

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**КАФЕДРА САУ**

**ОТЧЕТ**  
**по Лабораторной работе №1**  
**по дисциплине «МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В**  
**МЕХАТРОНИКЕ**  
**И РОБОТОТЕХНИКЕ»**  
**тема: Работа с механическим энкодером**

Студенты гр. 6492

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Мурашко А.С.  
Огурецкий Д.В.  
Спорыш И.В.

Преподаватель

\_\_\_\_\_

Девяткин А.В.

Санкт-Петербург

2019

**Цель работы** — освоить использование внешних прерываний в работе с механическим энкодером.

**Задание на лабораторную работу.** Написать программу вывода угла поворота энкодера на семисегментные индикаторы, используя внешнее прерывание №1 для определения угла поворота энкодера. Угол должен изменяться от -180 до 180 градусов без закликивания.

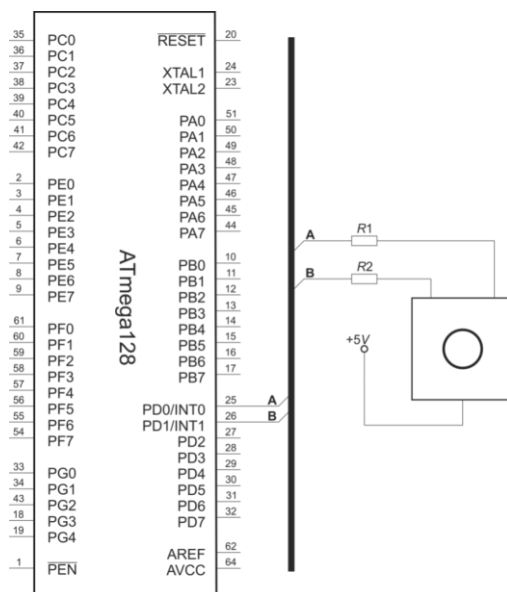


Рис. 1. Схема включения механического энкодера

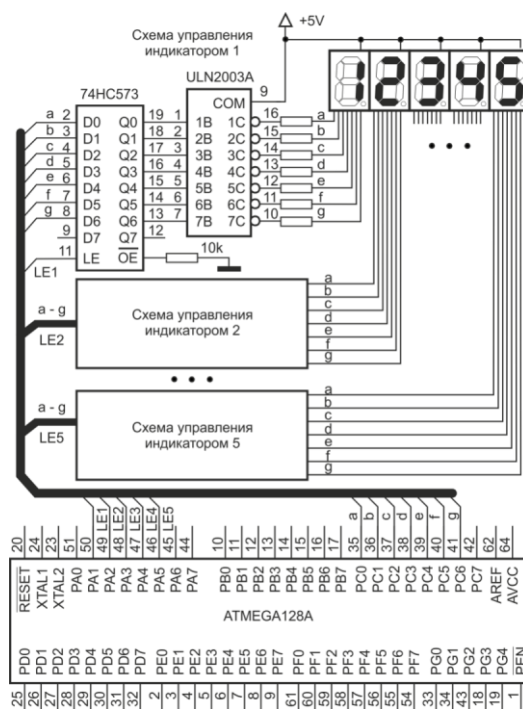


Рис. 2 Схема подключения семисегментных индикаторов

# Выполнение работы

## Описание библиотеки segments.c

### Блок-схема

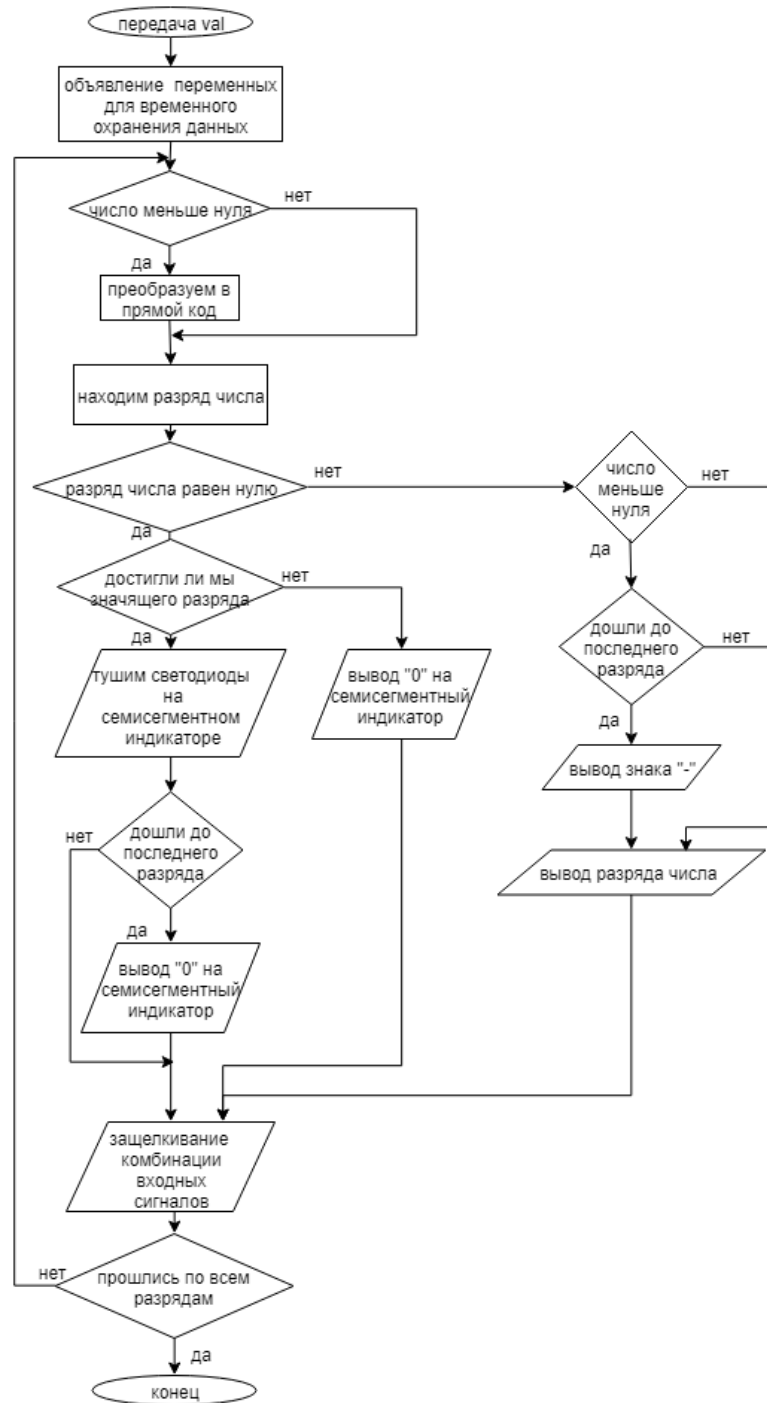


Рис. 3 Блок-схемы функций indic\_int и Digit библиотеки segments.h

## Код

Здесь есть некоторое несоответствие светодиодных индикаторов и номеров пинов порта PORTA, так крайний левый разряд (светодиодный индикатор) подключен к порту с номером один, но при этом он является старшим разрядом числа, а старшие разряды числа обычно имеют самый большой номер среди остальных.

```
1 // Массив кодов цифр
2 const unsigned char segments[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D,
3 0x07, 0x7F, 0x6F, 0x40, 0x00};
4 unsigned char Digit (unsigned int val, unsigned char m)
5 {
6     //[]-----[]
7     //| Назначение: выделение цифр из разрядов пятиразрядного |
8     //| десятичного положительного числа |
9     //| Входные параметры: |
10    //| d - целое десятичное положительное число |
11    //| m - номер разряда (от 1 до 5, слева направо) |
