# Проект на PIC микроконтроллере PIC16C84 (или PIC16F84)

[Яндекс.Директ](http://direct.yandex.ru/?partner)[Все объявления](http://direct.yandex.ru/search?from=http://www.rlocman.ru/shem/schematics.html%3Fdi=52951&ref-page=1124)

|  |
| --- |
| [Сторожевой таймер WatchDog](http://an.yandex.ru/count/Tk_TvZ-y1py40000Zh1zW0m5N8oL0fi1RaCh0OYzN1Ym0ecYgjHmc0UU0vsj7tC3dBM1Wr01bgsd2TYQfjY8YfAeIAABfYgAgNJdDAe1fQXoTG6HlmnS-GQFlHjP30UViNHOoW6JWWgP1aACa4mTe90WJg-GJ1sKdCOafvcF0QYgt0uHfB00000xhlqoRGn2EQ3G1B2g9BWAiG5_4G00?stat-id=3&test-tag=4458513)Автоматическая перезагрузка компьютера при зависании.Незаменим для серверов[Адрес и телефон](http://an.yandex.ru/count/Tk_TvXHIiIW40000Zh1zW0m5N8oL0fi1RaCh0OYzN1Ym0ecYgjHmc0UU0vsj7tC3dBM1Wr01bgsd2TYQfjY8YfAeIAABfYgAgNJdDAe7fQXoTG6HlmnS-GQFlHjP30UViNHOoW6JWWgP1aACa4mTe90WJg-GJ1sKdCOafvcF0QYgt0uHfB00000xhlqoRGn2EQ3G1B2g9BWAiG5_4G00?stat-id=3&test-tag=4458513) krug2000.ru |

David TaitПеревод Юрия Лысенкова

[PIC16C84](http://www.datasheet.ru/search.html?cx=005338834574777012439%3Aqye0jsmt8tw&cof=FORID%3A11&q=PIC16C84&ie=windows-1251&oe=windows-1251&sa=+%CF%EE%E8%F1%EA+) (или PIC16F84) фирмы [Microchip](http://www.rlocman.ru/cat/site.html?di=28757) - миниатюрный, но мощный микроконтроллер. Он основан на EEPROM или "FLASH" технологии, позволяющей перепрограммировать его буквально за секунды. Типовое количество циклов перезаписи - около 1000. Из его 18-ти выводов 13 могут использоваться как разряды ввода/вывода общего назначения. Когда они программируются на вывод, то допускают ток "1" до 20 мА и ток "0" до 25 мА (более чем достаточный для подключения, например, светодиодов). Это позволяет разрабатывать на данном микроконтроллере простые и недорогие электронные устройства и делает его идеальным для желающих изучить принципы работы микроконтроллеров.

Этот короткое руководство предназначено для людей, которые только что собрали или купили программатор для PIC микроконтроллера и хотят убедиться, что оба, программатор и микроконтроллер, работают. Для этого сначала необходимо обладать некоторыми знаниями о структуре и функционировании PIC микроконтроллера. Это цоколевка выводов (см.рис.):

 

Выводы RA\* и RB\* - это контакты ввода/вывода, связанные с регистрами микроконтроллера PORTA и PORTB соответственно (RA4 также может быть использоваться как вход внутреннего таймера, а RB0 может быть использован как источник прерываний). VDD и VSS - выводы питания (+Uпит и GND соответственно). Серия 16 x 84 работает в широком диапазоне питающих напряжений, но обычно VSS подключен к 0В, а VDD подключен +5 В. Вывод основного сброса /MCLR обычно подключен к VDD (напрямую или через резистор), потому что микроконтроллер содержит надежную схему сброса при включении питания - все, что вам надо, микроконтроллер выполнит сам. Выводы OSC1 и OSC2 подключаются к генератору тактовой частоты и могут быть сконфигурированы для различных его типов, включая режимы кварца и RC-генератора. Простая схема, которая используется как база для проекта с использованием PIC16C84 представлена на рисунке:



Более наглядная схема:



Схема содержит RC-генератор и один вывод (RB4) подключен к индикатору. Это - все, что необходимо для работы микроконтроллера. Charles Manning (Electronics Australia, April 1996) написал изумительно короткую (6 слов) программу для мигания индикатора, которую вы можете использовать с этой схемой: [http://www.microchip.com](http://www.microchip.com/)). Для использования программы вы должны извлечь ее из этого файла любым редактором, сохранить в другой файл (например LIGHTS.ASM), затем проассемблировать с помощью MPASM (используйте команду "**MPASM LIGHTS.ASM**") для получения HEX файла LIGHTS.HEX, который может быть загружен в микроконтроллер с помощью программатора. **Не обращайте внимания на замечания MPASM о том, что использование регистров TRIS и OPTION "не *рекомендуется"****.****Убедитесь в том, что сторожевой таймер "watchdog" включен и выбран RC-генератор. Если у вас еще нет MPASM, то здесь представлен HEX-вариант приведенной выше программы:***

