

# РАДИО

## начинающим

Александрю Сергеевичу Долгому посвящается...

## Программирование микроконтроллеров серии PIC для начинающих

А. КАРПАЧЕВ, г. Железногорск Курской обл.

*Предлагаемый вниманию читателей цикл статей предназначен для тех, кто не имеет представления о микроконтроллерах (МК) и совершенно не умеет программировать. Как правило, это совсем юные ребята или же умудрённые многолетним опытом ветераны радиотехники. Этим категориям радиолюбителей наиболее затруднительно освоить МК, поскольку им особенно сложно разобраться в серьёзных статьях о МК. Надеюсь, что, прочитав предлагаемый небольшой цикл статей, они смогут разрабатывать простые устройства на МК, а может, не остановившись на этом, пойдут дальше, всё глубже и глубже проникая в сложный, но увлекательный мир МК. В цикле статей пойдёт речь о МК серии PIC, производимых американской компанией Microchip Technology Inc. Название PIC является сокращением от англ. peripheral interface controller, что означает контроллер интерфейса периферии.*

Рассказывать о преимуществах применения МК не имеет смысла — это понимают даже те, кто никогда с ними не работал. Например, один МК может заменить с десяток обычных микросхем малой и средней степени интеграции: счётчики, регистры, логические элементы и т. п. Кроме того, даже самые простые и недорогие современные МК, как правило, содержат в своём составе аналого-цифровые преобразователи (АЦП), цифро-аналоговые преобразователи (ЦАП), регуляторы с широтноимпульсной модуляцией (ШИМ), компараторы и т. п. Благодаря этому при разработке многих устройств можно обойтись всего одним МК. А это значительное уменьшение объёма работы с “железом”, ведь развести, протравить, просверлить и залудить плату для десятка микросхем, а затем и распаять их довольно трудоёмко. Если они будут заменены одним МК, произойдет ещё и существенное уменьшение габаритов получившегося изделия. Согласитесь, в применении МК сплошные плюсы!

Но это одна “сторона медали”. А у всякой медали есть и вторая сторона

— появляется работа, связанная с программированием, и именно это является основным сдерживающим фактором применения МК радиолюбителями. Судите сами. Прошло уже около 20 лет, как в журнале была напечатана серия статей А. С. Долгого, где подробно рассказывалось о проектировании устройств на МК, их разработке и отладке [1, 2]. В этих источниках подробнейшим образом описаны практически все этапы, от создания алгоритма будущего устройства, до прошивки МК. Несмотря это, для подавляющего большинства радиолюбителей МК остаются пока Terra incognita. (Неизвестная земля). О справедливости данного утверждения свидетельствует относительно малый процент конструкций, выполненных на МК. Объясняется это тем, что для разработки устройства на МК недостаточно иметь навыки и опыт разработки, сборки и отладки устройства на обычных дискретных радиоэлементах и цифровых микросхемах. При создании устройства на МК необходимо ещё и суметь составить и отладить программу для него. А перенести опыт работы

электронщика в область работы программиста не так просто и совсем не легко.

Чтобы написать, отладить, скомпилировать и прошить программу в МК, необходимо научиться работать инструментами программиста. А это — различные программы — симуляторы, отладчики, компиляторы, а также программаторы. Также необходимо знать хотя бы один язык программирования, ассемблер или Си. В идеале, конечно, оба. Но это совсем непростые языки программирования. К тому же, чтобы писать даже простые программы на Ассемблере, необходимо представлять себе МК как минимум на уровне ячеек памяти, регистров и портов ввода—вывода. Я не говорю уже о том, что необходимо просто хорошо знать ПК, чтобы найти, установить и научиться пользоваться вышеперечисленными программами. Согласитесь, всего этого с лихвой хватит, чтобы отбить у радиолюбителя, незнакомого с программированием, всякое желание осваивать МК.

