

free42 Custom Menus

Mitch Richling

2021-03-19

Author: Mitch Richling
Updated: 2021-05-14 18:16:08

Copyright 2021 Mitch Richling. All rights reserved.

Contents

1	Metadata	1
2	Introduction	1
2.1	Customizing Custom Menu Table	1
3	Custom Menus	2
3.1	Menu Contents	2
3.2	Menu Code	2
4	Create system CUSTOM Menu	3
5	SPREF : Set some of my personal preferences	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
bin	-	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 consumed 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.

4 Create system CUSTOM Menu

KEY#	ASSIGNment	Only On
1	CUST	
2		
3		
4		
5	EEPFx	
6		
7	VirtV	DM42
8		
9		
10		
11		
12		
13	SST	
14	SST↑	
15	SST→	
16	GTO	
17	XEQ	
18	VIEW	DM42
18	PRV	!DM42

```
; ; The generated code uses the existence of GrMod to guess if we are on a DM42
(princ (message "))))))))) (ref:%s)\n" "MKCM"))
(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\" T0 %02d\n" da-assignment-fixed da-key))))
(princ "RTN\n")
```

3

5 SPREF: Set some of my personal preferences

6 END

7 EOF