free42 Custom Menus

Mitch Richling

2021-03-19

Author: Mitch Richling Updated: 2021-05-13 19:55:29

Copyright 2021 Mitch Richling. All rights reserved.

Contents

1	Metadata	1
2	Introduction 2.1 Customizing Custom Menu Table	1 1
3	Custom Menus 3.1 Menu Contents 3.2 Menu Code	2 2 2
4	Create system CUSTOM Menu	3
5	SPREF: Set some of my personal prefrences	4
6	END	4
7	EOF	4

1 Metadata

The home for this HTML file is: https://richmit.github.io/hp42/custom.html

A PDF version of this file may be found here: https://richmit.github.io/hp42/custom.pdf Files related to this document may be found on github: https://github.com/richmit/hp42 Directory contents:

src - The org-mode file that generated this HTML document
 src_42s - Ready to convert source listings for 42s code in this document

docs - This html document

 $\mbox{bin} \qquad \mbox{-} \quad \mbox{Importable RAW program files}$

2 Introduction

The custom menu on the 42s is a wonderful thing, but it is also very limited. It's just too small!! And it is only one level deep!! So instead of simply assigning functions to the menu keys, how about assigning programs to the keys which present a hierarchical menu of functions? Coding up such programs is tedious. So this org-mode document has a Little bit of Emacs lisp that will consume a table, and generate a 42s program implementing a hierarchical menu.

The flow to create your custom CUST program:

- Evaluate the "Generic menu generator" babel block in the hp42s-meta.org file. This will define the function MJR-generate-42-menu-code
- Evaluate the "For CUSTOM-type Menus" babel block in the hp42s-meta.org file. This will define the function MJR-custom-x-gen
- Define your menu in the Menu Contents section.
- Evaluate the babel block in the Menu Code section below.
- Load the program on your calculator

2.1 Customizing Custom Menu Table

The first column (which I labeled Menu) describes the menu keys. Colons are used to define a menu hierarchy. For example "top:mid:thingy" means we have a top level menu "top" containing a menu "mid" containing a function/program "thingy". Another entry like "top:thingy2" would add a function/program key "thingy2" to the "top" menu – yes menus can contain a mix of functions/programs and menus.

The only limitation on menu hierarchy depth or size comes from the use of two digit local labels in the generated code. A label is consume for each function/program and for each menu page.

The second column is the function/program name/label to call. If this is empty, then the menu name will be used. For example "SFUN:HYP:SINH" will call the SINH function if no Program is specified.

The other columns of the table are not used.

Menu	Function/Program	
SFUN		https://richmit.github.io/hp42/math.html
MATH:POLY		https://richmit.github.io/hp42/math.html
MATH:FUNC	FMENU	https://richmit.github.io/hp42/pgmforfun.html
MATH:VEC3		https://richmit.github.io/hp42/math.html
MATH:NLA		https://richmit.github.io/hp42/math.html
BASE		https://richmit.github.io/hp42/base.html
PHIL		https://richmit.github.io/hp42/phil.html
MEAS		https://richmit.github.io/hp42/meas.html
DATE	DMENU	https://richmit.github.io/hp42/date.html
CONS		https://richmit.github.io/hp42/cons.html
UNIT		https://richmit.github.io/hp42/unit.html
SIPFX		https://richmit.github.io/hp42/unit.html
UPROP		https://richmit.github.io/hp42/uprop.html

3 Custom Menus

3.1 Menu Contents

3.2 Menu Code

KEY 9 GTO 01

The menu program is generated via the following bit of elisp. You must first define the MJR-generate-42-menu-code and MJR-custom-x-gen by evaluating the code blocks in the hp42s-meta.org file.

