

Опубликовано в журнале Радиоаматор №1.2006 г.
(оригинал статьи)

--

Цель статьи – заинтересовать читателей писать программы для PIC микроконтроллеров на языке высокого уровня – PicBasic Pro, а также ознакомить с программным обеспечением, и средствами для отладки и записи программ в кристалл. Автор не является программистом, а статья написана по горячим следам в ходе освоения, возможно в этом ценность представленного материала для начинающих.

Урок программирования на PicBasic Pro

А. А. Кравченко

Черниговская обл.

В свое время автор писал программы на BASICe для компьютеров: Robik, Спектрум 48/128. Программы разных авторов публиковались в радиолобительских журналах. Изучать и дорабатывать эти программки было не менее интересно, чем разрабатывать что ни будь, сидя с паяльником. Но неумолимый ход времени и технического прогресса отбросил те старые компьютеры далеко в прошлое. И вот сейчас то, что раньше занимало плату с десятками микросхем, вмещается в одном кристалле, под названием микроконтроллер. Остается дополнить микроконтроллер устройством ввода информации, а также подключить графический LCD дисплей, LED индикатор, или, в крайнем случае, светодиоды, написать программу под все это «железо», и микрокомпьютер готов. Применение такого микрокомпьютера могут ограничить лишь пределы вашей фантазии: это различные устройства автоматизации, сигнализации, измерительной техники и др. А главное малые габариты, низкое энергопотребление плюс сторожевой таймер (watchdog timer или сокращенно WDT), который позволяет самостоятельно перезагружаться микроконтроллеру после случайных зависаний программы. Но если спать все это в кучу для многих не представляет труда, то написать самостоятельно программу, а особенно на ассемблере – это удел специалистов. Вот по этому, для программирования микроконтроллеров, автор выбирает язык высокого уровня - PicBasic. Матерые спецы называют его языком для ленивых. Да, его освоить так же просто, как и старый, знакомый многим Qbasic. За простоту мы платим замедлением скорости. Понятное дело, на ассемблере намного быстрее. Никакой язык не поможет, если вы не знаете тонкостей процессора. Но на PicBasic программы писать проще и быстрее, а места, где нужна скорость выполнения, можно при необходимости переписывать на ассемблере.

Существует несколько подобных языков Basic для PIC. Автор выбрал PicBasic Pro от microEngineering Labs, Inc. Описание компилятора языка на английском, можно скачать с [1] (на сегодняшний день там находится файл rbpm304.pdf размером 1,07 МБ), а русский перевод ранней версии описания - скачать с [2], перевод не идеален, по этому лучше параллельно пользоваться оригиналом на английском языке. Преимущество любого языка высокого уровня в том, что вы пишете программу готовыми подпрограммами «штампами», в Бейсике их обычно называю операторами. Благодаря готовым подпрограммам, сокращается время написания программы, значительно уменьшается размер текста программы, но снижается скорость выполнения, увеличивается занимаемый программой размер кода (после компиляции), по сравнению с подобной ассемблерной программой. В состав PicBasic Pro входит ряд операторов и функций, обеспечивающих выполнение большинства задач в радиоэлектронике. Это например, операторы обеспечивающие: управление LCD индикатором; ввод/вывод цифровых данных через порты; кнопочный ввод; счет импульсов на выводе; измерение длительности импульса;

чтение потенциометра с определенного вывода; генерация импульса заданной длительности; генерация телефонных DTMF сигналов; генерация двухтональной частоты; генерация звуковой посылки заданной частоты и длительности; управление широтно-импульсной модуляцией; асинхронный последовательный вход и выход; синхронный последовательный вход и выход; USB вход и выход; доступ к I²C устройствам; доступ к EEPROM; чтение аналого-цифрового преобразователя (АЦП встроен в некоторые модели PIC микроконтроллеров), математические операторы и многое другое. Полный список операторов и их подробное описание есть в описании компилятора PicBasic Pro. Лучше изучать язык, пользуясь готовыми программами, параллельно читая описание. Большое количество образцов программ на Pic Basic и PicBasic Pro найдете в [1], (на сегодняшний день смотрите разделе Resources, ссылка Sample Programs).

