***56.Типы устройств отображения информации в МПС. Сегментные и матричные табло. Принципы статической и динамической индикации.***

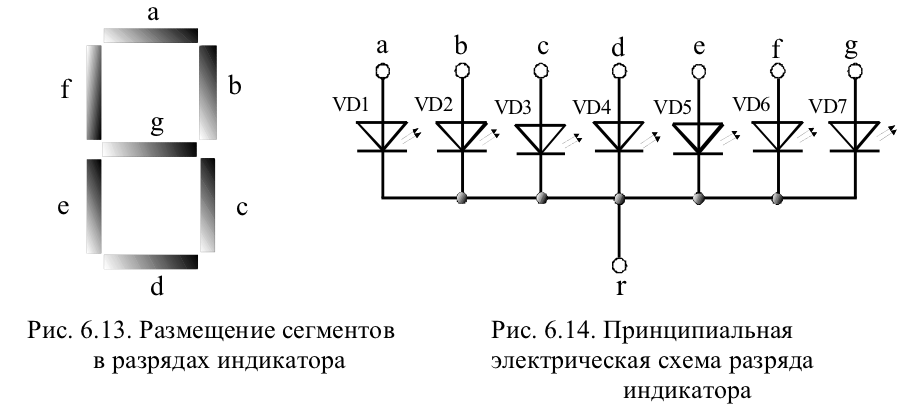
В микропроцессорных системах отображение информации осуществляется с помощью светодиодных или жидкокристаллических индикаторов. В зависимости от конфигурации отдельных сегментов и их взаимного размещения различают единичные, сегментные и матричные индикаторы.

В качестве единичного индикатора чаще всего используется светодиод. Его состояние «включен» обеспечивается пропусканием тока IД силой в несколько миллиампер (обычно 1–10 мА). Поскольку нагрузочная способность портов МК 25 мА, то светодиод можно подключать без дополнительного усилителя непосредственно к порту, предусмотрев лишь токоограничивающий резистор.

Из сегментных индикаторов ограничимся рассмотрением наиболее распространенного варианта – семисегментные индикаторы, предназначенные для отображения цифровых десятичных и шестнадцатеричных данных.

На рис. 6.13 показано размещение и условное обозначение светодиодных сегментов в цифровых разрядах индикаторов. Для отображения какой-либо цифры соответствующую комбинацию сегментов активизируют, пропуская через них ток. Величина рабочего тока сегмента составляет несколько миллиампер.

На рис. 6.14 показана принципиальная электрическая схема одноразрядного семисегментного светодиодного индикатора, приведен вариант схемы с общим катодом. Промышленность выпускает также варианты схем с общим анодом. (В дальнейшем все пояснения будем проводить для варианта индикатора с общим катодом без дополнительных оговорок). В процессе работы ток общего вывода r индикатора в 7 раз больше, чем ток одного сегмента. В схеме подключения индикатора к МК при недостаточной нагрузочной способности портов в цепях управления общими выводами разрядов индикатора необходимо предусматривать усилители.

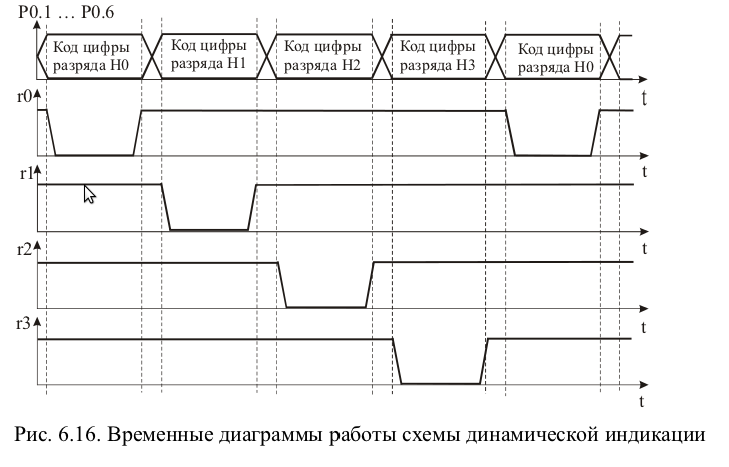
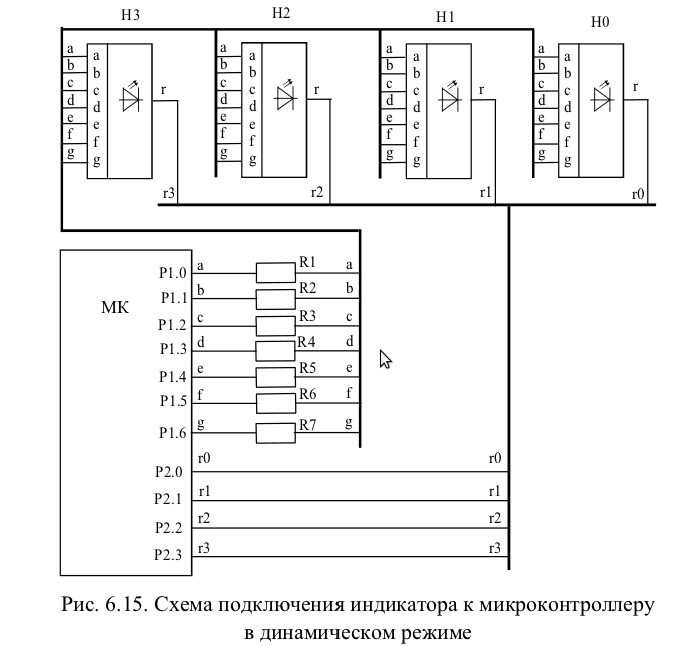