Но я утверждаю, что если радиолюбитель понимает принципы работы цифровых микросхем и разрабатывает

ЖУРНАЛ В ЖУРНАЛЕ



на них устройства, он сможет легко научиться писать простые программы для МК. То есть такие программы, которые, например, что-то включают/выключают в зависимости от времени или какого-то сигнала, выдерживают паузу, отсчитывают нужное число импульсов или генерируют сигнал с заданной частотой и скважностью. Такие программы, в которых нет сложных математических вычислений, прерываний и ветвлений, проверок различных флагов и т. п. Именно это я хочу показать на примере создания устройства с простым алгоритмом работы. Возможно, это станет вашим первоначальным толчком к применению МК, ну а дальше всё зависит от того, понравится ли вам создавать устройства, проводя основную часть времени не с паяльником и осциллографом, а с карандашом, бумагой и ПК.

Итак, вернёмся к основной причине, почему большинство радиолюбителей, несмотря на желание, не могут начать осваивать МК. Дело в том, что многим радиолюбителям сложно самостоятельно разобраться в серьёзных статьях на эту тему и в даташитах на МК. Большинство имеющихся публикаций на эту тему предназначено для тех, у кого уже есть даже не начальные, а вполне серьёзные знания в программировании. Даташиты на МК написаны очень сжато и столь серьёзным языком, что свободно ориентируются в них только программисты. К сожалению, эта же история повторяется и в Интернете — множество сайтов просто перепечатывают одну и ту же информацию, понять которую может либо программист, либо тот, кто уже имеет навыки работы с МК. В качестве начала обучения часто предлагается научиться мигать светодиодом. Конечно, проще этой задачи ничего придумать нельзя, но даже и эта задача невыполнима, если у вас нет хотя бы начальных знаний языка программирования и архитектуры МК, а также умения работать в среде разработки программ. Ну а конфигурирование самого МК — задача куда сложнее, чем мигание светодиодом! Об этом свидетельствуют встречающиеся на форумах просьбы начинающих выложить конфигурацию МК для того, чтобы они смогли начать писать программы для них!

Поскольку в начале изучения МК его настройка представляет наибольшую сложность, я приведу в этой статье листинг программы, в котором МК уже сконфигурирован и настроен. В этот листинг нужно будет просто вставлять программы своих конструкций. В данной статье вашему вниманию предлагается метод для начала изучения МК, в котором учтены все

вышеперечисленные задачи и сложности. В процессе обучения будет сказано, что нужно запомнить на этом этапе, а что можно пока проигнорировать без ущерба для понимания. Естественно, изучать только теорию абсолютно бессмысленно, необходимо изготавливать какое-либо устройство. Для опытного электронщика начинать изучение МК с бесполезного мигания светодиодам совсем уж примитивный и неинтересный путь. Но и разрабатывать сложный прибор, например, вольтметр, часы или MP3-плеер, тоже не получится — вы не сможете написать программу для него.

Приобретать различные стартовые наборы, с моей точки зрения, ни к чему, поскольку через неделю-другую они окажутся ненужными, да и цена на них неоправданно высока. Поэтому я подобрал конструкцию, которая не будет валиться без дела, а алгоритм её работы понятен каждому. Предлагаю начать изучение МК с изготовления переключателя гирлянд. Программу для такого устройства составить очень легко, и вам будет не очень трудно постигать азы программирования на языке ассемблера. Внося в программу несложные изменения, можно получить различные варианты переключения (бегущие огни, бегущая тень, мигание и т. п.). При этом вы будете нарабатывать опыт программирования. Ну а если переключатель гирлянд кажется вам бесполезным устройством, потому что перед Новым годом ими забыты все магазины, через некоторое время вы сможете превратить его в "светильник настроения". Описаний таких светильников в Интернете огромное число, а вот в магазинах их практически нет. Да и не всегда целесообразно тратить деньги на то, что можно сделать самому.

Есть ещё одна приятная новость. Изучив этот цикл статей, вы сможете сделать три конструкции: генератор прямоугольных импульсов с различной скважностью; таймер с практички с неограниченной выдержкой (звучит фантастически, но это так!); ну и, собственно переключатель гирлянд (с большим количеством вариантов переключений). Если я вас убедил, начнём.

