

# Аппаратные хитрости использования PIC-контроллеров

© Александр Торрес  
[altor@geocities.com](mailto:altor@geocities.com), 2:461/28

В этой статье описаны некоторые схемотехнические приемы, полезные при разработке приборов на базе PIC-контроллеров. Некоторые из них конечно можно применять и с другими микроконтроллерами. Разумеется, для некоторых опытных специалистов кое-что может показаться "прописными истинами", но думаю большое число начинающих работать с микроконтроллерами, узнают для себя что-то новое, да и опытным будет и полезно.

## Дежурный режим

Основное преимущество PIC-контроллеров заключается в их низком потреблении при высоком быстродействии, что очень полезно в приборах с автономным питанием. Еще большей экономии энергии можно достичь при использовании дежурного режима. При этом основное время микроконтроллер находится в режиме sleep, при котором приостанавливается выполнение программы, выключается тактовый генератор, но может продолжать свою работу Watch Dog таймер и некоторые периферийные устройства. Выход из этого режима может производиться по-разному, в зависимости от типа контроллера.

Например, можно запрограммировать Watch Dog таймер ("собаку") на определенный период, при срабатывании которого контроллер "проснется", опросит свои входы и, если нужно, перейдет к процедуре обработки. Если никаких действий совершить не нужно, он выполнит команду SLEEP и будет находиться в этом состоянии до следующего срабатывания таймера. При выполнении процедуры обработки следует периодически сбрасывать "собаку" для предотвращения ее повторного срабатывания. Используя подобную методику, можно строить приборы, не требующие выключателя питания, различные сигнализирующие и контрольные приборы.

Очень часто, прибор должен быть выполнен в виде пульта (к примеру - пульт ДУ телевизора), который большую часть времени находится в выключенном состоянии, а при нажатии кнопок управления производить соответствующие действия. Реализовать режим "пульт" можно несколькими способами, один из которых описан выше. Период "собаки" в этом случае следует выбирать в диапазоне 0.05-0.1 секунды.

Другие способы не основаны на использовании "собаки", т.е. контроллер не просыпается для опроса кнопок, а остается спать до их нажатия.

В PIC-контроллерах старого семейства (16C5х), для выхода из режима sleep нужно использовать специальную схемотехнику и программирование. При этом обеспечивается подача сигнала "сброс" на контроллер при нажатии любой кнопки. Рисунок 1.

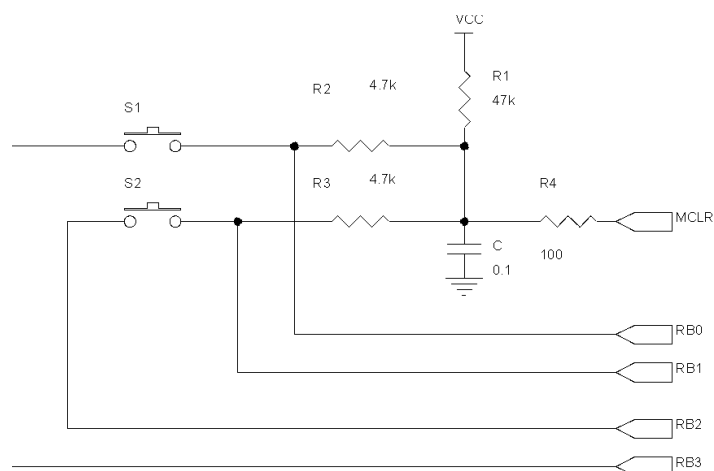


РИС 1.



В дежурном режиме выходы RB2 и RB3 находятся в состоянии "0", конденсатор С заряжен до уровня "1". При нажатии любой кнопки конденсатор быстро разряжается через резистор R2 или R3, что приводит к подаче сигнала "сброс". При сбросе, выходы переходят в высокоимпедансное состояние, и разряд конденсатора через R2 (R3) прекращается. Время разряда должно быть больше требуемого для сброса контроллера (около 18мс) но меньше минимальной длительности нажатия на кнопку (примерно 50-100мс). После снятия сигнала "сброс" (заряде конденсатора через R1) контроллер начинает выполнение программы. При этом на выходы RB2 и RB3 "0", на короткое (около 10мс) время, за которое конденсатор не успевает разрядиться, а с входов RB0 и RB1 считывается состояние кнопок. Аналогично можно подключить матрицу клавиатуры, рисунок 2.

