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

ОТЧЕТ

по Лабораторной работе №7
по дисциплине «МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В МЕХАТРОНИКЕ И РОБОТОТЕХНИКЕ»
тема: Работа с внешним АЦП по протоколу SPI

Студенты гр. 6492 _____ Мурашко А.С.
_____ Огурецкий Д.В.
_____ Спорыш И.В.

Преподаватель _____ Девяткин А.В.

Санкт-Петербург
2019

Цель работы — Изучение работы протокола SPI и работы с внешним АЦП AD7798.

Задание на лабораторную работу. Написать программу, которая бы выводила на семисегментные индикаторы старшие 5 бит результата измерения АЦП (с 20 по 16 биты), а в случае если они окажутся больше 16 (0b10000) подать звуковой сигнал на динамик

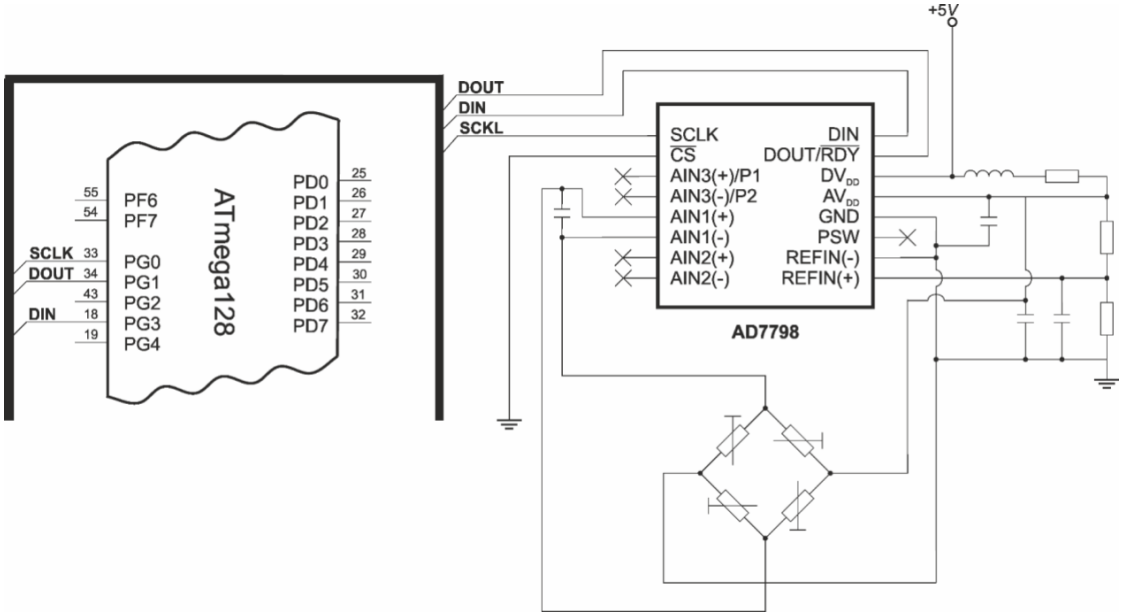


Рис. 1 Схема подключения АЦП к микроконтроллеру

Выполнение работы

Описание функций, реализующих работу SPI

Код

```
1  void SPI_init()
2  {
3       DDRG = _BV(MOSI) | _BV(SCLK) ;
4       BitSet(PORTG, SCLK) ;
5  }
6
7  #define pulse 10
8
9  void SPI_Write(char byte)
10 {
11     unsigned char i ;
12     for(i = (1 << 7) ; i > 0 ; i = (i>>1))
13     {
14         if(byte & i )
15             BitSet(PORTG, MOSI);
16         else
17             BitClr(PORTG, MOSI);
18         BitSet(PORTG, SCLK) ;
19         delay_us(pulse);
```

```
20          BitClr(PORTG, SCLK) ;
21          delay_us(pulse);
22      }
23  }
24
25  unsigned int SPI_Read()
26  {
27      unsigned char i, byte = 0 ;
28      for(i = (1 << 7) ; i > 0 ; i = (i>>1))
29      {
30          BitClr(PORTG, SCLK) ;
31          delay_us(pulse);
32          BitSet(PORTG, SCLK) ;
33          delay_us(pulse);
34          if( BIT_IS_SET(PING,MISO))
35          byte |= i ;
36      }
37      return byte ;
38  }
```