После выхода в свет статей [1—3], прошло уже много лет, и некоторые сложности, упоминавшиеся в них, к счастью, уже преодолены. Это касается отсутствия документации и справочной литературы на русском языке, а также отсутствия русскоязычных программных средств разработки и отладки программ для МК. Сейчас в Интернете можно скачать и даташиты, и справочную литературу, и различные самоучители. Там же можно купить и

программатор, а не собирать его самому, как 20 лет назад. Есть множество сайтов и форумов, где начинающим могут помочь советом и грамотным ответом, хотя частенько будут отправлять опять же к даташитам. Поэтому без небольших начальных знаний на форумах делать нечего.

Если вы не радиоинженер и не программист, значит, будете изучать МК в свободное от работы или учёбы время. Поэтому настройтесь на то, что процесс изучения займёт какое-то время, но никак не несколько дней. При этом подразумевается, что у вас имеется ПК и вы уверенный пользователь, потому что придётся скачивать и устанавливать на него необходимые программы, а также учиться ими пользоваться. Если же вы пользователь, но не очень уверенный, придётся искать того, кто вам поможет в этом, потому что здесь будут даваться начальные сведения о работе с МК, а не с компьютером и программами.

Впрочем, даже если у вас нет более опытного помощника, установить необходимые нам программы и научиться ими пользоваться можно с помощью Интернета. Ещё раз перечислю необходимые действия для начала изучения МК:

- иметь твёрдую уверенность и горячее желание научиться разрабатывать устройства на МК;
- скачать и установить на ПК среду разработки и отладки программ — MPLAB;
- приобрести программатор и установить его драйвер на ПК;
- если необходимо, скачать и установить на ПК программу-эмулятор COM-порта;
- скачать русскоязычную документацию на МК. В нашем случае — это PIC12F675.
- найти или скачать журналы, в которых напечатаны статьи, перечисленные в списке литературы.
- найти или скачать какой-нибудь самоучитель по программированию. Его нужно начинать читать одновременно с чтением данной статьи. Понадобится он и в дальнейшем, когда стадия "суперначинающего" будет вами преодолена и вы захотите двигаться дальше.

Согласитесь, что всего вышеперечисленного немало! Но, именно поэтому, первым пунктом и стоит горячее желание.

В самом начале пути необходимо определиться с программатором. Он нужен для записи созданной программы в МК. Если у вас достаточно опыта, можете сделать его самостоятельно, найдя описание, например, в Интернете. Можно купить и готовый программатор, тем более, что в Интернете он будет стоить дешевле,

чем комплектующие для него. Заказывая или изготавливая программатор, определитесь, нужен ли для его работы COM-порт. И если да, то вам придётся ставить на свой ПК ещё и программу-эмулятор COM-порта. Ведь в современных ПК, а тем более ноутбуках, уже давно нет разъёма COM-порта. Но самое главное, пока вы ждёте посылку с программатором, или изготавливаете его самостоятельно, — это не "остыть" к изучению МК.

Я применяю недорогой программатор, купленный в Интернете несколько лет назад. Приобретая его, я ещё мало что знал о МК и не знал, что некоторые МК имеют так называемую калибровочную константу, записанную на заводе-производителе. Также я не знал, что некоторые программаторы умеют самостоятельно считывать и вновь перезаписывать в МК эту константу, т. е. сохранять её. А некоторые программаторы, увы, не умеют этого делать. Так вот, чтобы облегчить себе жизнь, рекомендую выбрать программатор, который самостоятельно, без вашего участия, считывает и перезаписывает калибровочную константу.

Пока ожидаете или изготавливаете программатор, установите на ПК среду разработки и отладки программ — MPLAB. Подробно об этом можно прочитать в [1, № 8]. Там указано, где скачивать (приведены адреса сайтов), как устанавливать и как работать с этой программой. Поскольку со времени выхода статьи прошло уже около 20 лет, скачивать нужно, конечно же, более современную версию программы. Учиться работать с ней можно тоже с помощью Интернета, там есть множество сайтов, где работа с ней расписана подробнейшим образом.

