

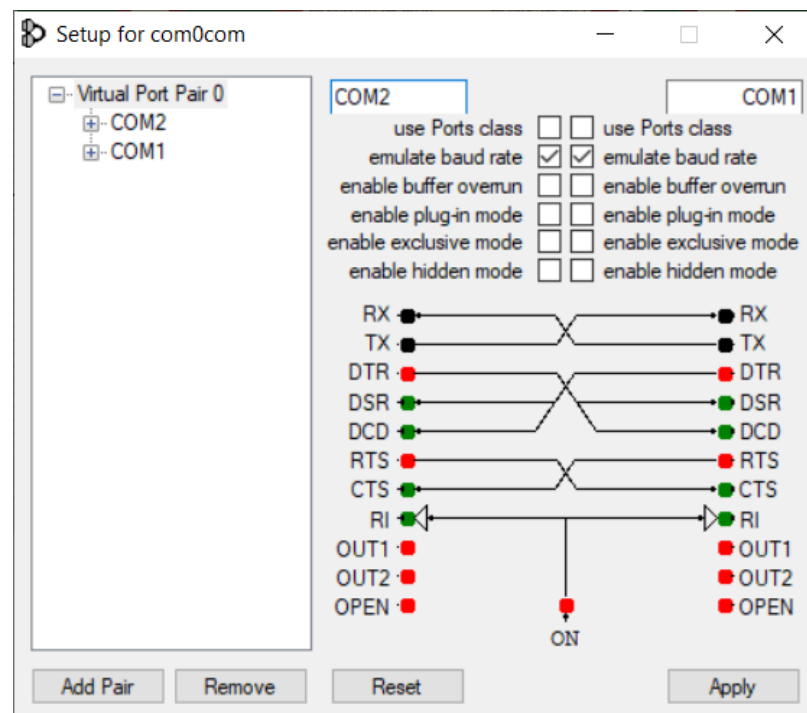
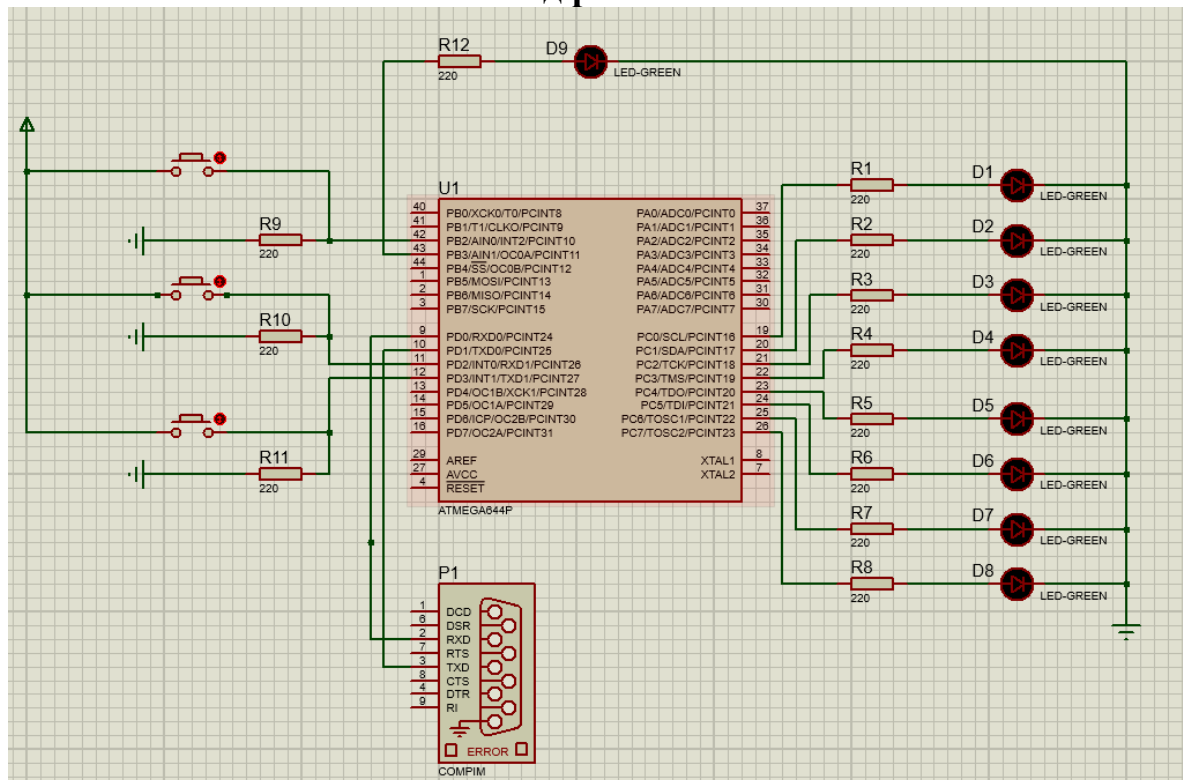
Орел, 2020 г