Описание функций библиотеки, реализующей работу LCD дисплея

Данная библиотека подробно описана в предыдущей лабораторной работе. Здесь добавлена только дополнительная функция.

Функция indic_5bit(char byte)

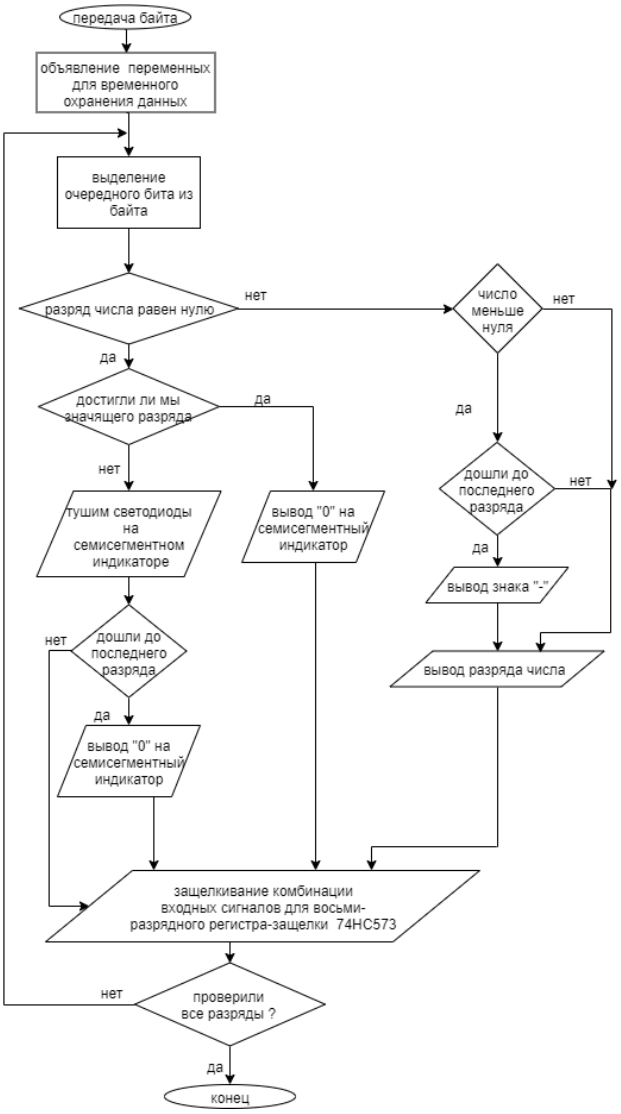
Данная функция введена в библиотеку “segments.h” специально для вывода 5 первых бит байта byte. Здесь есть некоторое несоответствие светодиодных индикаторов и номеров пинов порта PORTA, так крайний левый разряд (светодиодный индикатор) подключен к порту с номером один, но при этом он является старшим разрядом числа, а старшие разряды числа обычно имеют самый большой номер среди остальных.