12    //| Функция возвращает значение цифры в разряде m числа d |
13    //[]-----[]
14    unsigned char dig_num = 5, a ;
15    unsigned int d = val ;
16    while(dig_num)
17    { // цикл по разрядам числа
18        a = d%10; // выделяем очередной разряд
19        if(dig_num-- == m) break; // выделен заданный разряд - уходим
20        d /= 10; // уменьшаем число в 10 раз
21    }
22    return(a);
23 }
24
25 void indic_int (int val)
26 {
27     unsigned char dig_num = 1;
28     int flag_first_digit = 0;
29     int var = val;
30     if(val<0)
31     {
32         var = abs(val); ///перевод в прямой код отрицательного числа
33     }
34     do
35     {
36         unsigned int digit;
37         digit=Digit(var,dig_num);
38
39         if(digit==0)
40         {
41             if(flag_first_digit==0)
42             {
43                 PORTC = segments[VOID];
44                 if(dig_num==5)
45                 {
46                     PORTC = segments[0];
47                 }
48             }else
49             {
50                 PORTC = segments[0];
51             }
52         }
53     }
54 }
```

```

52     }else
53     {         if(val<0)
54         {
55             if(flag_first_digit==0)
56             {
57                 PORTC = segments[SIGN];
58                 BitSet(PORTA,dig_num-1) ;
59                 delay_us(1);
60                 BitClr(PORTA,dig_num-1) ;
61             }
62         }
63         flag_first_digit = 1;
64         PORTC = segments[digit];
65
66     }
67     BitSet(PORTA,dig_num) ;
68     delay_us(1) ;
69     BitClr(PORTA,dig_num) ;
70     dig_num++;
71 }
72 while (dig_num<=5);
73 }
74
75 void init_segments()
76 {
77     PORTA &= ~(_BV(1) | _BV(2) | _BV(3) | _BV(4) | _BV(5) ) ;
78     PORTC = 0 ;
79     DDRA = _BV(DDA1) | _BV(DDA2) | _BV(DDA3) | _BV(DDA4) | _BV(DDA5) ;
80     DDRC = _BV(DDC0) | _BV(DDC1) | _BV(DDC2) | _BV(DDC3) | _BV(DDC4) |
_BV(DDC5) | _BV(DDC6) | _BV(DDC7);
}

