**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Систем автоматического управления**

**отчет**

**по лабораторной работе №1**

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

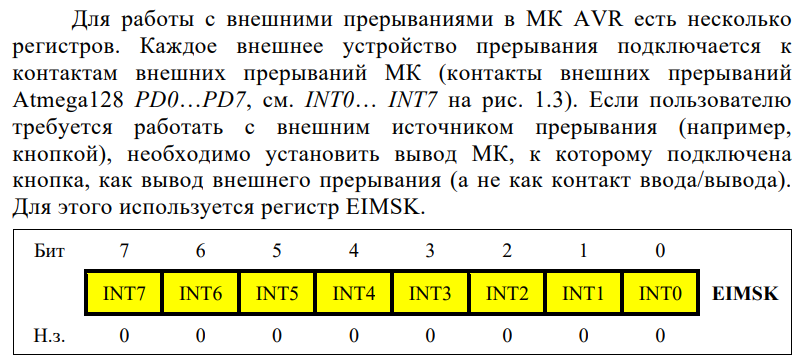
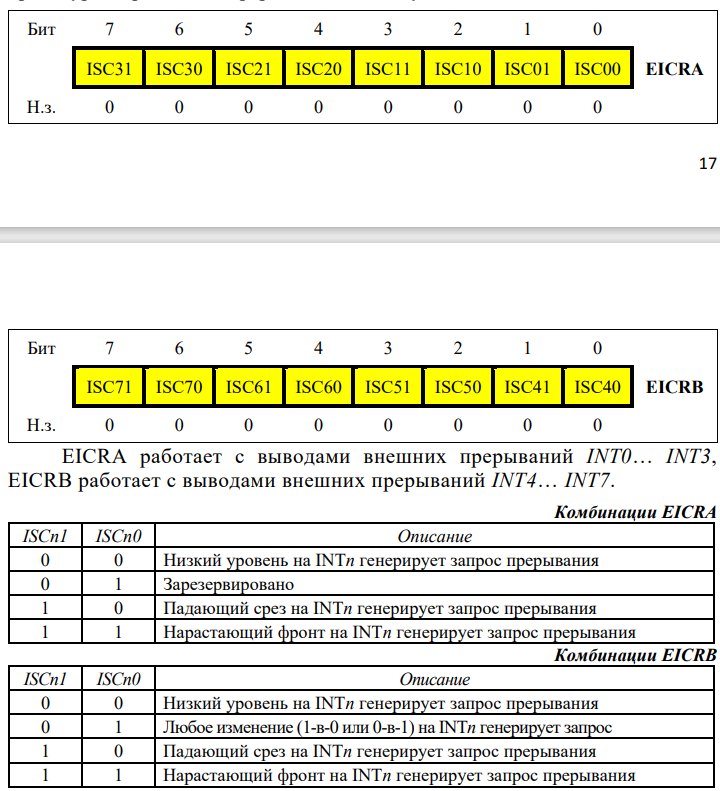
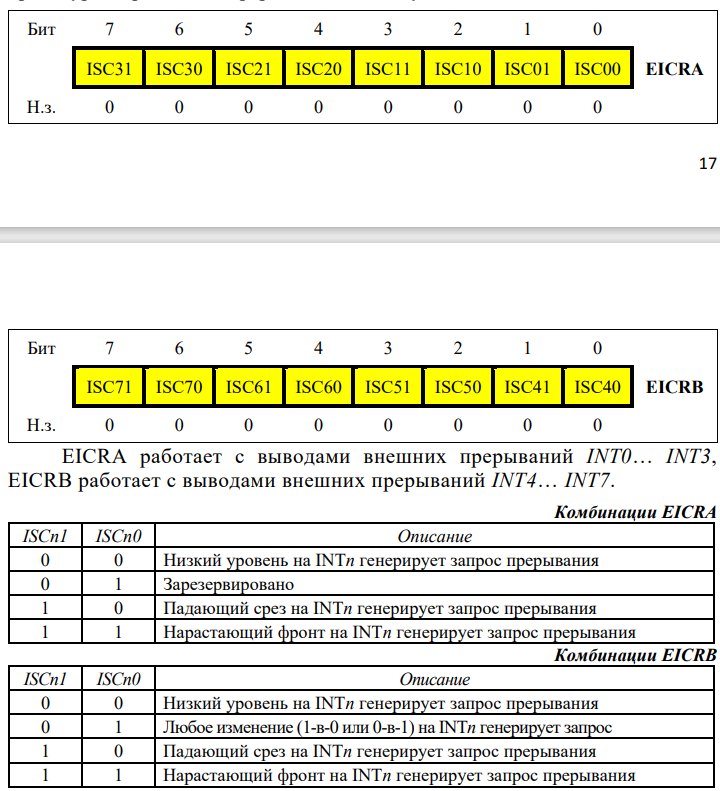
**Тема: «Работа с инкрементным энкодером»**

