```
/*
 * Copyright 2016-2021 NXP
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or
without modification,
 * are permitted provided that the following conditions are met:
 *
 * o Redistributions of source code must retain the above
copyright notice, this list
 *   of conditions and the following disclaimer.
 *
 * o Redistributions in binary form must reproduce the above
copyright notice, this
 *   list of conditions and the following disclaimer in the
documentation and/or
 *   other materials provided with the distribution.
 *
 * o Neither the name of NXP Semiconductor, Inc. nor the names
of its
 *   contributors may be used to endorse or promote products
derived from this
 *   software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR
 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

```
/**
```

```
 * @file    LPC802_Project_lab8.c
 * @brief   Application entry point.
 */
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC802.h"
#include "fsl_debug_console.h"
/* TODO: insert other include files here. */

/* TODO: insert other definitions and declarations here. */

#define red (9)
#define blue (12)
#define button (8)
void delay(void);

/*
 * @brief   Application entry point.
 */
int main(void) {

    /* Init board hardware. */
    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
    /* Init FSL debug console. */
    BOARD_InitDebugConsole();
#endif

    PRINTF("Hello World\n");
    SYSCON->SYSAHBCLKCTRL0  |= SYSCON_SYSAHBCLKCTRL0_GPIO0_MASK;
    //set data direction
    GPIO->DIRCLR[0] = (1UL << button);  //set port 8 to input
    GPIO->DIRSET[0] = (1UL << red);   //make red LED output
    GPIO->DIRSET[0] = (21UL << blue);  //make blue LED output
    //Before the loop turn both  off
    GPIO->CLR[0] = (1UL <<red);
    GPIO->CLR[0] = (1UL <<blue);

    /* Force the counter to be placed into memory. */

    /* Enter an infinite loop, just incrementing a counter. */
    while(1) {

        /* 'Dummy' NOP to allow source level single stepping of
            tight while() loop */
        //__asm volatile ("nop");
        //check if pin 8 is pushed
        if(GPIO->B[0][button] &1 ) { // it is high, not pushed
          GPIO->CLR[0] = (1UL<<red);  //  red LED is ON
          GPIO->SET[0] = (1UL<<blue);  // blue is off

        }
        else {  // it is low, pushed
          GPIO->NOT[0] = (1UL<<red);  //invert red
          //GPIO->CLR[0] = (1UL<<blue); //
          delay();
          GPIO->NOT[0] = (1UL<<blue);  //invert blue


        }
        }

    return 0 ;
}

void delay(void) {
     int a=0;
     for(a=0; a<400000; a++);
     asm("NOP");

}
```