*LIST P=16C84*

*;*

*MOVLW 0*

*TRIS 6*

*OPTION*

*LOOP SLEEP*

*INCF 6,F*

*GOTO LOOP*

*END*

*Эта программа написана для MPASM (бесплатный ассемблер от Microchip, доступный по адресу*

*:0C0000000030660062006300860A0328DE*

*Вы можете записать эти две строки в файл LIGHTS.HEX без использования MPASM. Если вы используете один из моих PIC программаторов (например*[*TOPIC*](http://yl.da.ru/pic/rpic-lnk.htm#chipprog)*или*[*PP*](http://yusoft.hotbox.ru/picfiles/pic84v04.zip)*), вы можете можете загрузить этот файл с корректной конфигурацией с помощью следующей команды:*

*:00000001FF*

***PP -RW8 LIGHTS.HEX (PP V-0.3)***

***PP -RW LIGHTS.HEX (PP V-0.4)***

*Программа использует таймаут "watchdog" таймера как как источник синхронизации для определения моментов включения и выключения светодиода; в результате вы можете заставить светодиод вспыхивать с различной частотой, подключая его к различным разрядам порта PORTB (RB0-RB7, выводы 6-13). Это необычное использование "watchdog" таймера. Обычно "watchdog" таймер используется, чтобы удостовериться, что PIC ведет себя в соответствии с заданной программой, и, если ваша программа специально не сконфигурирована для использования "watchdog" таймера, активизировать его было бы большой ошибкой. Простая программа LIGHTS использует его для выхода из режима "SLEEP" (т.е. режима "засыпания"); при выходе из этого режима PIC увеличивает содержимое регистра PORTB, что изменяет состояние RB0-RB7 и опять переходит в режим "засыпания" до следующего таймаута "watchdog" таймера. "Watchdog" таймер синхронизирован внутренним RC генератором, который имеет одинаковый период на всех PIC микроконтроллерах, следовательно, использование "watchdog" таймера для операций со временем гарантирует, что временные задержки будут измеряться стабильно независимо от конфигурации задающего генератора микроконтроллера или используемой частоты (желательно, чтобы частота составляла хотя бы несколько кГц). Это свойство делает программу LIGHTS очень удобной для начального тестирования большинства макетных плат для PIC. Схема может быть изменена для получения значительно большего количества эффектов путем добавления сведодиодов. Подключите первый из них к выводу RB0 (контакт 6), второй - к RB1 (контакт 7), третий - к RB2 (контакт 8) и т.д. Наилучшим вариантом является использование как минимум четырех светодиодов с увеличением до восьми (последний подключить к RB7, т.е. контакт 13). Каждый сведодиод подключается через резистор 470 Ом между ножкой микроконтроллера и "землей" (см.*[*схему*](http://rf.atnn.ru/pic/first.html#schematic3)*). Следующая программа реализует эффект "бегущего огня":*

*;****WALKLEDS.ASM****LIST P=16C84 ; PORTB EQU 6 TRISB EQU 86H OPTREG EQU 81H STATUS EQU 3 CARRY EQU 0 RP0 EQU 5**MSB EQU 3 ;номер бита для крайнего слева светодиода ; CLRF PORTB ;погасить все светодиды BSF STATUS,RP0 ;выбрать регистровый банк 1 CLRF TRISB^80H ;установить все разряды PORTB на вывод информации MOVLW 0AH MOVWF OPTREG^80H ;настроить предварительный делитель в WDT на (1:4) BCF STATUS,RP0 ;выбрать регистровый банк 0 INCF PORTB,F ;включить крайний справа светодиод BCF STATUS,CARRY ;очистить флаг CARRY LEFT SLEEP ;подождать таймаута WDT ("watchdog" таймера) RLF PORTB,F ;сдвинуть содержимое индикатора влево BTFSS PORTB,MSB ;достигли крайней левой позиции? GOTO LEFT ;если нет - цикл RIGHT SLEEP ;подождать таймаута WDT RRF PORTB,F ;сдвинуть содержимое индикатора вправо BTFSS PORTB,0 ;достигли крайней левой позиции? GOTO RIGHT ;если нет - цикл GOTO LEFT ;начать новый цикл ENDпрограмматор, указав ему следующие параметры:*

