

# Done at Home

Программирование микроконтроллеров avr

## Подключение 7 сегментных светодиодных индикаторов (динамическая индикация)

admin | 13.09.2013

0 Comment

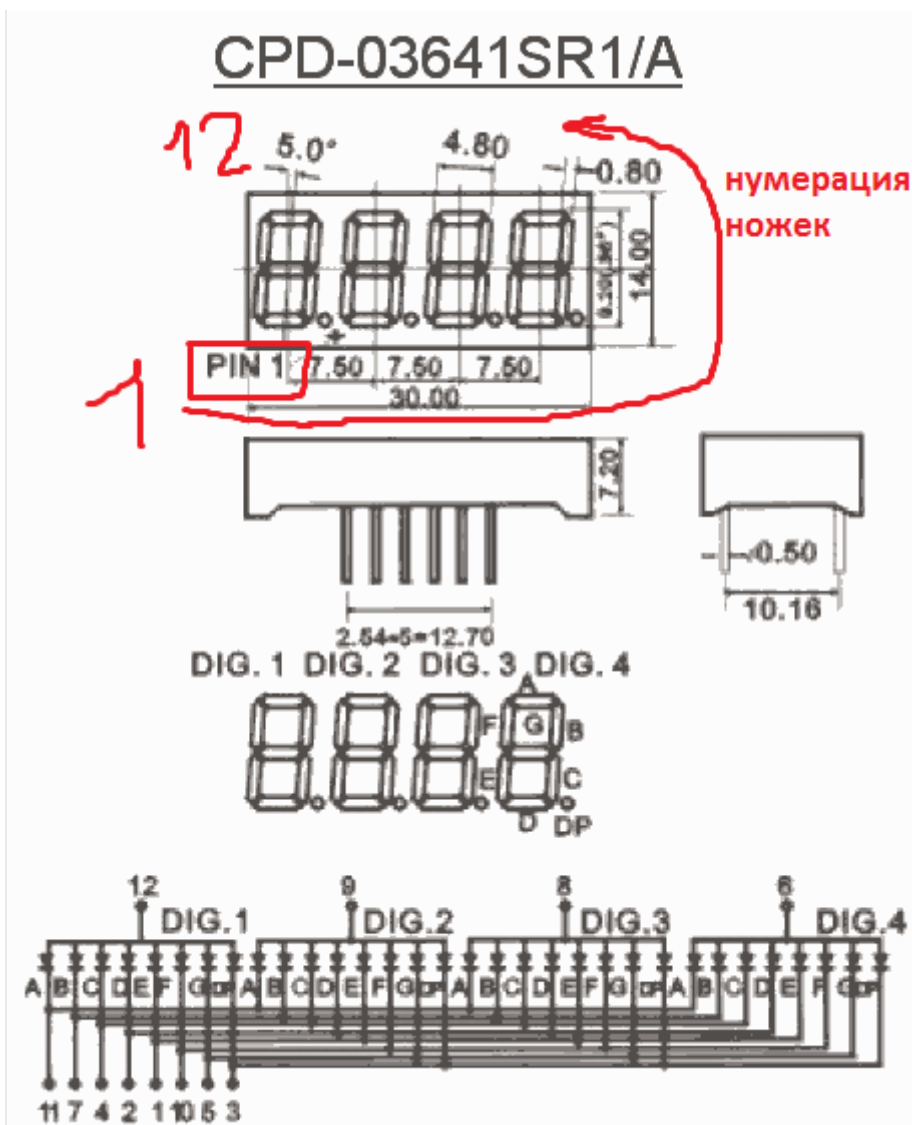
Подключение 7 сегментных светодиодных индикаторов (динамическая индикация)

Динамическая индикация - это метод отображения целостной картины через быстрое последовательное отображение отдельных элементов этой картины. (Благодаря инерционности человеческого зрения человек не видит что на самом деле в любой момент времени горит только часть изображения)

В этой статье будет использован: Светодиодные индикаторы (4 разряда):

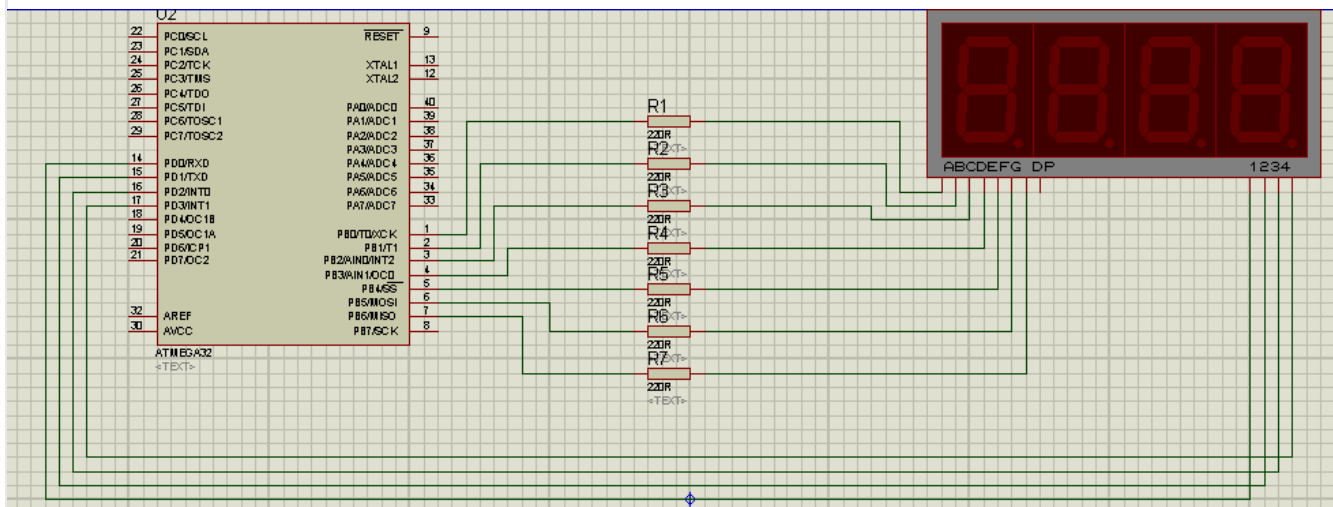
**CPD-03641SR1** ([ссылка](#))

