



РАДИОЛЮБИТЕЛЮ О МИКРОПРОЦЕССОРАХ И МИКРО-ЭВМ

ДИСПЛЕЙНЫЙ МОДУЛЬ

Теперь рассмотрим, каким должно быть программное обеспечение микро-ЭВМ, обслуживающей такой дисплейный модуль. Для этой цели необходима специальная подпрограмма.

Очевидно, что в любом месте основной программы, где возникает необходимость провести какие-либо операции с экраном (отобразить символ, переместить курсор, стереть символ), должна стоять команда вызова такой подпрограммы. Перед вызовом подпрограммы необходимо в какой-либо внутренний регистр микропроцессора или ячейку памяти поместить соответствующий код для операций с экраном, чтобы подпрограмма обслуживания «знала, что делать».

В табл. 2 дано соответствие между символами, отображаемыми на экране, и их кодами, записываемыми в ОЗУ страницы. Кроме кодов символов, в таблице приведены следующие управляющие коды перемещения курсора:

БК — в начало ряда знакомест (аналогичен возврату каретки в пишущей машинке).

ПС — в начало следующего ряда знакомест (аналогичен переводу строки для пишущей машинки);

→ — на одно знакоместо вправо;

← — влево;

↓ — вниз;

↑ — вверх;

↖ — в левый верхний угол экрана.

Существует также код **СТР** для стирания содержимого всего экрана с одновременным перемещением курсора в левый верхний угол. Не указанные в таблице коды используют для вывода на экран графических символов.

Теперь перечислим функции, возложенные на подпрограмму обслуживания.

Прежде всего программа анализирует, является ли выводимый код управляющим или же это код алфавитно-цифрового символа.

Если это код алфавитно-цифрового символа, то он сразу записывается в ячейку ОЗУ страницы дисплейного модуля. Адрес этой ячейки хранится в двух специально отведенных смежных ячейках памяти ОЗУ микро-ЭВМ. Значение этого адреса, соответствующее текущему положению на экране, будем обозначать далее через **ПК** (позиция курсора). Первоначально оно

равно **E800H**, т. е. начальному адресу ОЗУ дисплея. При каждом обращении к подпрограмме вывода это значение будет изменяться в соответствии с новым положением курсора на экране. Таким образом, вывод одного алфавитно-цифрового или графического символа сводится к записи его кода в ОЗУ страницы по адресу, равному **ПК**, увеличению на 1 содержимого

Таблица 2

00 -	20 - ПРОВОЛ	40 - S	60 - Ю
01 -	21 - !	41 - Я	61 - Я
02 -	22 - "	42 - В	62 - Б
03 -	23 - #	43 - С	63 - Ц
04 -	24 - \$	44 - D	64 - Д
05 -	25 - %	45 - E	65 - Е
06 -	26 - &	46 - F	66 - Ф
07 -	27 - '	47 - G	67 - Т
08 - <--	28 - (48 - H	68 - Х
09 -	29 -)	49 - I	69 - И
0A - ПС	2A - *	4A - J	6A - Й
0B -	2B - +	4B - K	6B - К
0C - ↖	2C - ,	4C - L	6C - Л
0D - БК	2D - -	4D - M	6D - М
0E -	2E - .	4E - N	6E - Н
0F -	2F - /	4F - O	6F - О
10 -	30 - 0	50 - P	70 - П
11 -	31 - 1	51 - Q	71 - Я
12 -	32 - 2	52 - R	72 - Р
13 -	33 - 3	53 - S	73 - С
14 -	34 - 4	54 - T	74 - Т
15 -	35 - 5	55 - U	75 - У
16 -	36 - 6	56 - V	76 - В
17 -	37 - 7	57 - W	77 - В
18 - -->	38 - 8	58 - X	78 - Ъ
19 - ↑	39 - 9	59 - Y	79 - Ы
1A - ↓	3A - :	5A - Z	7A - 3
1B -	3B - ;	5B - E	7B - Ш
1C -	3C - <	5C - \	7C - 3
1D -	3D - =	5D - ?	7D - Щ
1E -	3E - >	5E - ↑	7E - Ч
1F - СТР	3F - ?	5F - ←	7F -

ПРИМЕЧАНИЕ: ----- - ГРАФИЧЕСКИЙ СИМВОЛ

ПК и перемещению 1 в ОЗУ курсора на очередную позицию. При этом каждый раз проверяется, не достигнут ли конечный адрес ОЗУ дисплея (EFFFH), и если это так, то ПК вновь присваивается значение E800H.

Если же код управляющий, то содержимое ОЗУ страницы не меняется, а изменяется только содержимое пары ячеек ПК и происходит перемещение 1 в соответствующую ячейку ОЗУ курсора. Например, при получении подпрограммой управляющего кода «курсор вверх» (19H) происходят следующие операции: $ПК = ПК - ЧС$, где $ЧС = 64$ (число символов в строке), если новое значение ПК меньше E800H, то выполняется операция $ПК = ПК + РП$, где $РП = 800H$ (размер памяти дисплейного модуля).