Используются два способа управления сегментными индикаторами: статический и динамический.

При статическом способе на каждый цифровой разряд индикатора предусматривается свой регистр, в регистры записываются и хранятся на все время отображения семисегментные коды соответствующих цифр. Выходы регистров соединяются через токоограничивающие резисторы с входами соответствующих сегментов разрядов индикатора, общие выводы разрядов индикатора подключаются к общему проводу. Способ отличается простотой программного обслуживания, но требует повышенных аппаратных затрат (регистров и резисторов), применяется для малоразрядных индикаторов.

При большом количестве разрядов применяют динамический способ управления индикатором. Способ характеризуется минимальными аппаратными затратами, но сравнительно сложной программной поддержкой.

Схема подключения 4-разрядного индикатора в динамическом режиме к микроконтроллеру показана на рис. 6.15. В схеме одноименные сегменты всех разрядов объединяются и обслуживаются семью выводами одного из портов микроконтроллера. Общие выводы разрядов индикатора обслуживаются индивидуально четырьмя выводами другого порта микроконтроллера. Первый порт обеспечивает возбуждение определенной для каждой цифры комбинации сегментов, а второй порт активизирует тот или иной разряд индикатора. Суть динамического управления индикатором сводится к поочередному циклическому возбуждению разрядов индикатора. Если разряды индикатора будут подсвечиваться с частотой выше 25 Гц, то в силу инерционности человеческого зрения мерцания изображения заметны не будут. Временные диаграммы работы схемы динамической индикации приведены на рис. 6.16.

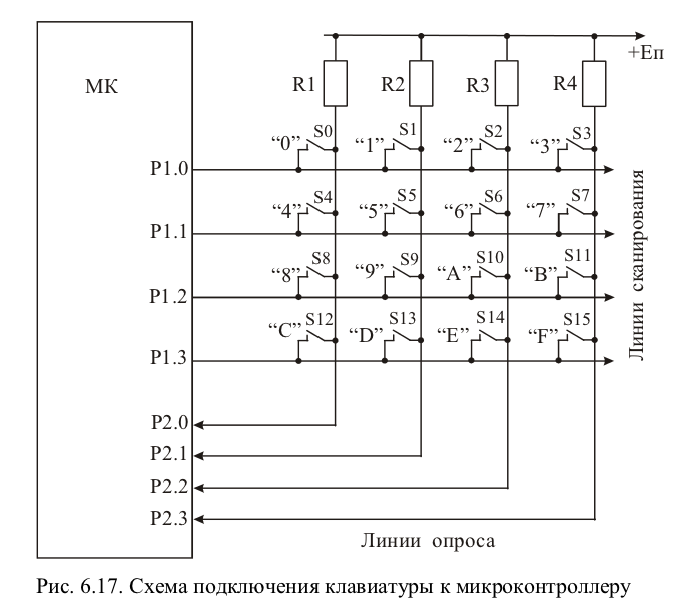


***57.Типы клавиатур и схемы их подключения. Алгоритмы опроса клавиатуры.***

Функцию ввода данных с клавиатуры в микропроцессорных системах реализуют как отдельный самостоятельный аппаратно-программный модуль. На него отводятся определенные аппаратные и программные ресурсы микроконтроллера.

Для небольшого количества клавиш для обслуживания каждой из них выделяется отдельный вывод микроконтроллера.

