Применение микроконтроллеров PICmicro для подключения к Интернет по протоколу PPP

Статья основывается на технической документации DS00724c компании Microchip Technology Incorporated, USA.

Москва - 2001

тел. (095) 737-7545
www.microchip.ru

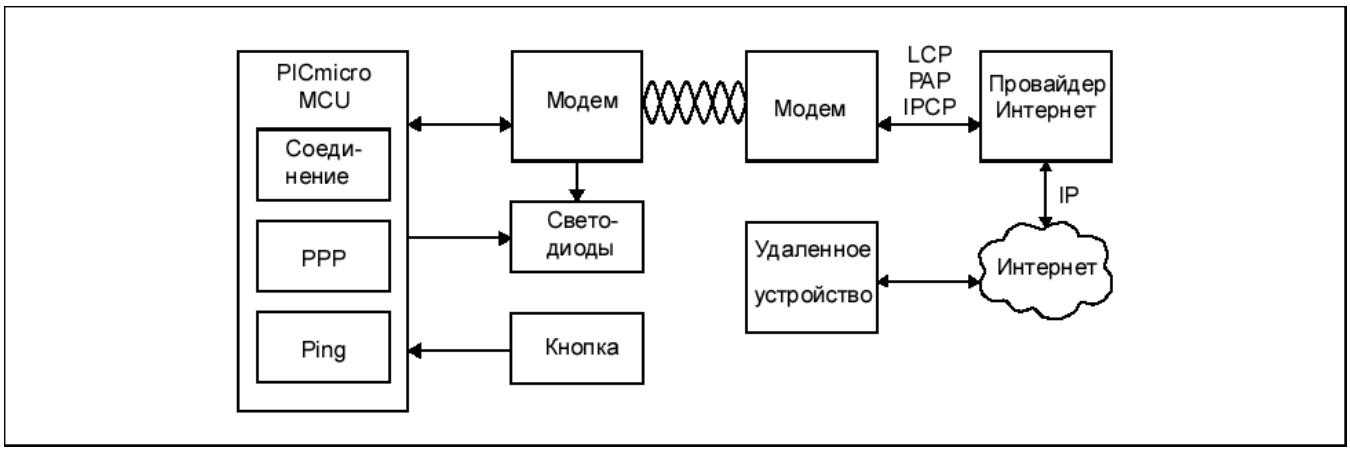
Применение микроконтроллеров PICmicro для подключения к Интернет по протоколу PPP

Статья основывается на технической документации DS00724b
компании Microchip Technology Incorporated, USA.

Микроконтроллеры PICmicro могут быть использованы для построения недорогого устройства подключения к Интернет. Как правило, Интернет интерфейс поддерживают персональные компьютеры, с установленной операционной системой и приложениями.

В данной статье будет рассмотрен пример устройства, которое подключается к Интернет по протоколу PPP и отвечает на поступившие Ping запросы. Пользователь может сам дополнить программное обеспечение алгоритмами передачи файлов TFTP или реализации простого протокола управления сетью SNMP.

Блок схема устройства



Протоколы PPP и SLIP используют модемы для подключения к Интернет. Протокол PPP был выбран потому, что он более универсален, не требует специфичной последовательности начала связи, проверяет контроль линии связи (не реализовано в данном примере).

Пример программы занимает 145 байт ОЗУ и 2170 слов ПЗУ. Загрузка процессора будет зависеть от скорости передачи данных. Алгоритм требует некоторого времени для передачи данных и обработки принятых пакетов. В алгоритм программы не включает протоколы Email, Telnet, FTP, Web. При обрыве соединения, алгоритм произведет попытку повторного подключения.

Протоколы Интернет

В сеть Интернет включено большое количество типов компьютеров и устройств, а правила их работы сведены в тысячи стандартов и соглашений. Данные в сети Интернет передаются пакетами от одного компьютера к другому.

В пакете указывается тип передаваемых данных (часть Web страницы, письмо Email и т.д.), для обработки соответствующей программой на компьютере. Пакеты бывают двух классов - UDP более простой, и TCP более сложный, с дополнительными пакетами начала/завершения связи и повтора потерянного сообщения.

Каждый компьютер (устройство) получает уникальный адрес в сети, например 10.241.45.12, и работает подобно обычному почтовому адресу.

В PPP протоколе требуется, чтобы был установлена последовательная передача 8-разрядных данных без бита паритета. В начале и в конце пакета передается знак тильды (~), шестнадцатеричное значение 0x7E. Подробное описание служебных символов в PPP протоколе смотрите в документации RFC 1662 “PPP in HDLC-like Framing”.