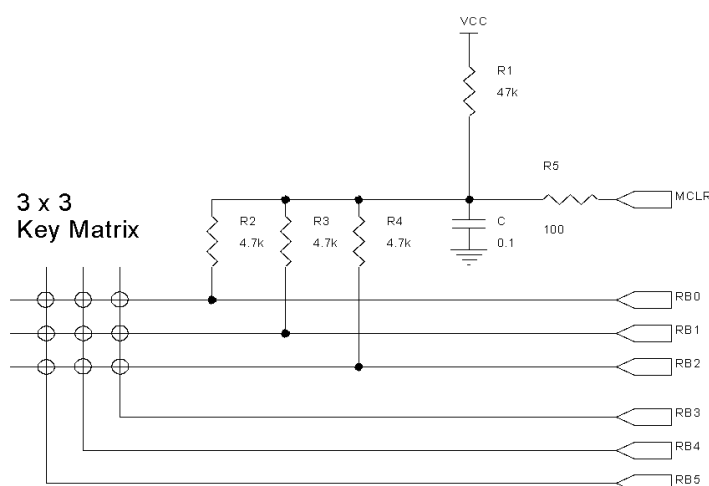
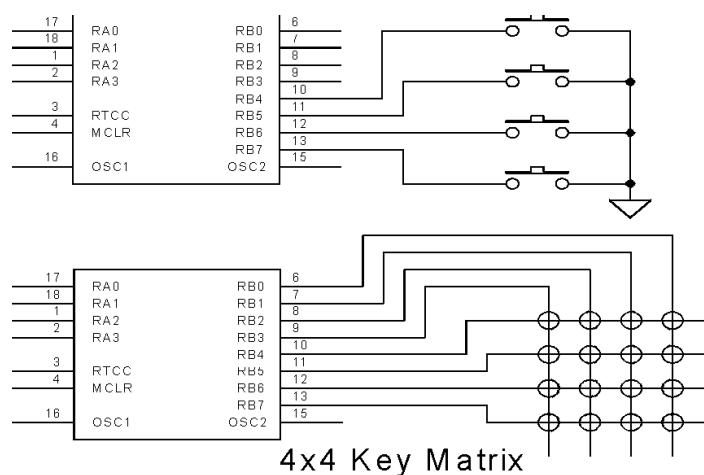


РИС 2.

Более новые PIC-контроллеры имеют возможность генерировать прерывание, выводящее из спящего режима при изменении состояния определенных входов, обычно RB4-RB7. При этом возможно как применение нескольких кнопок, так и матрицы клавиатуры (Рисунок 3.) Выводы RB4-RB7 запрограммированы на ввод, с включенными внутренними подтягивающими резисторами. Перед входом в режим sleep включается генерация прерываний по изменению состояния (в случае матрицы - выходы RB0-RB3 устанавливаются в "0"). При нажатии любой кнопки, вызывается прерывание, выводящее контроллер из режима sleep, после чего он может приступить к процедурам подавления дребезга контактов, определения нажатой кнопки (сканирования матрицы) и выполнению соответствующих нажатой кнопки действий. Работа "прерывания по изменению" имеет свои некоторые особенности, в связи с которыми его следует применять только для выхода из спящего режима, а при нормальной работе оно должно быть выключено во избежание ложных срабатываний.



4x4 Key Matrix  
РИС 3.



## Экономия выводов

Чего всегда не хватает при использовании микроконтроллеров, особенно в небольших корпусах? Числа портов ввода-вывода. Конечно, можно применить контроллер в 40-выводном корпусе, но это не всегда возможно по различным соображениям. К счастью существует ряд схемотехнических уловок, позволяющих многократно использовать одни и те же выводы для разных целей, или для управления несколькими устройствами. Рассмотрим некоторые из них.

### управление 2-мя светодиодами одним выходом

В отличие от микроконтроллеров семейства MCS-51, имеющих выходы с открытым стоком и не требующих переключения режима, в PIC-контроллерах выходные каскады активные, и для переключения режима работы (ввод или вывод) служит специальный регистр. Таким образом, каждый вывод любого порта может принимать 3 значения - "0", "1" и "вход", который эквивалентен высокоимпедансному ("Z") состоянию. Это позволяет управлять двумя светодиодами при помощи единственного вывода. Рисунок 4.

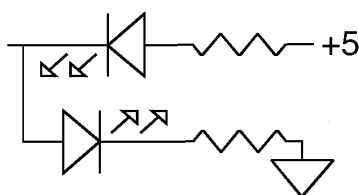


РИС 4.

При работе порта в режиме выхода, в зависимости от состояния "0" или "1" горит соответственно верхний или нижний светодиод. При переключении в режим входа, последний находится в Z-состоянии, и при соответствующем выборе резисторов ток через светодиоды весьма мал и их свечение почти незаметно.

### управление 6-ю светодиодами 3-мя выходами

Еще более интересно включение трех пар встречно-параллельных светодиодов по схеме "звезда", рисунок 5.

