This is a raw version of the HowTo - don't let that scare you though :D

FILE: app_defines.h

* find a free spot in the ram like

;; free: 0x6d..0x79

* add the variable to store the layer

CS_MENU_LAYER EQU 0x06d ; stores the active layer

* and possibly change the comment to match the new situation

;; free: 0x6e..0x79

FILE: setup_*.asm:

* in the CS_MENU_DIN_TABLE add the new buttons (SR + Pin might not match)

```
DIN_ENTRY   CS_MENU_BUTTON_Layer1,        2,     7
DIN_ENTRY   CS_MENU_BUTTON_Layer2,        1,     4
DIN_ENTRY   CS_MENU_BUTTON_Layer3,        1,     5
DIN_ENTRY   CS_MENU_BUTTON_Layer4,        1,     6
DIN_ENTRY   CS_MENU_BUTTON_Layer5,        1,     7
DIN_ENTRY   CS_MENU_BUTTON_Layer6,        1,     3
```

* in the CS_MENU_DOUT_TABLE add the layer LEDs (SR + Pin might not match)

```
DOUT_ENTRY  TMP3, 0,            1,  0    ; Layer 1 LED
DOUT_ENTRY  TMP3, 1,            1,  1    ; Layer 2 LED
DOUT_ENTRY  TMP3, 2,            1,  2    ; Layer 3 LED
DOUT_ENTRY  TMP3, 3,            1,  3    ; Layer 4 LED
DOUT_ENTRY  TMP3, 4,            1,  4    ; Layer 5 LED
DOUT_ENTRY  TMP3, 5,            1,  5    ; Layer 6 LED
```

FILE: cs_menu_buttons.inc:

* after

```
;; ---------------------------------------------------------------------
-
;; the button functions are defined here
;; ---------------------------------------------------------------------
-
```

* add the button handlers for the layer buttons

```
CS_MENU_BUTTON_Layer1
CS_MENU_BUTTON_Layer2
CS_MENU_BUTTON_Layer3
CS_MENU_BUTTON_Layer4
CS_MENU_BUTTON_Layer5
CS_MENU_BUTTON_Layer6
```

* now it should compile and run, but not do anything  * each of those “functions” handles the button presses for one layer button

* For example, the “Oscillator Layer” handler

CS_MENU_BUTTON_Layer2 ;; OSC 1 Layer

```
;; do nothing if button has been depressed
btfsc   MIOS_PARAMETER2, 0                    ;; check if button has been
released
return                                      ;; if so exit
;; open the menu
call    CS_MENU_GetMenuID_OSC                ;; goto OSC menu page
call    CS_MENU_BUTTON_Hlp_MenuChangeOk       ;; propagate menu change
;; store the active layer
movlw   0x02                                ;; put 2 (layer number) on the
WREG
movwf   CS_MENU_LAYER                        ;; and write it to the variable
return                                      ;; done
```

* if you only use the predefined menus, this is all there is to do for each layer. * I added a special menu for Filter and LFO, this requires some more work:

FILE: cs_menu_tables.inc:

* in CS_MENU_TABLES_IDS add another row like this:

```
db  CS_MENU_L_FL,  CS_MENU_ROOT,  CS_MENU_ROOT,  CS_MENU_ROOT
```

* CS_MENU_L_FL is the new [F]ilter+[L]fo menu * in CS_MENU_TABLES_L add a new row like this:

```
CS_MENU_T_ENTRY CS_MENU_TABLE_L_FL, CS_MENU_Page_Parameters,
CS_MENU_EXEC_GoToRoot, "F/L", 0, 0x00, PRINT_NOP
```

* right after that table there's a list of the menu IDs. To add our new menu add a line like this:

CS_MENU_L_FL EQU 0x0f

* now we want to add the menu to the menu structure. To do so you need to find CS_MENU_TABLE_L_ROOT, the “root table” * I wanted to have the new menu between the LFO and Envelope menus, so I put it between those two:

```
CS_MENU_ENTRY   CS_MENU_L_LFO,              "LFO",  0x000, PRINT_NOP,
EXEC_MENU,       R2PP2R_NOP
CS_MENU_ENTRY   CS_MENU_L_ENV,              "ENV",  0x000, PRINT_NOP,
EXEC_MENU,       R2PP2R_NOP
```

* resulting in this:

```
CS_MENU_ENTRY   CS_MENU_L_LFO,              "LFO",  0x000, PRINT_NOP,
EXEC_MENU,       R2PP2R_NOP
CS_MENU_ENTRY   CS_MENU_L_FL,               "F/L",  0x000, PRINT_NOP,
EXEC_MENU,       R2PP2R_NOP
CS_MENU_ENTRY   CS_MENU_L_ENV,              "ENV",  0x000, PRINT_NOP,
EXEC_MENU,       R2PP2R_NOP
```

* note that you can change the order of the menu items as you wish. * Now we're still missing the actual menu :D * At the end of the file insert something like this:

;

; The combined FIL/LFO menu ;

CS_MENU_TABLE_L_FL

```
db  (CS_MENU_TABLE_L_FL_End-CS_MENU_TABLE_L_FL)/CS_MENU_ENTRY_LEN, 0x00
```

```
;;              Register (00=dummy)         |<->|  max   print ix
exec ix          parameter transfer
CS_MENU_ENTRY   SID_Ix_L_Fx_CUTOFF_L,       "Cut", 0xfff, PRINT_Fx_CUTOFF,
EXEC_SELPAR,     R2PP2R_Fx_CUTOFF
CS_MENU_ENTRY   SID_Ix_L_Fx_RESONANCE,      "Res", 0x0ff, PRINT_Fx_RESON,
EXEC_SELPAR,     R2PP2R_Fx_RESON
CS_MENU_ENTRY   0x00,                       "   ", 0x000, PRINT_NOP,
EXEC_NOP,        R2PP2R_NOP
CS_MENU_ENTRY   SID_Ix_LFOx_RATE,           "Rte", 0x0ff, PRINT_LFOx_RATE,
EXEC_SELPAR,     R2PP2R_LFOx
CS_MENU_ENTRY   SID_Ix_LFOx_DEPTH,          "Dep", 0x0ff,
PRINT_LFOx_PMDEC8,EXEC_SELPAR,     R2PP2R_LFOx
```

CS_MENU_TABLE_L_FL_End

* this menu page has 5 entries, the third one being an empty dummy entry * Now the menu is done  * to make one layer button open this menu all we need to do is change the handler above to call this page * in this case that would look like this:

CS_MENU_BUTTON_Layer5 ;; FILTER/LFO LAYER

```
;; do nothing if button has been depressed
btfsc   MIOS_PARAMETER2, 0
```

```
return
movlw   0x10
movwf   CS_MENU_LAYER
call    CS_MENU_GetMenuID_FL    ; goto FILTER/LFO menu page
goto    CS_MENU_BUTTON_Hlp_MenuChange
return
```

One last thing to edit

The CS_MENU_GetMenuID_FL function, which returns the ID of the menu page needs to be defined. This should be done in cs_menu.inc in the section titled

```
;; ----------------------------------------------------------------------------
;;  This function returns the CS_MENU_x_xxx ID depending on selected engine
;; ----------------------------------------------------------------------------
```

which is around line 2056.