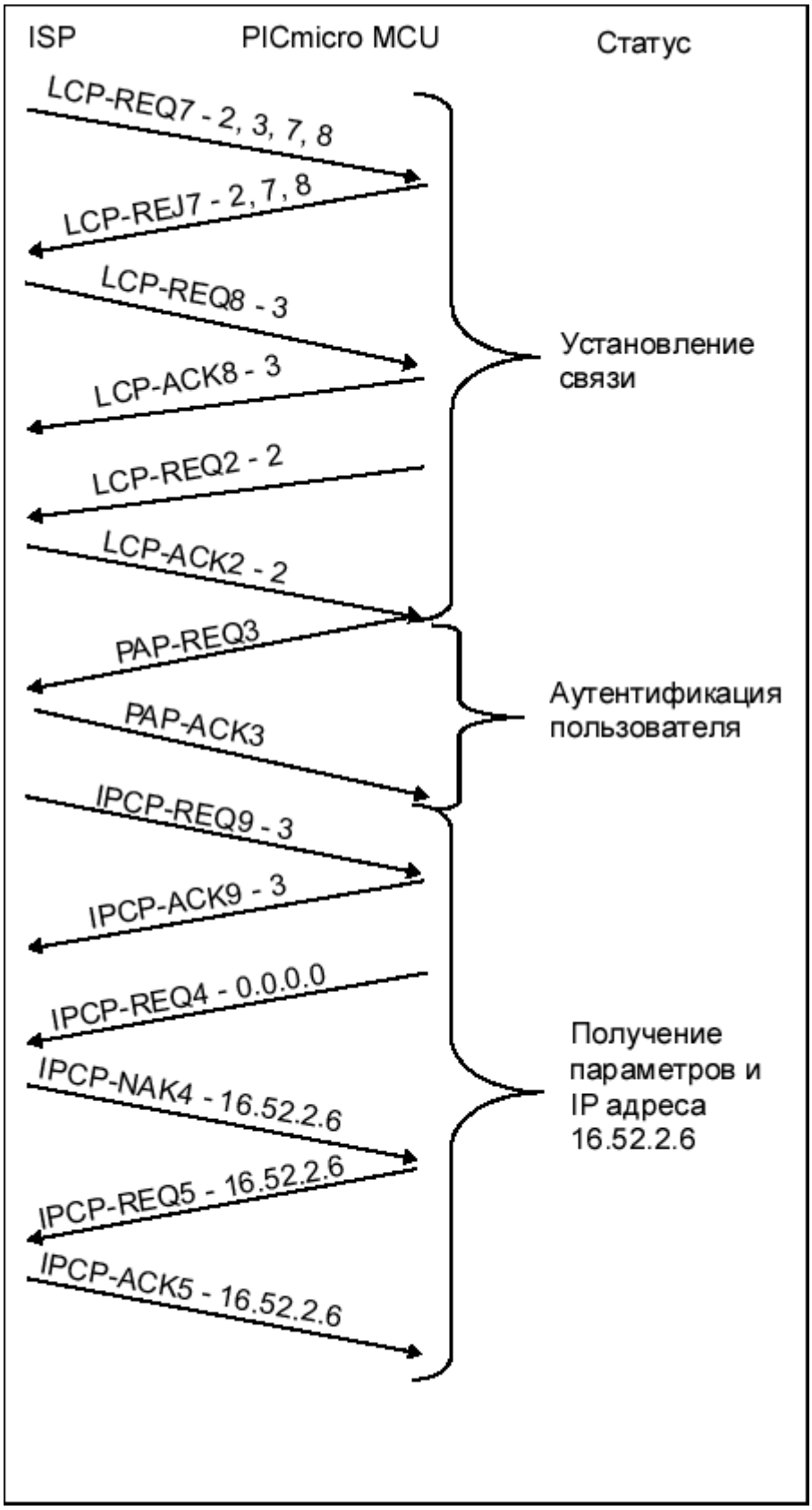
Выполнение PPP соединения состоит из нескольких стадий:
- обнаружение модема;
- установка соединения с удаленным компьютером (LCP);
- проверка имени и пароля доступа (PAP);
- согласование протокола сжатия данных (CCP);
- настройка IP параметров (IPCP).

Протоколы управления LCP, PAP, CCP, IPCP очень похожи друг на друга, но имеют различные значения на соответствующих стадиях подключения. Каждый пакет может запрашивать, отвергать или принимать список выбора. Соединение начинает пакетом запроса (REQ) списка параметров. Каждый список параметров состоит из:
- байта типа параметров;
- байта длины;
- список параметров.

После получения списка параметров должен быть сформирован пакет подтверждения (ACK). Если предложенные параметры не удовлетворяют, генерируется ответ NAK отвергающий весь список. Если некоторые параметры не удовлетворяют, формируется пакет REJ со списком параметров, которые не могут быть выполнены. Первая сторона модернизирует и передает новый список параметров до тех пор, пока не будет получено подтверждение ACK.

В алгоритме программы не реализован механизм отвержения списков параметров.