```

## Описание обработчика прерываний void int1Isr(void)

Прерывание происходит по изменению сигнала на входе INT1 (выход В энкодера).

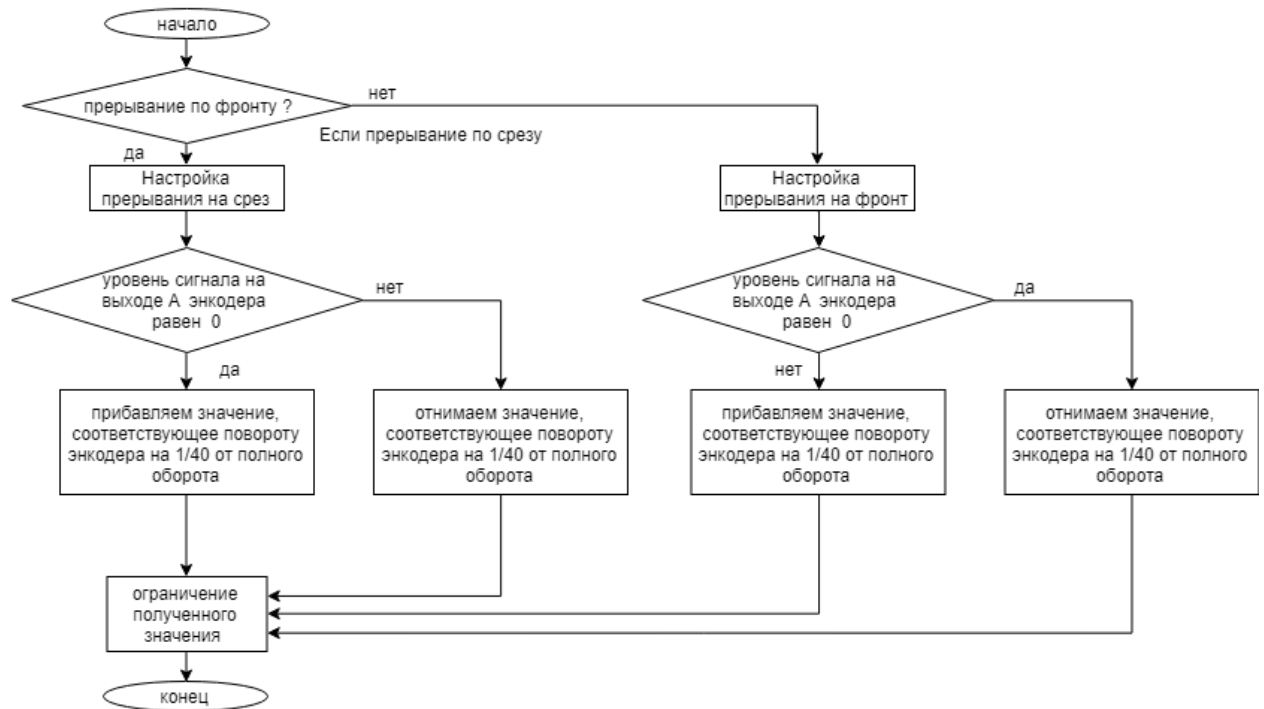


Рис. 4 Блок-схема обработчика прерываний void int1Isr

### Код

```
1. interrupt [EXT_INT1] void int1Isr(void)
2. {
3.     // Если прерывание по фронту
4.     if(BIT_IS_SET(PIND, 1))
5.     {
6.         // Настраиваем прерывание на срез
7.         EICRA = _BV(ISC11);
8.
9.         if(BIT_IS_CLEAR(PIND, 0))
10.        {
11.            angle+=9;
12.        }
13.        else // Если на второй ножке высокий уровень
14.        {
15.            angle-=9;
16.        }
17.    }
18.    else
19.    {
20.        // Если прерывание по срезу
21.        // Настраиваем прерывание на фронт
22.        EICRA = _BV(ISC11) | _BV(ISC10);
