**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Систем автоматического управления**

**отчет**

**по лабораторной работе №2**

**по дисциплине «Микропроцессорная техника в мехатронике и робототехнике»**

**ТЕМА: «РАБОТА С ТРЁХЦВЕТНЫМ СВЕТОДИОДОМ (ШИМ)»**

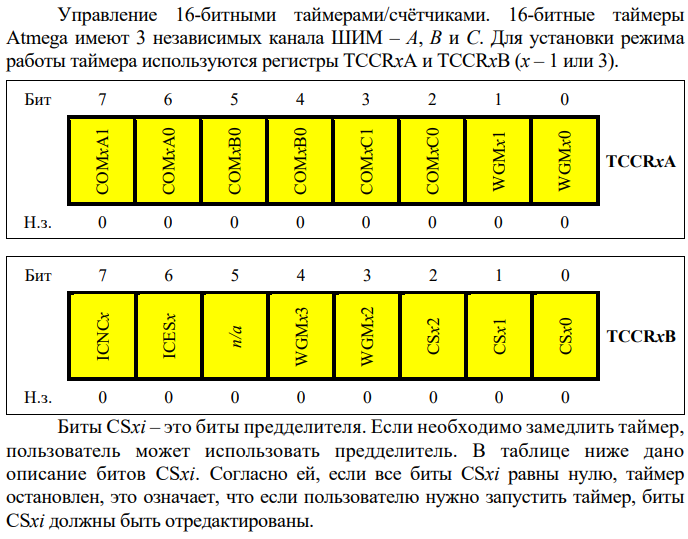
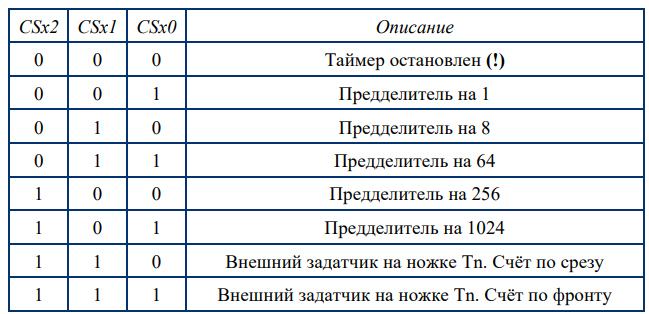
|  |  |  |
| --- | --- | --- |
| Студенты гр. 1492 |  | Старцев Н.А. |
| Преподаватель |  | Илатовская Е.В. |

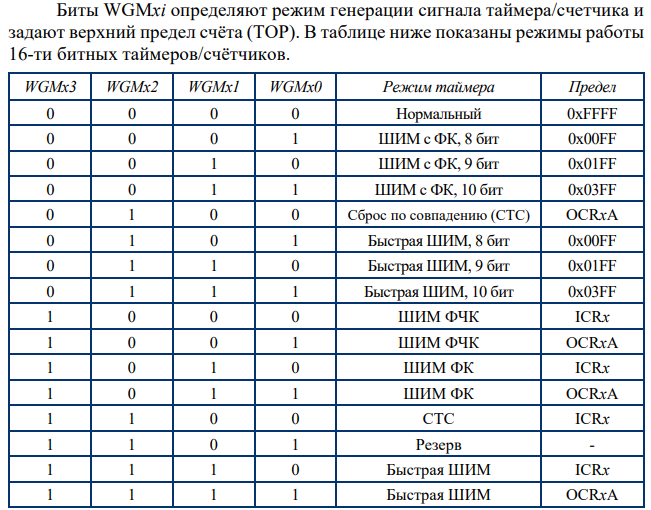
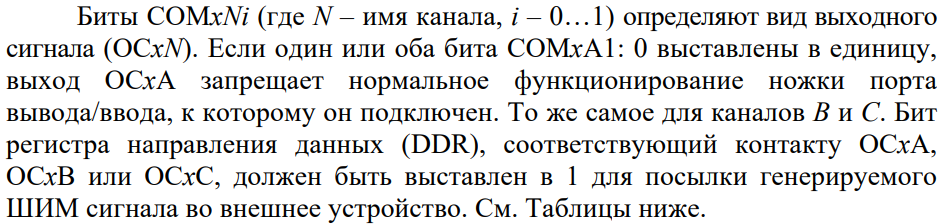
Санкт-Петербург