PPP подключение



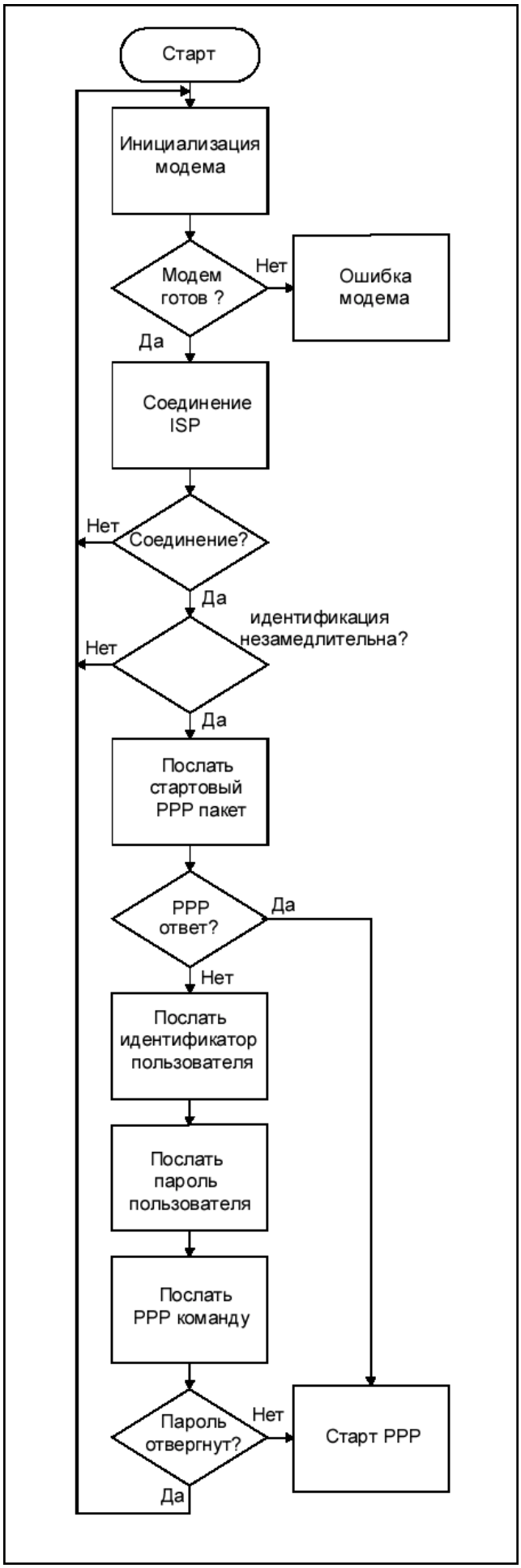
Основные термины:
ACK – подтверждение
CCP – протокол управление сжатием данных
CRC – циклическая контрольная сумма
CHAP – протокол установления подлинности соединения
DNS – система управления символьными адресами
DTR – готовность терминала данных
FTP – протокол передачи файлов
ICMP – Интернет протокол управления сообщениями
IP – Интернет протокол
ICIP – протокол настройки Интернет протокола
ISP – служба доступа в сеть Интернет
LCP – протокол управления соединением
MRU – максимальное принятое значение
NAK – отрицательное подтверждение
PAP – протокол проверки пароля
PPP – протокол соединения точка-точка
REJ – отключить
REQ – запрос
RFC – соглашение
SLIP – последовательный протокол связи с Интернет
SNMP – простой протокол управление сетевым оборудованием
TCP – протокол управление передачей
TFTP – простой протокол передачи файла
TTL – время жизни
UDP – пользовательский протокол датаграм

Физическое соединение

Нельзя начать работать в сети Интернет пока не будет выполнено соединение модемов на линии связи. В данном примере модем набирает телефонный номер и выполняет соединение. Программа посылает в модем цепочку команд и ожидает ответ в течение некоторого времени. Максимальное время ожидания 3 секунды, после этого истечения этого времени считается, что модем неисправен или отключен.

После тестовой проверки модема, программа указывает телефонный номер и дает команду выполнить соединение. В течение 30 секунд модем должен сформировать ответ о выполненном соединении, если такой ответ не поступает, модем сбрасывается в исходное состояние и производится повторная попытка соединения.

Выполнив успешное соединение с удаленным компьютером, программа в течение 10 секунд ожидает запрос имени и пароля пользователя. Формируются первые пакеты протокола PPP. После проверки имени и пароля для доступа в сеть Интернет, начинается работа по протоколу IPCP для получения IP адреса.



Структурная схема выполнения соединения

LCP параметры

Сначала согласуются LCP параметры для установления связи. Типовой пакет LCP показан на рисунке.

LCP пакет состоит из:
- стартовая посылка 0x7E, 0xFF, 0x03
- указатель протокола LCP 0xC0, 0x21
- байт значения пакета 0x01
- байт идентификации, который увеличивается при каждом правильно обработанном сообщении
- 16-разрядное число байтов в пакете LCP
- список параметров
- 16-разрядная контрольная сумма
- стоповая посылка 0x7E

Краткое описание параметров списка LCP

Maximum-Receive-Unit – имеет длину 2 байта и указывает максимальный размер PPP пакетов. Можно было сделать этот параметр очень маленьким, для упрощения обработки пакетов микроконтроллером. Однако минимальное значение допустимое в этом параметре – 576, что очень много для микроконтроллера. Поэтому этот параметр не подтверждается.

Примечание. Пакеты длиной более 47 байт будут отвергнуты, из-за ограниченного объема ОЗУ микроконтроллера.

Async-Control-Character-Map – 4 байтный параметр, каждый разряд которого определяет первый символ ASCII от 0 до 31 в управляющем пакете. Если бит, указывающий символ ASCII установлен и получен управляющий пакет с таким же первым символом, то этот пакет подлежит обработке. Такой алгоритм позволяет упростить программное обеспечение, выделяющее управляющие пакеты.

Authentication-Protocol – параметр, указывающий метод передачи пароля. Используется, если регистрация еще не была проведена. Значение параметра равное 0xC023 выбирает метод передачи идентификатора и пароля в виде простого текста. При значении равном 0xC223 идентификатор передается в виде простого текста, а пароль будет зашифрован (CHAP протокол).