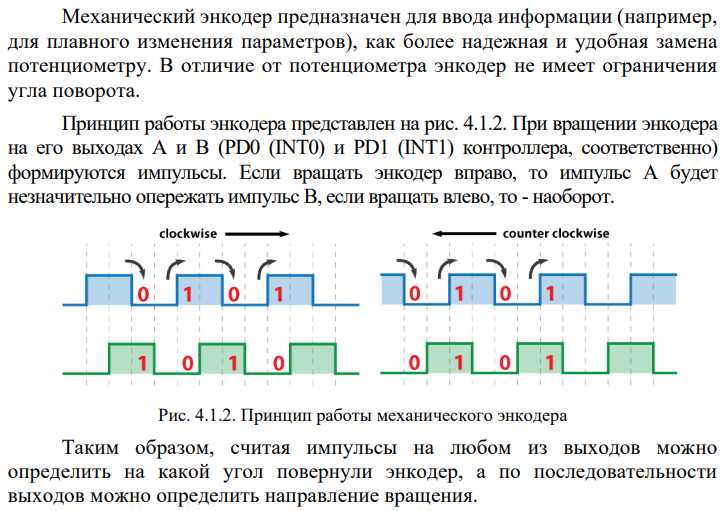
|  |  |  |
| --- | --- | --- |
| Студенты гр. 1492 |  | Старцев Н.А. |
| Преподаватель |  | Илатовская Е.В. |

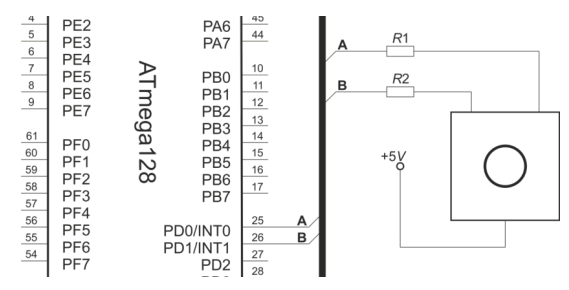
Санкт-Петербург

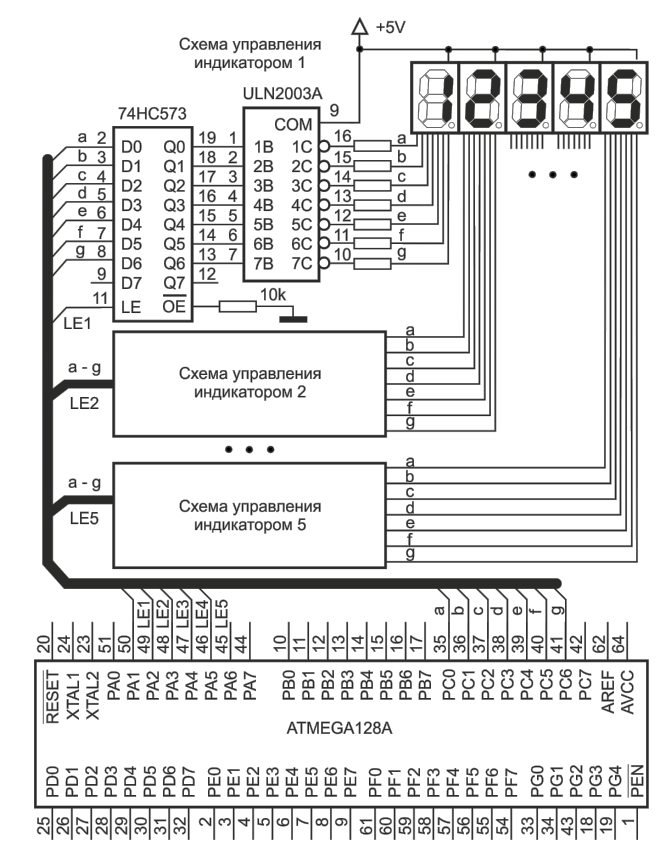
2024

**Цель работы:** получение навыков работы с энкодером, обработка данных от энкодера

**Основные сведения**

****

**Схема подключения**

****

**Результаты работы**

Написать программу вывода значения поворота энкодера на семисегментныые индикаторы, используя внешнее прерывание INT1 для определения направления вращения энкодера (на индикаторы должны выводиться как положительные, так и отрицательные значения углов поворота). По нажатию на кнопку энкодера, подключенную к INT2, счётчик оборотов должен обнулиться

