```
/*****************************************************************************
 *
 * Alternately toggle two LEDs when a push button is pressed.
 * ATmega328P (Arduino Uno), 16 MHz, AVR 8-bit Toolchain 3.6.2
 *
 * Copyright (c) 2018-2020 Tomas Fryza
 * Dept. of Radio Electronics, Brno University of Technology, Czechia
 * This work is licensed under the terms of the MIT license.
 *
 *****************************************************************************/

/* Defines -----------------------------------------------------------*/
#define LED_RED_0     PC0     // AVR pin where red LED 0 is connected
#define LED_RED_1     PC1     // AVR pin where red LED 1 is connected
#define LED_RED_2     PC2     // AVR pin where red LED 2 is connected
#define LED_RED_3     PC3     // AVR pin where red LED 3 is connected
#define LED_RED_4     PC4     // AVR pin where red LED 4 is connected
#define BTN         PD0
#define BLINK_DELAY 250
#ifndef F_CPU
#define F_CPU 16000000      // CPU frequency in Hz required for delay
#endif

/* Includes ----------------------------------------------------------*/
#include <util/delay.h>     // Functions for busy-wait delay loops
#include <avr/io.h>         // AVR device-specific IO definitions

/* Functions ---------------------------------------------------------*/
/**
 * Main function where the program execution begins. Toggle two LEDs
 * when a push button is pressed.
 */
int main(void)
{
    int i;
    /* LEDs */
    // Set pin as output in Data Direction Register...
    DDRC = DDRC | (1<<LED_RED_0);
    // ...and turn LED off in Data Register
    PORTC = PORTC | (1<<LED_RED_0);
    // Set pin as output in Data Direction Register...
    DDRC = DDRC | (1<<LED_RED_1);
    // ...and turn LED off in Data Register
    PORTC = PORTC & ~(1<<LED_RED_1);
    // Set pin as output in Data Direction Register...
    DDRC = DDRC | (1<<LED_RED_2);
    // ...and turn LED off in Data Register
    PORTC = PORTC & ~(1<<LED_RED_2);
    // Set pin as output in Data Direction Register...
    DDRC = DDRC | (1<<LED_RED_3);
    // ...and turn LED off in Data Register
    PORTC = PORTC & ~(1<<LED_RED_3);
    // Set pin as output in Data Direction Register...
    DDRC = DDRC | (1<<LED_RED_4);
    // ...and turn LED off in Data Register
    PORTC = PORTC & ~(1<<LED_RED_4);

    /* buton */

    DDRD = DDRD & ~(1<<BTN);
    PORTD = PORTD | (1<<BTN);


    // Infinite loop
    while (1)
    {

        for(i=0;i<4;i++)
        {
            // Pause several milliseconds
            _delay_ms(BLINK_DELAY);

            loop_until_bit_is_clear(PIND, BTN);

            PORTC = PORTC<<1;
        }

        for(i=0;i<4;i++)
        {
            // Pause several milliseconds
            _delay_ms(BLINK_DELAY);

            loop_until_bit_is_clear(PIND, BTN);

            PORTC = PORTC>>1;
       }
    }
    // Will never reach this
    return 0;
    }
```