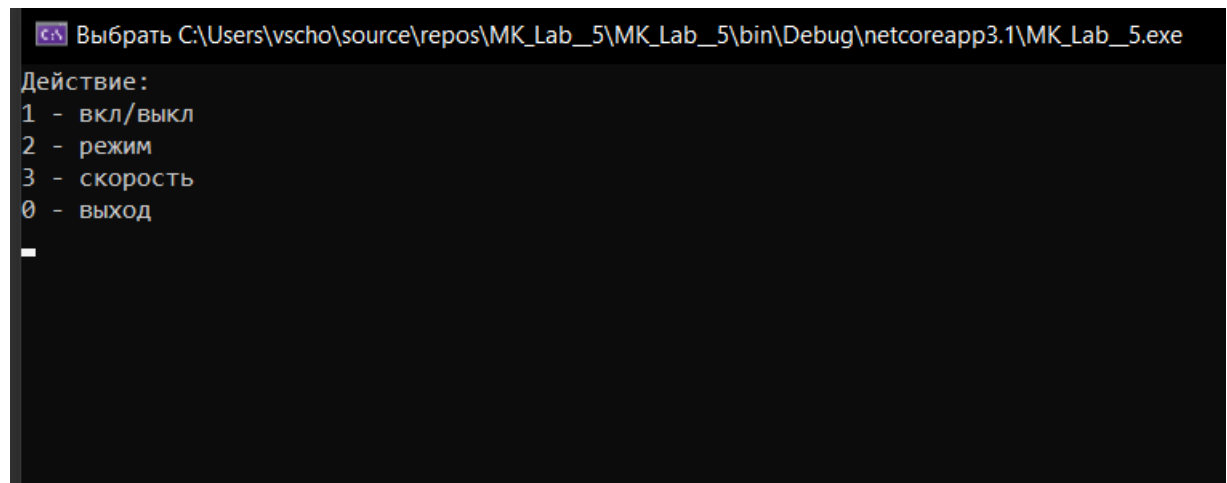
Задание на лабораторную работу

Разработать программу для микроконтроллера на языке Си для управления гирляндой (из лабораторной работы 4) с терминала.

Разработать терминал, позволяющий управлять гирляндой.

Ход работы





Код

Код микроконтроллера

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
```

```
int mainBtn = 0;
int modeOperating = 0;
int modeSpeed = 0;
int up = 1;
```

```
char i_LED = 0;
```

```
void ChangeSpeedMode()
{
    TCCR0B = 0x00;
    switch (modeSpeed)
    {
        case 0: TCCR0B = 0x05; break;
        case 1: TCCR0B = 0x04; break;
        case 2: TCCR0B = 0x03; break;
    }
}
```

```
void TimersSettings()
{
    //0x05 == 0101b
    switch (modeOperating)
    {
        case 0:
        {
            DDRB = 0;
            DDRC = 0xFF;
```

```

        TCCR0B = 0x00;
        TIMSK0 = 1 << TOIE0;
        ChangeSpeedMode();
        break;
    }
    case 1:
    {
        DDRB = 0;
        DDRC = 0xFF;
        TCCR0B = 0x00;
        TIMSK0 = 1 << OCIE0A;
        OCR0A = 200;
        PORTC = 0b11110000;
        ChangeSpeedMode();
        break;
    }
    case 2:
    {
        DDRC = 0;
        DDRB = 1 << 3;
        OCR0A = 200;
        TCCR0A = 1 << COM0A0;
        ChangeSpeedMode();
        break;
    }
}

