Lab1

Lab results

1. **GitHub**: https://github.com/FilipPaul/Digital-Electronics-2
2. **Blink example**.
    - Meaning of | is logical OR, in blink example is this operator used for writing right value to DDRB register (here you can set if GPIO is output or input).

```
3. DDRB = DDRB | (1 << LED_GREEN);
```

    - Meaning of & is logical AND, in blink example is this operator used together with ~ (logical NOT) and with <<(operator for shifting bits left) for setting 5th LSB bit in register PORTB to zero without changing other bits in register PORTB.

```
4. PORTB = PORTB & ~(1 << LED_GREEN);
```

    - Meaning of ^ is logical XOR, in sketch is used for toggling only 1 bit in register

```
5. PORTB = PORTB ^ (1 << LED_GREEN);
```

6. **Morse code. Submit**:
    - C code (main.c). **also can be found in GITHUB repository with firmware.hex!!**

```
7. #include <Arduino.h>
8.
9. /* Defines -----------------------------------------------------------*/
10.#define LED_GREEN PB5   // AVR pin where green LED is connected
11.#define SHORT_DELAY 300 // Delay in miliseconds
12.#define LONG_DELAY 600
13.#define BETWEEN_DELAY 300
14.#ifndef F_CPU
15.#define F_CPU 16000000 // CPU frequency in Hz required for delay func
16.#endif
17.
18./* Includes ----------------------------------------------------------*/
19.#include <util/delay.h> // Functions for busy-wait delay loops
20.#include <avr/io.h>     // AVR device-specific IO definitions
21.void morse(String letter)
22.{
23.  if (letter == "D")
24.  {
25.    PORTB = PORTB ^ (1 << LED_GREEN);
26.    _delay_ms(LONG_DELAY);
27.    PORTB = PORTB ^ (1 << LED_GREEN);
28.    _delay_ms(LONG_DELAY);
29.    PORTB = PORTB ^ (1 << LED_GREEN);
```

```
30.     _delay_ms(SHORT_DELAY);
31.     PORTB = PORTB ^ (1 << LED_GREEN);
32.     _delay_ms(SHORT_DELAY);
33.     PORTB = PORTB ^ (1 << LED_GREEN);
34.     _delay_ms(SHORT_DELAY);
35.     PORTB = PORTB ^ (1 << LED_GREEN);
36.     _delay_ms(SHORT_DELAY);
37.   }
38.   else if (letter == "E")
39.   {
40.     PORTB = PORTB ^ (1 << LED_GREEN);
41.     _delay_ms(SHORT_DELAY);
42.     PORTB = PORTB ^ (1 << LED_GREEN);
43.     _delay_ms(SHORT_DELAY);
44.   }
45.   else if (letter == "2")
46.   {
47.     PORTB = PORTB ^ (1 << LED_GREEN);
48.     _delay_ms(SHORT_DELAY);
49.     PORTB = PORTB ^ (1 << LED_GREEN);
50.     _delay_ms(SHORT_DELAY);
51.     PORTB = PORTB ^ (1 << LED_GREEN);
52.     _delay_ms(SHORT_DELAY);
53.     PORTB = PORTB ^ (1 << LED_GREEN);
54.     _delay_ms(SHORT_DELAY);
55.     PORTB = PORTB ^ (1 << LED_GREEN);
56.     _delay_ms(LONG_DELAY);
57.     PORTB = PORTB ^ (1 << LED_GREEN);
58.     _delay_ms(LONG_DELAY);
59.     PORTB = PORTB ^ (1 << LED_GREEN);
60.     _delay_ms(LONG_DELAY);
61.     PORTB = PORTB ^ (1 << LED_GREEN);
62.     _delay_ms(LONG_DELAY);
63.     PORTB = PORTB ^ (1 << LED_GREEN);
64.     _delay_ms(LONG_DELAY);
65.     PORTB = PORTB ^ (1 << LED_GREEN);
66.     _delay_ms(LONG_DELAY);
67.   }
68.}
69.
70./* Variables -----------------------------------------------------------*/
71.
72./* Function prototypes ------------------------------------------------*/
73.
74./* Functions -----------------------------------------------------------*/
75./**
76. * Toggle one LED and use the function from the delay library.
77. */
78.int main(void)
```

```
79.{
80.  // Set pin as output in Data Direction Register
81.  // DDRB = DDRB or 0010 0000
82.  DDRB = DDRB | (1 << LED_GREEN);
83.
84.  // Set pin LOW in Data Register (LED off)
85.  // PORTB = PORTB and 1101 1111
86.  PORTB = PORTB & ~(1 << LED_GREEN);
87.
88.  // Infinite loop
89.  while (1)
90.  {
91.    morse("D");
92.    _delay_ms(BETWEEN_DELAY);
93.    morse("E");
94.    _delay_ms(BETWEEN_DELAY);
95.    morse("2");
96.    _delay_ms(BETWEEN_DELAY);
97.  }
98.
99.  // Will never reach this
100.       return 0;
101.     }
102.
103.     /* Interrupt routines ----------------------------------------
    ----*/
104.
```