**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. /\* Defines -----------------------------------------------------------\*/
9. #define LED\_GREEN PB5   // AVR pin where green LED is connected
10. #define SHORT\_DELAY 300 // Delay in miliseconds
11. #define LONG\_DELAY 600
12. #define BETWEEN\_DELAY 300
13. #ifndef F\_CPU
14. #define F\_CPU 16000000 // CPU frequency in Hz required for delay func
15. #endif
16. /\* Includes ----------------------------------------------------------\*/
17. #include <util/delay.h> // Functions for busy-wait delay loops
18. #include <avr/io.h>     // AVR device-specific IO definitions
19. void morse(String letter)
20. {
21. if (letter == "D")
22. {
23. PORTB = PORTB ^ (1 << LED\_GREEN);
24. \_delay\_ms(LONG\_DELAY);
25. PORTB = PORTB ^ (1 << LED\_GREEN);
26. \_delay\_ms(LONG\_DELAY);
27. PORTB = PORTB ^ (1 << LED\_GREEN);
28. \_delay\_ms(SHORT\_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. }
36. else if (letter == "E")
37. {
38. PORTB = PORTB ^ (1 << LED\_GREEN);
39. \_delay\_ms(SHORT\_DELAY);
40. PORTB = PORTB ^ (1 << LED\_GREEN);
41. \_delay\_ms(SHORT\_DELAY);
42. }
43. else if (letter == "2")
44. {
45. PORTB = PORTB ^ (1 << LED\_GREEN);
46. \_delay\_ms(SHORT\_DELAY);
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(LONG\_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. }
66. }
67. /\* Variables ---------------------------------------------------------\*/
68. /\* Function prototypes -----------------------------------------------\*/
69. /\* Functions ---------------------------------------------------------\*/
70. /\*\*
71. \* Toggle one LED and use the function from the delay library.
72. \*/
73. int main(void)
74. {
75. // Set pin as output in Data Direction Register
76. // DDRB = DDRB or 0010 0000
77. DDRB = DDRB | (1 << LED\_GREEN);
78. // Set pin LOW in Data Register (LED off)
79. // PORTB = PORTB and 1101 1111
80. PORTB = PORTB & ~(1 << LED\_GREEN);
81. // Infinite loop
82. while (1)
83. {
84. morse("D");
85. \_delay\_ms(BETWEEN\_DELAY);
86. morse("E");
87. \_delay\_ms(BETWEEN\_DELAY);
88. morse("2");
89. \_delay\_ms(BETWEEN\_DELAY);
90. }
91. // Will never reach this
92. return 0;
93. }
94. /\* Interrupt routines ------------------------------------------------\*/