Код:

```
1   void indic_5bit(char byte)
2   {
3       char bit_mask = (1<<4);//маска для выбора разряда байта
4       char dig_num = 1; //номер светодиодного индикатора
5       int flag_first_digit = 0;
6       do
7       {
8          unsigned char digit;
9          if(byte & bit_mask)  //выделение бита из байта
10         digit=1;
11         else
12         digit=0;
13
14         if(digit==0)
15         {
16              if(flag_first_digit==0)
17              { //пока не появился первый значащий разряд,ничего не выводим
```

```
18                  PORTC = segments[VOID];
19                  if(dig_num==5)
20                  {  //когда свсе биты равны 0, выводим 0
21                      PORTC = segments[0];
22                  }
23              }else
24              { //отображаем значащие разряды числа , которые равны 0
25                  PORTC = segments[0];
26              }
27          }else
28          {    //отображаем разряд,который  не равен  нулю
29               flag_first_digit = 1;   //появился значащий разряд
30               PORTC = segments[digit];
31          }
32          //импульс для выбора светодиодного индикатора
33          BitSet(PORTA,dig_num) ;
34          delay_us(1)     ;
35          BitClr(PORTA,dig_num) ;
36          bit_mask=(bit_mask>>1);//переход к следующему биту
37          dig_num++;          //переход к следующему разряду
38      }
39      while (bit_mask>0); //дошли до последнего разряда
40  }
```

Блок-схема



Описание функций по работе с датчиком и main()

Работа программы подробно описана на блок-схеме.

Код

```
1  unsigned long readAD7798()
 2 {
 3     unsigned char byte1, byte2, byte3;
 4     unsigned long result;
 5     while(BIT_IS_SET(PING, MISO)); //ожидание начала передачи
 6     byte1 = SPI_Read();
 7     byte2 = SPI_Read();
 8     byte3 = SPI_Read();
 9     result = ((unsigned long)byte1 << 15) + ((unsigned long)byte2 << 7) +
10 (unsigned long)byte3;
11     return result;
12 }
13
14 void setAd7798()
```

```
15 {
16     SPI_Write(CONFIG_REG);
17     delay_us(90);
18     SPI_Write(_BV(G1) | _BV(G2) | _BV(UB));
19     SPI_Write(_BV(BUF));
20     delay_us(90);
21     SPI_Write(MODE_REG);
22     delay_us(90);
23     SPI_Write(_BV(MD2));
24     SPI_Write(_BV(FS0) | _BV(FS1) | _BV(FS2) | _BV(FS3));
25 
26     delay_us(90);
27     SPI_Write(MODE_REG);
28     delay_us(90);
29     SPI_Write(0x00);
30     SPI_Write(_BV(FS0) | _BV(FS1) | _BV(FS2) | _BV(FS3));
31     delay_us(90);
32     SPI_Write(_BV(RW) | _BV(RS1) | _BV(RS0) | _BV(CREAD));
33 }
34 
35 void resetAD7798()
36 {
37     unsigned char i;
38 
39     for(i = 0; i < 4; ++i)
40     {
41         SPI_Write(0xFF);
42     }
43 }
44 
45 #define TIMER_FREQ (_BV(CS02) | _BV(CS01))
46 void initTimer()
47 {   // Быстрый ШИМ
48     TCCR0 = _BV(WGM01) | _BV(WGM00) | _BV(COM01);
49     DDRB |= _BV(4);
50     OCR0 = 128;
51 }
52 void startAlarm()
53 {   //clkI/O/1024 (From prescaler)
54     TCCR0 |= TIMER_FREQ;
55 }
56 void stopAlarm()
57 {   //выключение таймера
58     TCCR0 &= ~TIMER_FREQ;
59 }
60 
61 void main(void)
62 {
63     unsigned long data;
64     init_segments();
65     initTimer();
66     SPI_init();
67     resetAD7798();//  сброс АЦП
68     setAd7798();  //настройка на постоянное измерение
69     while(1)
70     {
71        data = (char)(readAD7798()>>16); //сохраняем старший бай измерения
```

```
72          indic_5bit(data);
73          //включение звукового оповещния о первышении 16
74          if(data>16) {startAlarm();}else{stopAlarm();}
75          delay_ms(200);
76      }
    }
```

Блок-схема main()