Итак, программы установлены, даташит на русском языке скачан [4], программатор имеется, PIC12F675 приобретен, желание не пропало. Тогда идём дальше — начинаем изучать даташит. Ещё раз напоминаю, если вы радиолюбитель, для которого всё, что связано с программированием, — "тёмный лес", который вас пугает, не давая начинать осваивать МК. Не пугайтесь, пока пропускайте всё незнакомое. На что нужно обратить внимание, будет сказано.

В самом начале даташита, на с. 1 приведены основные параметры МК PIC12F629/675. Даже если вы далеки от программирования, согласитесь, основное понятно: максимальная тактовая частота — 20 МГц; напряжение питания — 2...5,5 В; шесть каналов ввода/вывода. Также в своём составе он имеет аналоговый компаратор, АЦП (только у PIC12F675), два таймера TMR0 и TMR1. Для начала это даже очень много. Скорее всего, у вас воз-

ник вопрос: сколько же смогли "повесить" на шесть функциональных выводов? Посмотрите на структурную схему МК, и вы увидите, что один его вывод может иметь от трёх до пяти функций. Это сделано, чтобы минимизировать общее число выводов МК. Именно поэтому практически любой вывод (кроме выводов питания) может быть настроен на множество функций: просто цифровой вход/выход, аналоговый вход АЦП или компаратора, вход счётчика или выход тактовой частоты и т. п. Это, конечно, рай для опытных разработчиков, но, увы, — сложность для начинающих, потому что такое обилие возможностей влечёт за собой необходимость тщательной настройки и МК в целом, и каждого его вывода. И вот это действительно не так просто!

Поэтому первый главный совет: пока не запоминайте, всё, чем может быть каждый вывод МК. Даже для такого простого МК, как PIC12F675, сделать это в начале изучения довольно сложно. Тем более, что, возможно, будут непонятны некоторые обозначения. Также не сильно пугайтесь схем, диаграмм и таблиц и пока не тратьте время на попытку глубоко их осмыслить. Второй главный совет — не нужно сразу пытаться выучить как пользоваться всеми возможностями МК и как настраивать его на это обилие возможностей. Это будем делать постепенно, по мере изучения. Поверьте, если вы в начале начнёте пытаться сразу всё выучить, запомнить и применить, то вас ждут два пути. Первый — если для вас всё окажется ясным и понятным, значит, вы очень умный и талантливый человек, и вас ждёт большое будущее на поприще программирования. А второй — вы просто запутаетесь и, скорее всего, оставите мысль о применении МК. Итак, запоминаете только то, что объяснено, остальное пропускаете.

Ну а потом у вас впереди опять два пути. Первый — от создания программ вы начнёте получать большее удовольствие, чем от работы с паяльником. Тогда вы сами всё глубже и глубже начнёте изучать и программирование, и МК, посвящая этому всё больше и больше своего времени. Второй путь — вы поняли, что программирование это не ваша стезя. Ну что ж, тогда у вас останется умение разрабатывать простые конструкции на МК. А это тоже немало.

Естественно, в самом начале изучения МК нас интересуют его выходы, чтобы чем-то управлять. Но понятно, что на будущее неплохо было бы узнать ещё и про входы, чтобы сам МК мог воспринимать какие-то внешние воздействия.

Посмотрите на таблицу 1-1 в даташите на с. 7. Она так и называется

"Назначение выводов МК PIC12F629/675". Следует запомнить, что наш МК имеет шесть универсальных выводов, пять из которых двуправленные, т. е. могут быть назначены либо входом, либо выходом. Для этого случая обозначаются они так: GP0 — вывод 7, GP1 — вывод 6, GP2 — вывод 5, GP4 — вывод 3 и GP1 — вывод 2. Ещё один вывод 4 может быть либо инверсным входом сброса, тогда он обозначается как MKLR с чертой наверху, либо только входом, тогда он обозначается GP3. Ещё запоминаем то, что GP0, GP1, GP2, GP3, GP4, GP5 называются каналами порта ввода/вывода. Называется этот порт GPIO, и мы видим, что он шестирядный. На будущее запомним, что настройка этого порта заключается в том, чтобы назначить выбранные выводы либо выходами, либо входами. Есть ещё одна приятная мелочь, если потребуются подтягивающие резисторы, знайте, что они уже встроены в МК! Их можно программно (т. е. написав специальные команды в программе) подключать к тем выводам, которые будут назначены входами. Но знайте, что есть и досадная мелочь — у входа GP3 такого резистора нет, поэтому при необходимости придётся ставить его самому, устанавливая резистор на плату. Согласитесь, что программно это сделать легче, — не надо паять резисторы, а это экономия времени, да и места на плате. Хотя не всё так однозначно, иногда может понадобиться установить внешний подтягивающий резистор, так как внутренний резистор имеет сопротивление около 40 кОм.

