

Lab 1: Pavlo Shelemba

My `Digital-electronics-2` GitHub repository:

<https://github.com/xshele01/Digital-electronics-2>

Blink example

1. C programming language contains following binary operators:

Opetaor	Symbol	Form	Operation
bitwise OR	<code> </code>	<code>x y</code>	each bit in x OR each bit in y
bitwise AND	<code>&</code>	<code>x & y</code>	each bit in x AND each bit in y
bitwise XOR	<code>^</code>	<code>x ^ y</code>	each bit in x XOR each bit in y
bitwise NOT	<code>~</code>	<code>~x</code>	all bits in x flipped
left shift	<code><<</code>	<code>x << y</code>	all bits in x shifted left y bits
right shift	<code>>></code>	<code>x >> y</code>	all bits in x shifted right y bits

2. Truth table for operators `|`, `&`, `^`, `~`:

b	a	b or a	b and a	b xor a	not b
0	0	0	0	0	1
0	1	1	0	1	1
1	0	1	0	1	0
1	1	1	1	0	0

Morse code

1. Listing of C code, which repeats one "dot" and one "dash" (letter `A`) on a LED:

```

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)
    {
        // Pause several milliseconds
        _delay_ms(SHORT_DELAY);

        // Dot
        PORTB = PORTB | (1<<LED_GREEN);

        _delay_ms(SHORT_DELAY * 2);

        PORTB = PORTB & ~(1<<LED_GREEN);

        _delay_ms(SHORT_DELAY);

        // Dash
        PORTB = PORTB | (1<<LED_GREEN);

        _delay_ms(SHORT_DELAY * 4);

        PORTB = PORTB & ~(1<<LED_GREEN);
    }
}

```

2. Scheme of Morse code application (connection of AVR device, LED, resistor, and supply voltage):