Magic-Number – 4-байтный случайный номер, для выполнения обмена данными по PPP протоколу. Параметр имеет высокую хаотичность. Значение параметра теряет уникальность после: трех попыток передачи данных, передачи и принятия данных, повторной передачи пакетов.

Protocol-Field-Compression - этот параметр не имеет никаких значений. Если требуется подтверждения компрессии данных, то отвечающая сторона формирует пакет с первым и вторым байтом не учтенными в протоколе обмена.

Address-and-Control-Field-Compression – параметр не имеет никаких значений. Для подтверждения отвечающая сторона должна сформировать PPP пакет с третьим и четвертым байтами 0xFF и 0x03 соответственно.

Работа по протоколу LCP завершается, после того как обе стороны подтверждают список параметров.

В таблице приведен список параметров протокола LCP и номера документов с их описанием.

Номер параметра	Название параметра	Номер документа
0	Vendor-Specific	RFC2153
1	Maximum-Receive-Unit	RFC1661
2	Async-Control-Character-Map	RFC1662
3	Authentication-Protocol	RFC1661
4	Quality-Protocol	RFC1661
5	Magic-Number	RFC1661
6	Quality-Protocol	Обсуждается
7	Protocol-Field-Compression	RFC1661
8	Address-and-Control-Field-Compression	RFC1661
9	FCS-Alternatives	RFC1570
10	Self-Describing-Pad	RFC1570
11	Numbered-Mode	RFC1663
12	Multi-Link-Procedure	Обсуждается
13	Callback	RFC1570
14	Connect-Time-Deprecated	
15	Compound-Frames	Обсуждается
16	Nominal-Data-Encapsulation	Обсуждается
17	Multilink-MRRU	RFC1990
18	Multilink-Short-Sequence-Number-Header	RFC1990
19	Multilink-Endpoint-Discriminator	RFC1990
20	Proprietary	
21	DCE-Identifier	RFC1976
22	Multi-Link-Plus-Procedure	RFC1934
23	Link-Discriminator-for-BACP	RFC2125
24	LCP-Authentication-Option	
25	Consistent-Overhead-Byte-Stuffing	(COBS)
26	Prefix-elision	
27	Multilink-header-format	
28	Internationalization	RFC2484
29	Simple-Data-Link-on-SONET/SDH	

В таблице приведен список кодов пакетов протокола LCP

Код	Тип пакета	Номер документа
0	Vendor-Specific	RFC2153
1	Configure-Request	RFC1661
2	Configure-Ack	RFC1661
3	Configure-Nak	RFC1661
4	Configure-Reject	RFC1661
5	Terminate-Request	RFC1661
6	Terminate-Ack	RFC1661
7	Code-Reject	RFC1661
8	Protocol-Reject	RFC1661
9	Echo-Request	RFC1661
10	Echo-Reply	RFC1661
11	Discard-Request	RFC1661
12	Identification	RFC1570
13	Time-Remaining	RFC1570

PAP протокол

Подробно протокол PAP описан в документации RFC1334. Алгоритм программы был упрощен к одному обмену пакетами. В LCP обмене был указано правило передачи идентификатора и пароля 0xC023. Устройство формирует пакет с идентификатором и паролем в текстовом формате. Если в ответ от сервера получает ACK пользователь зарегистрирован. В случае получения в ответ пакета NAK идентификатор или пароль неправильны, соединение прекращается.

Формат пакета протокола PAP.

IPCP протокол

После завершения работы по LCP и PAP протоколу необходимо настроить параметры работы в сети Интернет. Выбор IP протокола и параметров сжатия данных имеют большое количество нюансов описанных в документе RFC1332. Сервер посылает запрос, сопровождаемый IP адресом. Некоторые сервера передают параметры IP сжатия данных, однако, эти пакеты будут отвергнуты потому, что алгоритм сжатия не реализован в данном примере.

Формат пакета протокола IPCP.

Параметры IPCP протокола.