Как мы и договаривались, всю остальную информацию пока пропускаем, ведь для изготовления переключателя "бегущий огонь" знание только выходов уже достаточно. Для начала можно сделать переключатель на трёх светодиодах (трёхканальный переключатель), ну а в дальнейшем можно будет увеличить их число до пяти. Решим, что катоды светодиодов будут подключены к минусовой линии питания, тогда включаться они будут подачей на их аноды напряжения питания, что более привычно, а поэтому будет и более понятно при составлении программы.

Как вы знаете, эффект "бегущего огня" заключается в том, чтобы последовательно зажигать и гасить светодиоды. Собрать такое устройство можно всего на трёх транзисторах, на которых собран многофазный мультивибратор, или на одной микросхеме. В первом случае устройство получится простым, но с трудом поддающимся хоть какому-то усовершенствованию. Ведь чтобы изменить частоту пере-

ключения, придётся одновременно заменить три конденсатора. В случае применения микросхем его возможности расширяются, но только в рамках воплощённого алгоритма. Именно поэтому устройства на обычных микросхемах называются устройствами на жёсткой логике, т. е. чтобы устройство смогло выполнять ещё какие-то функции, оно должно быть существенно переработано. Устройства на МК называются устройствами на гибкой логике, т. е. изменяя не схему, а только

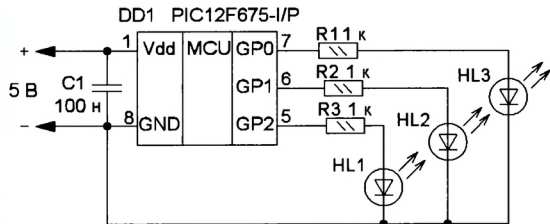


Рис. 1

программу, можно заставить одно и то же устройство работать совершенно по-другому. Естественно, в пределах разумного, так как получить из переключателя гирлянд цифровой вольтметр, конечно же, не получится.

Схема "бегущего огня" на МК показана на рис. 1. Поскольку в качестве выходов пять из шести выводов абсолютно равнозначны, решим, что светодиоды будут подключаться к выводам, которые обозначены как GP0, GP1 и GP2. Выводы GP4 и GP5 использованы пока не будут. Всё, схема собрана! Собрать её не просто, а очень просто. Для изменения частоты переключения не потребуется замена элементов. Но вот выходит на сцену "обратная сторона медали" — МК без программы, записанной в него, представляет собой просто кусочек кремния в пластике, поэтому собранное устройство, конечно же, не заработает.

Вот и настало время приступить к написанию программы. Первое, что нужно сделать, это просто составить для себя словесное описание того, что должно делать ваше устройство. Затем на основе этого описания нужно составить алгоритм его работы. Кстати, этот этап присутствует не только у программиста, при составлении программы, но и у радиолюбителя, при разработке устройства на дискретных элементах. Отличие между ними лишь в том, что радиолюбитель добивается выполнения алгоритма, соединяя между собой различные элементы с помощью паяльника, а программист пишет по заданному алгоритму программу, которую выполняет МК. Следует сразу предупредить, что это довольно существенное отличие! Вполне возможно, оно вам понравит-

ся, и вы начнёте всё больше и больше заниматься программированием, но возможно, что нет, тогда вы оставите это занятие. В конце концов, мир состоит не из одних программистов!

