

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 №7-8

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

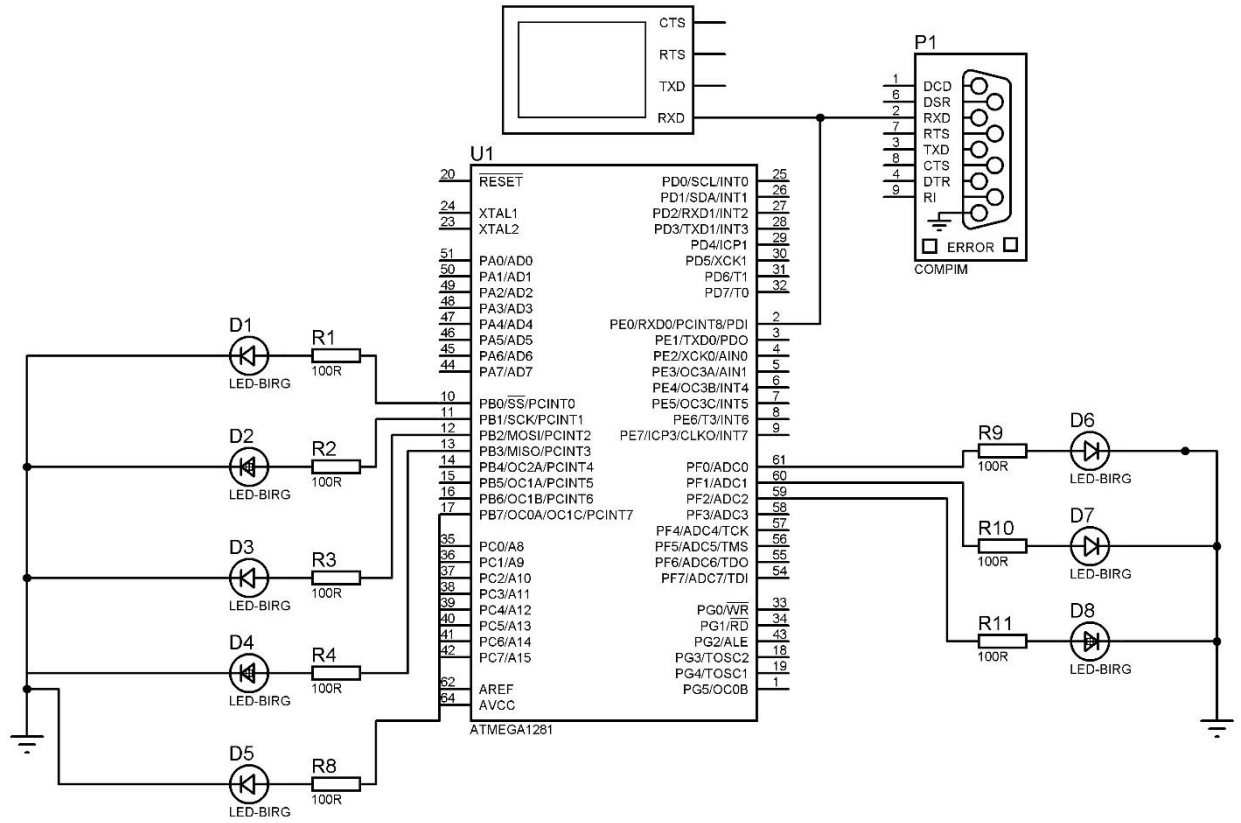
Проверили: Захарова О.В., Раков В.И.

Отметка о зачете:

Дата: «\_\_\_» \_\_\_\_\_ 2020 г.