Программы на PicBasic и PicBasic Pro можно писать в любом текстовом редакторе. Но удобнее пользоваться специальным редактором MicroCode Studio, последнюю версию можно скачать с [1]. В редакторе можно не только писать тексты программ, но и компилировать программы прямо с редактора, а также запустить с редактора программу, управляющую программатором (при наличии соответствующего программатора и программного обеспечения). Нам понадобится еще и компилятор. Это программа, преобразующая текст PicBasic программы в машинные коды - набор на первый взгляд бессмысленных цифр. Но только с этими кодами, а не PicBasic программой, сможет работать микроконтроллер. Демонстрационную версию компилятора можно скачать с [1]. Демо версия компилятора поддерживает ограниченное количество PIC микроконтроллеров: 16F627(A), 16F628(A), 16F84(A), 16F870, 16F871, 16F872, 16F873(A), 16F874(A), 16F876(A), 16F877(A). А также текст исходной программы не может превышать 31 строку (комментарии и интервалы не считаются). Но для изучения языка и получения минимальных навыков в программировании, демо версии вполне достаточно. Полнофункциональный компилятор PicBasic Pro поддерживает все существующие на сегодняшний день PIC микроконтроллеры, и не имеет ограничений в количестве строк PicBasic программы. Использование компилятора PicBasic Pro без лицензии является нарушением закона. Для того чтобы получить лицензию, вместе с самой последней версией продукта и документацией, необходимо обратиться к microEngineering Labs, Inc.

Папку с компилятором можно разместить в любом месте жесткого диска компьютера, важно чтобы имена папок в пути к компилятору не содержали символов кириллицы. При первом запуске MicroCode Studio, редактор автоматически найдет и пропишет в себе папку с компилятором. Вручную, демо компилятор, или любую другую версию компилятора, можно прописать в редакторе, выбрав пункт меню View/Compile and Program Options.

Напишем первую программку (blink.pbp) для мигания светодиода подключенного к PORTB.0 по схеме рис.1, для PIC16F628A. В дальнейшем, в программе поддержки программатора выставим биты конфигурации такими, чтобы в качестве генератора применить внутренний RC генератор частотой около 4 МГц (тип генератора: IntRC I/O). Времязадающие операторы (Pause в том числе) по умолчанию рассчитаны на работу с задающим генератором микроконтроллера частотой 4 МГц. Если в задающем генераторе будет установлен кварцевый резонатор с частотой отличающейся от 4 МГц, например на частоту 20 МГц, то в начале текста программы нужно записать строку:

```
DEFINE OSC 20.
```

Полный список DEFINE смотрите в описании компилятора.

```
'Программа для мигания светодиода, подключенного к выводу  
PORTB.0
```

```
loop: HIGH PORTB.0 ' Включить светодиод  
PAUSE 500 ' Задержка полсекунды
```

```

LOW PORTB.0 ' Выключить светодиод
PAUSE 500 ' Задержка полсекунды
GOTO loop ' Возврат в начало программы
END

```

Операторы в MicroCode Studio автоматически выделяются жирным шрифтом, а комментарии курсивом синего цвета, таким образом, улучшается читаемость программы. Словом loop в программе для мигания светодиода, обозначена метка (закладка), необходимая для выполнения переходов при помощи оператора GOTO или GOSUB. Перед комментариями ставится кавычка. Все что после кавычки компилятор пропускает, по этому размер комментариев на объем кода не влияет (кавычка находится на английской клавиатуре на месте русской буквы «э», вместо кавычки можно ставить точку с запятой «;»).

Теперь в редакторе, выберем тип микроконтроллера PIC16F628A, и нажмем кнопку "Compile Only" или F9. Если вы предварительно не сохранили файл, то программа предложит вам его сохранить, и после сохранения выполнит компиляцию. Имена папок и файлов, в пути к PicBasic файлу, не должны содержать символов кириллицы. При удачной компиляции текста программы blink.pbp для PIC16F628A, в нижней части окна редактора появится надпись "success: 59 words used", а в папке, куда ранее был сохранен файл blink.pbp, сгенерируются три дополнительных файла: blink.asm, blink.mac а также blink.hex – файл, предназначенный для программатора. Скриншот окна редактора MicroCode Studio (v.2.3.0.0) после компиляции программы, изображен на рис.2.

Для записи кодов программы в микроконтроллер понадобится программатор, и программа поддержки программатора. Микроконтроллер подключается к программатору с помощью таких цепей:

1. Напряжение питания (VDD) +5V.
2. Общий провод (VCC) 0 V.
3. Напряжение программирования (VPP), при программировании +12 В, при чтении +5 V.
4. Сигнал тактирования (CLC) импульсы размахом до 5 V.
5. Сигнал данных (Data I/O) импульсы размахом до 5 V.