(MJR-generate-42-menu-code "CUST" 0 tbl 0 1 'stay 'up #'MJR-custom-gen-lab #'MJR-custom-gen-sub)

```
(CUST)
0000 DSC: Auto-generated menu program
LBL "CUST"
LBL 01
                0000 Page 1 of menu CUST
CLMENU
"SFUN"
KEY 1 XEQ "SFUN"
"MATH"
KEY 2 GTO 03
"BASE"
KEY 3 XEQ "BASE"
"PHIL"
KEY 4 XEQ "PHIL"
"MEAS"
KEY 5 XEQ "MEAS"
"DATE"
KEY 6 XEQ "DMENU"
KEY 7 GTO 02
KEY 8 GTO 02
KEY 9 GTO 00
MENU
STOP
GTO 01
                @@@@ Page 2 of menu CUST
LBL 02
CLMENU
"CONS"
KEY 1 XEQ "CONS"
"UNIT"
KEY 2 XEQ "UNIT"
"SIPFX"
KEY 3 XEQ "SIPFX"
"UPROP"
KEY 4 XEQ "UPROP"
KEY 7 GTO 01
KEY 8 GTO 01
KEY 9 GTO 00
MENU
STOP
GTO 02
LBL 03
                @@@@ Page 1 of menu MATH
CLMENU
"POLY"
KEY 1 XEQ "POLY"
"FUNC"
KEY 2 XEQ "FMENU"
"VEC3"
KEY 3 XEQ "VEC3"
"NLA"
KEY 4 XEQ "NLA"
```

```
MENU
STOP
GTO 03
LBL 00 @@@@ Application Exit
EXITALL
RTN
@@@@ Free labels start at: 4
```

4 Create system CUSTOM Menu

Sometimes our carefully built in CUSTOM menu is lost — an errant program or system memory clear. What to do? How about a program that will recreate your CUSTOM menu when you need it? In the table below we have the definition of our custom menu, and a bit of lisp that will write a 42s program to ASSIGN each key.

KEY#	ASSIGNment	Only On
1	CUST	
2		
3		
4		
5	SIPFX	
6		
7	VirtV	DM42
8		
9		
10		
11		
12		
13	SST	
14	SST^{\uparrow}	
15	$SST \rightarrow$	
16	GTO	
17	XEQ	
18	VIEW	DM42
18	PRV	!DM42

Simply evaluate the followign lisp block, and the 42s program will be created.

```
;; The generated code uses the existance of GrMod to guess if we are on a DM42
(princ "@@@@ DSC: Autogenerated menu program\n")
(princ "LBL \"MKCM\"\n")
(princ "CLKEYS\n")
(cl-loop for (da-key da-assignment da-pform) in tbl
       for da-assignment-fixed = (replace-regexp-in-string "|" "|" da-assignment 't 't)
       for have-key = (not (zerop (length da-assignment-fixed)))
       when (and have-key (string-equal da-pform "DM42"))
       do (princ "SF 25\nRCL \"GrMod\"\nFS?C 25\n")
       when (and have-key (string-equal da-pform "!DM42"))
       do (princ "SF 25\nRCL \"GrMod\"\nFC?C 25\n")
       when have-key
       do (princ (message "ASSIGN \"%s\" TO %02d\n" da-assignment-fixed da-key))))
(princ "RTN\n")
(MKCM)
@@@@ DSC: Autogenerated menu program
LBL "MKCM"
CLKEYS
ASSIGN "CUST" TO 01
ASSIGN "SIPFX" TO 05
ASSIGN "||" TO 06
SF 25
RCL "GrMod"
FS?C 25
ASSIGN "VirtV" TO 07
ASSIGN "SST" TO 13
ASSIGN "SST↑" TO 14
ASSIGN "SST→" TO 15
ASSIGN "GTO" TO 16
ASSIGN "XEQ" TO 17
SF 25
RCL "GrMod"
FS?C 25
ASSIGN "VIEW" TO 18
SF 25
RCL "GrMod"
FC?C 25
```

```
ASSIGN "PRV" TO 18
```

5 SPREF: Set some of my personal prefrences

```
(SPREF)
0000 DSC: Set some of my personal prefrences
0000 IN: N/A
0000 OUT: N/A
@@@@ TST: free42_3.0.2
0000 BUG: This sets some of *my* favorite prefrences. You might not like them.;)
@@@@ UPD: 2021-04-05
LBL "SPREF"
FUNC 00
ALL
      0000 Display all digits
RECT
     @@@@ Complex number format
      0000 Angle mode
RAD
CPXRES @@@@ Complex results
      0000 Use periods
KEYASN @@@@ Custom menu
DECM @@@@ Make sure we are in decimal mode
      @@@@ WSIZE
64
FS? 78
BSIGNED @@@@ Unsigned integer mode
FS? 79
BWRAP @@@@ Don't wrap integers
DMY
      @@@@ D.MY mode
CL12
      @@@@ AM/PM mode
4STK
      0000 Four level stack
EXITALL @@@@ Exit menus
RTN
```

6 END

7 EOF