

Семисегментный индикатор. Вывод информации с микроконтроллера.

Семисегментный индикатор ввиду своей красочности часто применяется для отображения информации, например, значения температуры, величины напряжения либо тока.

Для начала рассмотрим, что собой представляет семисегментный индикатор. Внешне он имеет различные размеры. Главным идентификатором служит высота цифры, которая в справочниках приводится в дюймах. Высота цифры имеет стандартный ряд значений, который приводится в дюймах.



По количеству разрядов различают одно-, двух-, трех-, и четырехразрядные индикаторы. Бывает и более разрядов, но они встречаются довольно редко.

Семисегментный индикатор. Принцип работы семисегментного индикатора

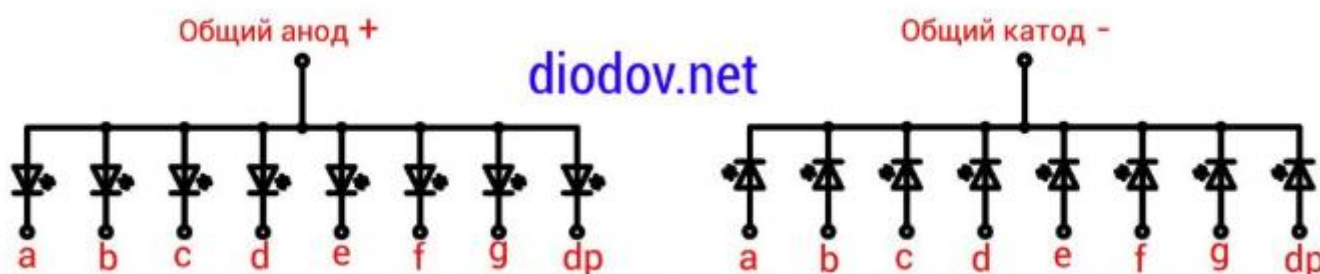


Любой семисегментный индикатор обязательно состоит из семи сегментов. Отсюда и происходит его название. Каждый сегмент – это обычный отдельный светодиод. Мощные семисегментники могут содержать в одном сегменте несколько, как правило, последовательно соединенных светодиодов.

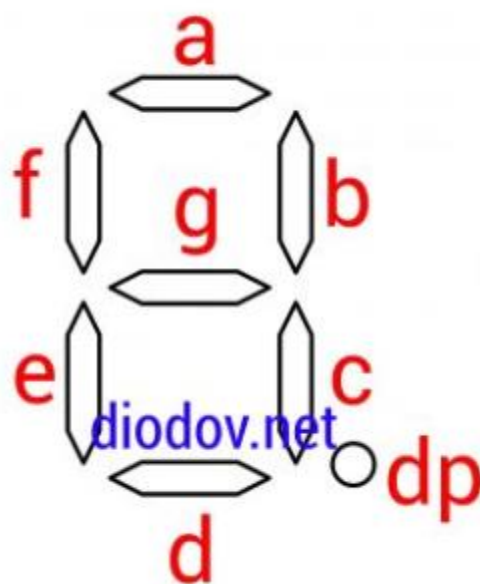
Кроме того, в корпусе помимо сегментов находится еще, и точка или запятая или другой символ.

С помощью семи сегментов можно изобразить десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 и некоторые буквы, как латиницы, так и кириллицы.