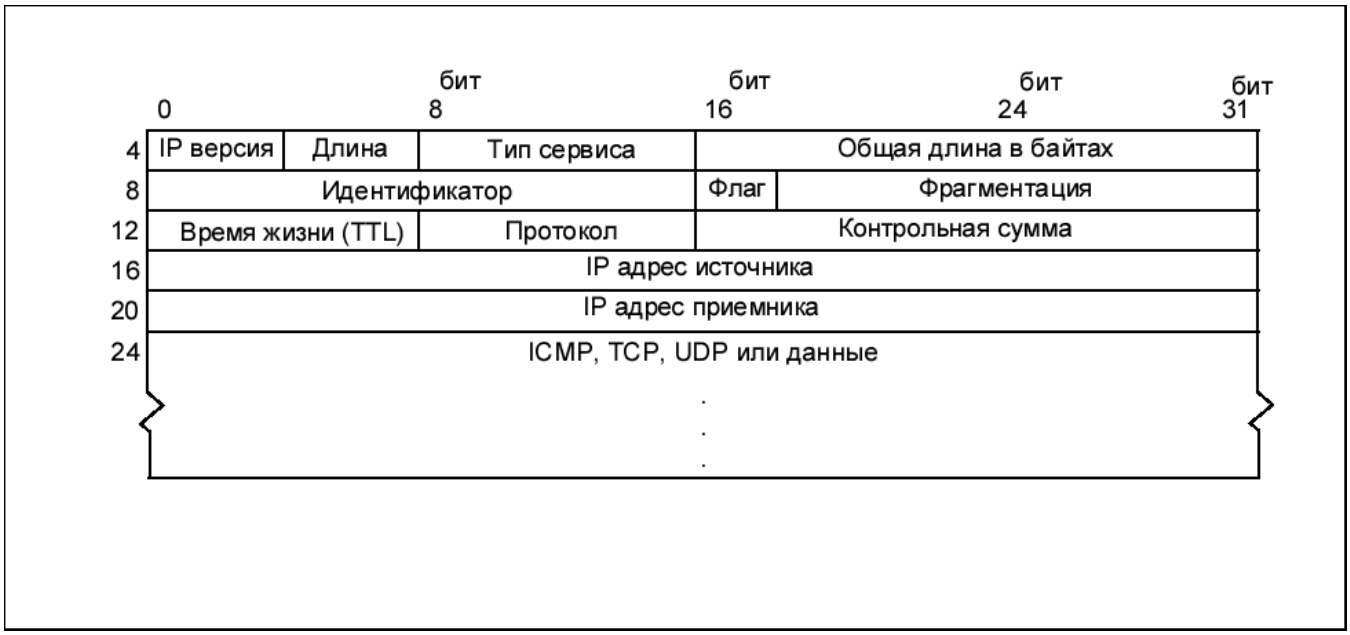
Код параметра	Название	Номер документа
1	IP-адрес	Обсуждается
2	IP-протокол сжатия	RFC1332
3	IP-адрес	RFC1332
4	Мобильный-IPv4	RFC2290
129	Адрес первого DNS сервера	RFC1877
130	Адрес первого NBNS сервера	RFC1877
131	Адрес второго DNS сервера	RFC1877
132	Адрес второго NBNS сервера	RFC1877

CCP протокол

Некоторые сервера будут пытаться указать параметры сжатия, но т.к. в алгоритме программы сжатие данных не реализовано эти пакеты будут отвергнуты. Алгоритмы сжатия данных сложны и требуют значительной производительности процессора, тем более что они эффективны при работе с пакетами большой длины.

ICMP соединение

Контрольное сообщение проверки соединения (Ping) посылается полным IP пакетом, пример показан на рисунке.



Этот протокол используется не только для проверки наличия устройства в сети, подробности можно найти в документации RFCS792 и RFCS950.

Работает протокол следующим образом: устройство формирует пакет проверки соединения с адресом удаленного устройства в сети и ожидает в течение 30 секунд ответ, для поддержания ISP связи. Устройство, которое сформировало сообщение, во время ожидания ответа, само может отвечать на удаленные запросы.

Пакет такого типа можно разбить на две основных части: стартовая IP часть, ICMP сообщение. Большую информативность содержит 20-байтная стартовая IP часть.

Первый байт разделен пополам: первая половина – версия IP, равная 4; вторая половина – количество 32-разрядных слов в IP части, в данном случае 5.

Второй байт – тип обслуживания пакета: оптимизация, минимизировать задержку, максимальная производительность, максимальная надежность передачи, минимизировать стоимость доставки пакета. Рекомендуется устанавливать тип обслуживания пакета – 0x00, что обозначает нормальное обслуживание без оптимизации.

Третий и четвертый байты указывает общее количество 16-разрядных слов, включая IP часть и ICMP сообщение.

Следующие 4 байта используются для фрагментации пакетов, т.к. в данном примере используются пакеты маленькой длины, поэтому этот алгоритм не реализован.

Девятый байт – время жизни пакета (TTL флаг). В этом байте указывается максимальное число маршрутизаторов, которые может пройти пакет, прежде чем будет отвергнут. Обычное значение этого байта 32 или 64.

Десятый байт указывает тип сообщения, которое передается вместе со стартовой IP частью.

В 11 и 12 байты записывается 16-разрядная контрольная сумма, описание смотрите в документации RFC1071.

Следующие 4 байт содержат IP адрес источника пакета.

И в последних 4 байтах указывается IP адрес приемника.

ICMP сообщение состоит из: байт типа сообщения, байт кода, контрольная сумма.

	бит 0		бит 8		бит 16	бит 24	бит 31
4	IP версия 0100	Длина 0101	Тип сервиса 0000 0000		Общая длина в байтах 00000000 00011100		
8	Идентификатор 1000 1000 0001 0000				Флаг 010	Фрагментация 00000 00000000	
12	Время жизни (TTL) 0111 1111		Протокол 0000 0001		Контрольная сумма 00110011 10100111		
16	IP адрес источника 11001101 11001000 00101101 01111100						
20	IP адрес приемника 11001111 10100001 01110101 01000011						
24	ICMP тип 0000 1000		ICMP код 0000 0000		ICMP контрольная сумма 11110111 11111110		
28	PING идентификатор 00000000 00000001				PING 00000000 00000000		

PPP пакет

```
0000 :  FF  03  00  21
0004 :  45  00  00  1C
0008 :  88  10  40  00
000C :  7F  01  33  A7
0010 :  CD  C8  2D  7C
0014 :  CF  A1  75  43
0018 :  08  00  F7  FE
001C :  00  01  00  00
0020 : 22   7C  7E
```

Байт типа имеет два значения: 8 – Ping запрос, 0 – Ping ответ.