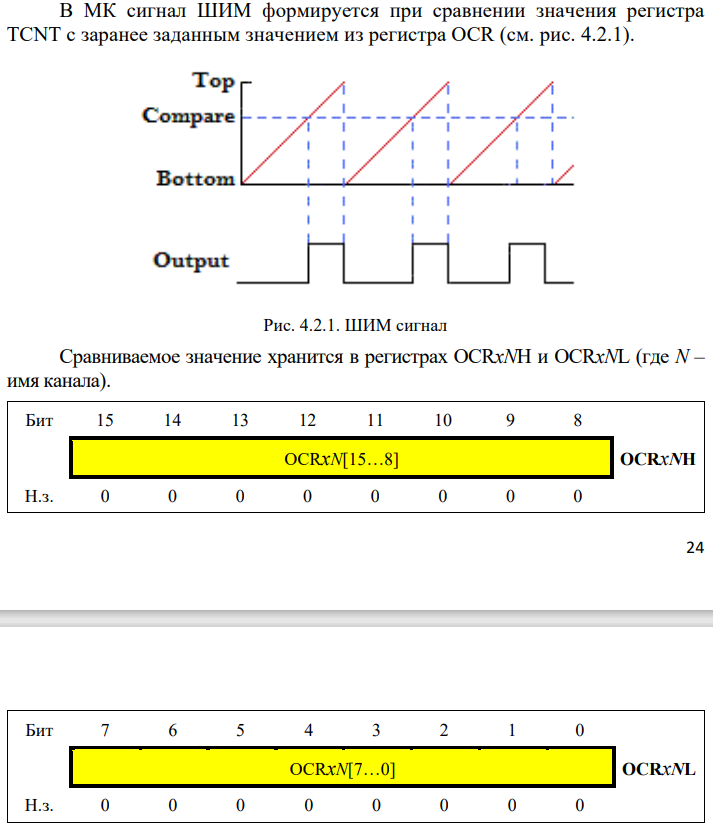
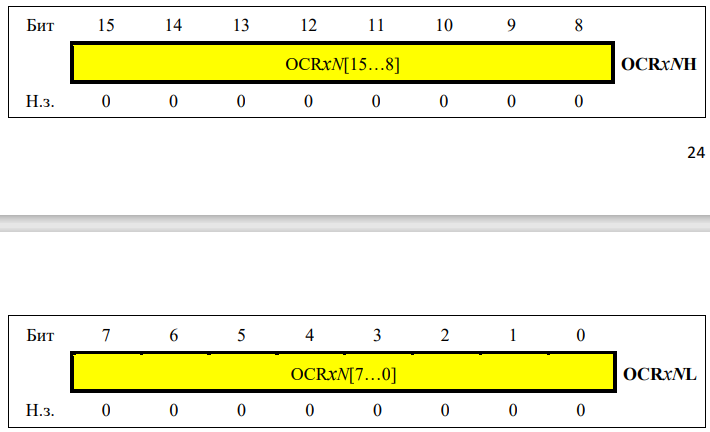
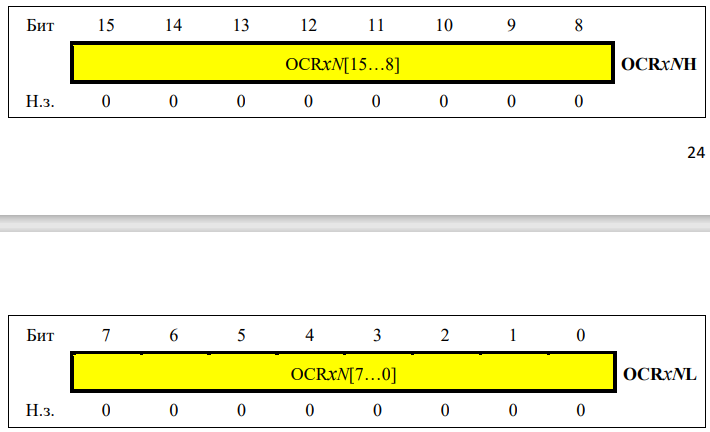
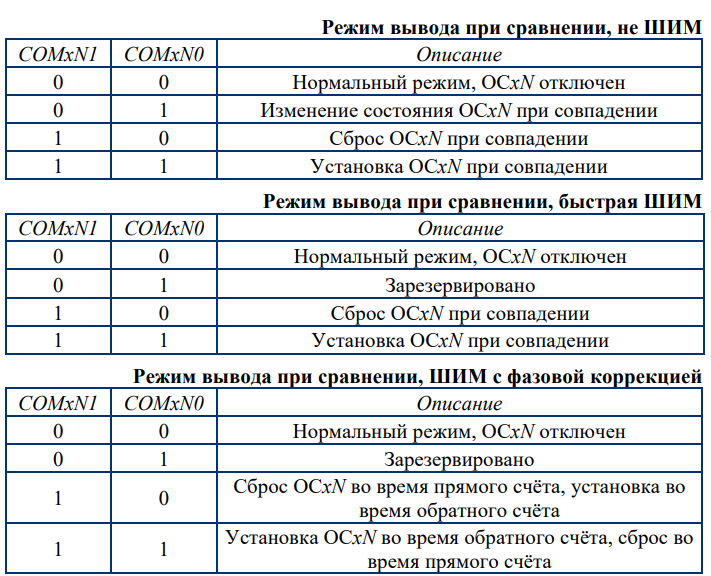
2024