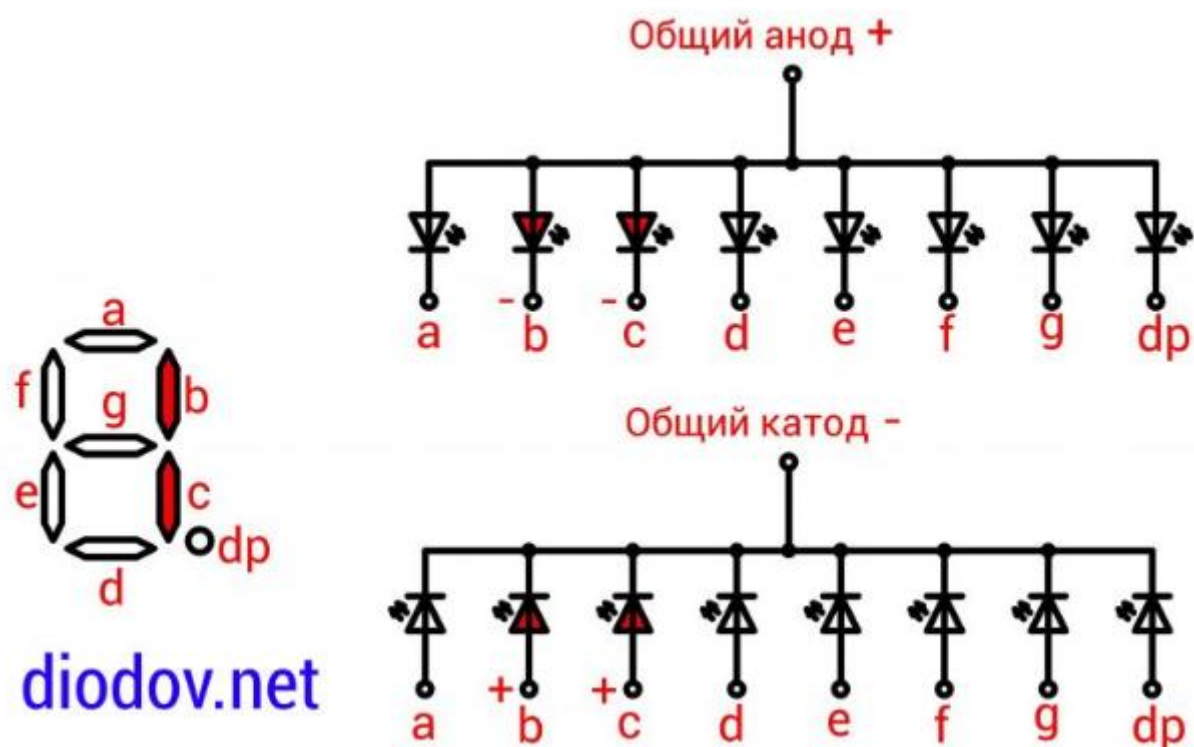
Светодиоды всех элементов соединяются одноименными выводами между собой или анодами, или катодами. Поэтому разделяют семисегментные индикаторы с общим анодом и общим катодом.



Вне зависимости от количества разрядов и размеров цифр каждый сегмент имеет название в виде одной из первых букв английского алфавита: a, b, c, d, e, f, g. Точка обозначается dp.

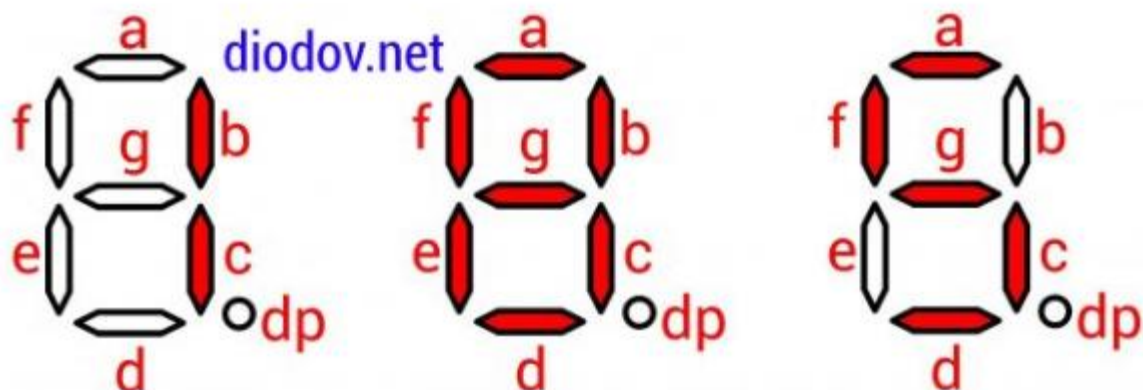


Для того чтобы засветить один из светодиодов семисегментного индикатора с общим анодом следует на общий вывод (анод) подать «+», а на соответствующий отдельный вывод – «-» источника питания.



