

```
1  /*****
2  System Interrupts File
3
4  File Name:
5      system_interrupt.c
6
7  Summary:
8      Raw ISR definitions.
9
10 Description:
11     This file contains a definitions of the raw ISRs required to support the
12     interrupt sub-system.
13
14 Summary:
15     This file contains source code for the interrupt vector functions in the
16     system.
17
18 Description:
19     This file contains source code for the interrupt vector functions in the
20     system. It implements the system and part specific vector "stub" functions
21     from which the individual "Tasks" functions are called for any modules
22     executing interrupt-driven in the MPLAB Harmony system.
23
24 Remarks:
25     This file requires access to the systemObjects global data structure that
26     contains the object handles to all MPLAB Harmony module objects executing
27     interrupt-driven in the system. These handles are passed into the individual
28     module "Tasks" functions to identify the instance of the module to maintain.
29 *****/
30
31 // DOM-IGNORE-BEGIN
32 /*****
33 Copyright (c) 2011-2014 released Microchip Technology Inc. All rights reserved.
34
35 Microchip licenses to you the right to use, modify, copy and distribute
36 Software only when embedded on a Microchip microcontroller or digital signal
37 controller that is integrated into your product or third party product
38 (pursuant to the sublicense terms in the accompanying license agreement).
39
40 You should refer to the license agreement accompanying this Software for
41 additional information regarding your rights and obligations.
42
43 SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND,
44 EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF
45 MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.
46 IN NO EVENT SHALL MICROCHIP OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER
47 CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR
48 OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES
49 INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR
50 CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF
51 SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES
52 (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.
53 *****/
54 // DOM-IGNORE-END
55
56 // *****/
57 // *****/
58 // Section: Included Files
59 // *****/
60 // *****/
61
62 #include "system/common/sys_common.h"
63 #include "app.h"
64 #include "app_sdcard.h"
65 #include "system_definitions.h"
66
67 // *****/
68 // *****/
69 // Section: System Interrupt Vector Functions
70 // *****/
71 // *****/
72
73
```

```

74
75
76 static uint8_t compt_Link = 0;
77
78
79
80
81
82
83
84
85 /*
86
87 void __ISR(_CHANGE_NOTICE_VECTOR, ipl1AUTO) _IntHandlerChangeNotification(void)
88 {
89     // TODO: Add code to process interrupt here
90     PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_CHANGE_NOTICE_A);
91 }*/
92
93
94
95 void __ISR(_TIMER_1_VECTOR, ipl1AUTO) IntHandlerDrvTmrInstance0(void)
96 {
97
98
99     PLIB_PORTS_PinToggle(PORTS_ID_0, PORT_CHANNEL_B, PORTS_BIT_POS_8);
100     PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_TIMER_1);
101 }
102
103 //timer 100ms
104 void __ISR(_TIMER_2_VECTOR, ipl1AUTO) IntHandlerDrvTmrInstance1(void)
105 {
106
107     if(compt_Link < 19)
108     {
109         compt_Link++;
110     }
111     else
112     {
113         APP_UpdateState(APP_STATE_LINK_XBEE);
114         compt_Link = 0;
115     }
116     PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_TIMER_2);
117 }
118
119 void __ISR(_SPI_1_VECTOR, ipl1AUTO) _IntHandlerSPIInstance0(void)
120 {
121     DRV_SPI_Tasks(sysObj.spiObjectIdx0);
122 }
123 void __ISR(_SPI_2_VECTOR, ipl1AUTO) _IntHandlerSPIInstance1(void)
124 {
125     DRV_SPI_Tasks(sysObj.spiObjectIdx1);
126 }
127 /*****
128 End of File
129 */
130

```