```
 1 /*
 2  * Copyright (c) 2015, Freescale Semiconductor, Inc.
 3  * Copyright 2016-2017 NXP
 4  * All rights reserved.
 5  *
 6  * SPDX-License-Identifier: BSD-3-Clause
 7  */
 8
 9 #include "board.h"
10 #include "fsl_debug_console.h"
11 #include "fsl_gpio.h"
12
13 #include "pin_mux.h"
14 #include <stdbool.h>
15 /*******************************************************************************
   **
16  * Definitions
17
   ******************************************************************************
   */
18 #define APP_BOARD_TEST_LED_PORT 1U
19 #define APP_BOARD_TEST_LED_PIN 10U
20 #define APP_SW_PORT BOARD_SW1_GPIO_PORT
21 #define APP_SW_PIN  BOARD_SW1_GPIO_PIN
22
23 #define SPEED1_TEST_LED_PORT 1U
24 #define SPEED1_TEST_LED_PIN 1U
25
26 #define SPEED2_TEST_LED_PORT 1U
27 #define SPEED2_TEST_LED_PIN 3U
28
29 #define SPEED3_TEST_LED_PORT 1U
30 #define SPEED3_TEST_LED_PIN 7U
31
32 #define SPEED4_TEST_LED_PORT 1U
33 #define SPEED4_TEST_LED_PIN 9U
34
35
36
37
38
39
40
41
42
43 /*******************************************************************************
```

```
  **
44  * Prototypes
45
  ******************************************************************************
  */
46 /*!
47  * @brief delay a while.
48  */
49 void delay(void);
50
51 /
  ******************************************************************************
  **
52  * Variables
53
  ******************************************************************************
  */
54
55 /
  ******************************************************************************
  **
56  * Code
57
  ******************************************************************************
  */
58 void delay(void)
59 {
60     volatile uint32_t i = 0;
61     for (i = 0; i < 100000; ++i)
62     {
63         __asm("NOP"); /* delay */
64     }
65 }
66
67 /*!
68  * @brief Main function
69  */
70 int main(void)
71 {
72     uint32_t port_state = 0;
73     uint32_t port_state_1 = 0;
74     uint32_t port_state_2 = 0;
75     uint32_t port_state_3 = 0;
76     uint32_t port_state_4 = 0;
77
78
79
80     /* Define the init structure for the output LED pin*/
```

```
 81    gpio_pin_config_t led_config = {
 82        kGPIO_DigitalOutput, 0,
 83    };
 84
 85    /* Board pin, clock, debug console init */
 86    /* attach 12 MHz clock to FLEXCOMM0 (debug console) */
 87    CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);
 88    /* enable clock for GPIO*/
 89    CLOCK_EnableClock(kCLOCK_Gpio0);
 90    CLOCK_EnableClock(kCLOCK_Gpio1);
 91
 92    BOARD_InitPins();
 93    BOARD_BootClockFROHF48M();
 94    BOARD_InitDebugConsole();
 95
 96    /* Print a note to terminal. */
 97    PRINTF("\r\n GPIO Driver example\r\n");
 98    PRINTF("\r\n The LED is taking turns to shine.\r\n");
 99
100    /* Init SW GPIO PORT. */
101    GPIO_PortInit(GPIO, SPEED1_TEST_LED_PORT);
102    GPIO_PortInit(GPIO, SPEED2_TEST_LED_PORT);
103    GPIO_PortInit(GPIO, DIRECTION1_TEST_LED_PORT);
104    GPIO_PortInit(GPIO, DIRECTION2_TEST_LED_PORT);
105
106    /* Init output LED GPIO. */
107    GPIO_PortInit(GPIO, APP_BOARD_TEST_LED_PORT);
108    GPIO_PinInit(GPIO, APP_BOARD_TEST_LED_PORT, APP_BOARD_TEST_LED_PIN,
   &led_config);
109    GPIO_PinWrite(GPIO, APP_BOARD_TEST_LED_PORT, APP_BOARD_TEST_LED_PIN, 1);