Схем программаторов существует множество. Автор пользуется ProPic2 (подключается к LPT порту, и использует внешнее питание), его схему можно скачать с [3]. В ProPic2 установлена одна 40 контактная ZIF панелька, но ее можно заменить панелькой собранной из DIP40 и двух панелек DIP20. Разные PIC микроконтроллеры вставляются в разные места панельки программатора, начиная с первого вывода или со смещением (см. схему ProPic2). Чтобы обеспечить возможность пользоваться одной панелькой при программировании разных микросхем, в ProPic2 используются три напряжения программирования: VPP1, VPP2 и VPP3. При выборе в программе поддержки программатора типа программируемой микросхемы, программа включает только одну определенную цепь напряжения VPP (для PIC16F84 и PIC16F628 включается VPP1). Схему и рисунок печатной платы более простого программатора JDM можно скачать с [8], он подключается к COM порту компьютера и не требует внешнего питания (питается от COM порта). На плате JDM программатора установлена панелька DIP18, по этому он способен программировать только 18-ти выводные микроконтроллеры, а также ряд микросхем памяти. Но если в JDM программаторе вместо DIP панельки установить разъем, к которому подключать переходники с панельками под разные микроконтроллеры, то тогда можно запрограммировать любой PIC микроконтроллер. Последнюю версию программы IcProg (поддерживает много разных программаторов, в том числе ProPic2 и JDM) можно скачать с [4].

После подключения программатора ProPic2 к LPT порту компьютера (или JDM к COM порту), запустить IcProg. Настроить его для работы с имеющимся программатором. Скриншот окна «Hardware settings» программы IcProg (v.1.05D), для работы с

программатором ProPic2 изображено на рис.3. Вставить в панельку программатора PIC микроконтроллер, открыть с помощью IcProg файл проекта с расширением HEX. Перед записью кодов в память PIC микроконтроллера необходимо проверить, и правильно установить биты конфигурации для каждого конкретного проекта (назначение битов конфигурации смотрите в документации для каждого PIC микроконтроллера, PDF файлы документации можно скачать, например с [5]). Скриншот окна IcProg с открытым файлом blink.hex, и с установленными битами конфигурации, для нашего проекта, изображен на рис.4. Нажать кнопку "Program All". Программа запишет, считает и верифицирует коды, и если запись произошла без ошибок, появится окно с надписью "Device successfully verified!". Теперь наш PIC переставляем из панельки программатора в панельку схемы рис.1, подаем питание, и, работает!

Существует ряд программаторов, программы управления которыми можно прописать в редакторе MicroCode Studio. Тогда при нажатии на кнопку "Compile and Program" в окне редактора, запустится программа поддержки программатора, в которой уже будет открыт HEX файл, только что сгенерированный при компиляции. В [1] представлены программаторы: EPIC™ Programmer и melabs Serial Programmer, а также beta программное обеспечение к ним. Программу поддержки программатора можно прописать в MicroCode Studio, выбрав пункт меню View/Compile and Program Options. У автора нет схем подобных программаторов, не выставляют схем на сайты и фирмы разработчики.

В [1] также предлагается ряд плат для отладки программ на PIC микроконтроллерах. На этих платах обычно установлены: LCD индикатор, разъем для подключения платы отладки к программатору, схема развязки (MAX232) для связи PIC микроконтроллера с компьютером через COM порт, и минимум других средств, необходимых для проверки работы и отладки записанной в PIC программы, это, например: кнопки, потенциометры, светодиоды, элементы телефонной схемотехники и др. Разные платы, предназначены для разных направлений проектов. Удобство состоит в том, что микросхема – PIC микроконтроллер остается в панельке платы отладки, как при программировании, так и при запуске записанной в кристалл программы. При этом все необходимые для программирования и чтения памяти сигналы (список сигналов приведен выше) подведены от программатора к плате отладки при помощи короткого кабеля. Автор, для отладки своих программ разработал упрощенную, но более универсальную плату, схема рис.5. Дорогостоящий LCD индикатор решил не устанавливать, установил ряд светодиодов для индикации логических состояний на выводах портов микроконтроллера; разъем, к которому подключены все выводы портов микроконтроллера. Микросхему MAX232 заменил дешевой схемой на двух транзисторах. Плата предназначена для 18-ти выводных микроконтроллеров, включая популярный PIC16F84A и новый PIC16F62x (автор проводил эксперименты на плате только с PIC16F84 и PIC16F628A).