Код программы:

#include <avr/io.h>

// #define F\_CPU 11059200

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

#include <avr/interrupt.h>

uint8\_t my\_znak[] = {0b00111111, 0b00000110, 0b01011011,

                     0b01001111, 0b01100110, 0b01101101, 0b01111101, 0b00000111,

                     0b01111111, 0b01101111, 0b00000000}; // рисунки цифр

// функция для возведения 10 в нужную степень

int16\_t angle = 0;

ISR(INT0\_vect)

{

    if ((PIND & (1 << 0)) != 0)

    {

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

        if ((PIND & (1 << 1)) != 0)

            angle+=90;

        else

            angle-=90;

    }

    else

    {

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

        if ((PIND & (1 << 1)) != 0)

            angle-=90;

        else

            angle+=90;

    }

    if (angle>9999){angle=-9999;}

    if (angle<-9999){angle=9999;}

}

ISR(INT2\_vect){angle=0;}

uint16\_t raz(uint8\_t k) // определяет разрядность числа

{

    uint16\_t a = 1;

    for (uint16\_t i = 0; i < k; i++)

    {

        a \*= 10;

    }

    return a;

}

int main(void)

{

    sei();

    EIMSK |= (1 << INT0);

    EIMSK |= (1 << INT2);

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

    DDRA = 0xFF; // теперь ножки это выход

    DDRC = 0xFF;

    int16\_t abs = 0; // задаем кол-во шагов

    while (1)

    {

        // костыль чтобы нули до первой не нулевой цифры не рисовались

        uint8\_t k = 0;

        uint8\_t flag = 1;

        if (angle < 0){ // Рисуем минус

            PORTC = 0b01000000;

            PORTA = 1 << (1);

            \_delay\_us(3);

            PORTA = 0x00;

            abs = -angle/10;

        }

        else

        {

            PORTC = 0b00000000; // не рисуем минус

            PORTA = 1 << (1);

            \_delay\_us(3);

            PORTA = 0x00;

            abs = angle/10;

        }

        for (uint8\_t i = 2; i < 6; i++)

        {                                // бежим по всем разрядам

            k = (abs / raz(5 - i)) % 10; // вычисляем цифру в разряде

            if (k > 0)

            {

                flag = 0;

            }

            // подменям ноль на пустоту, если цифры уже были

            if (flag && i!=5)

            {

                k = 10;

            }

            PORTC = my\_znak[k]; // вгружаем в по порту нашу цифру

            // выбираем разряд в который будем рисовать

            PORTA = 1 << (i);

            \_delay\_us(3);