*:100000008601831686010A3081008312860A031056 :100010006300860D861D08286300860C061C0C28CC :020020000828AE :00000001FF*

*Снова вы должны с указать программатору включить "watchdog" таймер и RC генератор. Если вы запишете указанные выше четыре HEX строки в файл (например, WALKLEDS.HEX), вы можете загрузить программу, используя*

*PP -RW8 WALKLEDS.HEX (PP V-0.3) PP -RW WALKLEDS.HEX (PP V-0.4) TOPIC -RWG WALKLEDS.HEX (TOPIC V-0.2)*

*Программа "бегущий огонь" предназначена для использования четырех светодиодов, но вы можете изменять значение*[*MSB*](http://rf.atnn.ru/pic/first.html#MSB)*для использования большего количества светодиодов: для количества светодиодов 5, 6, 7 и 8 значения MSB должны быть 4, 5, 6 и 7 соответственно.*

*В программе не используются команды TRIS и OPTIONS, не рекомендованные к применению фирмой****"MicroChip"****, т.к. они могут не поддерживаться в будущих микроконтроллерах. Таким образом, в отличие от*[*предыдущей программы*](http://rf.atnn.ru/pic/first.html#program1)*, во время ассемблирования не будут генерироваться предупреждения Для предотвращения генерации MPASM'мом сообщений о корректном использовании регистровых банков необходимо инвертировать старший значащий бит в любом адресе, приходящемся на банк 1 (например, использовать TRISB^80H вместо обычного TRISB, где оператор "^" означает битовое "исключающее ИЛИ"). Это - одна из уловок, которые используются для подавления вывода предупреждающих сообщений MPASM.*

*В качестве заключительного примера рассмотрим программу, когорая демонстрирует такие же эффекты, как и программа*[*WALKLEDS*](http://rf.atnn.ru/pic/first.html#program2)*на 4-х светодиодах. Вы обратите внимание на то, что она значительно больше по объему и ее нельзя назвать примеров эффективного программирования. Она всего лишь предназначена для демонстрации нескольких ключевых идиом и технологий PIC микропроцессоров. Кроме прочего, она содержит обработчик прерываний, процедуры записи и чтения данных встроенного FLASH EEPROM и демонстрирует, как в PIC реализован принцип табличного поиска. Программа содержит примеры одной из наиболее удобных особенностей MPASM, такой как два вида макрокоманд. Она также показывает, как отменить заданное по умолчанию основание системы счисления (шестнадцатеричное) для чисел и как внедрить информацию о конфигурации микроконтроллера. По крайней мере, по стилю она больше напоминает "настоящую" программу для микроконтроллера.*

*;****PATTERN.ASM***

***TOPIC -RWG LIGHTS.HEX (TOPIC V-0.2)****Проассемблируйте эту программу с помощью MPASM для получения ее HEX представления:*

*; Программа разработана для демонстрации процедур чтения/записи данных EEPROM и обработки*

*; прерываний таймера. Таблица значений записана в EEPROM процессор выполняет "холостой"*

*; цикл. Когда таймер переполняется, происходит прерывание работы процессора и следующее*

*; значение таблицы считывается из EEPROM и записывается в порт B, т.е. отображается на*

*; светодиодах. С помощью изменений в таблице может быть изображен любой образец длиной до*

