

2.5. Лабораторная работа № 5.

Отображение информации в системах с МК-51

Цель работы: научиться подключать к микроконтроллеру средства отображения информации для визуализации показаний.

В данной лабораторной работе в качестве средств визуализации рассматриваются семисегментные индикаторы. В базе данных Multisim также имеются 15-сегментные светоиндикаторы и многопозиционные дисплеи.

2.5.1. Общие сведения о семисегментных индикаторах

Семисегментные индикаторы удобны в управлении, имеют высокую яркость, широкий диапазон рабочих температур и низкую стоимость [6, 7].

Устройство состоит из семи светодиодов продолговатой формы, размещенных таким образом, чтобы, зажигая их в разных сочетаниях, можно было бы отобразить любую десятичную арабскую цифру от 0 до 9. Кроме семи основных сегментов, индикатор чаще всего дополняют восьмым маленьким сегментом, который предназначен для отображения десятичной точки (запятой). Если расположить в ряд несколько таких индикаторов, можно отображать любое десятичное число с плавающей запятой.

Внешний вид семисегментного индикатора приводится на рис. 2.41, где каждый сегмент индикатора обозначается буквой латинского алфавита. Такой индикатор обычно выполняется в виде отдельного самостоятельного компонента и имеет 9 выводов.

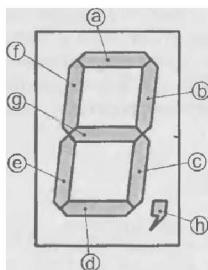


Рис. 2.41. Семисегментный цифровой индикатор

По внутренней схеме включения семисегментные индикаторы подразделяются на индикаторы с общим анодом (рис. 2.42) и индикаторы с общим катодом (рис. 2.43).

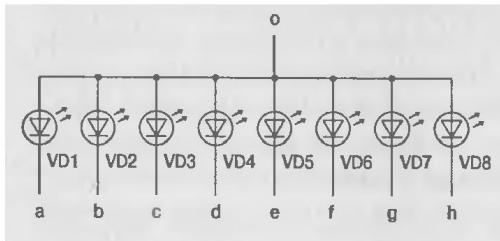


Рис. 2.42. Схема индикатора с общим анодом

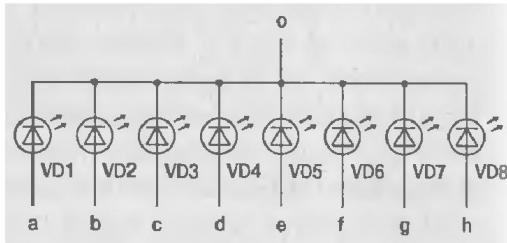


Рис. 2.43. Схема индикатора с общим катодом

В первом случае на общий вывод светодиодов подается плюс источника питания, а во втором – минус, в некоторых схемах семисегментный индикатор имеет 10 выводов (общий вывод дублируется).

Для подключения семисегментного индикатора повышенной яркости необходимо применять буферные элементы, например, регистры-зашелки или преобразователь двоичного кода в код семисегментного индикатора.

Пример подключения индикатора с общим анодом к МК показан на рис. 2.44, где использовано непосредственное подключение устройства к выводам микроконтроллера.

Для преобразования цифрового кода в код семисегментного индикатора существует два способа.

1. *Программный вариант* (см. рис. 2.44). Для подключения одного индикатора используются все 8 линий порта. Необходимо программно задать таблицу включения цифровых кодов семисегментного индикатора, которая для схемы с общим анодом имеет следующий вид:

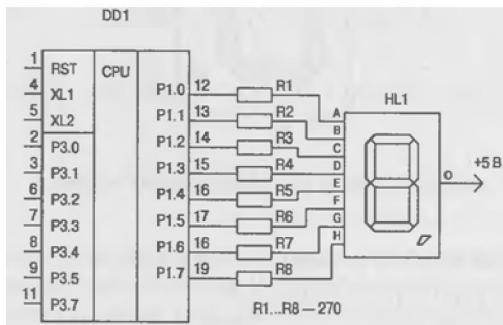


Рис. 2.44. Подключение семисегментного индикатора с общим анодом

```

db 11000000b      ; символ «0»
db 11111001b      ; символ «1»
db 10100100b      ; символ «2»
db 10110000b      ; символ «3»
db 10011001b      ; символ «4»
db 10010010b      ; символ «5»
db 10000010b      ; символ «6»
db 11111000b      ; символ «7»
db 10000000b      ; символ «8»
db 10010000b      ; символ «9»

```

Для светоиндикаторов с общим катодом коды цифр необходимо инвертировать.

