```c
/*
 * Copyright (c) 2016, Freescale Semiconductor, Inc.
 * Copyright 2016-2017 NXP
 * All rights reserved.
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include "fsl_debug_console.h"
#include "board.h"
#include "fsl_sctimer.h"

#include "pin_mux.h"
#include <stdbool.h>
/*******************************************************************************
 * Definitions
 ******************************************************************************/

#define SCTIMER_CLK_FREQ CLOCK_GetFreq(kCLOCK_BusClk)
#define DEMO_FIRST_SCTIMER_OUT kSCTIMER_Out_4
#define DEMO_SECOND_SCTIMER_OUT kSCTIMER_Out_2

/*******************************************************************************
 * Prototypes
 ******************************************************************************/

/*******************************************************************************
 * Variables
 ******************************************************************************/

/*******************************************************************************
 * Code
 ******************************************************************************/

/*!
 * @brief Main function
```

```
37 */
38 int main(void)
39 {
40     sctimer_config_t sctimerInfo;
41     sctimer_pwm_signal_param_t pwmParam;
42     uint32_t event;
43     uint32_t sctimerClock;
44 
45     /* Board pin, clock, debug console init */
46     /* attach 12 MHz clock to FLEXCOMM0 (debug console) */
47     CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);
48 
49     BOARD_InitPins();
50     BOARD_BootClockFROHF48M();
51     BOARD_InitDebugConsole();
52 
53     sctimerClock = SCTIMER_CLK_FREQ;
54 
55     /* Print a note to terminal */
56     PRINTF("\r\nSCTimer example to output 2 center-aligned PWM signals\r\n");
57     PRINTF("\r\nProbe the signal using an oscilloscope");
58 
59     SCTIMER_GetDefaultConfig(&sctimerInfo);
60 
61     /* Initialize SCTimer module */
62     SCTIMER_Init(SCT0, &sctimerInfo);
63 
64     /* Configure first PWM with frequency 24kHZ from first output */
65     pwmParam.output = DEMO_FIRST_SCTIMER_OUT;
66     pwmParam.level = kSCTIMER_HighTrue;
67     pwmParam.dutyCyclePercent = 25;
68     if (SCTIMER_SetupPwm(SCT0, &pwmParam, kSCTIMER_CenterAlignedPwm, 24000U,
   sctimerClock, &event) == kStatus_Fail)
69     {
70         return -1;
71     }
72 
73     /* Configure second PWM with different duty cycle but same frequency as
   before */
74     pwmParam.output = DEMO_SECOND_SCTIMER_OUT;
75     pwmParam.level = kSCTIMER_HighTrue;
76     pwmParam.dutyCyclePercent = 50;
77     if (SCTIMER_SetupPwm(SCT0, &pwmParam, kSCTIMER_CenterAlignedPwm, 24000U,
   sctimerClock, &event) == kStatus_Fail)
78     {
79         return -1;
80     }
81 
```

```
82      /* Start the timer */
83      SCTIMER_StartTimer(SCT0, kSCTIMER_Counter_L);
84
85      while (1)
86      {
87      }
88  }
89
```