**Светодиодный индикатор 30×14 мм, красный, 4 разряда, общий анод**



Из рисунка видно, что для индикации какого либо символа необходимо подать 5V на ножку для определенного знакового места плюс выставить нули на ножках (A,B,C...) , чтобы зажечь соответствующие части числа.

Так вот схема подключения:



- 1) Резисторы номиналом 220ОМ для защиты от перегорания.
- 2) Точки не подключены т.к не будут использованы в этой статье

Общие принципы работы:

- 1) Мы настроим таймер счетчик на прерывание через один и тот же временной промежуток для посимвольной индикации.
- 2) Создадим глобальные переменные для хранения каждого из разрядов выводимого числа
- 3) Напишем несколько функций необходимых для работы

Код:

```

1 //Начнем с подключаемых заголовочных файлов
2
3 #define F_CPU 1000000UL //определяем тактовую частоту как 1МГц
4 #include <avr/io.h> //основной заголовочный файл
5 #include <util/delay.h> //место хранения _delay_us()
6 #include <avr/interrupt.h> //заголовочный файл для работы с прерываниями
7
8 //функция для инициализации ножек подключенных к индикатору
9 void Port_init (void)
10 {
11 //определяем порт DiB на вывод информации
12 DDRB=0xFF;
13 PORTB=0x00;
14
15 DDRD=0xFF;
16 PORTD=0x00;
17 }
18
19 //функция принимает значение цифры которую необходимо вывести. Другими словами принимая в
20 void segment (char N)
21 {
22 switch (N)
23 {
24 case 0: PORTB&=~63; break; // число 0
25 case 1: PORTB&=~6; break; // число 1
26 case 2: PORTB&=~91; break; // число 2
27 case 3: PORTB&=~79; break; // число 3
28 case 4: PORTB&=~102; break; // число 4
29 case 5: PORTB&=~109; break; // число 5
30 case 6: PORTB&=~125; break; // число 6
31 case 7: PORTB&=~7; break; // число 7
32 case 8: PORTB&=~127; break; // число 8
33 case 9: PORTB&=~111; break; // число 9
34 case 10: PORTB&=~64; break; // число 10 это -
35 case 11: PORTB&=~0; break; // пустое место
36 };
37 }
38
39 //создаем глобальные переменные для хранения разрядов выводимого числа
40 char A1=0;
41 char A2=0;
42 char A3=0;
43 char A4=0;
44
45 //Эта функция принимает как аргумент число которое необходимо вывести. Эта функция меняет
46 void print_led (int chislo)
47 {
48 A1=chislo%10; // 1ый разряд 
49 A2=(chislo%100)/10; // 2ой разряд
50 A3=(chislo%1000)/100; // 3ий разряд
51 A4=(chislo%10000)/1000; // 4ий разряд
52 }
53

```

```
54 // инициализируем 1ый 16 битный таймер счетчик на прерывание по совпадению
55 void timer_init (void)
56 {
57   TCCR1B |= (1<<WGM12); // устанавливаем режим CTC (сброс по совпадению)
58   TIMSK |= (1<<OCIE1A); // устанавливаем бит разрешения прерывания 1ого счетчика по совпадению
59   OCR1AH = 0b00000111; // определяем число сравнения. Определяем как часто делать прерывания
60   OCR1AL = 0b00000000;
61   TCCR1B |= (1<<CS10); //запуск таймера
62 }
63
64 //переменная будет отвечать чтобы каждое следующее прерывание обрабатывало следующий разряд
65 char N=0;
66
67 // програма прерывания по совпадению 16ти разрядного счетчика
68 ISR (TIMER1_COMPA_vect)
69 {
70   //определяем какой разряд на очереди. очищаем порт. переходим на следующее знако место. рис
71   if(N==0) { PORTB|=127; PORTD=8; segment(A1); }
72   if(N==1) { PORTB|=127; PORTD=4; segment(A2); }
73   if(N==2) { PORTB|=127; PORTD=2; segment(A3); }
74   if(N==3) { PORTB|=127; PORTD=1; segment(A4); }
75
76   N+=1;
77   if(N==4)N=0;
78 }
79
80 //главная функция
81 int main(void)
82 {
83   //инициализируем порт DiB
84   Port_init();
85   //инициализируем и запускаем таймер
86   timer_init();
87
88   //глобальное разрешение прерываний если эту функцию не вызвать прерывания не будут работать
89   sei();
90   char i;
91
92   //вот суть примера. Создаем число и выводим его на наш индикатор каждую сек оно увелич на 1
93   for(i=0;i<9000;i++)
94   {
95     print_led(i);
96     _delay_ms(1000);
97     if(i==9000)i=0;
98   }
99 }
```

## Подключение 7 сегментных светодиодных ...



Рубрика: Все посты AVR Подключаем к AVR

harves.ru