Байт кода всегда равен нулю.

Контрольная сумма вычисляется с учетом стартовой IP части.

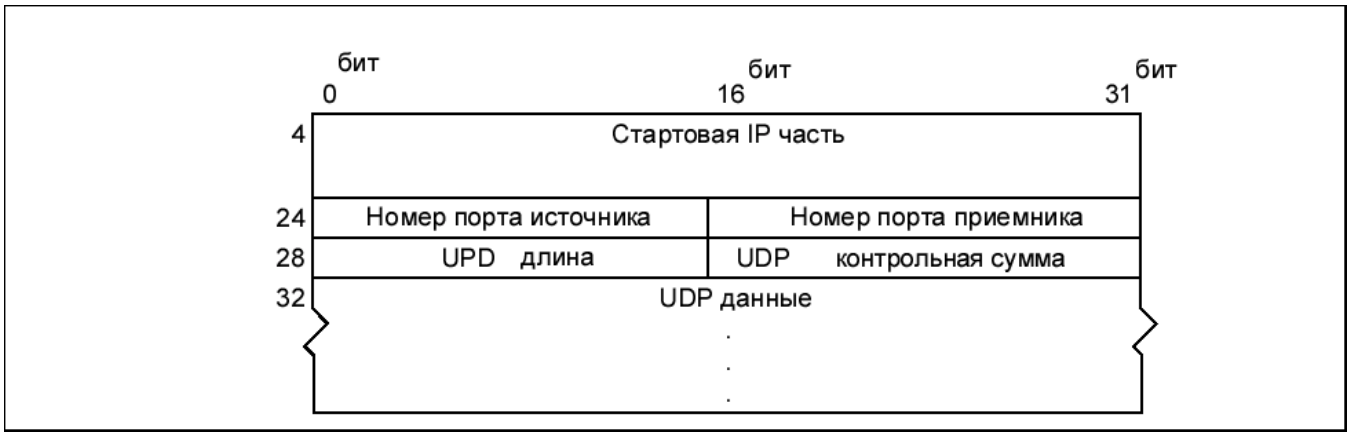
В случае выполнения передачи Ping пакетов в ICMP сообщение может быть включена произвольная информация. Алгоритм, реализованный в данном примере, отвечает на Ping пакеты, включая в ответ дополнительную информацию, которую пользователь может использовать по своему усмотрению.

UDP протокол

UDP протокол используется для передачи файлов TFTP, преобразования символьных имен узлов в цифровые адреса с помощью DNS серверов, работа с SNMP. Подробное описание протокола UDP - RFC768.

Алгоритм данного примера не использует UDP протокол, но это описание поможет его добавить в программу.

UDP – является надежным протоколом передачи данных, с возможностью повторной передачи потерянных пакетов. UDP пакет состоит из: стартовой 20-байтной IP части, стартовой 8-байтной UDP части и UDP данных.



Первые 2 байта – источник пакета. Следующие 2 байта – номер порта приемника. Например, порт 69 всегда используется для TFTP.

Пятый и шестой байты указывают длину UDP сообщения, 8 байт стартовой части UDP плюс данные.

Последние 2 байта стартовой части UDP – 16-разрядная контрольная сумма, включающая стартовую часть UDP и данные.

Контрольная сумма не обязательна UDP сообщении. Значение контрольной суммы 0x0000 означает, что контрольная сумма не проверяется. Если вычисленное значение контрольной суммы равно 0x0000, то она должна быть инвертирована к 0xFFFF.

При нечетном количестве байт в передаваемых данных, необходимо дополнить пакет байтом 0x00 до четного числа байт.

Формат UDP данных будет зависеть от номера порта, по которому выполняется соединение и рабочего протокола обмена. Протокол TFTP описан в документации RFC1350.

TCP протокол

TCP протокол используется для работы с FTP, Email, Telnet, HTTP. Первоначальная версия документации на TCP протокол – RFC793. Усовершенствованное описание TCP можно найти в документации RFC1122 и RFC1123.

Для реализации TCP протокола требуется большой объем ОЗУ и ПЗУ, поэтому в данном примере не реализуется.

Существенным отличием от UDP протокола является – возможности организации передачи пакетов через определенный промежуток времени, повторной передачи предыдущих пакетов, организация большого числа одновременных связей и др.

Аппаратная реализация

Этот пример был разработан для микроконтроллера PIC16C63A, чтобы показать, как небольшой PPP алгоритм связи может быть сокращен до необходимого минимума. Программа использует 151 из 192 байт ОЗУ, 2.2Кслов программы и 6 портов ввода/вывода. В микроконтроллере остались свободные ресурсы для размещения приложений пользователя.

В примере использован модем Ceremtek CH1786LC, с максимальной скоростью передачи данных 2400 бод. Для более скоростных приложений используйте другой модем (например CH1794 – 14.4 кбод). При выборе модема, убедитесь, что он поддерживает тип набора номера, Ваше телефонной станции. Дополнительным требованием к модему является – максимальный потребляемый ток 50мА.

Устройство работает от щелочной батарейки 9В емкостью 560мА/ч. Такой емкости хватит примерно на 9 часов непрерывной работы устройства. Если длительность звонка будет около 1 минуты, то батарейки хватит примерно на 500 звонков. Методом снижения общего энергопотребления, может быть дополнительная цепь отключения модема от питания во время ожидания и установка DC-DC преобразователя.