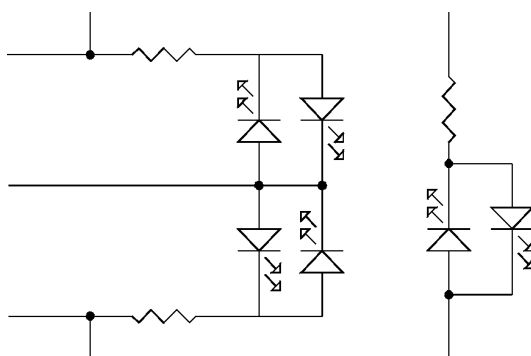


РИС 5.

Соответствующими сигналами микроконтроллера можно добиться свечения любого из шести светодиодов, а также их комбинаций. Используя динамическую индикацию, можно получить любые комбинации.

### Ввод аналогового сигнала с выводом стробов

В PIC-контроллерах со встроенным АЦП, одни и те же выводы могут использоваться как аналоговый вход, цифровой вход или цифровой выход. Возможно одновременное использование таких выводов для разных целей. В качестве примера, на рисунке 6. показано подключение 7-сегментных индикаторов, при котором выходы сканирования используются как аналоговые входы.



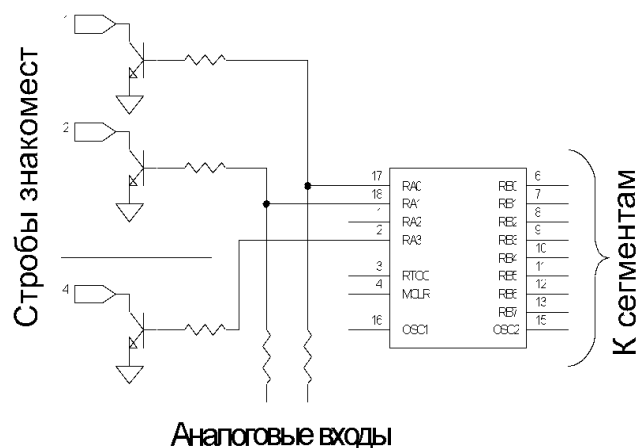


РИС 6.

В процессе сканирования индикатора, выходы, генерирующие стробы знакомест кратковременно переключаются в режим аналогового ввода. Необходимая развязка осуществляется резисторами. Для исключения влияния транзисторных ключей, желательно применение полевых транзисторов.

## Расширение числа входов-выходов:

### мультиплексорами

При необходимости опроса большого количества цифровых входов, если позволяют временные характеристики, удобно использовать мультиплексоры. К примеру, для ввода 8 сигналов при помощи 561КП2 требуется всего 4 вывода контроллера вместо 8. В случае использования контактных датчиков, подтягивающий резистор может быть один. На рисунке 7 показано подключение 4-х кнопок, 3-х позиционного переключателя и логического сигнала TTL/CMOS уровня.

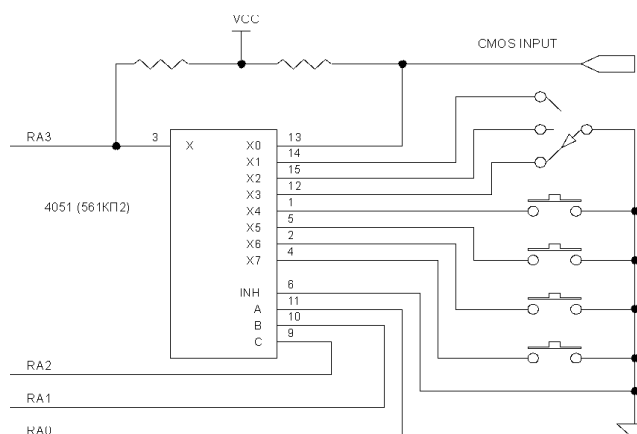


РИС 7.

### сдвигowymi регистрами

Для экономии выводов при выводе можно использовать сдвиговые регистры, для управления которыми требуется 2 сигнала (данные и строб) и, при необходимости, сигнал защелкивания. На рисунке 8 для простоты показано использование сдвигового регистра 4015 (561ИР2) и защелок 4042 (561ТМ3) для вывода восьми разрядов, но, разумеется, выгоднее использовать более современные регистры с защелками, например великолепный 8-разрядный регистр 74НС595.



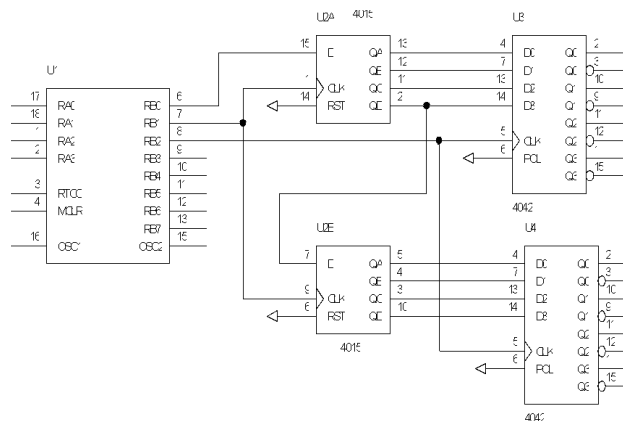


РИС 8.

