

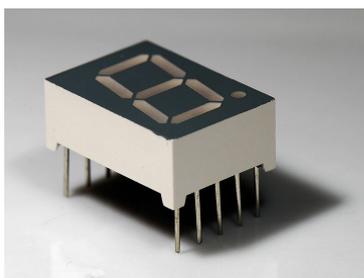
СЕМИСЕГМЕНТНЫЙ ИНДИКАТОР

Цель: получить общее представление об семисегментном индикаторе, научиться выводить на него информацию в мультиплексорном режиме.

Задание: настроить отладочную плату, согласно задания на программирование, разработать и отладить программу в среде программирования AVR Studio на языке ANSI-C.

ОБЩАЯ ИНФОРМАЦИЯ

Назначение



Семисегментный индикатор – устройство отображения цифровой информации. Это — наиболее простая реализация индикатора, который может отображать арабские цифры (Рис. 1). Для отображения букв используются более сложные многосегментные и матричные индикаторы.

Рис. 1. Семисегментный светодиодный индикатор с десятичной запятой.

Семисегментный индикатор, как говорит его название, состоит из семи элементов индикации (сегментов), включающихся и выключающихся по отдельности. Включая их в разных комбинациях, из них можно составить упрощённые изображения арабских цифр. Часто семисегментные индикаторы делают в курсивном начертании, что повышает читаемость.



Рис. 2. Различные форматы светодиодных индикаторов.

Кроме семисегментных индикаторов, выпускаются различного вида светодиодные сборки (Рис. 2) служащие для отображения символов.

Индикаторы выпускаются различных размеров и цветов.

Типы индикаторов и схемы включения

Семисегментный индикатор представляет собой сборку из 8-и светодиодов (точку на индикатор добавили несколько позже).

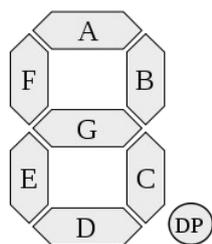


Рис. 3. Расположение сегментов.

Сегменты обозначаются буквами от А до G; восьмой сегмент (DP) – десятичная запятая, предназначенная для отображения дробных чисел.

Светодиодные индикаторы имеют предельно простую форму, так как в них применяются светодиоды, отлитые в форме сегментов, и чем меньше разных типов светодиодов, тем дешевле устройство.

В обычном светодиодном индикаторе девять выводов: один идёт к катодам всех сегментов, и остальные восемь – к аноду каждого из сегментов. Эта схема называется «схема с общим катодом», существуют также схемы с общим анодом (Рис. 4).