Индикацию состояния соединения отображают три светодиода – статус соединения, передача данных, прием данных.

Светодиод статуса имеет следующие состояния:

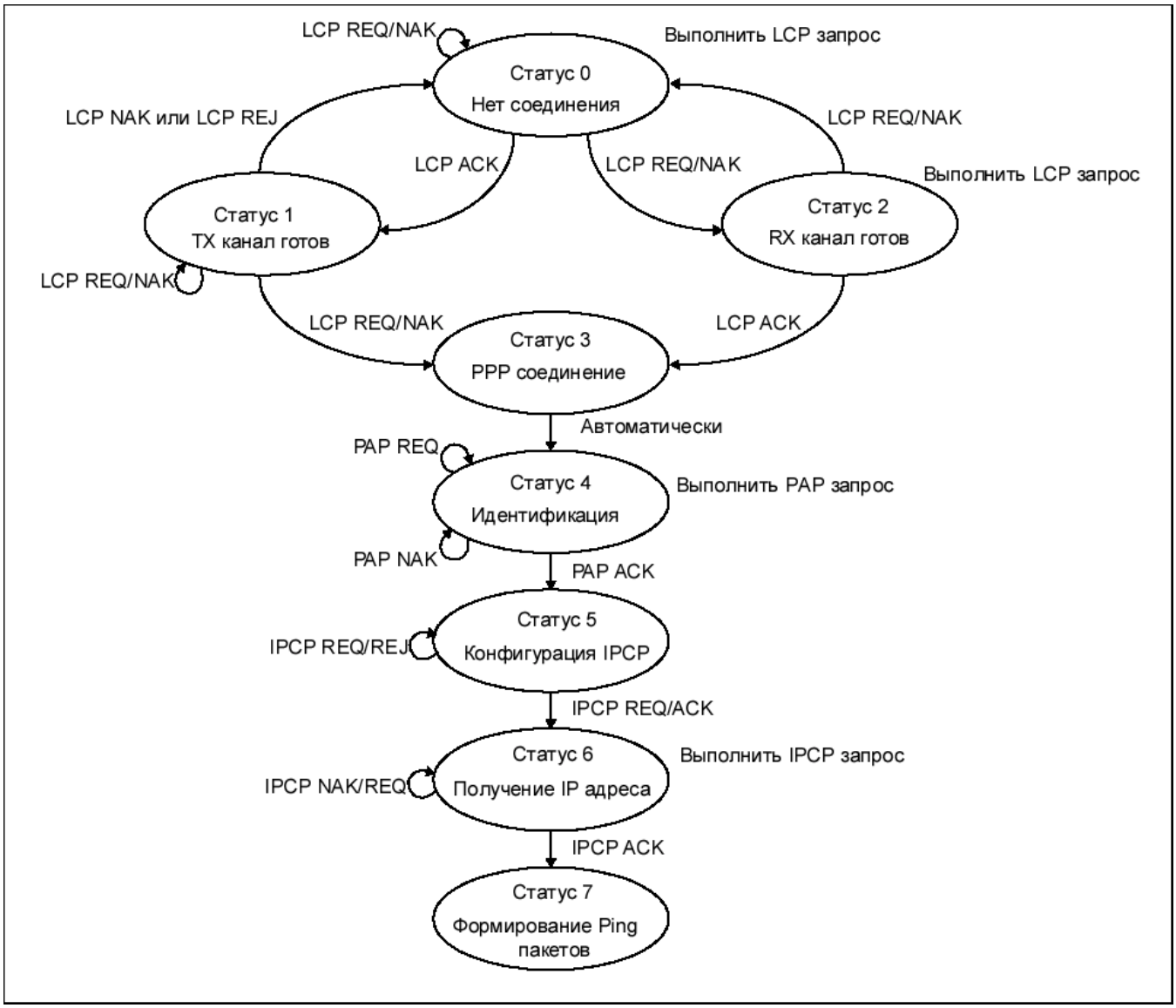
Выключен - модем не инициализирован;

Быстро мигает – выполняется набор номера;

