**https://github.com/PetrDockalik/Digital-electronics-2**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Blink a LED and use 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.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Defines -----------------------------------------------------------\*/

#define LED\_GREEN PB5 // AVR pin where green LED is connected

#define SHORT\_DELAY 300 // Delay in milliseconds

#define LONG\_DELAY 1500 // Delay in milliseconds

#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 one LED

\* and use 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);

// Infinite loop

while (1)

{

// Invert LED in Data Register

// PORTB = PORTB xor 0010 0000

PORTB = PORTB ^ (1<<LED\_GREEN); //On Line

\_delay\_ms(LONG\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //Off

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //On

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);´//Off

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //On

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //Off

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //On

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //Off

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //On

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //Off

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //On

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN); //Off

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);

\_delay\_ms(LONG\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);

\_delay\_ms(LONG\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);

\_delay\_ms(SHORT\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);

\_delay\_ms(LONG\_DELAY);

PORTB = PORTB ^ (1<<LED\_GREEN);

\_delay\_ms(3000);

}

// Will never reach this

return 0;

}