*; 64 значений.*

*; Copyright (C) 1997 David Tait (david.tait@man.ac.uk)*

*PROCESSOR 16C84 \_\_CONFIG 03FF3 ; RC генератор PCL equ 2 STATUS equ 3 ; адреса стандартных регистров PORTB equ 6 EEDATA equ 8 EEADR equ 9 INTCON equ 0BH OPTREG equ 081H TRISB equ 086H EECON1 equ 088H EECON2 equ 089H RP0 equ 5 Z equ 2 GIE equ 7 T0IE equ 5 T0IF equ 2 WREN equ 2 WR equ 1 RD equ 0 #define bank0 bcf STATUS,RP0 ;выбрать Bank 0 #define bank1 bsf STATUS,RP0 ;выбрать bank 1 magic macro ;"магическая" последовательность записи в EEPROM movlw 55H movwf EECON2^80H movlw 0AAH movwf EECON2^80H endm cblock 0CH ;блок переменных n\_vals n\_tmp endc ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*; ; Основная точка входа в программу ; ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*; org 0 goto start ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*; ; Точка входа в прерывание ; ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*; org 4 ; Обычно содержимое необходимо сохранять перед процедурой обработки прерывания и ; восстанавливать после нее, но в данной программе в этом нет необходимости, т.к. ; процессор ничего не делает между прерываниями. Смотрите PIC datasheet для ; рекомендуемой процедуры. movf EEADR,w xorwf n\_vals,w btfsc STATUS,Z ;EEADR == n\_vals? clrf EEADR ;если да, то начать с нуля call ee\_rd movf EEDATA,w ;читать EEPROM movwf PORTB ;отобразить байт incf EEADR,f ;новый адрес bcf INTCON,T0IF ;очистить флаг прерывания retfie start clrf PORTB bank1 clrf TRISB^80H ;все разряды порта B на вывод movlw B'00000111' movwf OPTREG^80H ;таймер 0 предварительный делитель 256:1 bsf EECON1^80,WREN ;разрешить запись в EEPROM bank0 call ee\_init ;пересылка таблицы в EEPROM bank1 bcf EECON1^80H,WREN ;запретить запись в EEPROM bank0 bsf INTCON,T0IE ;разрешить прерывания от таймера bsf INTCON,GIE ;разрешить все прерывания loop goto loop ;"холостой" цикл ; ee\_init ; ; инициализация EEPROM из таблицы ee\_init clrw call lut ;получить кол-во элементов в таблице movwf n\_vals ;и сохранить movwf n\_tmp ;сохранить еще раз clrf EEADR decf EEADR,f ;EEADR = -1 ee\_in1 incf EEADR,f ;следующий адрес movf EEADR,w addlw 1 call lut ;получить соответствующее значение таблицы movwf EEDATA call ee\_wr ;запись в EEPROM decfsz n\_tmp,f ;есть еще? goto ee\_in1 ;да clrf EEADR ;нет, завершить return ; lut ; ; просмотр таблицы lut addwf PCL,f ;добавить W к PCL для получ. адреса эл-та таблицы retlw D'12' ;число элементов в таблице retlw B'1000' ;первый элемент retlw B'1000' retlw B'0100' retlw B'0100' retlw B'0010' retlw B'0010' retlw B'0001' retlw B'0001' retlw B'0010' retlw B'0010' retlw B'0100' retlw B'0100' ;последний элемент ; ee\_wr ; ; Записать байт из EEDATA в EEPROM по адресу в EEADR. Прерывания ; должны быть запрещены перед вызовом ee\_wr. ee\_wr bank1 magic ;вызов "волшебной" последовательности bsf EECON1^80H,WR ;начать запись ee\_wr1 btfsc EECON1^80H,WR ;запись завершена? goto ee\_wr1 ;нет bank0 return ; ee\_rd ; ; Прочитать байт EEPROM из EEPROM по адресу EEADR в EEDATA ee\_rd bank1 bsf EECON1^80H,RD ;начать запись bank0 return ;при возвращении чтение должно быть завершено end Это HEX файл, сформированный MPASM (запишите его как PATTERN.HEX): :020000000E28C8 :0800080009080C060319890127 :10001000442008088600890A0B110900860183160E :10002000860107308100081583121C2083160811F1 :1000300083128B168B171B2800012C208C008D003F :1000400089018903890A0908013E2C2088003A2089 :100050008D0B22288901080082070C3408340834EB :1000600004340434023402340134013402340234DE :1000700004340434831655308900AA30890088146A :100080008818402883120800831608148312080079 :02400E00F33F7E :00000001FF При использовании моего программатора это может быть выполнено с помощью следующей команды: PP PATTERN.HEX (PP V-0.4) TOPIC -G PATTERN.HEX (TOPIC V-0.2)*

*2nd Edition 4/Feb/97*

*; Эти программы могут показаться не слишком совершенными, но если вы только что собрали или купили программатор PIC и поспешно собрали простую испытательную схему, то такая программа для включения и выключения светодиодов вам пригодится.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Для комментирования материалов с сайта и получения полного доступа к нашему форуму Вам необходимо*[***зарегистрироваться***](http://www.rlocman.ru/forum/register.php?s=)*.* | *Начало формы*

|  |  |  |
| --- | --- | --- |
| *Имя* |  | *Запомнить?* |
| *Пароль* |  |  |  |

*Конец формы* |

|  |
| --- |
| **Рекомендуемые публикации по теме:** |
|

|  |  |
| --- | --- |
| [Форум](http://www.rlocman.ru/forum/) |  **»** |

 | [Ресурсы для проектов на микроконтроллерах PIC и AVR](http://www.rlocman.ru/forum/showthread.php?t=8204) |
|