Орёл, 2020

# Cxema



## Код

```
;
; lab7_8.asm
;
; Created: 27.04.2020 18:07:00
; Author : Nikita
;

.include "m1281def.inc"

.equ CRYSTAL_FREQ = 1000000
.equ BAUD = 9600

.def shouldRunReg = r16
.def modeReg = r17
.def tmp = r18
.def delayReg = r19
.def lightStateReg = r20
.def udrReg = r21

.cseg
.org 0 jmp START
.org $0032 jmp USART0_RX

START:
    rcall init
    MAIN_LOOP:
        in tmp, TCNT0
        cpi tmp, 0
        breq main_loop_tick

        jmp MAIN_LOOP

    main_loop_tick:
        call make_tick

    rjmp MAIN_LOOP

init:
    ser tmp
    out DDRB, tmp
    out DDRF, tmp

    ldi tmp, 0b00111111
    sts EICRA, tmp

    ldi tmp, 0b00000111
    out EIMSK, tmp

    ldi tmp, 1
    out PORTF, tmp

    ldi delayReg, 250
    out OCR0A, delayReg

    ldi tmp, 0x02
    out TCCR0A, tmp

    ldi tmp, 0x05
    out TCCR0B, tmp
```

```

    ldi tmp, 0b10010000
    sts UCSR0B, tmp

    ldi tmp, 6
    sts UBRR0L, tmp

    ldi shouldRunReg, 1
    ldi lightStateReg, 0xff
    out PORTB, lightStateReg

    sei
ret

make_tick:
    cpi shouldRunReg, 1
    breq tick

    ;; Shut down
    clr tmp
    out PORTB, tmp
    ret

    ;; Make tick
tick:
    cpi modeReg, 0
    breq mode_0

    cpi modeReg, 1
    breq mode_1

    cpi modeReg, 2
    breq mode_2

    mode_0:
        com lightStateReg
        out PORTB, lightStateReg
        ret

    mode_1:
        ser lightStateReg
        out PORTB, lightStateReg
        ret

    mode_2:
        com lightStateReg
        out PORTB, lightStateReg
        ret

ret

switch_light:
    cpi shouldRunReg, 0
    breq turn_on

    ;; If we here, then light is turned on. Need to turn off
    ldi shouldRunReg, 0
    ret

    turn_on:
        ldi shouldRunReg, 1
ret

change_mode:
    inc modeReg

    cpi modeReg, 3

```

```

    breq reset_mode

    jmp set_mode
reset_mode:
    clr modeReg

set_mode:
    cpi modeReg, 0
    breq set_mode_0

    cpi modeReg, 1
    breq set_mode_1

    cpi modeReg, 2
    breq set_mode_2

set_mode_0:
    ldi tmp, 2
    out TCCR0A, tmp

    ldi tmp, 5
    out TCCR0B, tmp

    ldi lightStateReg, 0xff
    out PORTB, lightStateReg

    ldi tmp, 1
    out PORTF, tmp
    ret

set_mode_1:
    ldi tmp, 0x40
    out TCCR0A, tmp

    ldi tmp, 5
    out TCCR0B, tmp

    ldi lightStateReg, 0xff
    out PORTB, lightStateReg

    ldi tmp, 2
    out PORTF, tmp
    ret

set_mode_2:
    ldi tmp, 0b01000011
    out TCCR0A, tmp

    ldi tmp, 0b00001101
    out TCCR0B, tmp

    ldi lightStateReg, 0b10101010
    out PORTB, lightStateReg

    ldi tmp, 4
    out PORTF, tmp
    ret

ret

change_delay:
    cpi delayReg, 250
    breq reset_delay

    jmp increase_delay
reset_delay:

```

```

        ldi delayReg, 0

increase_delay:
        ldi tmp, 50
        add delayReg, tmp
        out OCR0A, delayReg
ret

USART0_RX:
        cli

        lds udrReg, UDR0

        cpi udrReg, 0x6d
        breq usart_change_mode

        cpi udrReg, 0x72
        breq usart_switch_light

        cpi udrReg, 0x73
        breq usart_change_delay

usart_change_mode:
        call change_mode
        sei
        reti

usart_switch_light:
        call switch_light
        sei
        reti

usart_change_delay:
        call change_delay
        sei
        reti

        sei
reti

```