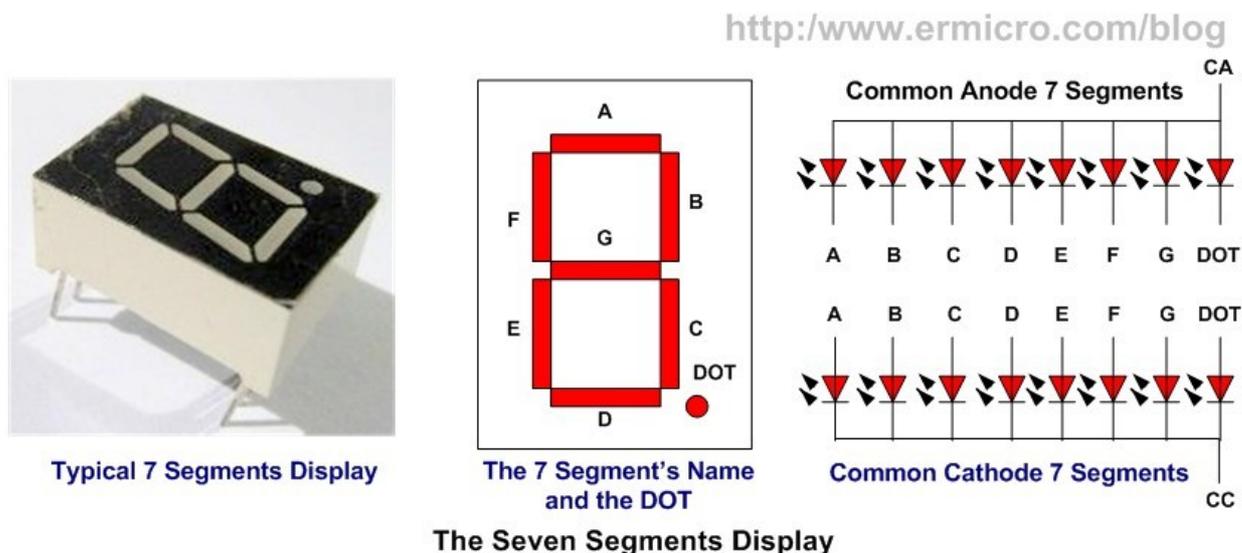


Рис. 4. Схема с общим анодом/катодом.

