

## Lab assignment - 01

---

1. Submit the link to your Digital-electronics-2 repository on GitHub.
  - <https://github.com/xsigmu06/Digital-electronics-2>
2. Blink example. Submit:
  - Answers to questions: What is the meaning of |, &, ^, ~, << binary operators? Write a truth table and explain the use of operators with examples.

- |... bitwise OR

○ A	○ B	○ A B
○ 0	○ 0	○ 0
○ 0	○ 1	○ 1
○ 1	○ 0	○ 1
○ 1	○ 1	○ 1

- copies a bit if it exists in either operand
- $0101 \mid 0001 = 0101 \dots (5 \rightarrow 5)$

- &... bitwise AND

○ A	○ B	○ A&B
○ 0	○ 0	○ 0
○ 0	○ 1	○ 0
○ 1	○ 0	○ 0
○ 1	○ 1	○ 1

- copies a bit if it exists in both operands
- $0101 \& 0001 = 0001 \dots (5 \rightarrow 1)$

- ^... bitwise XOR

○ A	○ B	○ A^B
○ 0	○ 0	○ 0
○ 0	○ 1	○ 1
○ 1	○ 0	○ 1
○ 1	○ 1	○ 0

- copies a bit if it exists in exactly one operand
- $0101 \wedge 0001 = 0100$  ... (5 -> 4)

- ~... bitwise COMPLEMENT (NEGATION) - unary

○ A	○ ~A
○ 0	○ 1
○ 1	○ 0

- flips a bit to it's opposite value
- $\sim 0101 = 1010$  ... (5 -> 10)

- <<... bitwise left shift

- moves a bit to the left by number of bits specified by left operand
- $0101 \ll 0001 = 1010$  ... (5 -> 10)

### 3. Morse code application. Submit:

- C code (main.c).

- <https://github.com/xsigmu06/Digital-electronics-2/blob/master/Labs/01-tools/blink/blink/main.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.
*
*****/

/* Defines -----*/

                                                                    // # ... pred

kompilaci

#define LED_GREEN    PB5      // AVR pin where green LED is connected
                             // preimenuje PB5
#define SHORT_DELAY 100      // Delay in milliseconds
#define DOT_DELAY    200      // Delay for dot in milliseconds
#define DASH_DELAY    400     // Delay for dash in milliseconds
#define LEN 50              // max length of input characters
#ifndef F_CPU
#define F_CPU 16000000      // CPU frequency in Hz required for delay func
                             // procesor bezi na 16 MHz
#endif

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

/* Variables -----*/

/* Function prototypes -----*/

/* Functions -----*/
/**
 * Toggle one LED and use the function from the delay library.
 */

void morse(char code[LEN]) // Function for detecting dash or dot in char array
and blinking correspondingly
{
    for(int i = 0; i<=LEN; i++)
    {
        if(code[i]=='-')
        {
            PORTB = PORTB ^ (1<<LED_GREEN);
            _delay_ms(DASH_DELAY);
            PORTB = PORTB ^ (1<<LED_GREEN);
            _delay_ms(DOT_DELAY);
        }
        else if(code[i]=='.')
        {

```

```

        PORTB = PORTB ^ (1<<LED_GREEN);
        _delay_ms(DOT_DELAY);
        PORTB = PORTB ^ (1<<LED_GREEN);
        _delay_ms(DOT_DELAY);
    }
    else
    {
        _delay_ms(SHORT_DELAY);
    }
}

}

int main(void)
{
    // Set pin as output in Data Direction Register
    // DDRB = DDRB or 0010 0000
    DDRB = DDRB | (1<<LED_GREEN);

    // data direction register B, jestli je
    vstupni nebo vyst. (...nahrajeme 1) port

    //OR - nastaveni
    na 1... ( aby se nam neprepsaly pripadne 1 ktere tam uz jsou)
    // Set pin LOW in Data Register (LED off)
    // PORTB = PORTB and 1101 1111
    PORTB = PORTB & ~(1<<LED_GREEN);

    //<< binary shift, posune se o PB5 - 5 pozic
    (PB5 je na 5. pinu)

    //AND -
    nastaveni na nulu, negace ~
    // Infinite loop
    // blinking
    while (1)
    {
        char code[LEN] = "-...,.----";
        // D-E-2 ...(- . .; .; . . - - -)

        morse(code);
        // Pause several milliseconds
        _delay_ms(SHORT_DELAY);

        // Invert LED in Data Register
        // PORTB = PORTB xor 0010 0000
        PORTB = PORTB ^ (1<<LED_GREEN);
    }

    // Will never reach this

```

```
    return 0;  
}  
/* Interrupt routines -----*/
```