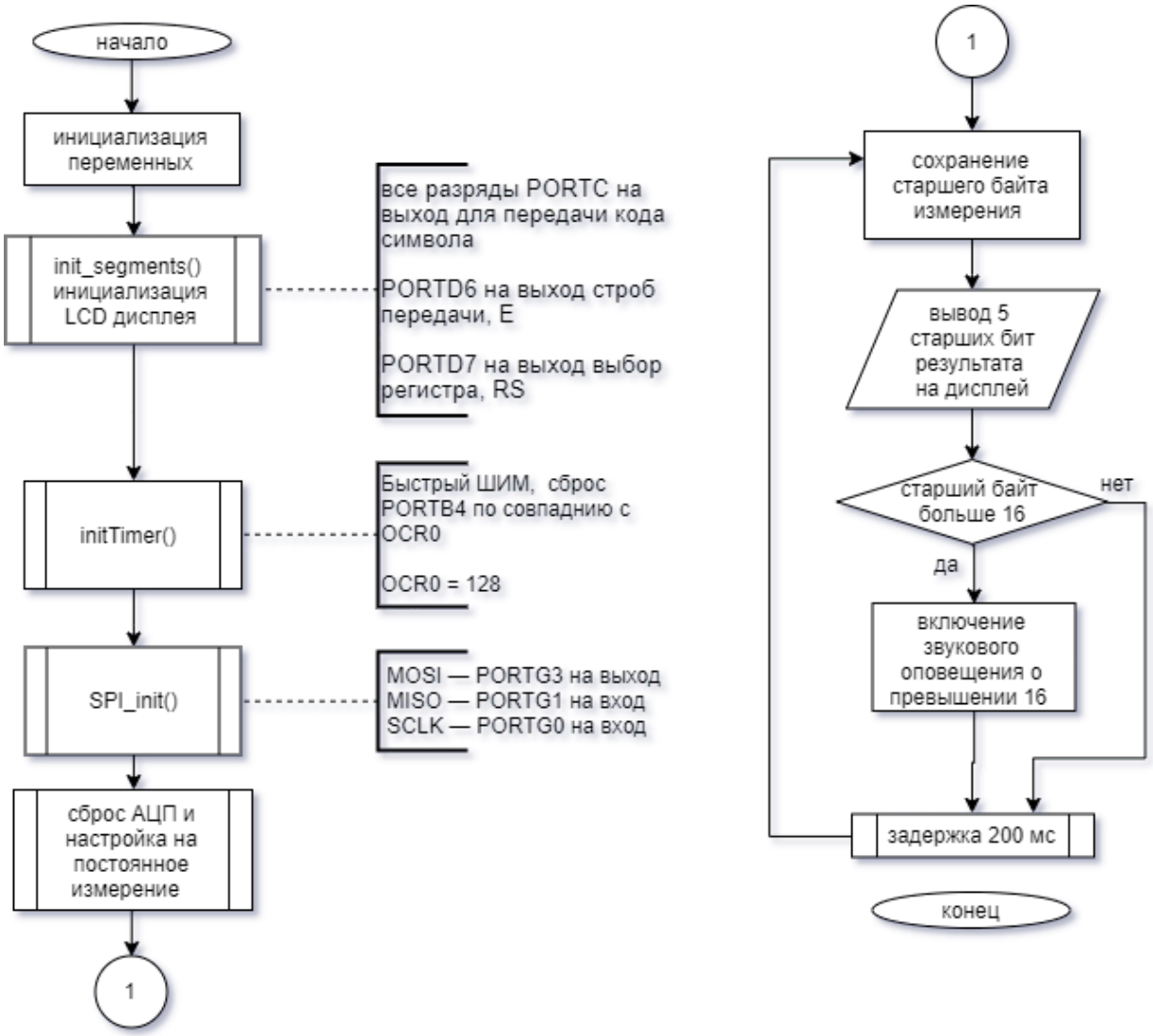


Рис. 2 Блок-схема main()

Вывод: с помощью микроконтроллера создать программный SPI и считывать показания с тензо-датчика.