Если применяется общий катод, — то наоборот – минус подается на общий, а плюс на отдельный вывод.

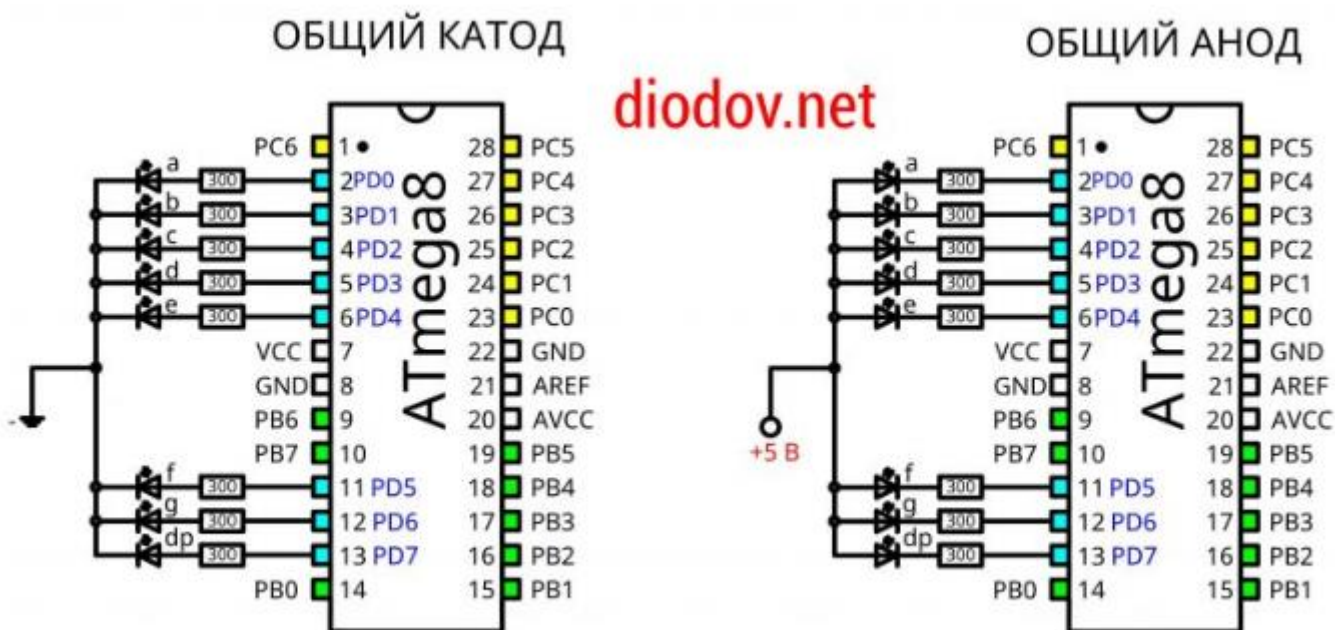
Чтобы отобразить на индикаторе цифру или букву следует засветить несколько сегментов. Например, для отображения единицы 1 задействуются сегменты b и c. При отображении восьмерки 8 задействуются все символы от a до g. Пятерка получается из таких символов: a, c, d, f, g.