Микроконтроллер вставляется в панельку. Нужный кварц с конденсаторами, или RC цепочка для RC генератора PIC16F84, подключаются к плате при помощи малогабаритных штекеров J7, J8, J9, J10 (J7 предназначен для подключения резистора RC цепочки). К выводам портов микроконтроллера, с помощью перемычек JP6-JP21, подключаются светодиоды – для индикации логических состояний на выводах. Плата соединяется с программатором кабелем длиной 20 см, через разъем J11 (контакт 2 разъема J11 соединен с цепью VPP1 программатора ProPic2). Теперь при отладке программы и многократных перезаписях кодов в памяти микроконтроллера, не требуется переставлять дорогостоящую микросхему с панельки в панельку. Единственное неудобство: при записи программы в PIC16F84 нужно отключить кварц или RC цепочку - сорвать генерацию задающего генератора (иначе возможны ошибки при записи кодов), а в PIC16F628A программа удачно записывается и читается и с подключенными элементами генератора. Также, при обращении программатора к PIC, выводы CLOCK и DATA микроконтроллера (выводы портов RB6 и RB7), не должны быть задействованы в проекте, и должны соединяться только с программатором (перемычки JP6, JP7 отключить, а перемычки JP1,

JP2 подключить, а также не использовать выводы RB6 и RB7 разъема J1). Штекерный разъем J1 (все выводы портов микроконтроллера) предназначен для подключения внешней схемы проекта, это может быть готовая плата проекта, или макет, с индикатором, кнопками ввода данных, и всеми необходимыми для конкретного проекта элементами. Питательные напряжения к внешней схеме проекта можно подключить от платы отладки с помощью штекеров: J7, J3 (+5 В и +12 В); J4, J5, J6 (GND). В случае если потребуется запрограммировать микроконтроллер, вставленный в панельку платы отладки при отсутствии внешнего питания для платы, то необходимо установить переключку JP22 (рис.5). При этом на время чтения или записи кодов, напряжение +5 В поступит от программатора к плате через контакт 1 разъема J11. Если вы хотите использовать один источник питания для платы и программатора (источник питания платы), тогда установите переключку JP23 и напряжение +12 В поступит от платы к программатору через контакт 3 разъема J11.

PIC16F628(A) и ряд других PIC микроконтроллеров поддерживают специальную функцию ICD_(In-Circuit Debugger) – внутри схемная отладка. Для функционирования отладки необходимо PIC микроконтроллер связать с компьютером через COM порт (для PIC16F62X, вывод 7 - RX, вывод 8 - TX). После компиляции PicBasic программы в режиме ICD и записи кодов в память микроконтроллера, запустить программу в микроконтроллере из окна редактора MicroCode Studio. Примечание: MicroCode Studio поддерживает только одну модель ICD, это PIC16F628. Для поддержки большего количества моделей, необходимо приобрести MicroCode Studio Plus, текущей или ранней версии. После запуска программы в микроконтроллере, выполняемая в каждый момент времени строка PicBasic программы будет подсвечиваться в окне редактора, синим цветом. Программу, также, можно запускать по шагам. В отдельно открытом окне редактора MicroCode Studio можно контролировать переменные, регистры, память, EEPROM микроконтроллера. Особенно это удобно в случае с вычислениями, на этапе освоения языка, когда вы не понимаете, что должен выдавать тот или иной математический оператор. Данную функцию предпочтительнее применять для отладки отдельных участков программы проекта, так как код, при компиляции ICD имеет больший объем, а также программа работает очень медленно. Для увеличения скорости выполнения программы можно увеличить частоту задающего генератора - установив кварц на более высокую частоту, но не выше максимально допустимой.

Плата отладки программ, соединена проводами с разъемом J2, который необходимо подключить к свободному COM порту компьютера. Автор использует кабель в виде трех сплетенных одиночных проводов длиной около 1 метра. Переключки на разьеме J2 (рис.5), подключенном к COM порту компьютера, запаивать обязательно! Обычно для согласования периферийных устройств с COM портом, применяют специализированную микросхему MAX232, автор применил более дешевый вариант - схему на двух транзисторах VT1 VT2 (рис.5).

Для ознакомления с функцией ICD напишем еще одну программку (icd-test.pbp) в которой значение двух переменных динамически изменяется при каждом повторении цикла программы, переменная Counter увеличивается на единицу, а переменная Junk - уменьшается на пять. Для визуальной работы программы с помощью мигающего светодиода на плате отладки, в середину цикла новой программы вставим текст предыдущей, уменьшив время задержки в операторах Pause до 0,1 секунды.