Этот способ, как и ввод через мультиплексор, никакой особой хитрости не представляет, поэтому рассмотрим более интересный способ вывода. Иногда, когда экономия выводов важнее быстродействия, можно использовать ОДИН вывод для полного управления сдвиговым регистром, используя широтно-импульсную модуляцию. На рисунке 9. показан фрагмент схемы для сдвигового регистра, со сдвигом и защелкиванием по положительному фронту.

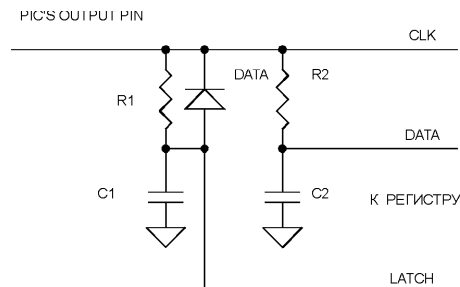


РИС 9.

Выход микроконтроллера постоянно находится в состоянии "0". Подробно процесс записи показан на рисунке 10. Диаграмма а) отображает сигнал на выходе контроллера. Информационный "0" передается коротким импульсом и длинной паузой, "1" передается наоборот - длинным импульсом при короткой паузе. На диаграмме б) показано напряжение на конденсаторе C2. Постоянная времени R2C2 выбрана таким образом, чтобы это напряжение логически соответствовало сигналу диаграммы с) на входе данных. Состояние входа данных, в момент сдвига отмечено крестиком.

Аналогично формируется сигнал защелкивания (LATCH) цепочкой R1C1 и диодом, ускоряющим разряд C1. Соответствующие диаграммы д) и е).

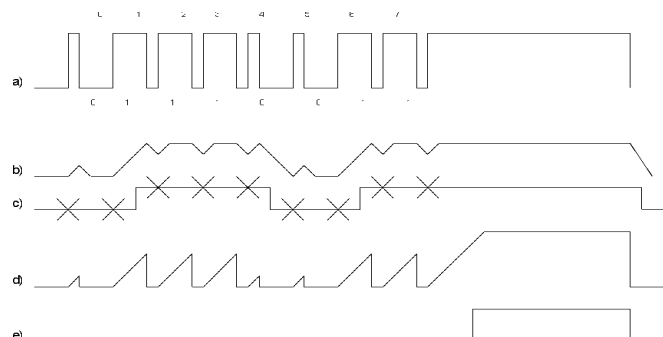


РИС 10.

### Интерфейс с высоковольтными сигналами

Очень часто требуется вводить сигналы с напряжением, многократно превышающим напряжение питания контроллера и, соответственно, допустимые логические уровни. Это может быть ввод с RS232, рассмотренный ниже, или, к примеру, ввод сетевого напряжения, с целью



измерения его частоты, фазы или синхронизации с ним. Благодаря наличию защитных диодов на всех выводах PIC-контроллеров, это не составляет большого труда. (Рисунок 11.)

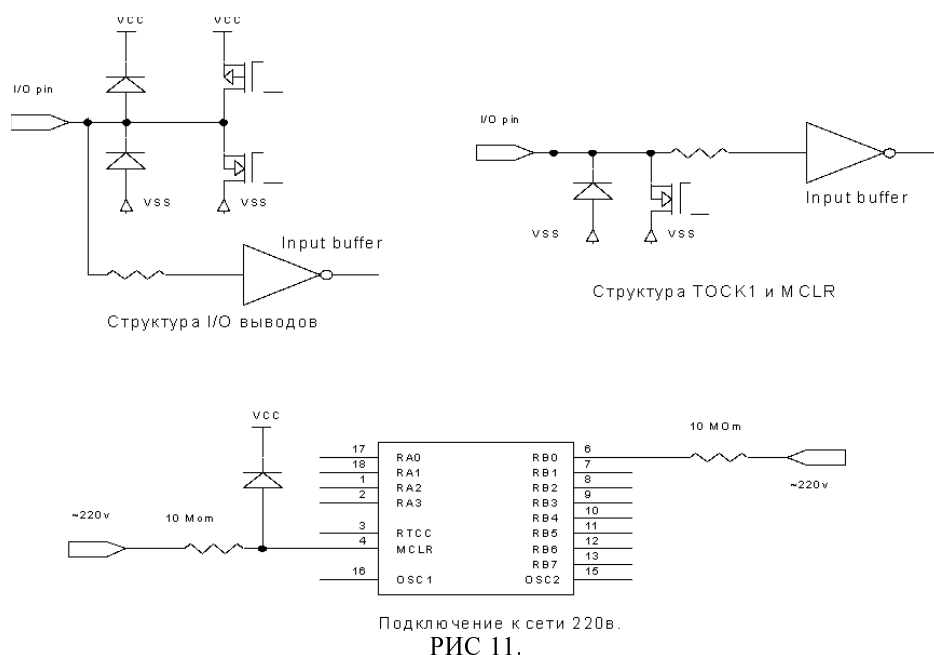


РИС 11.