ОПИСАНИЕ ПРИНЦИПИАЛЬНОЙ СХЕМЫ

Принципиальная схема

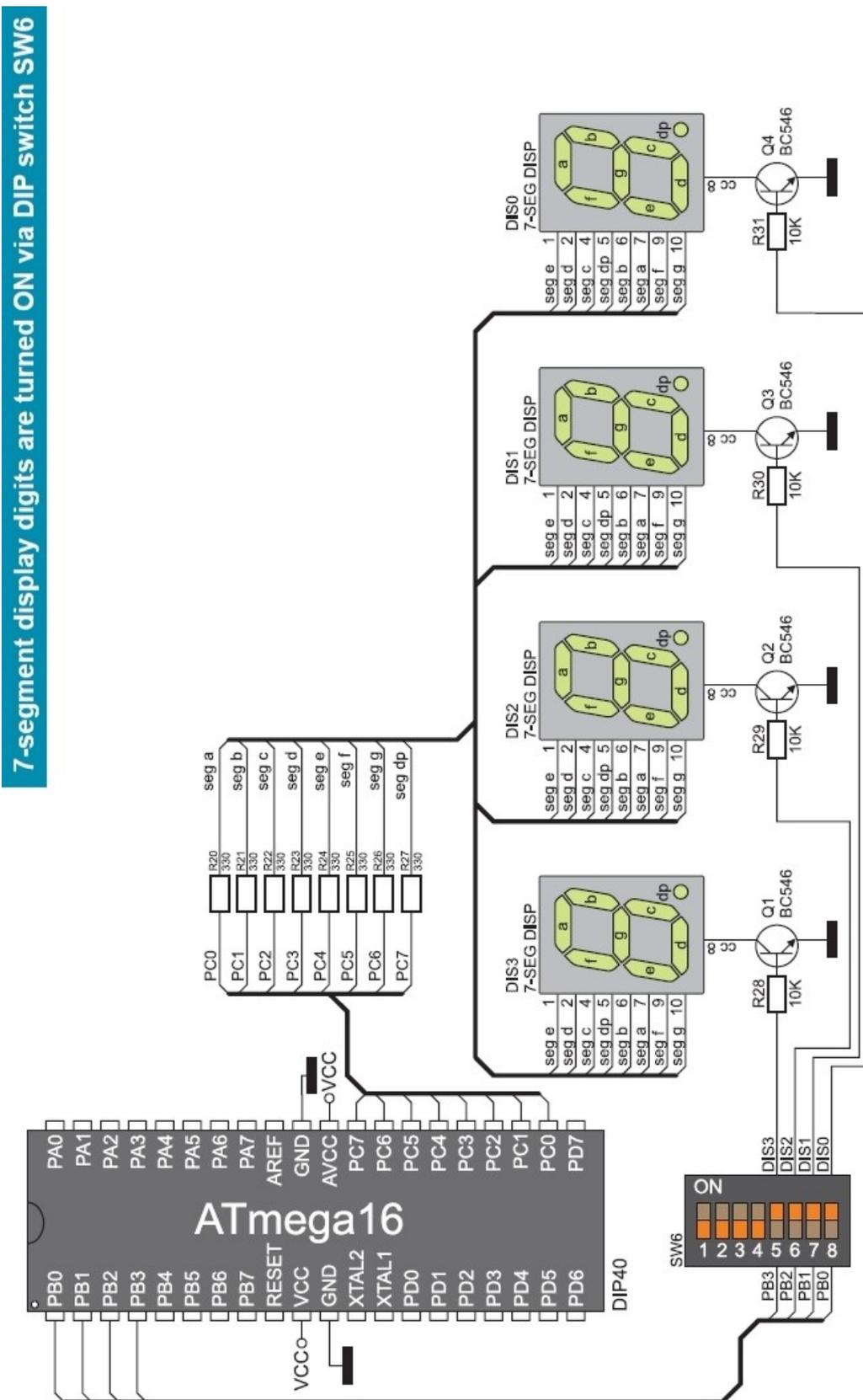


Рис. 5. Включение семисегментных индикаторов.

Детальное описание принципиальной схемы можно найти в документации на отладочную плату EASYAVR5A. В данном разделе будут изложены лишь некоторые общие положения...

Отладочная плата EASYAVR5A содержит 4 семисегментных индикатора (DIS3, DIS2, DIS1, DIS0) включенных в мультиплексорном режиме. Линии данных подключены к PORTC, а линии выбора индикаторов к PORTB (PB0, PB1, PB2, PB3). Для включения индикаторов используется ключ SW6 (5, 6, 7, 8). Запаянные индикаторы работают по схеме с «общим катодом». Принципиальная схема представлена на Рис. 5.

Внимание!!!

В документации (v. 100) опечатка – линии данных подключены к PORTC. В обновленном руководстве (v.101) данная неточность устранена.

Внимание!!!

Пример программирования индикаторов

Для примера рассмотрим программу выводящую на индикаторы символ «L». Индикаторы DIS3, DIS2, DIS0 должны отображать символ, а DIS1 – должен быть выключен.

```
//
// Настраиваем порты ввода/вывода
//

// Линии данных для 7-сегментных индикаторов
DDRC = 0xFF;

// Линии выбора для 7-сегментных индикаторов
DDRB = 0x0F;

//
// Включаем DIS3, DIS2 и DIS0
// Выводим на индикаторы символ "L"
//

PORTC = 0b00111000;           // "L"
PORTB = 0b00001101;           // DIS3, DIS2, DIS0
```

Результаты выполнения данных строк отображены на Рис. 6 и Рис. 7.