Как подключить семисегментный индикатор к микроконтроллеру

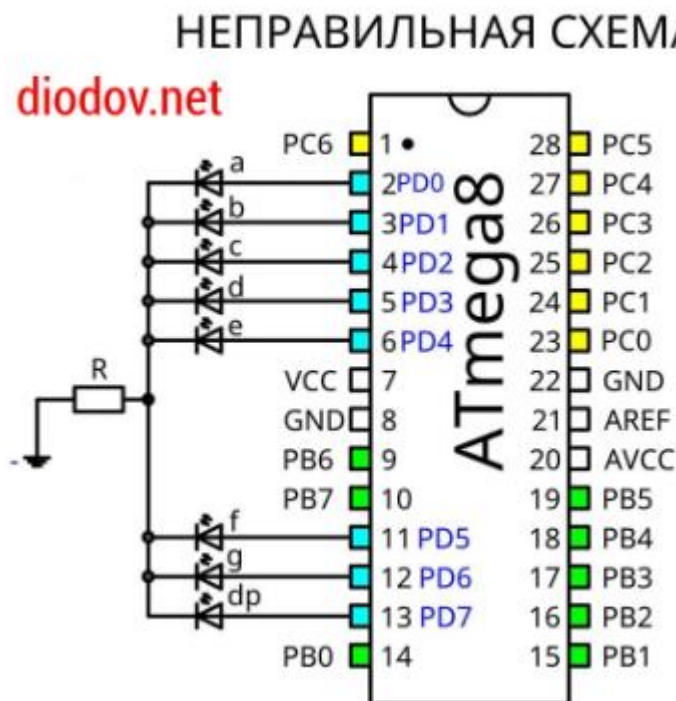
Теперь рассмотрим, как подключить семисегментный индикатор к микроконтроллеру АТмега8. Подключим его к порту D. Данный порт имеет все восемь бит, что очень удобно сочетается с количеством выводов одноразрядного семисегментного индикатора, у которого их также восемь с учетом вывода для точки.

Схемы подключения с общим анодом ОА и общим катодом ОК аналогичны, только общий вывод подключается соответственно к плюсу или минусу источника питания.



Все светодиоды подключаются к выводам микроконтроллера через отдельные резисторы сопротивлением 220...330 Ом.

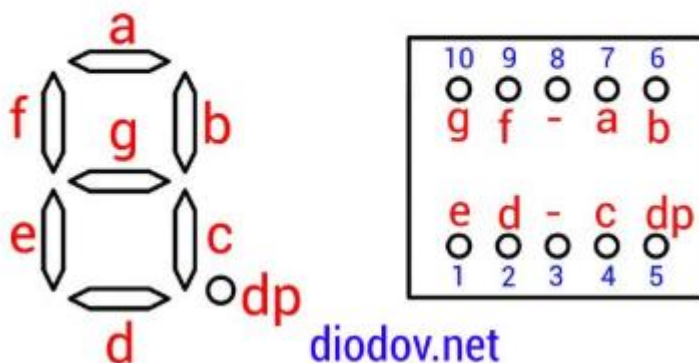
Не стоит экономить на резисторах и подключать все элементы через один общий резистор. Поскольку в таком случае с изменением числа



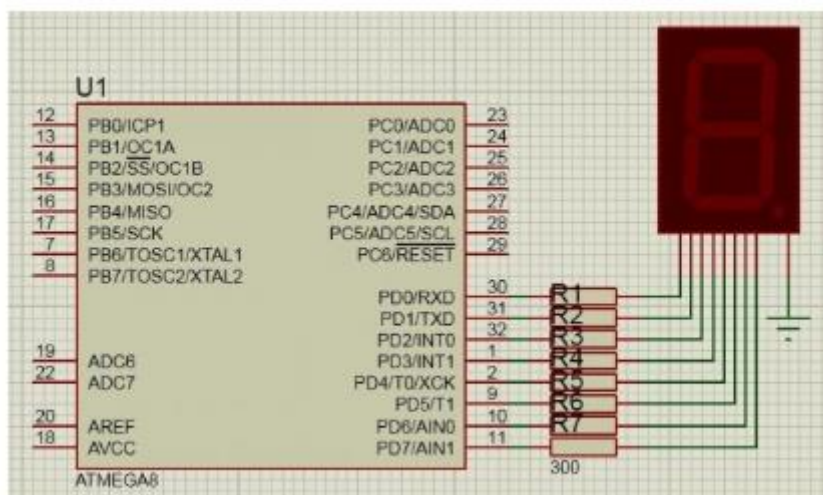
задействованных сегментов будет изменяться величина тока, протекающего через них. Поэтому цифра 1 будет светиться ярче, чем 8.