|  |  |
| --- | --- |
| [Форум](http://www.rlocman.ru/forum/) |  **»** |

 | [Проекты на микроконтроллерах ПЛИС](http://www.rlocman.ru/forum/showthread.php?t=7579) |
|

|  |  |
| --- | --- |
| [Схемы](http://www.rlocman.ru/shem/search.html) |  **»** |

 | [Модифицируем кабель передачи данных T610 для использования его в проектах на микроконтроллерах](http://www.rlocman.ru/shem/schematics.html?di=45665) |
|

|  |  |
| --- | --- |
| [Новости](http://www.rlocman.ru/news/search.html) |  **»** |

 | [Интегрированная среда разработки Atmel Studio 6 поддерживает разработку проектов на микроконтроллерах ARM Cortex-M и AVR](http://www.rlocman.ru/news/new.html?di=114985) |
|

|  |  |
| --- | --- |
| [Книги](http://www.rlocman.ru/book/search.html) |  **»** |

 | [Справочник по PIC-микроконтроллерам (PICmicro Microcontroller)](http://www.rlocman.ru/book/book.html?di=28180) |

**Микроконтроллер выполняет роль вольтметра**

[**Microchip**](http://www.rlocman.ru/shem/search.html?fr=Microchip)**» PIC16F84A-20P, PIC16F628A**

PIC-микроконтроллер и полевые транзисторы образуют четырехзначный вольтметр.



Схема на рис.1 - развитие предыдущей идеи конструкции по использованию аналогового входа в микроконтроллере, не имеющего встроенного АЦП, а так же используются технические приемы из другой идеи конструкции по управлению семисегментным светодиодным индикатором без внешних ключевых транзисторов. Данная схема имеет последовательный канал, и нужна только витая пара для передачи измеренных значений на персональный компьютер.

Последовательный канал был протестирован с использованием программы компании Microsoft Hyper Terminal сконфигурированной параметрами 115,200 бод; 8 бит, четность, 1 стоп-бит; без аппаратного контроля.

Коротко, программа управляет одним светодиодным семисегментным индикатором за раз по линиям RA0 и RB7. Установка выхода RA0 в единицу и использование RB7, как входа активизирует индикатор с общим анодом DS3. Установка выхода RA0 в ноль и использование RB7 как входа, активизирует индикатор с общим катодом DS2. Использование RA0 как входа и установка выхода RB7 в единицу активизирует индикатор с общим анодом DS1, а при использовании RA0 как вход и установке выхода RB7 в ноль активизирует индикатор с общим катодом DS0. После успешной активизации одного индикатора, только одна из линий RB0 … RB6, конфигурируется как выход для управления одним светодиодным сегментом. Эта схема больше не имеет ограничения на питающее напряжение VDD - 3В или ниже - так как светодиоды включены встречно-параллельно, таким образом, прямое падение напряжения на одном светодиоде ограничивает обратное напряжение на другом. Использование красных светодиодов требует 1,6 В.



Рис.2 иллюстрирует новые аспекты идеи конструкции. Q1, R5, и R6 работают как эквивалентный переменный резистор, RX, который заряжает конденсатор C3. Вместо подключения RX к земле, просто подключите его к одной линии ввода-вывода – например RB0 – микроконтроллера. Если RB0 включен как выход в нулевом состоянии, значит первый аналоговый канал активизирован и измерительная подпрограмма подсчитывает импульсы заряда до величины 66% от VDD; затем, по таблице полученная величина задержки переводится в величину милливольт из трех цифр. Для увеличения количества аналоговых входов, вы можете подключить до семи цепей переменного резистора в параллель – таким образом, что каждый подключен между C3 и одной линией ввода-вывода, RB1 … RB7. Важно, что линии ввода-вывода подключены к индикаторам и так же активируют или отключают аналоговые каналы. Когда один аналоговый канал активизирован линией ввода-вывода выходом в низком состоянии, другие линии имеют высокое сопротивление и работают как входы, что отключает все остальные каналы. Соответственно, индикаторы отключены.