110
111    /* Port masking */
112    GPIO_PortMaskedSet(GPIO, APP_BOARD_TEST_LED_PORT, 0x0000FFFF);
113    GPIO_PortMaskedWrite(GPIO, APP_BOARD_TEST_LED_PORT, 0xFFFFFFFF);
114    port_state = GPIO_PortRead(GPIO, APP_SW_PORT);
115    port_state_1 = GPIO_PortRead(GPIO,SPEED1_TEST_LED_PORT);
116    port_state_1 = GPIO_PortRead(GPIO,SPEED2_TEST_LED_PORT);
117    port_state_1 = GPIO_PortRead(GPIO,DIRECTION1_TEST_LED_PORT);
118    port_state_1 = GPIO_PortRead(GPIO,DIRECTION2_TEST_LED_PORT);
119
120    PRINTF("\r\n Standard port read: %x\r\n", port_state);
121    PRINTF("\r\n Standard port read: %x\r\n", port_state_1);
122    PRINTF("\r\n Standard port read: %x\r\n", port_state_2);
123    PRINTF("\r\n Standard port read: %x\r\n", port_state_3);
124    PRINTF("\r\n Standard port read: %x\r\n", port_state_4);
125
126
127    port_state = GPIO_PortMaskedRead(GPIO, APP_SW_PORT);
```

```
128     port_state_1 = GPIO_PortMaskedRead(GPIO, SPEED1_TEST_LED_PORT);
129     port_state_2 = GPIO_PortMaskedRead(GPIO, SPEED2_TEST_LED_PORT);
130     port_state_3 = GPIO_PortMaskedRead(GPIO, DIRECTION1_TEST_LED_PORT);
131     port_state_4 = GPIO_PortMaskedRead(GPIO, DIRECTION2_TEST_LED_PORT);
132     PRINTF("\r\n Masked port read: %x\r\n", port_state);
133
134     while (1)
135     {
136         port_state = GPIO_PortRead(GPIO, APP_SW_PORT);
137         if (!(port_state & (1 << APP_SW_PIN)))
138         {
139             PRINTF("\r\n Port state: %x\r\n", port_state);
140             PRINTF("\r\n Port state:")
141             GPIO_PortToggle(GPIO, APP_BOARD_TEST_LED_PORT, 1u <<
    APP_BOARD_TEST_LED_PIN);
142         }
143         port_state_1 = GPIO_PortRead(GPIO, SPEED1_TEST_LED_PORT );
144         if (!(port_state_1 & (1 << SPEED1_TEST_LED_PORT)))
145          {
146             PRINTF("\r\n Port state: %x\r\n", port_state_1);
147             PRINTF("\r\n SPEED1 FOR M1:")
148             GPIO_PortToggle(GPIO, APP_BOARD_TEST_LED_PORT, 1u <<
    APP_BOARD_TEST_LED_PIN);
149          }
150          port_state_2 = GPIO_PortRead(GPIO, SPEED2_TEST_LED_PORT );
151          if (!(port_state_2 & (1 << SPEED2_TEST_LED_PORT)))
152            {
153             PRINTF("\r\n Port state: %x\r\n", port_state_1);
154             PRINTF("\r\n SPEED2 FOR M2:")
155             GPIO_PortToggle(GPIO, APP_BOARD_TEST_LED_PORT, 1u <<
    APP_BOARD_TEST_LED_PIN);
156           }
157           port_state_3 = GPIO_PortRead(GPIO, DIRECTION1_TEST_LED_PORT );
158           if (!(port_state_3 & (1 << DIRECTION1_TEST_LED_PORT)))
159           {
160             PRINTF("\r\n Port state: %x\r\n", port_state_1);
161             PRINTF("\r\n DIRECTION1 FOR M1:")
162             GPIO_PortToggle(GPIO, APP_BOARD_TEST_LED_PORT, 1u <<
    APP_BOARD_TEST_LED_PIN);
163            }
164           port_state_4 = GPIO_PortRead(GPIO, DIRECTION2_TEST_LED_PORT );
165           if (!(port_state_4 & (1 << DIRECTION2_TEST_LED_PORT)))
166           {
167              PRINTF("\r\n Port state: %x\r\n", port_state_1);
168              PRINTF("\r\n DIRECTION2 FOR M2:")
169              GPIO_PortToggle(GPIO, APP_BOARD_TEST_LED_PORT, 1u <<
    APP_BOARD_TEST_LED_PIN);
170            }
```

```
171
172         delay();
173     }
174 }
175
```