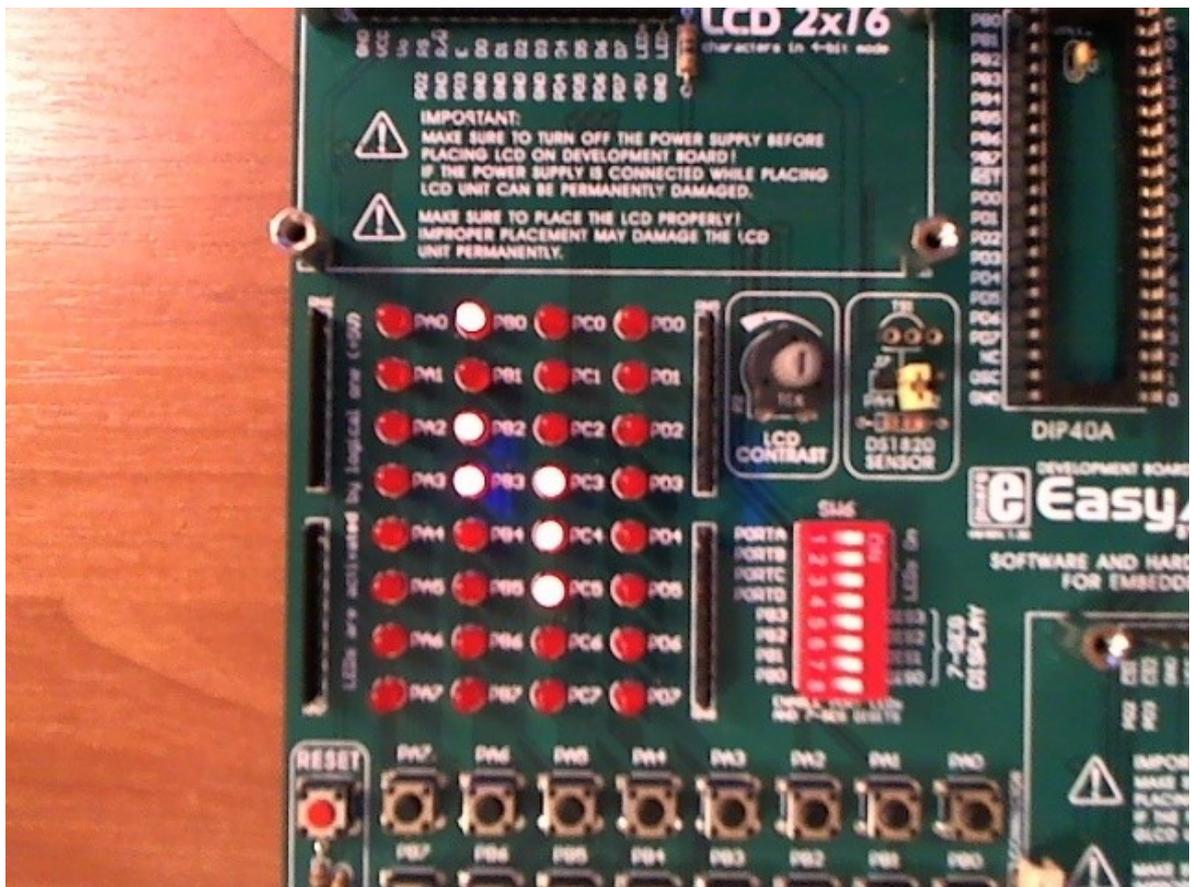


Рис. 6. Состояние портов ввода/вывода.