2. *Аппаратный вариант* (рис. 2.45). Подключение светоиндикатора производится через преобразователь (десифратор) двоичного кода в код семисегментного индикатора. При таком подключении используются всего 4 линии выходного порта.

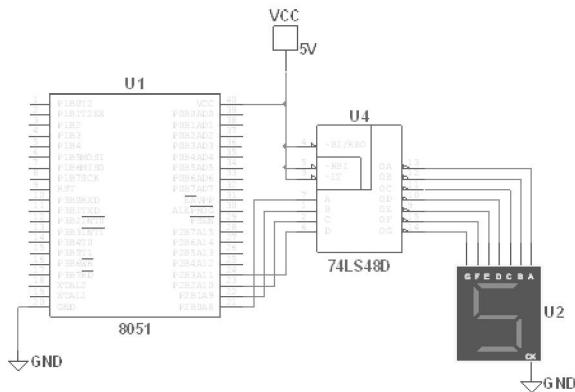


Рис. 2.45. Подключение семисегментного индикатора с общим катодом через десифратор

2.5.2. Ход выполнения лабораторной работы

Рассмотрим два способа преобразования цифрового кода в код семисегментного индикатора на примере следующей задачи: требуется вывести на индикатор, подключенный к порту P1, цифры от 9 до 1.

При выполнении лабораторной работы можно воспользоваться любым из показанных ниже вариантов преобразования информации.

Программный вариант. Открываем и сохраняем новый схемный проект Circuit 5. Устанавливаем на рабочем поле МК-51 (создаем программный файл prog5.asm), семисегментный индикатор с общим анодом, 8 токоограничивающих резисторов по 270 Ом, землю и питание (рис. 2.46).

Активируем закладку программного файла, щелкнув по ней ЛК мыши либо выбрав программный файл в окне разработки. Затем помещаем в окно программного файла отлаженный, например, в среде PRO View ассемблерный (или Си) программный код и сохраняем его.

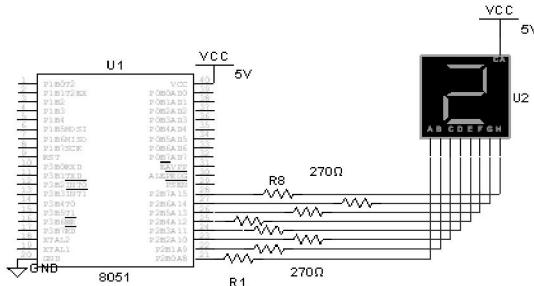


Рис. 2.46. Программный вариант преобразования информации в код индикатора

Приведем **код программы на ассемблере:**

```
$MOD51           ;подключение МК-51
org 20h          ;начинаем программу с адреса 20h
mov r2,#9        ;загрузить в регистр r2 число 9
ind:
    mov a,r2      ;загрузить аккумулятор операндом из регистра r2
    mov r3,#255    ;r3 – счетчик для реализации временной задержки
    ki   mov dptr,#zg  ;загрузка начала таблицы знакогенератора
    movc a,@a+dptr ;считывание кода цифры в ACC
    ind1:
        mov p2,a   ;вывод полученного кода впорт p2
        nop         ;пустая операция
        nop         ;пустая операция
        djnz r3,ind1 ;организация цикла для устойчивого отображения
                      ;каждой цифры
        djnz r2,ind  ;переход к отображению следующей цифры
                      ;(от 9 до 0)
    org 0100h       ;таблица кодов расположена с адреса 100h
    zg:             ;знакогенератор
        db 11000000b ;символ «0»
        db 11111001b ;символ «1»
        db 10100100b ;символ «2»
        db 10110000b ;символ «3»
        db 10011001b ;символ «4»
        db 10010010b ;символ «5»
        db 10000010b ;символ «6»
```

```
db 1111100b      ;символ «7»  
db 10000000b    ;символ «8»  
db 10010000b    ;символ «9»  
END
```