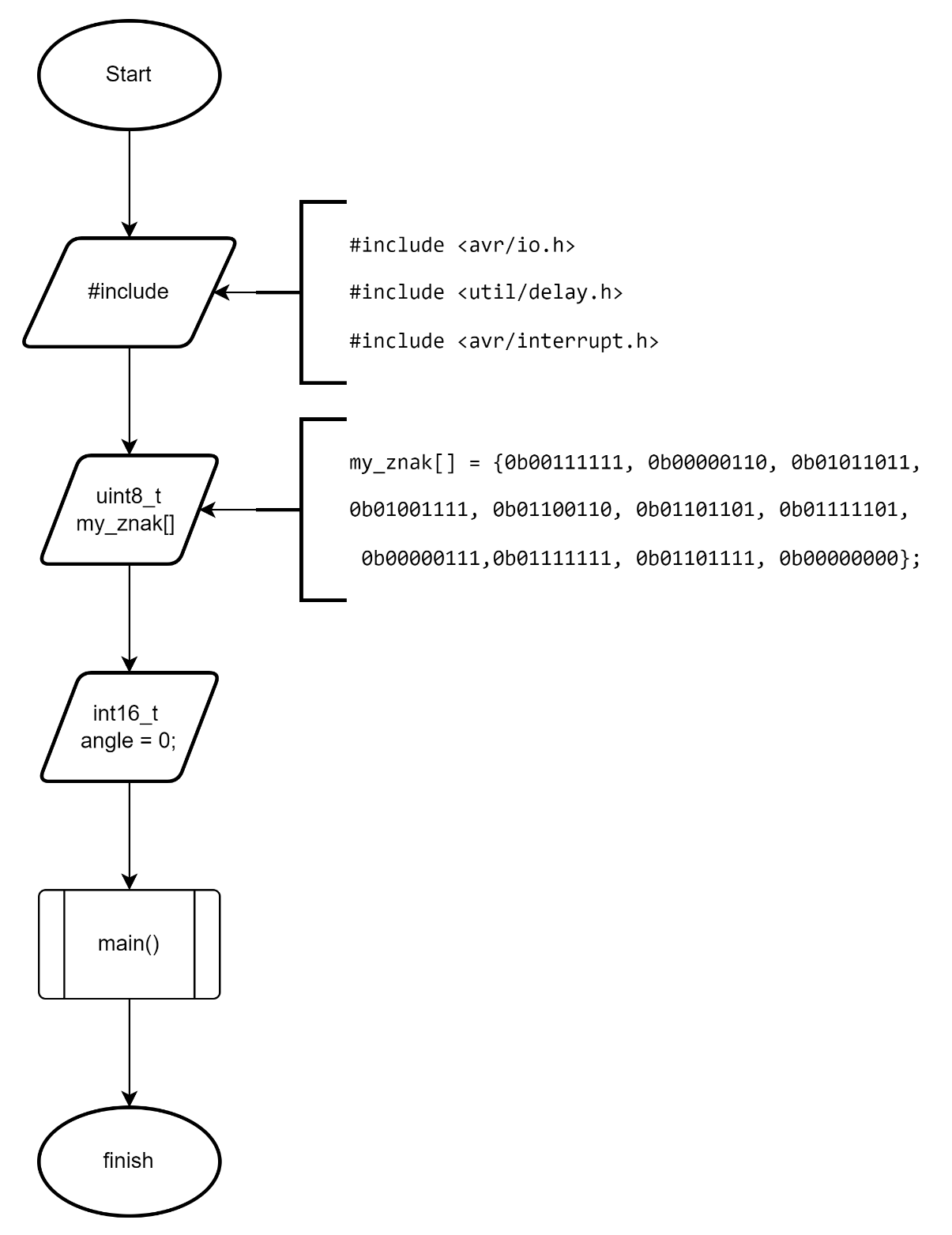
            PORTA = 0x00;