Для большого количества клавиш используют матричную схему, которая требует значительно меньшего количества выводов микроконтроллера. Схема подключения к микроконтроллеру (МК) 16-кнопочной клавиатуры показана на рис. 6.17. Как видно, для обслуживания 16 кнопок требуется всего лишь 8 выводов МК. Выводы P1.0 ... P1.3 микроконтроллера настраиваются как выходы и являются линиями сканирования состояния клавиатуры. Выводы Р2.0 ... Р2.3 настраиваются как входы и являются линиями опроса состояний кнопок клавиатуры. Линии сканирования образуют строки, а линии опроса – столбцы матрицы клавиатуры. Потенциалы линий опроса с помощью резисторов R1 ... R4 подтягиваются до уровня напряжения питания. При опросе состояния клавиатуры при незамкнутых клавишах на этих линиях читаются логические единицы. В случае нажатия какой-либо клавиши происходит электрическое замыкание линий строки и столбца, на пересечении которых находится нажатая кнопка. Если при этом на эту линию строки выведен логический ноль, то на линии столбца с нажатой кнопкой будет читаться также логический ноль. Таким образом, при опросе состояния клавиатуры микроконтроллер по линиям опроса будет читать для каждой нажатой кнопки свой уникальный код возврата KW, который может быть впоследствии преобразован в двоичный код нажатой кнопки KNK. Код возврата не чувствителен к строке, в которой находится нажатая кнопка, поэтому необходима корректировка KNK. Она может быть выполнена по формуле KNK = KNK + 4 i, где i – номер строки с нажатой кнопкой, принимает значения от 0 до 3.



*Алгоритм контроля и ожидания освобождения буфера клавиатуры* может базироваться на контроле значения флага буфера клавиатуры FBK. Если FBK = 0, то буфер пуст, если FBK = 1, то буфер занят. Процедура KLAW заносит в буфер код нажатой клавиши, возводит в единицу FBK, а другие процедуры забирают оттуда код и сбрасывают флаг в ноль. При этом пока флаг остается возведенным в единицу, процедура KLAW игнорирует нажатие кнопок.

*Определение факта нажатия кнопки* удобно сделать через дополнительную переменную – флаг нажатия кнопки – FNK, которая отражает текущее состояние кнопок: нажата – FNK = 1 или не нажата – FNK = 0, как показано на рис. 6.18. Производя логический анализ двух соседних значений FNK (текущего FNKI и предыдущего FNKJ), легко фиксировать момент нажатия кнопок. Нажатие будет иметь место, если выполняется условие FNKi ∧ FNKj = 1 .



Значения FNK можно определить по следующему алгоритму:

- по линиям сканирования выставить логические нули;

- прочитать код возврата KW на линиях опроса;

- если KW = F, то FNK = 0, кнопки не нажаты;

- если KW ≠ F, то FNK =1, одна или более кнопок нажаты.

*Устранение дребезга контактов* лучше всего делать путем опроса состояния клавиатуры с частотой (5 ... 20 Гц). При этом все переходные процессы обычно завершаются до следующего опроса состояния кнопок. Периодический опрос клавиатуры лучше всего организовать по прерываниям от переполнения счетчика-таймера. В реальных микропроцессорных системах опрос клавиатуры 126

обычно совмещают в цикле с обслуживанием индикатора в динамическом режиме. *Определение кода нажатой кнопки* KNK проводят методом последовательной проверки гипотез: нажатая кнопка находится на линии 0-й строки, 1-й строки, 2-й строки, 3-й строки. Для проверки, находится ли нажатая кнопка в i-й строке (где i = 0 ... 3), на ней выставляется логический нуль, а на всех других – логические единицы. Читают код возврата KW на линиях опроса. Если KW = F, то нажатая кнопка не в этой строке. Если KW ≠ F, то нажатая кнопка в этой строке. Далее преобразуют позиционный код нажатой кнопки в двоичный по правилам: если KW = E, то KNK = 0; если KW = D, то KNK = 1; если KW = B, то KNK = 2; если KW = 7, то KNK = 3. И последний шаг этой процедуры – корректировка KNK с учетом веса строки с нажатой кнопкой по формуле KNK = KNK + 4 i, где i – номер строки с нажатой кнопкой. *Занесение в буфер кода нажатой кнопки* обеспечивается операцией BK = KNK. *Возведение флага буфера клавиатуры* реализуется операцией FBK = 1.

1. ***Понятие интерфейса. Параллельная и последовательная передача данных.Синхронный и асинхронный обмен данными.***