Для согласования высоковольтного напряжения с логическим входом служит резистор, ограничивающий ток через защитные диоды на безопасном уровне (для особо ответственных применений желательно этот резистор составить из двух последовательных резисторов, для защиты микроконтроллера в случае пробоя резистора). При оценке точности состояния логического сигнала (в момент перехода через ноль) следует учитывать входную емкость (5пФ) и сопротивление защитного резистора, которое обычно принимается 10 МОм для напряжения 220в (при этом ток ограничивается на уровне примерно 30мкА). Выводы MCLR или T0CK1 имеют только один защитный диод, соединенный с общим проводом, поэтому при подаче на эти выводы высоковольтных сигналов, кроме резистора нужно включить дополнительный защитный диод к питанию.

## Интерфейс с RS232C

Основная задача при соединении устройства на микроконтроллере с компьютером (или другим оборудованием) через RS232 заключается в преобразовании уровня сигналов TTL/CMOS (0-5в) в биполярный сигнал и обратно. При этом очень часто в приборе отсутствует источник повышенного положительного и отрицательного напряжений. Решить эту проблему можно несколькими способами, наиболее удобные из которых описаны ниже.

### использование ADM202

В настоящее время существует множество микросхем-преобразователей уровня, но наиболее удобно применение микросхем, имеющих внутри преобразователь напряжения. Таким микросхемам требуется питание только +5в, от которого питается сам микроконтроллер. Необходимые для RS232 напряжения (обычно +10 и -10) вырабатываются емкостным преобразователем, для чего используются внешние конденсаторы (обычно 4 конденсатора, емкостью 0.1 или 1.0 мкф). Такие микросхемы выпускаются различными фирмами, с различным количеством входов-выходов и разным быстродействием. Типичный представитель - микросхема ADM202 фирмы Analog Devices (функциональный аналог микросхемы MAX202 производства MAXIM). Она имеет по две линии ввода-вывода. Пример использования показан на рисунке 12.



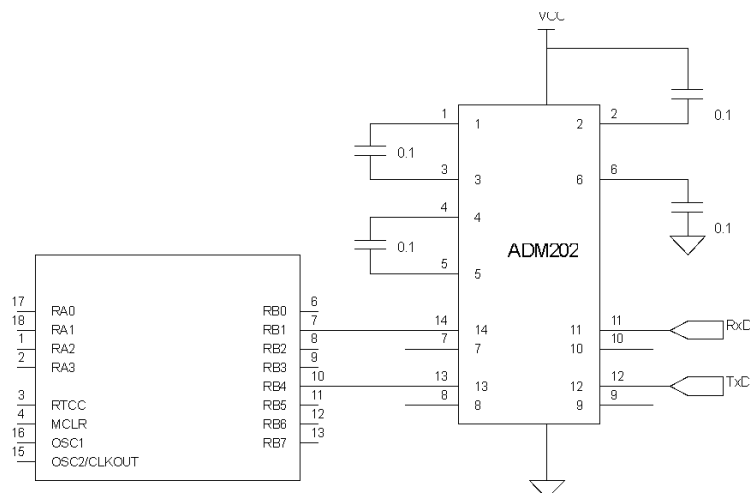


РИС 12.

### схема на транзисторе

Во многих случаях не требуется работа на больших скоростях и на большие расстояния, и достаточно только сигналов передачи данных (TxD и RxD). В этом случае можно воспользоваться простой схемой на одном транзисторе, для преобразования уровня и избавиться от необходимости иметь источник отрицательного напряжения или преобразователь. Для сигнала "0" вполне достаточно напряжения +5в, а для сигнала "1" используется отрицательное напряжение с выхода компьютера. Это напряжение через развязывающий диод заряжает конденсатор, который служит источником отрицательного напряжения независимо от сигнала на выводе TxD компьютера. Для ввода сигнала в контроллер используется наличие защитных диодов на входах самого контроллера, что позволяет вводить биполярный сигнал просто через резистор. Схема данного варианта сопряжения показана на рисунке 13. При экспериментах можно вообще отказаться от преобразования уровня выхода, т.к. почти все современные последовательные порты компьютеров могут воспринимать однополярный сигнал с TTL/CMOS уровнями, но применять это в серийных приборах не рекомендуется, так как этот режим не соответствует стандартам, и могут возникнуть проблемы при работе с другим компьютером.