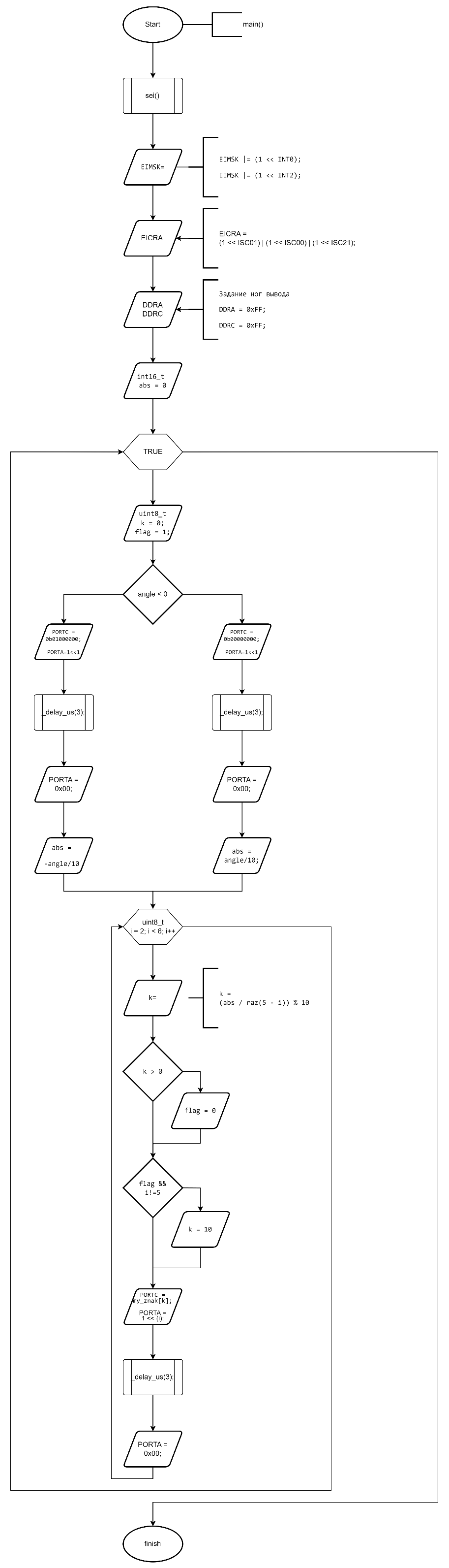
        }

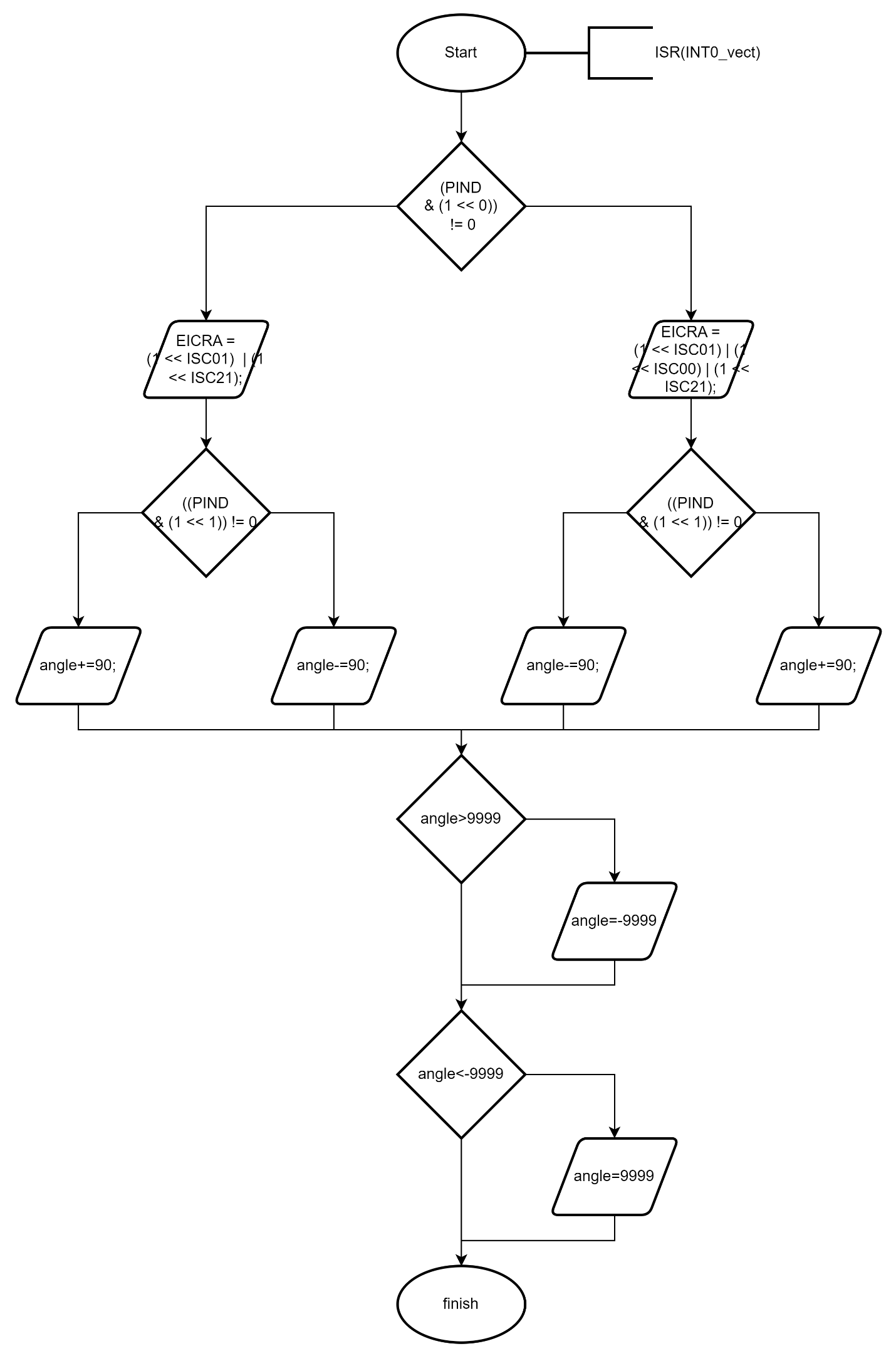
    }

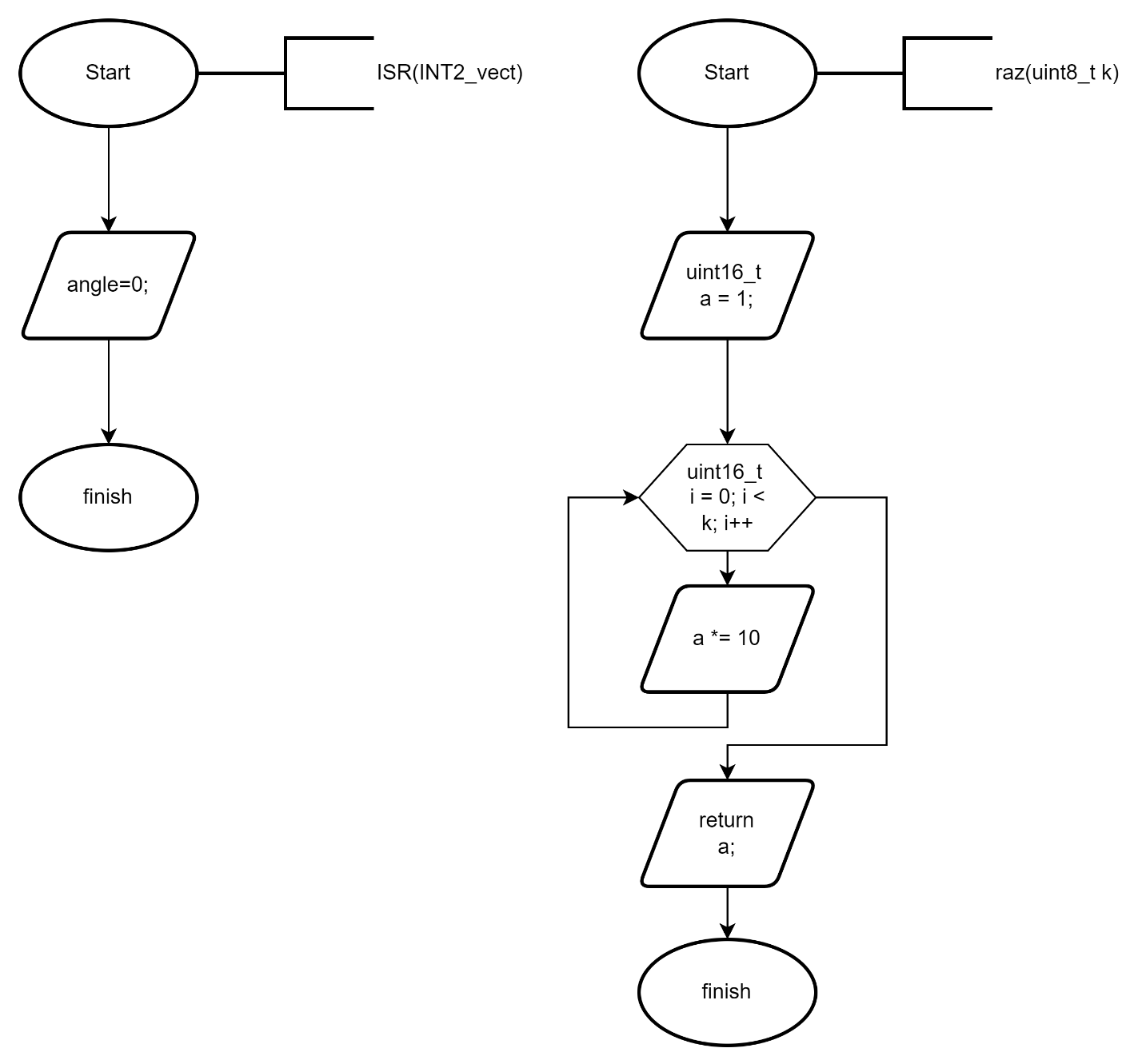
}

Блок схема программы:



****

****

****

**Вывод:**

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