В схему на рис.1 так же добавлен простейший последовательный канал без добавления внешних компонентов. Если вы подключите две линии ввода-вывода, RA1 и RA2, сконфигурированные как выходы к RXD (Выв 2) и GND (Выв 5) разъема RS 232, вы сможете создавать, с помощью программы, положительное и отрицательное напряжение относительно земли порта RS 232 в ПК. Когда RA1 в единице, а RA2 в ноле, RXD имеет положительный потенциал 5 В относительно земли порта RS 232 в ПК. Когда RA1 в ноле, а RA2 в единице, RXD имеет отрицательный потенциал -5 В относительно земли порта RS 232 в ПК. [Listing 1](http://www.rlocman.ru/i/File/2007/10/di4049.zip%22%20%5Ct%20%22_blank) содержит практический пример для [PIC16F84A-20P](http://www.datasheet.ru/search.html?cx=005338834574777012439%3Aqye0jsmt8tw&cof=FORID%3A11&q=PIC16F84A-20+P&ie=windows-1251&sa=+%CF%EE%E8%F1%EA+). Он не оптимизирован, но полностью прокомментирован для облегчения задачи перевода на другие микросхемы средней сложности [компании Microchip](http://www.rlocman.ru/cat/site.html?di=28757), например[PIC16F628A](http://www.datasheet.ru/search.html?cx=005338834574777012439%3Aqye0jsmt8tw&cof=FORID%3A11&q=PIC16F628A&ie=windows-1251&sa=+%CF%EE%E8%F1%EA+), которая поддерживает частоту работы до 20 МГц и имеет больше линий ввода-вывода.

[*Терраэлектроника*](http://www.rlocman.ru/rds.php?mem=terraelectronica&rd=http%3A%2F%2Fwww.terraelectronica.ru%2F)

На английском языке: **[Microcontroller Functions as Voltmeter](http://www.radiolocman.com/shem/schematics.html?di=32833" \t "_blank)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Для комментирования материалов с сайта и получения полного доступа к нашему форуму Вам необходимо [**зарегистрироваться**](http://www.rlocman.ru/forum/register.php?s=). | Начало формы

|  |  |  |
| --- | --- | --- |
| Имя |  | Запомнить? |
| Пароль |  |  |  |

Конец формы |

|  |  |
| --- | --- |
| **Фрагменты обсуждения:** | [**Полный вариант обсуждения »**](http://www.rlocman.ru/forum/showthread.php?t=7082) |

* Где прошивка?
* В тексте статьи есть ссылка на архив (Listing 1) с исходником на ассемблере. Компилируйте и будет вам прошивка.
* Не понятно какое максимальное число индицируется на индикаторе ... . :(
* Некоторые из проектов http://kentar.net63.net/pic--voltmeter-ammeter-.html http://320volt.com/en/pic16f876-lcd-...e-devresi-pbp/ http://www.iw5blt.it/Pic\_Index.html http://vasileelettronic.altervista.o...ore%20news.htm

|  |
| --- |
| **Рекомендуемые публикации по теме:** |
|

|  |  |
| --- | --- |
| [Схемы](http://www.rlocman.ru/shem/search.html) |  **»** |

 | [Интеллектуальная система контроля температуры на базе микроконтроллера AVR. Часть 2 - Описание задач, выполняемых функциями в программе микроконтроллера, настройка и работа с системой.](http://www.rlocman.ru/shem/schematics.html?di=136011) |
|

|  |  |
| --- | --- |
| [Форум](http://www.rlocman.ru/forum/) |  **»** |

 | [Обсуждение: Миниатюрный вольтметр на микроконтроллере ATmega8L](http://www.rlocman.ru/forum/showthread.php?t=6909) |
|

|  |  |
| --- | --- |
| [Схемы](http://www.rlocman.ru/shem/search.html) |  **»** |

 | [Бесконтактный тахометр на микроконтроллере AVR. Часть 2. Программное обеспечение для микроконтроллера](http://www.rlocman.ru/shem/schematics.html?di=77580) |
|

|  |  |
| --- | --- |
| [Схемы](http://www.rlocman.ru/shem/search.html) |  **»** |

 | [DDS генератор на микроконтроллере ATmega16. Часть 2. Программное обеспечение для микроконтроллера](http://www.rlocman.ru/shem/schematics.html?di=70215) |
|

|  |  |
| --- | --- |
| [Схемы](http://www.rlocman.ru/shem/search.html) |  **»** |

 | [12-кнопочная клавиатура с двухпроводным интерфейсом. Часть 3. Подключение интерфейса клавиатуры к микроконтроллеру, основные моменты в ПО микроконтроллера, демонстрация работы.](http://www.rlocman.ru/shem/schematics.html?di=100107) |