Чтобы знать какой из выводов отвечает тому или иному сегменту нам понадобится распиновка семисегментного индикатора. Отсчет выводов, как и у микросхем, начинается с левого нижнего и продолжается против часовой стрелки. При этом лицевая сторона индикатора должна быть направлена вверх, а выводы вниз.

Обозначение выводов семисегментного индикатора



Семисегментный индикатор в Proteus находится в категории (Category) Optoelectronics (Оптоэлектроника). Ниже в подкатегории (Sub-category) следует кликнуть по строке 7-Segment Displays. После этого в окне результатов (Results) выбираем одноразрядный семисегментный индикатор 7SEG-MPX1-CC.



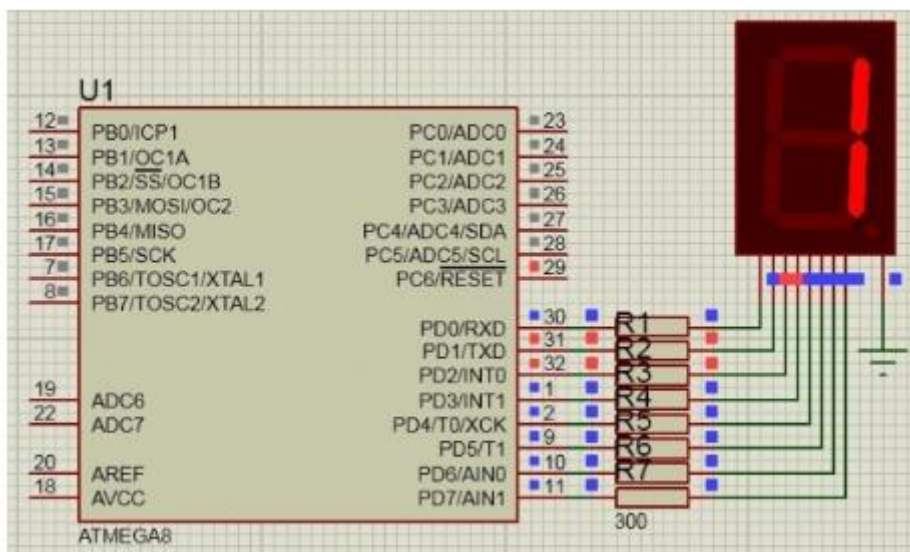
Код для микроконтроллера АТmega8

Теперь пишем код. Сначала настраиваем порт D полностью на выход. Для отображения единицы 1 задействуются сегменты b и c, выводы которых подключены к PD1 и PD2. Поэтому соответствующие биты регистр PORTD нужно установить в единицу.

```
#include <avr/io.h>

int main(void)
{
    DDRD = 0b11111111;
    while (1)
    {
        PORTD = 0b00000110; //1
    }
}
```

После компиляции кода и прошивки кода результаты мы видим в Proteus.



Аналогичным образом формируются все цифры.

Давайте сделаем программу более интересной, так, чтобы цифры изменялись в порядке нарастания от нуля до девяти с паузой 0,3 секунды.

```
#define F_CPU 1000000L
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRD = 0b11111111;

    while (1)
    {
        PORTD = 0b00111111; //0
        _delay_ms(300);
        PORTD = 0b00000110; //1
        _delay_ms(300);
        PORTD = 0b01011011; //2
        _delay_ms(300);
        PORTD = 0b01001111; //3
        _delay_ms(300);
    }
}
```

```
PORTD = 0b01100110; //4
  _delay_ms(300);
PORTD = 0b01101101; //5
  _delay_ms(300);
PORTD = 0b01111101; //6
  _delay_ms(300);
PORTD = 0b00000111; //7
  _delay_ms(300);
PORTD = 0b01111111; //8
  _delay_ms(300);
PORTD = 0b01101111; //9
  _delay_ms(300);
  }
}
```

Данный код можно значительно упорядочить.

Задание на выполнение:

1. Составить конспект по данной лекции.
2. Изучить конфигурацию микроконтроллера АТМЕГА8.
3. Переделать программу с использованием массива.

Результат программы переслать на e-mail: rasov@rambler.ru с темой СемисегментныйИндикатор_ФИО