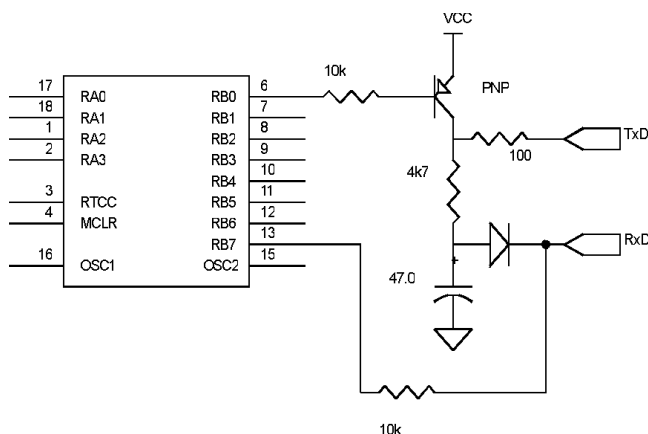


РИС 13.

### Расширение памяти

#### Программной - использование интерпретаторов:

Небольшой размер программной памяти вместе с невозможностью использования внешней памяти программ очень часто является большим ограничением PIC-контроллеров младшего и среднего уровня, несмотря на довольно компактные программы, особенно, если они пишутся на ассемблере. Там, где не требуется большое быстроедействие, в программную память микроконтроллера можно записать базовые подпрограммы и интерпретатор, а саму программу разместить во внешней памяти с последовательным доступом. Это может быть Форт-



интерпретатор, Бейсик-интерпретатор, или вообще нечто свое. Наружную память "программы" при этом можно наращивать в принципе до бесконечности. Возможно также размещение во внешней последовательной памяти различных таблиц данных (разумеется, это возможно и без интерпретатора, но с учетом увеличенного времени доступа).

### Рабочей:

Ячейки ОЗУ (или регистры) подобны программной памяти - их тоже часто не хватает, а иногда требуется просто хранить большие массивы данных. Поэтому применяются различные способы решения этой проблемы, кроме конечно иногда вполне справедливого, но не рассматриваемого здесь способа "смени микроконтроллер".

## Расширение ОЗУ последовательными EEPROM

В некоторых случаях, когда не хватает внутренних регистров микроконтроллера, время обращения не обязательно мало и запись осуществляется нечасто, есть смысл использовать последовательные EEPROM с шиной i2c или SPI, даже тогда, когда не требуется хранить данные при отсутствии питания. Микросхемы серий 24Cxx и 93Cxx очень доступны и относительно дешевы, чего не скажешь про микросхемы ОЗУ с последовательным доступом. Необходимо только иметь в виду, что количество циклов перезаписи EEPROM ограничено, в остальном же применение последовательных EEPROM сложностей не вызывает. Рисунок 14.

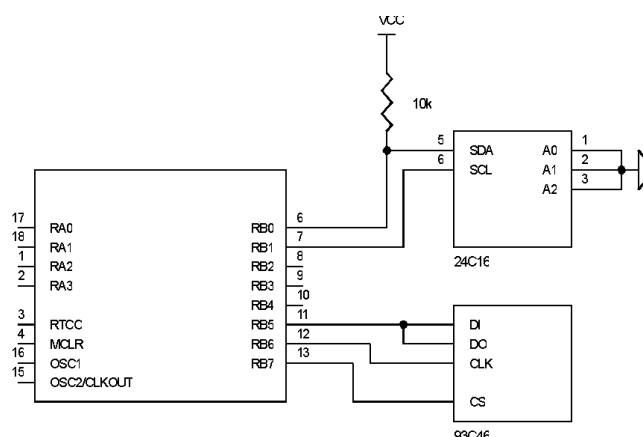


РИС 14.

### Замечания по работе с i2c:

Работа с шиной i2c имеет некоторые особенности. По полному стандарту i2c управление нужно осуществлять двунаправленными выводами, которые при работе на выход имеют открытый коллектор (сток) с подтягивающими резисторами. Это связано с вопросами арбитража и синхронизации. Это выполняется в семействе MCS-51, но не выполняется в PIC-контроллерах, выходы которых активны как в состоянии "0", так и в состоянии "1" (за исключением одного выхода, имеющего открытый сток). Этот вопрос можно решить применением "виртуального открытого стока". Обычно вывод переключают или на вход или на выход регистром TRIS и при выводе просто записывают "0" или "1" в регистр данных. При "виртуальном открытом стоке" в регистре данных постоянно записан "0", сам вывод подтянут резистором к питанию. И при вводе и при выводе "1" вывод находится в режиме входа, и переключается регистром TRIS только при выводе "0". Нужно иметь в виду, что такая работа нужна только при необходимости соблюдения стандартов i2c, в основном в multi-master среде или с устройствами, которые могут выставлять неготовность по линии синхронизации. При обычной работе с последовательной памятью этого, как правило, не требуется.

