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 № 4

на тему: «Разработка программы на языке Си, работающей с внешними прерываниями и таймерами»

по дисциплине: «Программирование микроконтроллеров»

Вариант № 18

Выполнил: Шорин В.Д. Шифр: 171406

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.

Орел, 2020 г

**Задание на лабораторную работу**

Задание из лабораторной работы 1-2 выполнить с использованием внешних прерываний и таймеров.

При выполнении лабораторной работы режимы работы гирлянды реализовать различными режимами работы таймера/счетчика.

Один из режимов работы гирлянды реализовать с использованием схемы вывода сигнала совпадения.

**Ход работы**

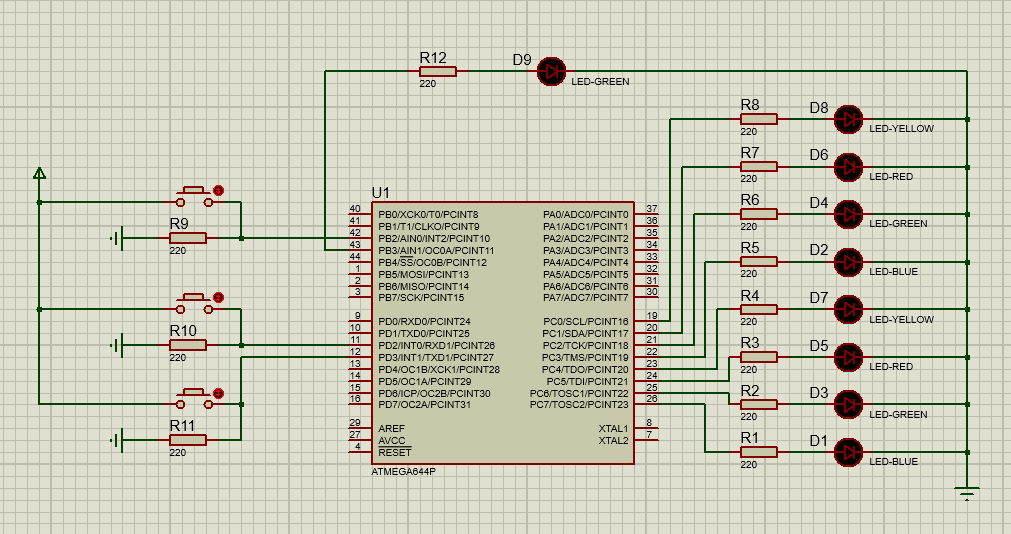
****

Рисунок 1 – Схема микроконтроллера

**Код**

#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()

{

//0х05 == 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 = 0b00000111;

sei();

}

// 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();

while (1) { }

}