```c
/***********************************************************************
 *
 * Control LEDs using functions from GPIO and Timer libraries. Do not
 * use delay library any more.
 * 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_D1  PB5
#define LED_D2  PB4
#define LED_D3  PB3
#define LED_D4  PB2

/* Includes ----------------------------------------------------------*/
#include <avr/io.h>         // AVR device-specific IO definitions
#include <avr/interrupt.h>  // Interrupts standard C library for AVR-GCC
#include "gpio.h"           // GPIO library for AVR-GCC
#include "timer.h"          // Timer library for AVR-GCC

/* Function definitions ----------------------------------------------*/
/**
 * Main function where the program execution begins. Toggle one LED
 * on the Multi-function shield using the internal 8- or 16-bit
 * Timer/Counter.
 */
int main(void)
{
    /* Configuration of LED(s) */
    GPIO_config_output(&DDRB, LED_D1);
    GPIO_write_low(&PORTB, LED_D1);
    GPIO_config_output(&DDRB, LED_D2);
    GPIO_write_low(&PORTB, LED_D2);
    GPIO_config_output(&DDRB, LED_D3);
    GPIO_write_low(&PORTB, LED_D3);

    /* Configuration of 8-bit Timer/Counter0 */
    TIM0_overflow_16ms();
    TIM0_overflow_interrupt_enable();

    /* Configuration of 16-bit Timer/Counter1
     * Set prescaler and enable overflow interrupt */
    TIM1_overflow_262ms();
    TIM1_overflow_interrupt_enable();

    /* Configuration of 8-bit Timer/Counter2 */
    TIM2_overflow_4ms();
    TIM2_overflow_interrupt_enable();

```

```
54
55      // Enables interrupts by setting the global interrupt mask
56      sei();
57
58      // Infinite loop
59      while (1)
60      {
61          /* Empty loop. All subsequent operations are performed exclusively
62           * inside interrupt service routines ISRs */
63      }
64
65      // Will never reach this
66      return 0;
67  }
68
69  /* Interrupt service routines ----------------------------------------*/
70  /**
71   * ISR starts when Timer/Counter0 overflows. Toggle D1 LED on
72   * Multi-function shield.
73   */
74  ISR(TIMER0_OVF_vect)
75  {
76      // Configuring a GPIO pin for output and toggling it.
77      GPIO_toggle(&PORTB, LED_D1);
78  }
79
80  /**
81   * ISR starts when Timer/Counter1 overflows. Toggle D2 LED on
82   * Multi-function shield.
83   */
84  ISR(TIMER1_OVF_vect)
85  {
86      // Configuring a GPIO pin for output and toggling it.
87      GPIO_toggle(&PORTB, LED_D2);
88  }
89
90  /**
91   * ISR starts when Timer/Counter2 overflows. Toggle D3 LED on
92   * Multi-function shield.
93   */
94  ISR(TIMER2_OVF_vect)
95  {
96      // Configuring a GPIO pin for output and toggling it.
97      GPIO_toggle(&PORTB, LED_D3);
98  }
```