Ниже приводится **программа решения подобной задачи на С** для случая бесконечного вывода цифр в прямом порядке (от 0 до 9) и паузой после вывода цифры 9.

```
#include <8051.h>  
void main()  
{  
    unsigned char i,j;      // 8-битные переменные  
    unsigned char massiv [11]=  
    {  
        0xC0,           //массив кодов семисегментного индикатора  
        от 0 до 9  
        0xF9,  
        0xA4,  
        0xB0,  
        0x99,  
        0x92,  
        0x82,  
        0xF8,  
        0x80,  
        0x90,  
        0xFF           //код выключения  
    };  
    P1=massiv [10];        //сначала выключим индикатор  
    for(i=0;i<10;i++)     //затем выводим код в цикле в порт 2  
    {  
        P2=massiv[i];      //коды от 0 до 9  
        for(j=0;j<100;j++)  
            //временная задержка для устойчивого горения каждой цифры  
            continue;  
    }  
    P2=massiv[10];        //выключить индикатор  
    while(1);  
}
```

Покажем *аппаратный вариант преобразования* числа в код индикатора. Открываем и сохраняем новый схемный проект Circuit 5–2. Устанавливаем на рабочем поле МК-51 (создаем программный файл prog5–2), семисегментный индикатор с общим катодом, преобразователь двоичного кода в семисегментный индикатор 74LS48D для светоиндикатора с общим катодом (данная модель дешифратора уже содержит подтягивающие сопротивления), землю и питание (см. рис. 2.45).

Код программы на С

```
#include <8051.h>
void main()
{
    unsigned char i,j;      //8-битные переменные
    for(i=0;i<10;i++)       //затем выводим код в цикле в порт2
    {
        P2=i;                //коды от 0 до 9
        for(j=0;j<100;j++)
            //временная задержка для устойчивого горения каждой цифры
            continue;
    }
    while(1);
}
```

Для вывода байта (числа от 0 до 255) на семисегментные индикаторы может потребоваться включение в схему нескольких индикаторов. При этом необходимо использовать следующую *ассемблерную программу* преобразования байтового числа в коды BCD:

mov B,#100	;загрузить в В число 100 для вычисления количества сотен в числе
div AB	;аккумулятор содержит число сотен, т. е. старшую цифру
mov r0,A	;пересылка в R0 старшей цифры
xch A,B	;пересылка остатка исходного числа в аккумулятор
mov B,#10	;загрузить в В число 10 для вычисления количества десятков в числе
div AB	;аккумулятор содержит число десятков, В – число единиц
swap A	;размещение числа десятков в старшей тетраде аккумулятора
add A,B	;суммирование остатка (числа единиц), теперь аккумулятор содержит две младшие цифры

Для подключения нескольких семисегментных индикаторов к одному порту микроконтроллера используется прием, называемый «динамической индикацией» [6].

2.5.3. Задания для лабораторной работы

Задание 1. Байтовую переменную VAL вывести на трехразрядный семисегментный индикатор (использовать 3 индикатора, подключенных к одному из портов МК). Варианты заданий приведены в табл. 2.14.

Таблица 2.14

Варианты задания 1

Параметры	1	2	3	4	5	6
VAL	0FFh	0A7h	4Ch	210	100	150
Выходной порт	P1	P2	P3	P2	P3	P1

Задание 2. Сложить два числа a и b , представленных в двоично-десятичном коде (BCD). Результат вывести на двухразрядный семисегментный светоиндикатор, подключенный к одному из портов МК. Варианты заданий приведены в табл. 2.15.

Таблица 2.15

Варианты задания 2

Параметры	1	2	3	4	5	6
a	23	77	36	16	45	64
b	52	18	54	61	37	35
Выходной порт	P1	P2	P3	P3	P1	P2

2.5.4. Содержание отчета

1. Наименование и цель работы.
2. Описание особенностей работы семисегментного индикатора, используемого в схеме.
3. Копия схемного файла во время моделирования с указанием позиционных обозначений элементов.
4. Копия программного файла (на ассемблере или на С) с подробными комментариями.
5. Полученные результаты и выводы по работе.

2.5.5. Вопросы для самоконтроля

1. Расскажите о схемных особенностях использования порта Р0 в качестве выходного.
2. Как настроить порт МК на прием информации?
3. Можно ли отдельные линии порта настроить на ввод, а другие на вывод?
4. В чем отличие при представлении байта кодом ВСД по сравнению с двоичным кодом?
5. Каковы схемные особенности регистра-защелки информации? Покажите на примерах, имеющихся в БД Multisim.
6. Расскажите о схемотехнических отличиях, связанных с использованием программного преобразования двоичного числа в код семисегментного индикатора (десятичное число), по сравнению с аппаратным преобразованием.
7. Какие регистры МК по умолчанию используются для операций умножения и деления?
8. Как преобразовать байт в десятичные цифры, выводимые на семисегментные индикаторы?
9. Расскажите о схемных особенностях использования элементов с общим коллектором.