; Программа, динамически изменяющая значение двух переменных, и для мигания светодиода

```
Counter VAR WORD ; Определим двухбайтовую переменную Counter  
Counter = 0 ; Запишем в переменную Counter число 0  
Junk VAR WORD ; Определим двухбайтовую переменную Junk
```

```

Junk = 10000 ; Запишем в переменную Junk десятичное число 10000
begin: ; Метка с именем "begin"
FOR Counter = 0 TO 10000 ; Счет от 0 до 10000
Junk = junk - 5 ; Уменьшим значение переменной Junk на пять

HIGH PORTB.0 ' Включить светодиод
PAUSE 100 ' Задержка 0,1 сек
LOW PORTB.0 ' Выключить светодиод
PAUSE 100 ' Задержка 0,1 сек

NEXT Counter ; Берем следующее значение переменной Counter
GOTO begin ; Возврат в начало программы к метке begin

```

Для компиляции программы в режиме ICD, нажать кнопку «ICD compile».

Примечание: компилятор Demo версии не поддерживает функцию ICD.

Так как MicroCode Studio поддерживает только одну модель, это PIC16F628, а автор проводил эксперименты с PIC16F628A, то ICD компиляция выполнялась для PIC16F628 (архитектура микроконтроллеров подобна), а при записи кодов в кристалл, в окне IcProg выставлялся тип микроконтроллера: PIC16F628A.

По окончании записи кодов в память микроконтроллера, в окне MicroCode Studio нажать кнопку "ICD Run". Программа в микроконтроллере запустится, обмениваясь через СОМ порт компьютера информацией с MicroCode Studio. При этом будет создаваться иллюзия работы PicBasic программы в окне редактора - синяя подсветка строк программы будет перемещаться от строки к строке по ходу выполнения. А мигание светодиода LED8 подтвердит работу программы в самом PIC микроконтроллере. Но из-за большего объема кода программы при компиляции в режиме ICD, а также необходимости обмена информацией микроконтроллера с MicroCode Studio, программа в микроконтроллере будет работать медленнее, чем после обычной компиляции. Замедление работы программы можно увидеть, сравнивая частоту мигания светодиода при ICD компиляции, с частотой мигания после обычной компиляции. Строки программы Pause 100, при ICD компиляции, будут обеспечивать паузу, значительно большую чем 100 миллисекунд. В дополнительно открывшемся окне редактора MicroCode Studio, можно наблюдать, как увеличивается переменная Counter, и уменьшается переменная Junk, (рис.6). Это окно можно переключить для контроля регистров, памяти или EEPROM. После нажатия кнопки ICD Pause, кнопкой ICD Step можно управлять программой по шагам. Скомпилированная в режиме ICD и записанная в микроконтроллер программа, самостоятельно (без MicroCode Studio) работать не будет.

Удобно то, что программу проекта можно отладить с помощью ICD на PIC микроконтроллере, который поддерживает функцию ICD, а потом оперативно переписать программу, и скомпилировать ее для другого PIC, более дешевого, или такого, который ICD не поддерживает. Важно только чтобы в используемом микроконтроллере хватило выводов портов, и нужных для проекта функций. На время отладки программы с помощью функции ICD, выводы PIC микроконтроллера RX и TX в проекте не задействовать.

Внимание! При работе с программатором и платой отладки на незаземленном компьютере могут появиться ошибки при записи (чтении) кодов, как следствие воздействия наводок. У автора появлялись ошибки при записи в PIC16F84 по причине наводок на цепь платы отладки, соединенную с выводом микроконтроллера OSC1/CLKIN. Сбои в работе программатора исчезли после заземления компьютера, или при подключении на время программирования, резистора 4,7 кОм между выводом OSC1/CLKIN микроконтроллера PIC16F84 и корпусом платы отладки.

Дополнительная информация:

- [1] <http://melabs.com/>
<http://www.microengineeringlabs.com> Сайты microEngineering Labs, Inc
- [2] <http://www.ra-publish.com.ua/arc-ra.html> Приложения к статьям журнала «Радиоаматор»
- [3] <http://www.propic2.com> Сайт посвящен программатору ProPic2
- [4] <http://www.ic-prog.com> Сайт посвящен программе IcProg
- [5] <http://www.microchip.ru:80/lit/pic> PIC-микроконтроллеры фирмы Microchip
<http://www.microchip.ru:80/lit/?mid=1x0> Статьи на русском языке
- [6] <http://www.mecanique.co.uk> Сайт Mecanique
- [7] <http://www.rentron.com> Сайт Reynolds Electronics
- [8] <http://picbasic.narod.ru> Русскоязычный сайт, посвящен программированию микроконтроллеров
- [9] <http://www.nnov.rfnet.ru/pic/first.html> David Tait, «Мой первый проект на PIC микроконтроллере» (русский перевод статьи).