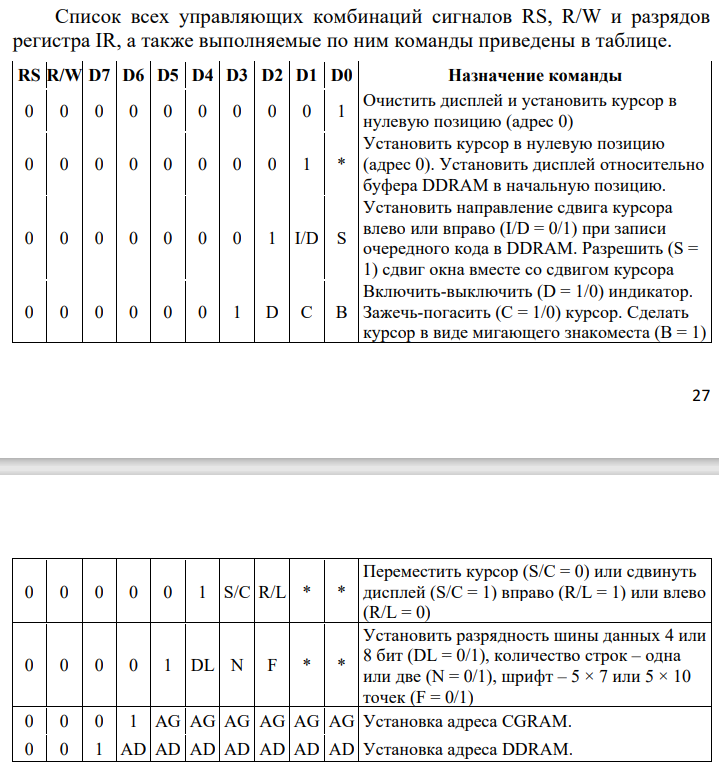
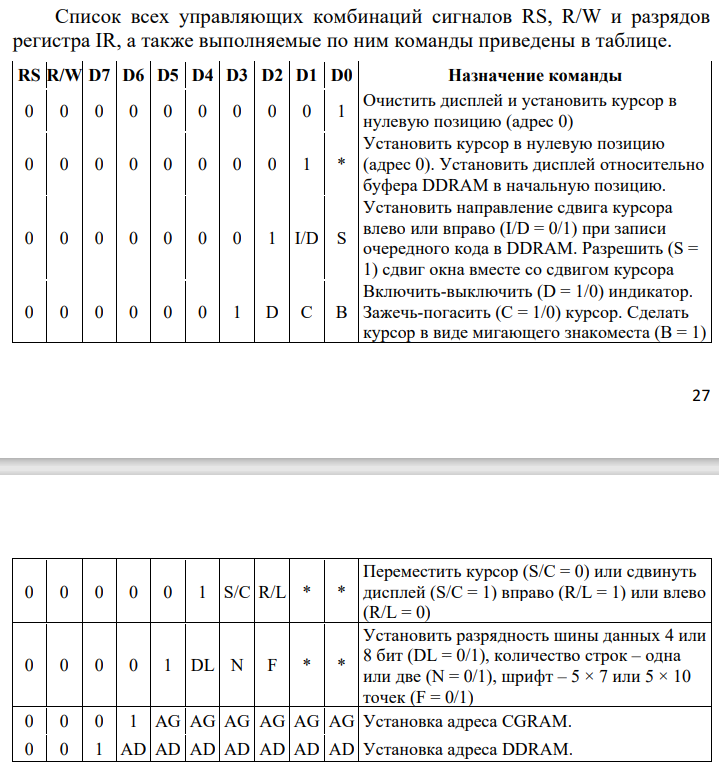
**Цель работы:** получение навыков работы с таймер счетчиками, использование таймеров для шим управления яркостью и цветом трехцветного светодиода

