

# Summary

This page is for describing the functions found in the file seq\_buttons.inc in the MBSEQ v2.4a application.

seq\_buttons.inc contains the handler functions for button presses. USER\_DIN\_NotifyToggle calls SEQ\_Button\_Handler which first checks to see if the button press came from one of the two shift registers which are assigned in main.asm to be the 16 general purpose buttons.

## Functions

SEQ_BUTTON_Handler
SEQ_BUTTON_Handler_ChkGP1
SEQ_BUTTON_Handler_ChkGP2
CS_MENU_BUTTON_Handler_NoGP
SEQ_BUTTON_Handler_Loop
SEQ_BUTTON_Handler_End
SEQ_BUTTON_GP
SEQ_BUTTON_GP1
SEQ_BUTTON_GP2
SEQ_BUTTON_GP3
SEQ_BUTTON_GP4
SEQ_BUTTON_GP5
SEQ_BUTTON_GP6
SEQ_BUTTON_GP7
SEQ_BUTTON_GP8
SEQ_BUTTON_GP9
SEQ_BUTTON_GP10
SEQ_BUTTON_GP11
SEQ_BUTTON_GP12
SEQ_BUTTON_GP13
SEQ_BUTTON_GP14
SEQ_BUTTON_GP15
SEQ_BUTTON_GP16
SEQ_BUTTON_GP_Cont
SEQ_BUTTON_Select
SEQ_BUTTON_Exit
SEQ_BUTTON_Left
SEQ_BUTTON_LeftRemote
SEQ_BUTTON_Right
SEQ_BUTTON_RightRemote
SEQ_BUTTON_StartStop
BUTTON_Control4SetStart
SEQ_BUTTON_StartStop_Stop

SEQ_BUTTON_Pause
SEQ_BUTTON_Rew
SEQ_BUTTON_Fwd
SEQ_BUTTON_Play
SEQ_BUTTON_Stop
SEQ_BUTTON_Continue
SEQ_BUTTON_Edit
SEQ_BUTTON_Mute
SEQ_BUTTON_Pattern
SEQ_BUTTON_Song
SEQ_BUTTON_Song_Toggle
SEQ_BUTTON_Song_Toggle_1
SEQ_BUTTON_Song_Toggle_0
SEQ_BUTTON_Song_Toggle_Cont
SEQ_BUTTON_Song_NoToggle
SEQ_BUTTON_Menu
SEQ_BUTTON_Metronome
SEQ_BUTTON_Scrub
SEQ_BUTTON_Solo
SEQ_BUTTON_All
SEQ_BUTTON_Fast
SEQ_BUTTON_Fast_Loop
SEQ_BUTTON_LayerA
SEQ_BUTTON_LayerB
SEQ_BUTTON_LayerC
SEQ_BUTTON_Layerx
SEQ_BUTTON_Layerx_MPS
SEQ_BUTTON_Layerx_Edit
SEQ_BUTTON_Layer_Depr
SEQ_BUTTON_Track1
SEQ_BUTTON_Track2
SEQ_BUTTON_Track3
SEQ_BUTTON_Track4
SEQ_BUTTON_Trackx_Cont
SEQ_BUTTON_Trackx_MPS
SEQ_BUTTON_Trackx_Edit
SEQ_BUTTON_Trackx_Toggle
SEQ_BUTTON_Trackx_Radio
SEQ_BUTTON_Trackx_End
SEQ_BUTTON_Trackx_Hlp_Sav
SEQ_BUTTON_F1
SEQ_BUTTON_F2
SEQ_BUTTON_F3
SEQ_BUTTON_F4

# Function Descriptions

SEQ\_BUTTON\_Handler\_ChkGP1 first moves MIOS\_PARAMETER1 into W, AND's W with 0xf8 and then xorlw's against "(DEFAULT\_GP\_DIN\_SR0 - 1) « 3" which, presumably results in zero if the registers match. Perhaps the AND statement converts the button number in MIOS\_PARAMETER1 into a value representing which SR it belongs to.

SEQ\_BUTTON\_Handler searches through mios\_tables.inc for a DIN pin that matches MIOS\_PARAMETER1 and then calls the appr. function

From the comments:

```
GP button number is stored in MIOS_PARAMETER1, button value in
MIOS_PARAMETER2 (1 for released, 0 for pressed)
```

If the button press did come from one of the GP's then the program branches to SEQ\_GP\_Button in seq\_gp.inc, not to be confused with SEQ\_Button\_GP in seq\_button.inc.

From:

<http://wiki.midibox.org/> - **MIDIbox**

Permanent link:

[http://wiki.midibox.org/doku.php?id=seq\\_buttons](http://wiki.midibox.org/doku.php?id=seq_buttons)

Last update: **2006/10/15 10:35**