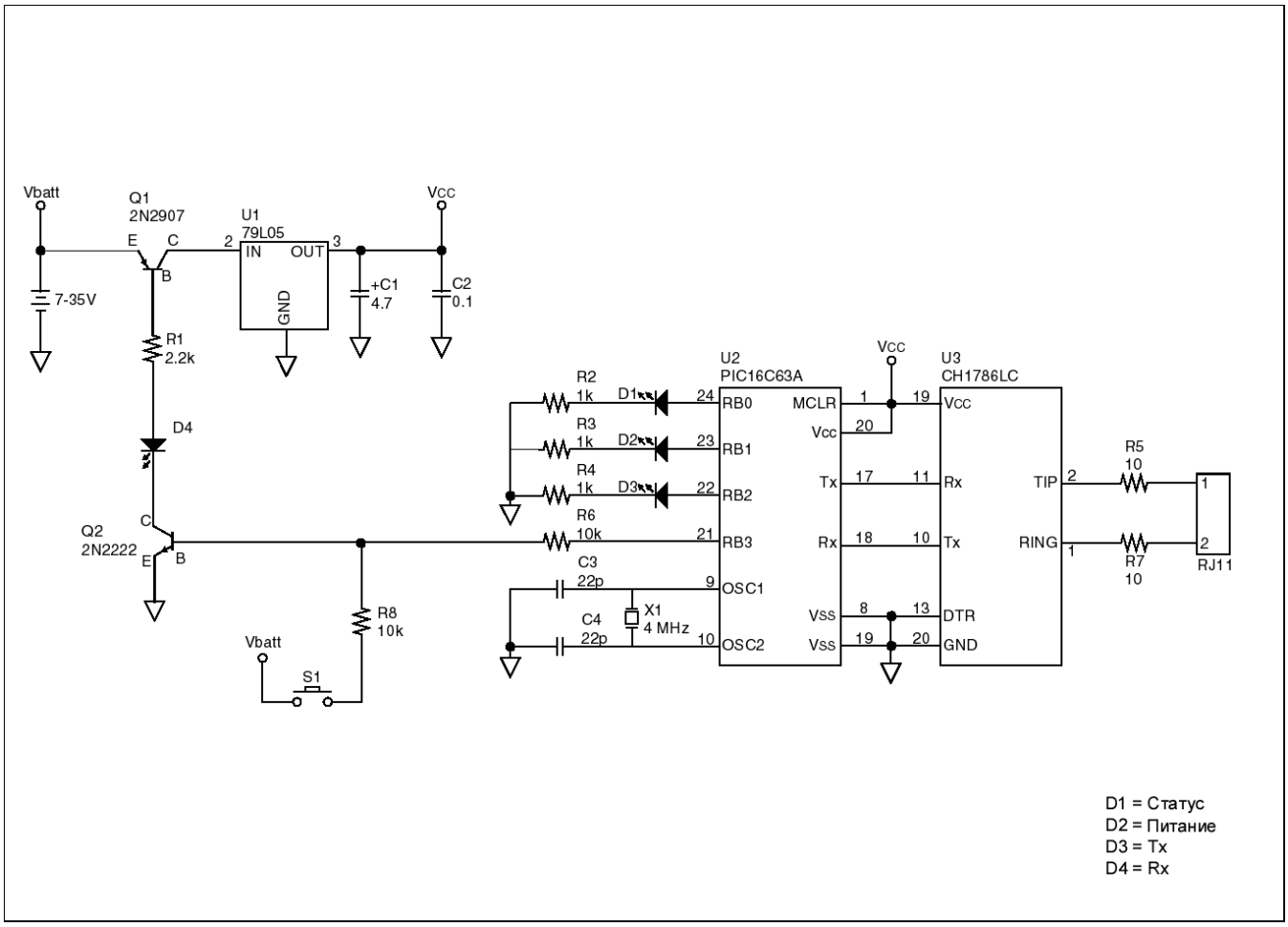
Светиться непрерывно – установлено PPP соединение.

Через две секунды после соединения светодиод гаснет и начинает передавать 32-битный IP адрес устройства. Длинная вспышка – 1, короткая вспышка – 0. Преобразовав полученную последовательность, сначала в шестнадцатеричные байты, а затем в десятичное значение получите текущий IP адрес устройства в сети Интернет.

Пример преобразования



Принципиальная электрическая схема устройства



Программное обеспечение

Программное обеспечение написано на языке С, и состоит из двух основных частей – управление модемом и работа с протоколами.

Нажав на кнопку (смотрите принципиальную электрическую схему) микроконтроллер включиться и через короткую задержку (около 250 мс) установит высокий уровень сигнала на выводе RB3. Пока пользователь удерживает кнопку нажатой, или сохраняется высокий уровень сигнала на выводе RB3 – схема остается включенной. Программное обеспечение переведет вывод RB3 в третье состояние (выключит устройство) после удачного соединения и передачи пакетов Ping, или после 20 неудачного соединения. Если кнопка все еще нажата, программное обеспечение будет повторять попытки соединения.

Программное обеспечение будет пытаться установить соединение до 20 раз с интервалом в 30 секунд. После установки соединения передается пакет PPP с идентификационной информацией. Если идентификация не происходит, соединение завершается, и алгоритм начинает выполняться сначала. Весь процесс соединения выполняется последовательно с учетом статуса алгоритма.

После выполнения соединения модемом статус алгоритма имеет значение 0. Когда сервер Интернет подтверждает LCP конфигурацию - устанавливается бит 0. Когда алгоритм подтверждает LCP конфигурацию - устанавливается бит 1. Пока бит 0 сброшен, алгоритм посылает запросы LCP каждую секунду. Когда оба бита установлены - статус алгоритма получает значение 4.

После подтверждения идентификационной информации – значение статуса алгоритма 5, а после подтверждения IPCP параметров – алгоритму присваивается уровень 6.

На уровне 6 – алгоритм посылает IP запросы с адресом 0.0.0.0 каждую секунду. Сервер должен ответить пакетом NAK с правильным IP адресом, чтобы перевести алгоритм в заключительное состояние 7.

В состоянии 7 выдается текущий IP адрес на светодиод, и формируются Ping пакеты каждые 30 секунд. После положительного ответа на Ping пакеты, устройство отключается, если кнопка питания не нажата.

Текст программы на языке С – 00724.zip

Статья основывается на технической документации DS00724b
компании Microchip Technology Incorporated, USA.