```

```
23.
24.     if (BIT_IS_CLEAR(PIND, 0))
25.     {
26.         angle-=9;
27.     }
28.     else // Если на второй ножке высокий уровень
29.     {
30.         angle+=9;
31.     }
32. }
33. if (angle>180)
34. {
35.     angle=180;
36. }
37. if (angle<-180)
38. {
39.     angle=-180;
40. }
41.
42. }
```

## Описание main()

### Блок-схема

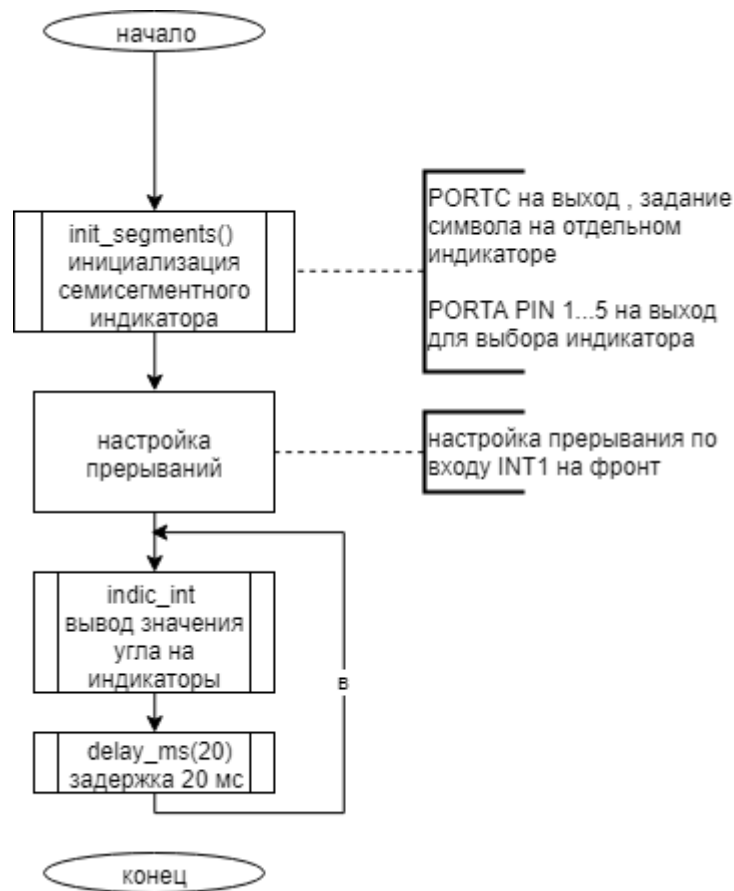


Рис. 5 Блок-схема main()

### Код

```
1. void main(void)
2. {
3.     init_segments() ;
4.     #asm("sei"); //разрешение прерываний
5.     EICRA = _BV(ISC11) | _BV(ISC10) ;//на фронт
6.     EIMSK = _BV(INT1);
7.
8.     while (1)
9.     {
10.         indic_int(angle);
11.         delay_ms(20);
12.     }
13. }
```

Вывод: мы освоили использование внешних прерываний в работе с механическим энкодером.