

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе №5

по дисциплине: «Программирование микроконтроллеров»

вариант 1

Выполнил: Евдокимов Н.А.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

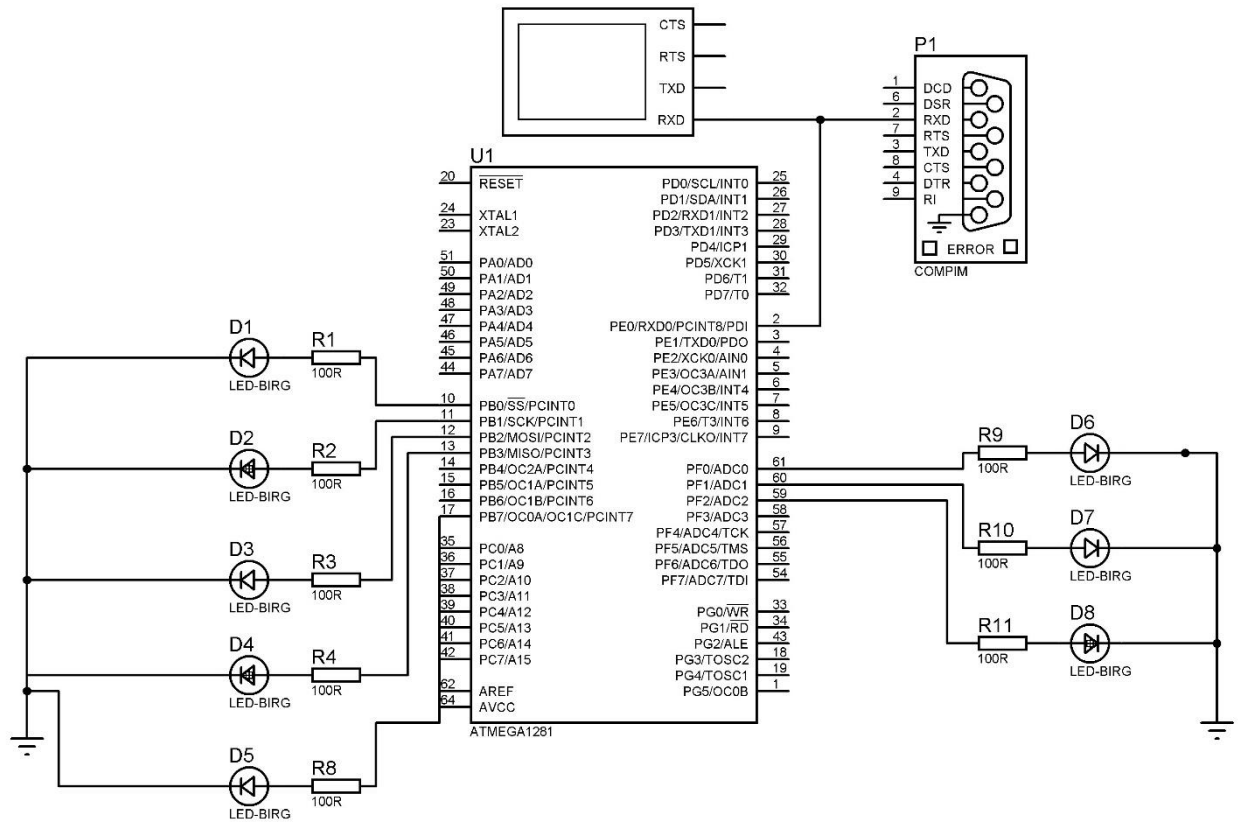
Проверили: Захарова О.В., Раков В.И.

Отметка о зачете:

Дата: «\_\_\_» \_\_\_\_\_ 2020 г.