if (!mainBtn)
{
    TCCR0B = 0x00;
    PORTC = 0;
    PORTB = 0;
}

}

void INTSettings()
{
    // Настройка прерываний INT0, INT1, INT2 (передний фронт)
    EICRA = 0b00111111;
    // разрешение прерываний INT0, INT1, INT2
    EIMSK = 0b000000111;

    sei();
}

```

```

ISR(USART0_RX_vect)
{
    char receivedSymbol = UDR0;

    switch (receivedSymbol)
    {
        case '1':
        {
            PORTC = 0xFF;
            mainBtn = (mainBtn == 0) ? 1 : 0;
            TimersSettings();
            break;
        }
        case '2':
        {
            if (modeOperating == 2)
            {
                modeOperating = 0;
            }
            else
            {
                modeOperating++;
            }

            TimersSettings();
            break;
        }
        case '3':
        {
            if (up)
            {
                modeSpeed++;
                up = (modeSpeed == 2) ? 0 : 1;
            }
            else
            {
                modeSpeed--;
                up = (modeSpeed == 0) ? 1 : 0;
            }

            ChangeSpeedMode();
            break;
        }
    }
}

```

```

// change modeoperating
ISR (INT0_vect)
{
    cli();
    if (modeOperating == 2)
    {
        modeOperating = 0;
    }
    else
    {
        modeOperating++;
    }

    TimersSettings();

    sei();
}

// change modespeed
ISR (INT1_vect)
{
    cli();

    if (up)
    {
        modeSpeed++;
        up = (modeSpeed == 2) ? 0 : 1;
    }
    else
    {
        modeSpeed--;
        up = (modeSpeed == 0) ? 1 : 0;
    }

    ChangeSpeedMode();

    sei();
}

// on/off btn
ISR (INT2_vect)
{
    cli();
    mainBtn = (mainBtn == 0) ? 1 : 0;
}

```

```

        TimersSettings();
        sei();
    }

ISR (TIMER0_OVF_vect)
{
    PORTC = 1 << i_LED;
    i_LED++;
    i_LED = i_LED % 8;
}

ISR (TIMER0_COMPA_vect)
{
    PORTC = ~PORTC;
    TCNT0 = 0;
}

int main(void)
{
    DDRC = 0xFF;
    DDRB = 0x00;

    INTSettings();

    UBRR0 = 0x06;
    UCSRB = 1 << RXCIE0 | 1 << RXEN0; // Разрешение прерываний по
передаче и Вкл. прием по USART

    while (1)    {
    }
}

```

Код терминала

```

using System;
using System.IO.Ports;

namespace MK_Lab__5
{
    class Program
    {
        static void Main(string[] args)
        {
            SerialPort _serialPort;

            try
            {

```

```

        // настройки порта
        _serialPort = new SerialPort("COM1", 9600, Parity.None, 8, Stop-
Bits.One);
        _serialPort.Handshake = Handshake.None;
        _serialPort.WriteTimeout = 1000;
        _serialPort.Open();
    }
    catch (Exception e)
    {
        Console.WriteLine("ERROR: невозможно открыть порт:" +
e.ToString());
        return;
    }

    bool flag = true;

    while (flag)
    {
        Console.WriteLine("Действие:");
        Console.WriteLine("1 - вкл/выкл");
        Console.WriteLine("2 - режим");
        Console.WriteLine("3 - скорость");
        Console.WriteLine("0 - выход");

        string operand = Console.ReadLine();

        switch (operand)
        {
            case "1":
            {
                try
                {
                    if (!_serialPort.IsOpen)
                        _serialPort.Open();
                    _serialPort.Write("1");
                }
                catch (Exception ex)
                {
                    Console.WriteLine("Error opening/writing to serial port :: "
+ ex.Message, "Error!");
                }
                break;
            }
            case "2":
            {

```



```

        try
        {
            if (!(_serialPort.IsOpen))
                _serialPort.Open();
            _serialPort.Write("2");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error opening/writing to serial port :: "
                               + ex.Message, "Error!");
        }
        break;
    }
    case "3":
    {
        try
        {
            if (!(_serialPort.IsOpen))
                _serialPort.Open();
            _serialPort.Write("3");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error opening/writing to serial port :: "
                               + ex.Message, "Error!");
        }
        break;
    }
    case "0":
    {
        flag = false;
        return;
    }
}

_serialPort.Close();
}
}
}

```