## Параллельным ОЗУ

Разумеется, при использовании контроллеров с соответствующим количеством выводов, возможно применение обычных микросхем статической или динамической памяти, имитируя портами контроллера шины адреса, данных и управления. Динамическая память имеет большую емкость и позволяет сократить число выводов контроллера, за счет мультиплексирования шины адреса и использования 1-битной или 4-х битной шины данных, но требует регенерации. Очень



выгодно ее использование в приборах типа "самописец", при условии, что при записи или воспроизведении их частота обеспечивает требуемый период регенерации. В режиме хранения регенерацию можно производить холостым воспроизведением или использовать внутреннюю регенерацию, имеющуюся во многих микросхемах динамической памяти.

### **"Интеллектуальный ключ"**

Существует большое количество переносных приборов, в которых по каким-либо причинам нежелательно иметь тумблер или кнопку питания с фиксацией. Удобнее включение-выключение производить или нажатием любой кнопки или специальной кнопки "Вкл" (не фиксируемой). Реализовать это можно различными способами, например, с использованием дежурного режима, но очень часто подобная аппаратура питается от одной-двух пальчиковых батарей через преобразователь, и в целях экономии хочется иметь именно ключ, отключающий все или почти все. Для ключевого элемента лучше всего использовать низковольтный полевой транзистор с малым падением напряжения, но они достаточно дороги и дефицитны, поэтому в случае питания от 2-х и более батареек в качестве самого ключа можно использовать.... другой микроконтроллер, например дешевый 16C505 (0.49\$) или восьминожку 12C508. Рисунок 15.

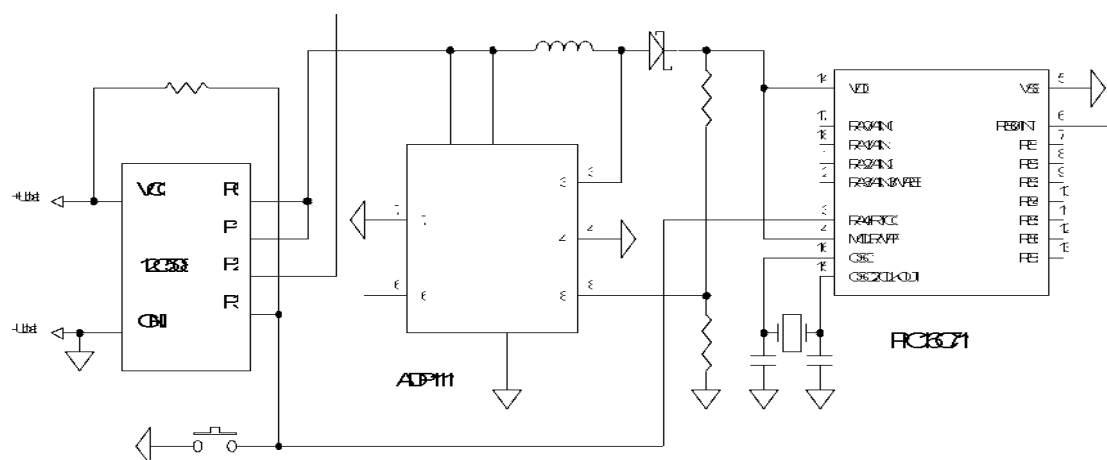


РИС 15.

"Ключевой" контроллер работает от внутреннего генератора, или от внешней RC-цепочки, если нужна его пониженная тактовая частота для уменьшения потребления и находится в спящем дежурном режиме. В качестве "выхода" ключа используется один или несколько выходов, соединенных вместе. Один вход можно использовать для кнопки "Вкл.", на другой можно подать сигнал с основного микропроцессора, например для программного выключения. Вообще же, если у "ключевого" контроллера еще остаются свободные выводы то ему можно поручить еще какую-нибудь работу во включенном состоянии прибора, например - выполнять роль еще одного Watch Dog, или генерировать импульсы прерывания на основной микропроцессор. В общем, на что хватит фантазии. В приведенной схеме выход основного контроллера RA4 с открытым стоком используется для блокировки включенного состояния, а выход P2 вспомогательного - для генерации прерываний.

### **Микро потребление**

В аппаратуре с батарейным питанием необходимо экономить каждый миллиампер-час батарей. Это легко достигается благодаря низкому потреблению самих микроконтроллеров, возможности использования режима sleep и т.д., но при конструировании нужно учитывать различные мелочи.

Так для уменьшения потребления, нужно использовать микроконтроллеры с буквой "L" в обозначении, на как можно меньшей тактовой частоте и по возможности с низким напряжением питания. При наличии свободных выходов, периодически использующиеся дополнительные устройства (например, последовательные EEPROM) нужно запитывать сигналом "1", включая его только на момент обращения к устройству. Рисунок 16.