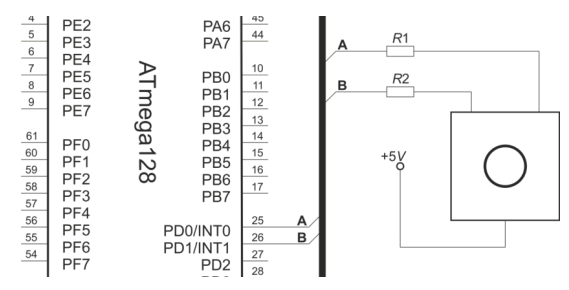
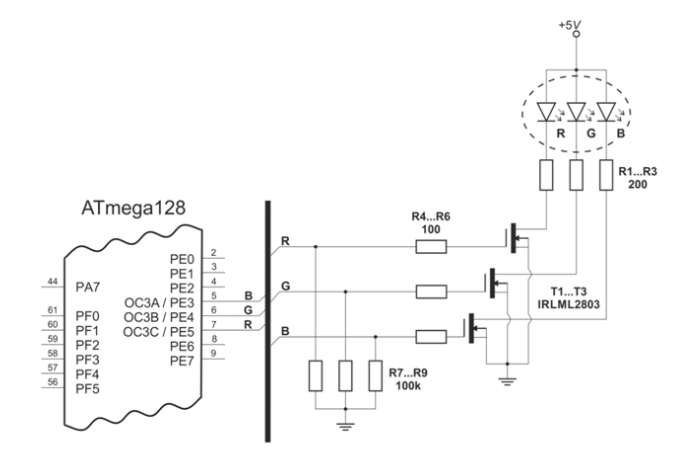
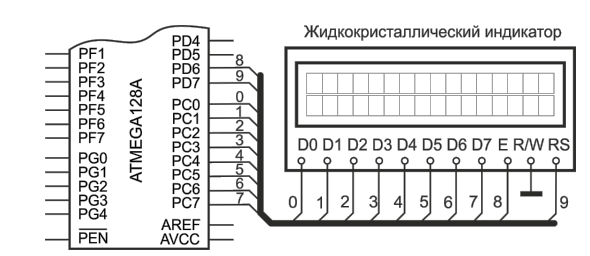
**Основные сведения**

****

****

****

****

**Схема подключения**

**Результаты работы**

написать программу управления цветом и яркостью светодиода с помощью механического энкодера. В таблице заданий 0....255 означает, что при вращении энкодера по часовой стрелке яркость данного цвета увеличивается, а против часовой стрелки – уменьшается. На ЖКИ выводятся значения яркости (0…255) для каждого из трёх каналов RGB диода в виде: R: xxx G: xxx B: xxx

Код программы:

#include <avr/io.h>

// #define F\_CPU 11059200

#include <util/delay.h> //для использования пауз

#include <avr/interrupt.h>

#define E 6

#define RS 7

uint8\_t my\_color = 0;

uint8\_t myColorRed = 0;

uint8\_t myColorGreen = 0;

uint8\_t myColorBlue = 0;

void lcdCmd(uint8\_t cmd)