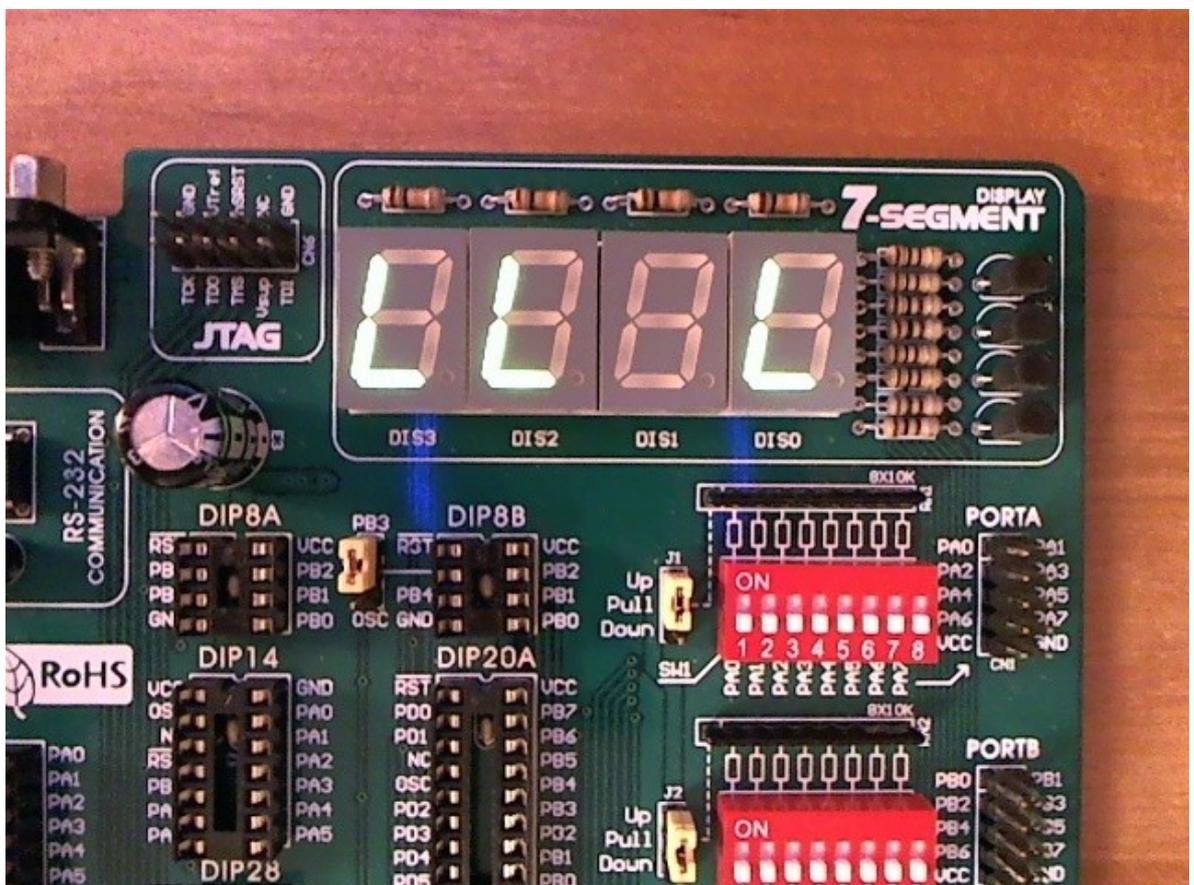


Рис. 7. Состояние индикаторов.

Мультиплексорный режим работы индикаторов

При включении 4-х индикаторов в мультиплексорном режиме, аноды А, В, С, D, Е, F, G и DP всех индикаторов объединяются в группы (по имени вывода) и подключаются к МК. Катоды коммутируются управляющими сигналами через транзисторные ключи (Q1, Q2, Q3, Q4) на «землю».

При работе в данном режиме индикаторы включаются последовательно (по кругу). Но за счет того, что их переключение происходит сравнительно быстро, создается впечатление, что они работают одновременно.

Дополнительную информацию по подключению индикаторов можно найти в следующем источнике:

- Баранов В.Н. «Применение микроконтроллеров AVR: схемы, алгоритмы, программы» (глава 4).

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Набрать текст программы (приведен в приложении).
2. Отладить ее и изучить на симуляторе принцип работы.
3. Запрограммировать в МК и проверить работоспособность.
4. Модифицировать программу согласно своему заданию (выдает преподаватель).
5. Отладить программу.
6. Запрограммировать в МК и протестировать.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое семисегментный индикатор.
2. Зачем используется мультиплексорный режим включения индикаторов.
3. Последовательность действий необходимая, для отображения символов на 4-х индикаторах (один цикл работы).

ПРИЛОЖЕНИЕ 1

Текст программы

/*

ЛАБОРАТОРНАЯ РАБОТА №3

Программа предназначена для изучения принципов работы и вывода информации на семисегментный индикатор

Описание работы программы

На 4-х семисегментных индикаторах отображается следующее:

- при нажатой кнопке PD0 - "0123";
- при нажатой кнопке PD1 - линия вверху;
- при нажатой кнопке PD2 - линия посередине;
- при нажатой кнопке PD3 - линия внизу

^ - для корректной работы симулятора нужно выключить оптимизацию кода
(Project -> Configuration Options -> General -> Optimization -> -00)

Предварительные настройки на плате

Настройка семисегментных индикаторов...

Сегменты индикаторов A, B, C, D, E, F, G и DP подключены к PORTC
Перебор индикаторов идет при помощи выводов PB0, PB1, PB2, PB3

Отключаем подтягивающие резисторы для PORTB, PORTC
Потенциал на выводах портов будет устанавливать МК
(SW2 и SW3 все разряды в положение OFF)

Включение индикации...

