```c
/************************************************************************
 *
 * Blink a LED and use the function from the delay library.
 * 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.
 *
 ************************************************************************/

/* Github ------------------------------------------------------------*/
// https://github.com/venca611/Digital-electronics-2


/* Defines -----------------------------------------------------------*/
#define LED_GREEN   PB5     // AVR pin where green LED is connected
#define SHORT_D 200     // Delay in miliseconds
#define LONG_D SHORT_D*3 // Delay in miliseconds


#ifndef F_CPU
#define F_CPU 16000000      // CPU frequency in Hz required for delay func
#endif

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

/* Variables ---------------------------------------------------------*/
/* Function prototypes -----------------------------------------------*/
void blink(char);
void morse_char(char);
void morse_code(char *);

/* Functions ---------------------------------------------------------*/
void blink(char charD){
    PORTB = PORTB ^ (1<<LED_GREEN);
    if(charD == 'S')
        _delay_ms(SHORT_D); // tecka
    else
        _delay_ms(LONG_D); // carka

    PORTB = PORTB ^ (1<<LED_GREEN);
    _delay_ms(SHORT_D);
    return;
}

void morse_char(char character){
    switch(character){
        case 'D': blink('L'); blink('S'); blink('S'); break;
        case 'E': blink('S'); break;
```

```
        case '2': blink('S'); blink('S'); blink('L'); blink('L'); blink('L');
          break;
    }
    _delay_ms(LONG_D);
    return;
}

void morse_code(char *name){
    for(int i = 0; i < strlen(name); i++)
        morse_char(name[i]);
    return;
}



/**
 * Toggle one LED and use the function from the delay library.
 */
int main(void)
{
    // Set pin as output in Data Direction Register
    // DDRB = DDRB or 0010 0000
    DDRB = DDRB | (1<<LED_GREEN);

    // Set pin LOW in Data Register (LED off)
    // PORTB = PORTB and 1101 1111
    PORTB = PORTB & ~(1<<LED_GREEN);

    char *name = "DE2";
    // Infinite loop
    while (1)
    {
        morse_code(name);
        for(int i = 0; i < 7; i++)
            _delay_ms(SHORT_D);
    }
    // Will never reach this
    return 0;
}

/* Interrupt routines ----------------------------------------------*/
```