{                                   // посыл команды на экран

    DDRC = 0xFF;                    // все разряды PORTC на выход

    DDRD |= ((1 << E) | (1 << RS)); // разряды PORTD на выход

    PORTD &= ~(1 << RS);            // выбор регистра команд RS=0

    PORTC = cmd;                    // записать команду в порт PORTC

    PORTD |= (1 << E);              // \ сформировать на

    \_delay\_us(5);                   // | выводе E строб 1-0

    PORTD &= ~(1 << E);             // / передачи команды

    \_delay\_ms(10);                  // задержка для завершения записи

}

void lcdInit(void)

{                                   // инициализация (ВКЛ) экрана

    DDRC = 0xFF;                    // все разряды PORTC на выход

    DDRD |= ((1 << E) | (1 << RS)); // разряды PORTD на выход

    \_delay\_ms(100);                 // задержка для установления питания

    lcdCmd(0x30);                   // \ вывод

    lcdCmd(0x30);                   // | трех

    lcdCmd(0x30);                   // / команд 0x30

    lcdCmd(0x38);                   // 8 разр.шина, 2 строки, 5 × 7 точек

    lcdCmd(0x0C);                   // включить ЖКИ

    lcdCmd(0x06);                   // инкремент курсора, без сдвига экрана

    lcdCmd(0x01);                   // очистить экран, курсор в начало

}

void lcdData(uint8\_t data)

{ // посыл данных на экран

    DDRC = 0xFF;

    DDRD |= ((1 << E) | (1 << RS));

    PORTD |= (1 << RS);

    PORTC = data;

    PORTD |= (1 << E);

    \_delay\_us(5);

    PORTD &= ~(1 << E);

    \_delay\_ms(1);

}

ISR(INT0\_vect) // прерывания по инкодеру

{

    int8\_t delta = 0;

    if ((PIND & (1 << 0)) != 0)

    {

        EICRA = (1 << ISC01) | (1 << ISC21);

        if ((PIND & (1 << 1)) != 0)

            delta = 5;

        else

            delta = -5;

    }

    else

    {

        EICRA = (1 << ISC01) | (1 << ISC00) | (1 << ISC21);

        if ((PIND & (1 << 1)) != 0)

            delta = -5;

        else

            delta = 5;

    }

    switch (my\_color)

    {

    case 0:

        myColorRed += delta;

        break;

    case 1:

        myColorGreen += delta;

        break;

    case 2:

        myColorBlue += delta;

        break;

    default:

        break;

    }

}

ISR(INT2\_vect) { my\_color = (my\_color + 1) % 3; }

void my\_print(void)

{

    char c[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

    lcdCmd((1 << 7) | 2);

    lcdData(c[myColorRed / 100]);

    lcdData(c[(myColorRed / 10) % 10]);

    lcdData(c[myColorRed % 10]);

    lcdCmd((1 << 7) | 7);

    lcdData(c[myColorGreen / 100]);

    lcdData(c[(myColorGreen / 10) % 10]);

    lcdData(c[myColorGreen % 10]);

    lcdCmd((1 << 7) | 12);

    lcdData(c[myColorBlue / 100]);

    lcdData(c[(myColorBlue / 10) % 10]);

    lcdData(c[myColorBlue % 10]);

}

int main(void)

{

    sei();

    lcdInit();

    EIMSK |= (1 << INT0);

    EIMSK |= (1 << INT2);

    EICRA = (1 << ISC01) | (1 << ISC00) | (1 << ISC21);

    DDRE = (1 << 3) | (1 << 4) | (1 << 5);

    /\* Инициализация таймера №3. 8-ми битная быстрая

    ШИМ, преддедитель на 8 \*/

    TCCR3A = (1 << COM3A1) | (1 << COM3B1) | (1 << COM3C1) | (1 << WGM30);

    TCCR3B = (1 << WGM32) | (1 << CS31);

    OCR3AH = 0;

    OCR3AL = 128;

    char Start\_str[] = {'R', ':', ' ', ' ', ' ', 'G', ':', ' ', ' ', ' ', 'B', ':'};

    lcdCmd((1 << 7) | 0);

    for (uint8\_t i = 0; i < 12; i++)