Если подпрограммой получен код стирания информации с экрана (1FH), то область памяти от E800H до EFFFH

заполняется кодом символа «пробел» (20H), а в ПК заносится число E800H.

Перейдем теперь к описанию клавиатуры нашего дисплея. Заметим сразу, что мы стремились максимально упростить схему клавиатуры и сделать ее практически независимой от типа применяемых контактных устройств.

Каким же требованиям должна отвечать клавиатура дисплея? Прежде всего она должна формировать коды всех символов, приведенных в табл. 2. Кроме того, должна быть предусмотрена защита от одновременного нажатия на несколько клавиш и дребезга контактов. Все эти действия в нашем случае в основном возложены на программу.

На рис. 4 показана принципиальная электрическая схема клавиатуры. Она подключена к микро-ЭВМ с помощью ППА КР580ВВ55. Все три ка-

нала ППА настроены для работы в режиме 0, причем канал А — на вывод, а каналы В и С — на ввод информации. На микросхемах D2 и D3 собран дешифратор состояния адресной шины, формирующий сигнал ВМ для ППА. Диоды V1—V8 служат для защиты линий канала А от повреждения при одновременном нажатии на несколько клавиш.

Клавиатура представляет собой матрицу (7×8) нормально разомкнутых контактов и отдельную группу из трех контактов. Контакты замыкаются при нажатии на клавиши клавиатуры. На рис. 5 показано расположение клавиш, принятое в большинстве промышленных дисплеев.

В процессе сканирования (опроса) контактов клавиатуры подпрограмма обслуживания последовательно формирует нулевой уровень на каждой из линий порта А. При этом на других семи линиях формируются единичные уровни. Сразу после этого программа обслуживания считывает и анализирует содержимое порта В. Если ни одна из клавиш не нажата, то во всех разрядах порта В будут записаны единицы, так как на соответствующие линии через резисторы R1—R7 подано напряжение 5 В. Когда какая-либо клавиша будет нажата, то нулевой уровень с соответствующей линии канала А будет подан на одну из входных линий канала В. При этом необходимо помнить, что даже за самое кратковременное нажатие клавиши оператором микропроцессор способен неоднократно просканировать все контакты. Подпрограмма обслуживания определяет номер нажатой клавиши и формирует соответствующий 7-разрядный код.

При нажатии на каждую из клавиш могут формироваться три различных кода в зависимости от того, была ли одновременно нажата какая-либо из трех дополнительных клавиш модификации кода: РУС, УС и СС. Это позволяет сократить общее число клавиш. Клавиша РУС служит для формирования кодов русских букв, клавиша УС для формирования кодов различных управляющих и графических символов, а клавиша СС — для формирования кодов специальных символов. Клавиша РУС должна быть с фиксацией. Клавиши модификации кодов подключены к порту С через RS-триггеры, устраняющие дребезг контактов. Дребезг остальных контактов устраняется программно.

Подпрограммы обслуживания дисплея и клавиатуры входят в основную управляющую программу микро-ЭВМ, которая будет описана в одной из следующих статей.

Г. ЗЕЛЕНКО,
В. ПАНОВ, С. ПОПОВ

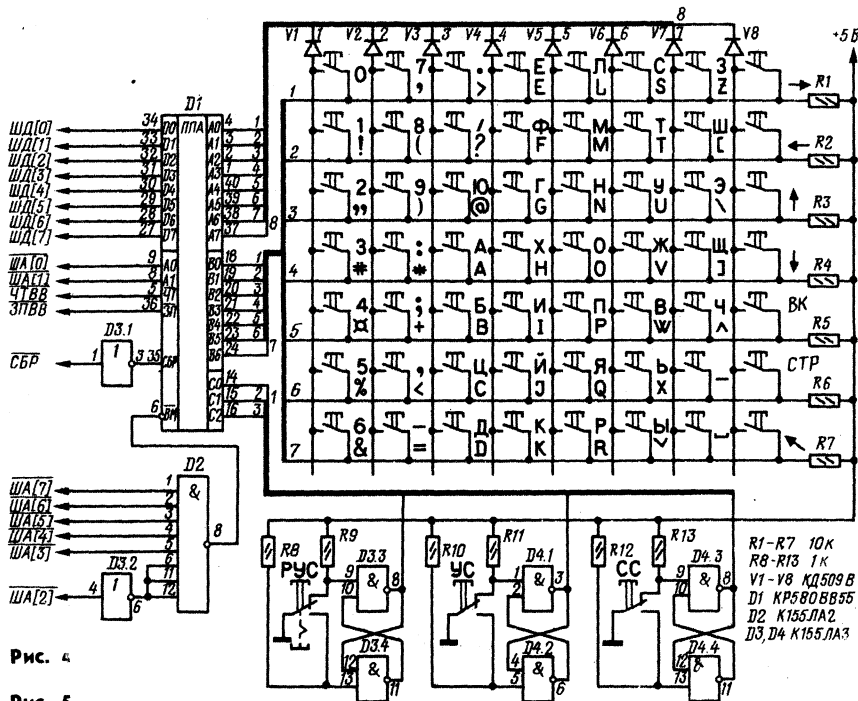


Рис. 4

Рис. 5