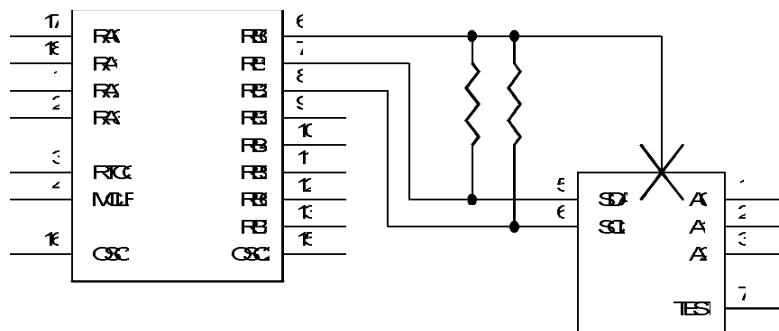


РИС 16.

Все неиспользованные выводы портов следует перевести в режим вывода, в режиме sleep нужно перевести все выходные сигналы в состояние с наименьшим током в нагрузке, выключить неиспользуемые внутренние периферийные устройства (например, встроенные АЦП), все входные сигналы должны находиться в четких логических уровнях, не допускать висящих в воздухе входов или подачи на цифровые входы напряжений ниже логической "1" или выше логического "0". (последнее относится ко всем без исключения CMOS микросхемам, не только к микроконтроллерам).

Не следует забывать, что микросхемы 16C84 и 16LC84 имеют ошибку, в результате которой у них повышенное потребление в спящем режиме, поэтому нужно использовать только 16F84 и 16LF84, у других PIC-контроллеров эта ошибка отсутствует.

Если используется Watch Dog и режим sleep, то в спящем режиме можно устанавливать максимально допустимый в данной задаче период, что уменьшает средний потребляемый ток.

При работе в RC-режиме есть возможность переключения тактовой частоты "высокая низкая", как показано на рисунке 17.

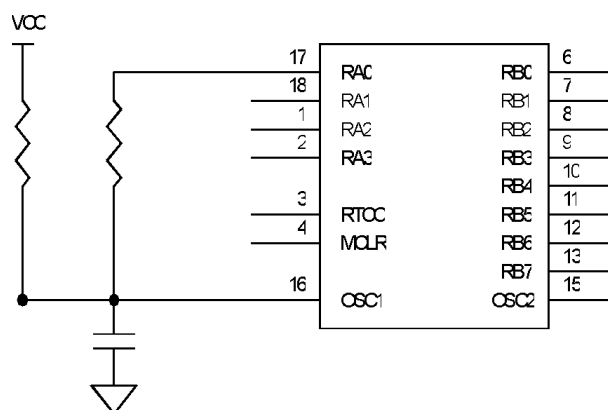


РИС 17.

Это полезно когда от микроконтроллера не все время требуется высокое быстродействие. Вывод, обозначенный "I/O pin" находится все время в режиме ввода (высокоимпедансное состояние) и только на определенное время переключается в режим вывода "1", при этом постоянная времени RC-цепочки уменьшается и тактовая частота увеличивается.

Напоследок приведу похожее решение для кварцев. На рисунке 18 показано подключение двух кварцев - часового 32768 Гц. (LOW) и более высокочастотного (HIGH), коммутируемых микроконтроллером через аналоговый ключ. Следует заметить, что сам я эту схему не проверял, но авторы утверждают, что она у них работает. Возможно, потребуется экспериментальное определение сопротивления резисторов и емкости конденсаторов для конкретных кварцев.



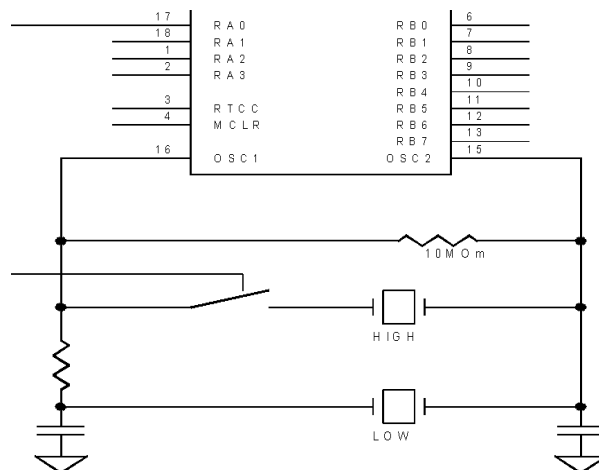


РИС 18.

В качестве часовых нужно применять старые отечественные "ромбики" или импортные "спички", отечественные "спички", как правило, очень плохо работают (а часто вообще не работают) в генераторах PIC-контроллеров.

Литература и полезные ссылки:

Microchip Microcontrollers Databook

Embedded Control Handbook

Microchip World Magazine

<http://www.geocities.com/SiliconValley/Way/5807>

<http://www.geocities.com/SiliconValley/Lab/6311>

г. Харьков.