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

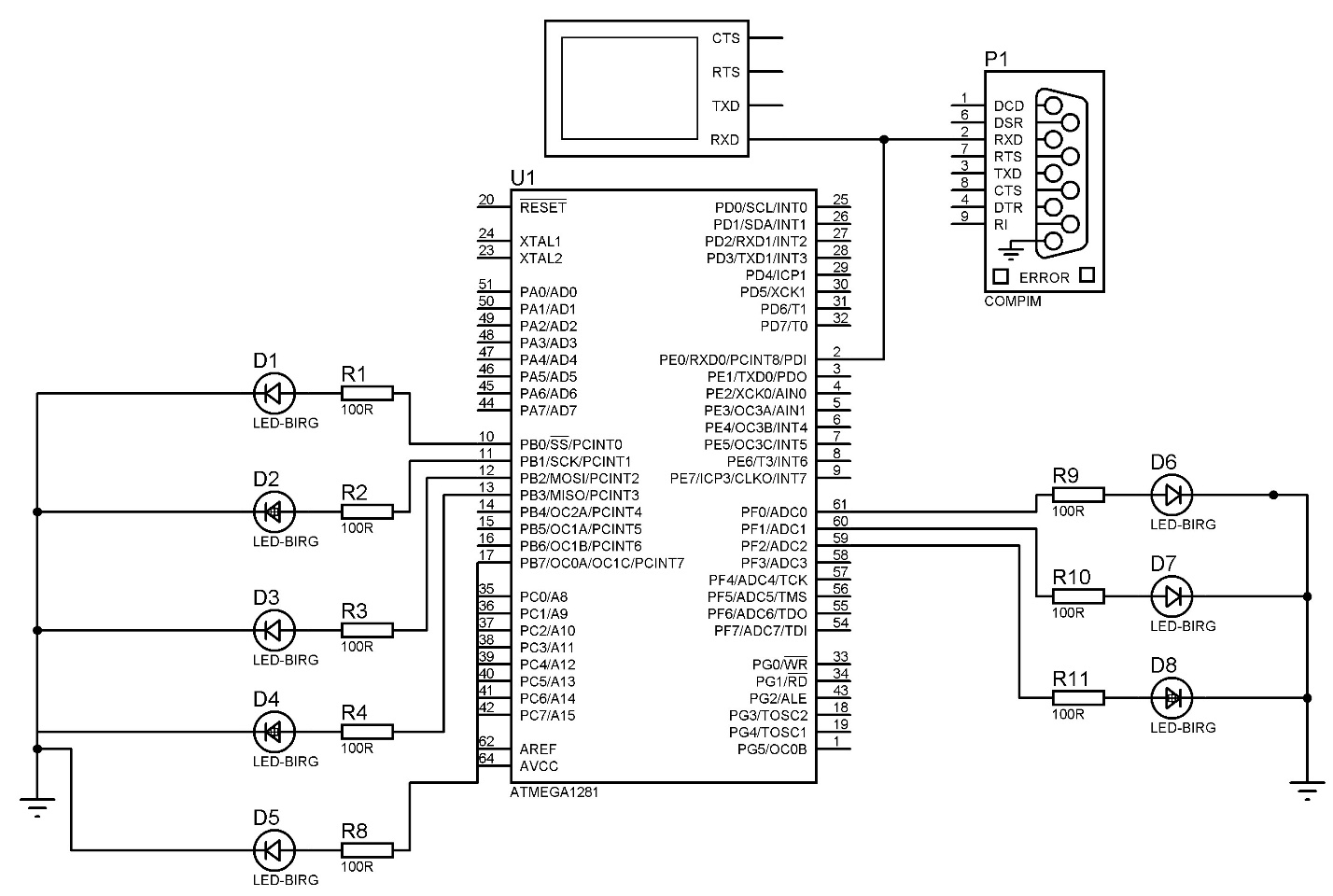
Проверили: Захарова О.В., Раков В.И.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Орёл, 2020

**Схема**



**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