Как должен работать наш переключатель, совершенно ясно. Включается первый светодиод и светит некоторое время, затем гаснет и включается второй светодиод, который тоже светит некоторое время. После того как он погаснет, включается третий светодиод и тоже светит такое же время. После погасания третьего светодиода вновь включается первый, и далее всё повторяется. Конечно, программист напишет такую простую программу даже без составления алгоритма. Но, во-первых, вы пока ещё совсем начинающий программист, а во-вторых, привыкайте работать, составляя алгоритм, это правило хорошего тона, как и делать подробные комментарии к составляемой программе. Пренебрегающие

ной, поищите в Интернете что-нибудь попроще.

Получился вот такой алгоритм. Для очень внимательных я сразу скажу, что в этом алгоритме есть небольшое упущение. Сделано это специально, чтобы показать, как отлаживать программу:

1. Включить первый светодиод, т. е. подать лог. 1 на выход GP0.
2. Выключить первый светодиод, т. е. подать лог. 0 на выход GP0.
3. Включить второй светодиод, т. е. подать лог. 1 на выход GP1.
4. Выключить второй светодиод, т. е. подать лог. 0 на выход GP1.
5. Включить второй светодиод, т. е. подать лог. 1 на выход GP2.
6. Выключить второй светодиод, т. е. подать лог. 0 на выход GP2.
7. Перейти на строку № 1.

Это запись алгоритма словами. Но обычно его изображают графически, так как рисунок более наглядно показывает нам последовательность действий, связи, переходы и т. п. Графическое изображение нашего алгоритма приведено на рис. 2.

Кстати, замечу, что сейчас это тоже программа, только написана для человека на понятном ему человеческом языке, поэтому в таком виде более точно называть её инструкцией. Если вы посадите человека за выключатели и дадите ему эту инструкцию, получите "бегущие огни", правда, ненадолго, потому что человек скоро устанет, начнёт сбиваться и путаться в выключателях. Лучше, конечно, поручить эту задачу МК. Но для этого нужно написать эту инструкцию, точнее уже программу, на языке, понятном для него. Сделать это по такому подробному алгоритму будет очень легко даже начинающему. А именно это нам и нужно.

#### ЛИТЕРАТУРА

1. Долгий А. Разработка и отладка устройств на МК. — Радио, 2001, № 5, с. 17—19; № 6, с. 24—26; № 7, с. 19—21; № 8, с. 28—31; № 9, с. 22—24; № 10, с. 14—16; № 11, с. 19—21; № 12, с. 23—25. Радио, 2002, № 1, с. 1—19.

2. Долгий А. О настройке тактового RC-генератора, встроенного в МК PIC12F629 и PIC12F675. — Радио, 2009, № 3, с. 22, 23.

3. Гладштейн М. Проектируем устройства на микроконтроллерах. — Радио, 2000, №11, с. 25, 26; № 12, с. 20—23.

4. Однокристалльные 8-разрядные FLASH CMOS микроконтроллеры компании Microcip Tecnolodgi Incorporated PIC 12F629, 12F675. — URL: [http://www.microchip.ru/files/dsheets-rus/PIC12F629\\_675.pdf](http://www.microchip.ru/files/dsheets-rus/PIC12F629_675.pdf) (27.11.20).

(Продолжение следует)

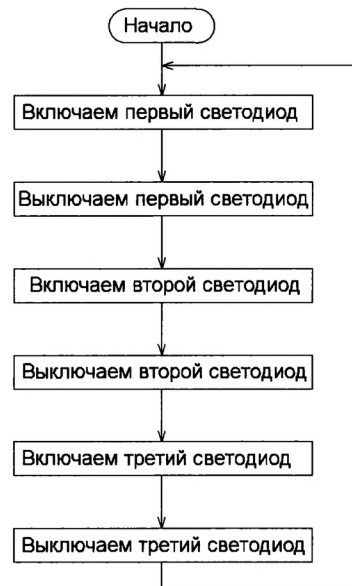


Рис. 2

этимися правилами начинающие программисты через некоторое время сами с трудом разбираются в своих программах.

Составляя алгоритм по описанию работы устройства, необходимо описать всё, до мельчайших подробностей и учесть все возможные варианты работы устройства. Если захотите глубже разобраться в этом, попробуйте изучить статью [3]. Если она покажется вам слож-