    {

        lcdData(Start\_str[i]);

    }

    while (1)

    {

        my\_print();

        OCR3A = myColorBlue;

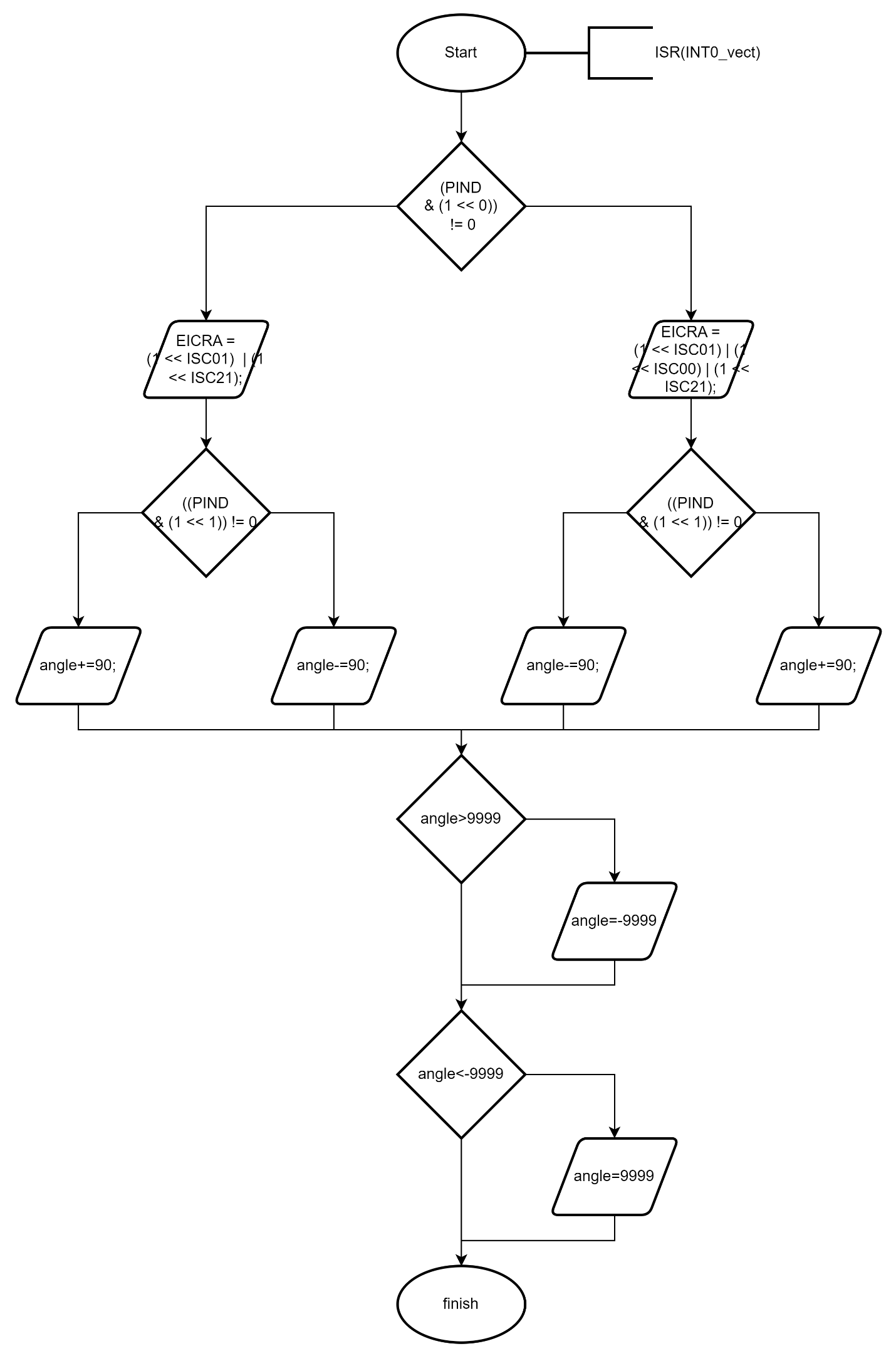
        OCR3B = myColorGreen;

