARD

2008 HP-20b Introduced

2009 Free42 Open-Source Calculator Introduced

2010 HP-30b Introduced

SwissMicros

2011 SwissMicros Founded

2011 WP34S Open-Source Project Introduced

2015 WP34C produced on HP20b/HP-30b

2017 SwissMicros DM42 Introduced using Free42

2019 WP43C Open-Source Project Introduced targeting SwissMicros DM42

2021 WP43 Prototype on SM DM42

2022 C43 Open-Source Forked

2023 C47 Developed for SwissMicros DM42 Hardware Platform

2024 Keyboard Layout Defined
R47 Prototype produced by SwissMicros

2025 R47 2nd Prototype Hardware produced by SwissMicros

What can the R47 Do?

How can you say more than 1400 features, for a calculator?

Count of Item		INCLUDE		
MainType	SubType	TRUE	FALSE	Grand Total
Command	internal	5	13	18
	nonpgm	39	4	43
	special	1		1
	strike	9	1	10
	TAM	13		13
	legacy		2	2
	PEM	23		23
	submnu	2		2
	Command	344	13	357
	cyclic ; nonpgm	1		1
	HW	6	1	7
	nonpgm ; HW	1		1
Command Total		444	34	478
Function	legacy	7		7
	proposal		1	1
	linked ; monadic	186		186
	legacy ; monadic	4		4
	triadic	12		12
	monadic	249	1	250
	cyclic ; monadic	3		3
	internal ; strike		1	1
	monadic ; dyadic	2		2
	dyadic	63		63
	niladic	1		1
Function Total		527	3	530
MENU	ASM	8		8
	internal	1	2	3
	MENU	123	4	127
	strike	2		2
	TAM	14		14
	item	17		17
	43		4	4
	47	2		2
	ASM ; legacy		2	2
	TAM ; ASM	3		3
	435		1	1
	TAM ; proposal		1	1
MENU Total		170	14	184
Setting	cyclic	2		2
	internal		1	1
	pgm	119	2	121
	Setting	70	16	86
	SIM	1		1
	strike	3		3
	legacy	2		2
	pgm ; deprecated		1	1
	pgm ; stack	2		2
	cyclic ; stack	1		1
	pgm ; HW	3		3
	HW	2		2
Setting Total		205	20	225
Grand Total		1346	71	1417

- high-resolution display with 4 stack levels, 3 rows of menus, and status bar always shown
- 34 digits of precision; exponents to ±6144; up to 1000 named variables
- full mathematical support for real and complex numbers (rectangular and polar)
- scientific calculator functions like trigonometry, hyperbolics, logs, and antilogs
- support for arbitrary bases from 2 to 16 and bitwise operations
- fraction support
- matrix/vector math with full screen matrix editing capability
- 2-variable statistics with best-fit modelling and basic plotting
- equation writer with support for solving, numeric integration and derivatives, and basic plotting
- time value of money operations
- keystroke programmable with tests, flags, looping, and branching
- date and time math as well as built-in clock
- built-in unit conversions and library of constants
- user-customizable keyboard and softkey-based menus
- I/O to built-in flash memory or via USB for backing up and restoring states, programs, and configuration

R47 Feature Counts

Count of Item		INCLUDE		
MainType	SubType	TRUE	FALSE	Grand Total
Command	internal	5	13	18
	nonpgm	39	4	43
	special	1		1
	strike	9	1	10
	TAM	13		13
	legacy		2	2
	PEM	23		23
	submnu	2		2
	Command	344	13	357
	cyclic ; nonpgm	1		1
	HW	6	1	7
	nonpgm ; HW	1		1
Command Total		444	34	478

MENU	ASM	8		8
	internal	1	2	3
	MENU	123	4	127
	strike	2		2
	TAM	14		14
	item	17		17
	43		4	4
	47	2		2
	ASM ; legacy		2	2
	TAM ; ASM	3		3
	43S		1	1
	TAM ; proposal		1	1
MENU Total		170	14	184

Function	legacy	7		7
	proposal		1	1
	linked ; monadic	186		186
	legacy ; monadic	4		4
	triadic	12		12
	monadic	249	1	250
	cyclic ; monadic	3		3
	internal ; strike		1	1
	monadic ; dyadic	2		2
	dyadic	63		63
	niladic	1		1
Function Total		527	3	530

Setting	cyclic	2		2
	internal		1	1
	pgm	119	2	121
	Setting	70	16	86
	SIM	1		1
	strike	3		3
	legacy	2		2
	pgm ; deprecated		1	1
	pgm ; stack	2		2
	cyclic ; stack	1		1
	pgm ; HW	3		3
	HW	2		2
Setting Total		205	20	225