Включаем индикацию состояния выводов порта и семисегментные индикаторы
(SW6 = 11111111)

Настройка кнопок на PORTD по Pull-Down схеме...

Для выводов PD0, PD1, PD2 и PD3 включаем Pull-Down резисторы
(J4 = Pull-Down; SW4 = 11110000)

При замыкании кнопок выводы МК соединяются с VCC
(J6 = VCC)

Дополнительные настройки...

Ключи SW7, SW8 и SW9 все разряды в положение OFF

*/


```

unsigned char DIS1_Symbol = Space;
unsigned char DIS2_Symbol = Space;
unsigned char DIS3_Symbol = Space;

// Читаем кнопки PD3, PD2, PD1 и PD0
ButtonsState = PIND & 0x0F;

//
// В зависимости от замкнутой кнопки, определяем символы
// которые будут выведены на индикаторы
//
switch (ButtonsState)
{
    // Если замкнута PD0
    case 0x01:
    {
        // На индикаторы будет выведено "0123"
        DIS3_Symbol = sym_0;
        DIS2_Symbol = sym_1;
        DIS1_Symbol = sym_2;
        DIS0_Symbol = sym_3;
        break;
    }

    // Если замкнута PD1
    case 0x02:
    {
        // На индикаторы будет выведена линия вверху
        DIS3_Symbol = UpLine;
        DIS2_Symbol = UpLine;
        DIS1_Symbol = UpLine;
        DIS0_Symbol = UpLine;
        break;
    }

    // Если замкнута PD2
    case 0x04:
    {
        // На индикаторы будет выведена линия посередине
        DIS3_Symbol = MdLine;
        DIS2_Symbol = MdLine;
        DIS1_Symbol = MdLine;
        DIS0_Symbol = MdLine;
        break;
    }

    // Если замкнута PD3
    case 0x08:
    {
        // На индикаторы будет выведена линия внизу
        DIS3_Symbol = LwLine;
        DIS2_Symbol = LwLine;
        DIS1_Symbol = LwLine;
        DIS0_Symbol = LwLine;
        break;
    }

    // Во всех других случаях
    default:
    {
        // На индикаторы выводятся пробелы
        DIS3_Symbol = Space;
        DIS2_Symbol = Space;
        DIS1_Symbol = Space;
    }
}

```

```

        DIS0_Symbol = Space;
        break;
    }
}

/*
Последовательно выводим на индикаторы соотв. символы
(DIS3 -> DIS2 -> DIS1 -> DIS0 -> ...)
В каждый момент времени включен только один индикатор
Но за счет того, что происходит их быстрая смена кажется,
что они горят одновременно
*/

//
// Вывод символа на DIS3
//
PORTB = 0x00;           // Отключаем все индикаторы
PORTC = DIS3_Symbol;   // Задаем символ для вывода
PORTB = 0x08;         // Включаем индикатор DIS3
_delay_ms(Delay_Const); // Задержка

//
// Вывод символа на DIS2
//
PORTB = 0x00;           // Отключаем все индикаторы
PORTC = DIS2_Symbol;   // Задаем символ для вывода
PORTB = 0x04;         // Включаем индикатор DIS2
_delay_ms(Delay_Const); // Задержка

//
// Вывод символа на DIS1
//
PORTB = 0x00;           // Отключаем все индикаторы
PORTC = DIS1_Symbol;   // Задаем символ для вывода
PORTB = 0x02;         // Включаем индикатор DIS1
_delay_ms(Delay_Const); // Задержка

//
// Вывод символа на DIS0
//
PORTB = 0x00;           // Отключаем все индикаторы
PORTC = DIS0_Symbol;   // Задаем символ для вывода
PORTB = 0x01;         // Включаем индикатор DIS0
_delay_ms(Delay_Const); // Задержка
}
}

```

ПРИЛОЖЕНИЕ 2

Диаграмма активности (для программы из приложения 1).