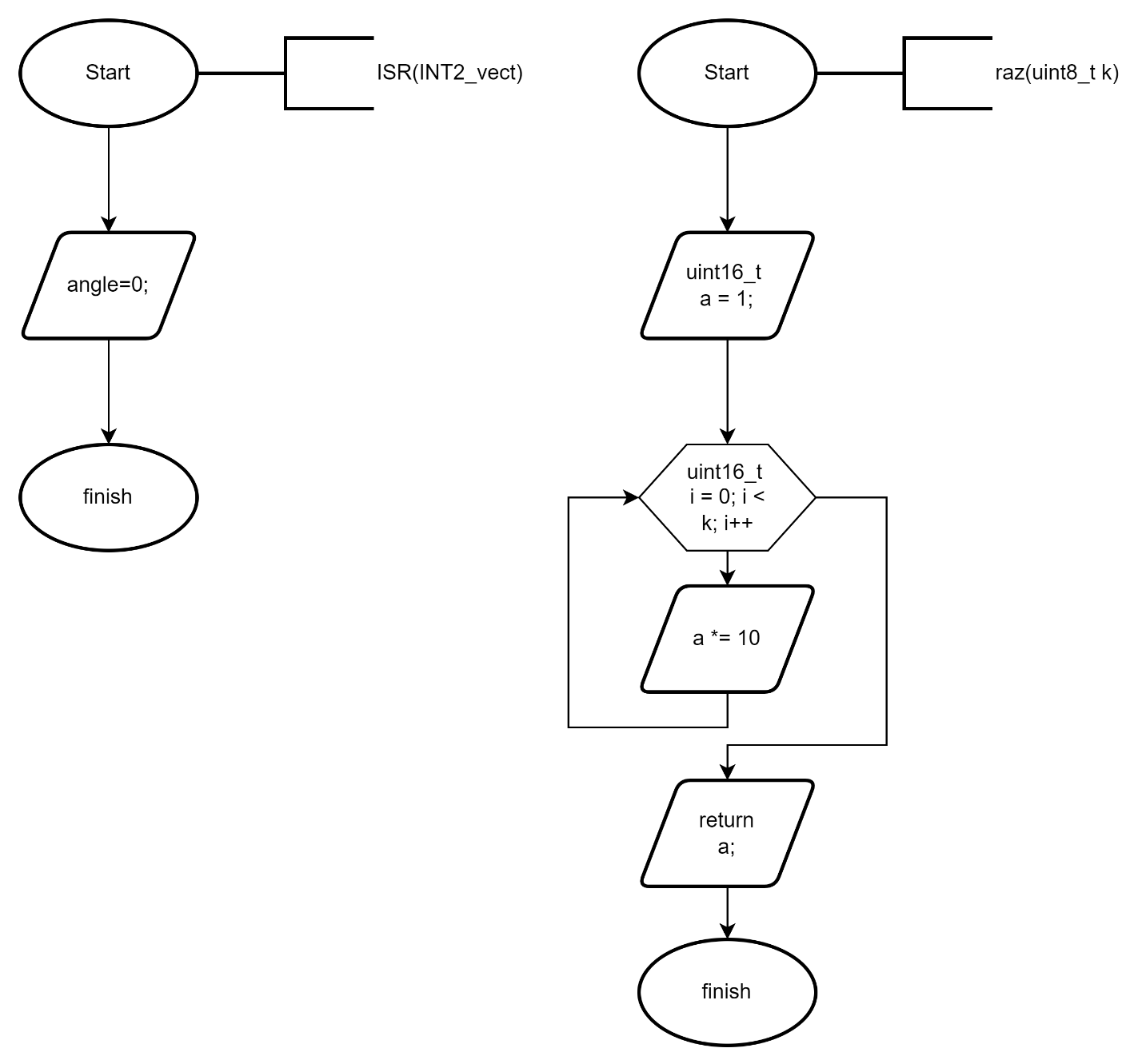
        OCR3C = myColorRed;

        \_delay\_us(100);

    }

}

****

****

**Вывод:**

В ходе лабораторной работы были освоены навыки взаимодействия с внешними прерываниями, обработка прерываний энкодера и кнопки.