Орёл, 2020

## Схема



## Скриншот терминала

```
0) Change mode
1) Turn on or off
2) Change speed
0
Bytes wrote: 1
```

## Код микроконтроллера

```
#include <avr/io.h>
#include <avr/delay.h>
#include <stdbool.h>
#include <avr/interrupt.h>

#define F_CPU 1000000UL
#define BITRATE 9600
#define BAUD F_CPU / (16 * BITRATE) - 1

bool shouldRun = true;
uint8_t mode = 0;
uint8_t delay = 50;
char comMsg;

void increaseDelay() {
    if (delay >= 250) {
        delay = 0;
    }

    delay += 50;
    OCR0A = delay;
}

void makeTick() {
    if (!shouldRun) {
        PORTB = 0;
        return;
    }

    switch(mode) {
        case 0: {
            PORTB = ~PORTB;
            break;
        }

        case 1: {
            PORTB = 0xff;
            break;
        }

        case 2: {
            PORTB = ~PORTB;
            break;
        }
    }
}

void changeMode() {
    mode = (mode + 1) % 3;

    switch(mode) {
        case 0: {
            TCCR0A = 0x02; // CTC
            TCCR0B = 0x05;
            PORTB = 0xff; // Blink
            PORTF = 0b00000001;
            break;
        }

        case 1: {
```

```

        TCCR0A = 0x40; // Normal
        TCCR0B = 0x05;
        PORTB = 0xff; // Constant light
        PORTF = 0b00000010;
        break;
    }

    case 2: { // Moving light
        TCCR0B = (1 << WGM02) | 0x05;
        TCCR0A = (1 << WGM01) | (1 << WGM00) | (1 << COM0A0); // Fast PWM
        PORTB = 0b10101010;
        PORTF = 0b00000100;
        break;
    }
}

}

char udr;
ISR(USART0_RX_vect) {
    cli();

    udr = UDR0;
    switch(udr) {
        case 'm': {
            changeMode();
            break;
        }

        case 'r': {
            shouldRun = !shouldRun;
            break;
        }

        case 's': {
            increaseDelay();
            break;
        }

        default: {}
    }

    sei();
}

int main(void) {
    DDRA = 0;
    DDRB = 0xff;
    DDRF = 0xff;
    EICRA = 0b00111111;
    EIMSK = 0b00000111;
    PORTF = 0b00000001;

    OCR0A = 50;
    TCCR0A = 0x02; // CTC
    TCCR0B = 0x05;

    UCSRB = 0b10010000;
    UBRR0L = 0x06;

    sei();

    while (1) {
        if (TCNT0 == 0) {
            makeTick();
        }
    }
}

```

```

    }
}

```

## Код терминала

```

#include <iostream>
#include "Serial.h"

using namespace std;

int main() {
    tstring portName(TEXT("COM1"));
    Serial port(portName, 9600);

    char messages[] = {'m', 'r', 's'};

    cout << "0) Change mode" << endl;
    cout << "1) Turn on or off" << endl;
    cout << "2) Change speed" << endl;

    int userInput;
    while(true) {
        cin >> userInput;
        if (userInput < 0 || userInput > 2) {
            cout << "Incorrect value" << endl;
            continue;
        }

        char* msg = (char*) &messages[userInput];

        int bytesWrote = port.write(msg, 1);
        cout << "Bytes wrote: " << bytesWrote << endl;
        port.flush();
        Sleep(1000);
    }

    return 0;
}

#include <iostream>
using namespace std;

#include "Serial.h"

Serial::~Serial()
{
    CloseHandle(commHandle);
}

Serial::Serial(tstring &commPortName, int bitRate)
{
    commHandle = CreateFile(commPortName.c_str(),
        GENERIC_READ|GENERIC_WRITE, 0, NULL, OPEN_EXISTING,
        0, NULL);

    if(commHandle == INVALID_HANDLE_VALUE)
    {
        throw("ERROR: Could not open com port");
    }
    else
    {
        // set timeouts
        COMMTIMEOUTS cto = { MAXDWORD, 0, 0, 0, 0};
        DCB dcb;
    }
}

```

```

        if(!SetCommTimeouts(commHandle,&cto))
        {
            throw("ERROR: Could not set com port time-outs");
        }

        // set DCB
        memset(&dcb,0,sizeof(dcb));
        dcb.DCBlength = sizeof(dcb);
        dcb.BaudRate = bitRate;
        dcb.fBinary = 1;
        dcb.fDtrControl = DTR_CONTROL_ENABLE;
        dcb.fRtsControl = RTS_CONTROL_ENABLE;

        dcb.Parity = NOPARITY;
        dcb.StopBits = ONESTOPBIT;
        dcb.ByteSize = 8;

        if(!SetCommState(commHandle,&dcb))
        {
            throw("ERROR: Could not set com port parameters");
        }
    }
}

int Serial::write(const char *buffer)
{
    DWORD numWritten;
    WriteFile(commHandle, buffer, strlen(buffer), &numWritten, NULL);

    return numWritten;
}

int Serial::write(const char *buffer, int buffLen)
{
    DWORD numWritten;
    WriteFile(commHandle, buffer, buffLen, &numWritten, NULL);

    return numWritten;
}

int Serial::read(char *buffer, int buffLen, bool nullTerminate)
{
    DWORD numRead;
    if(nullTerminate)
    {
        --buffLen;
    }

    BOOL ret = ReadFile(commHandle, buffer, buffLen, &numRead, NULL);

    if(!ret)
    {
        return 0;
    }

    if(nullTerminate)
    {
        buffer[numRead] = '\\0';
    }

    return numRead;
}

#define FLUSH_BUFFSIZE 10

```

```
void Serial::flush()
{
    char buffer[FLUSH_BUFFSIZE];
    int numBytes = read(buffer, FLUSH_BUFFSIZE, false);
    while(numBytes != 0)
    {
        numBytes = read(buffer, FLUSH_BUFFSIZE, false);
    }
}
```