

Микроконтроллеры и сети Ethernet

Олег Вальпа (Челябинская обл.)

В статье описано простое и недорогое решение для реализации сетевых устройств Ethernet на базе трёх семейств микроконтроллеров.

ВВЕДЕНИЕ

В настоящее время устройства с интерфейсом Ethernet востребованы как никогда. Они позволяют осуществлять дистанционный контроль и управление удалёнными объектами по сети с

удобным и интуитивно понятным интерфейсом и решать множество новых задач, например, мониторинг объектов, голосовую связь (IP-телефония), автоматизацию зданий, медиасервис и т.п.

Используя технологии WiFi, можно подключить сетевые устройства к беспроводной точке доступа или мосту. Технологии Интернет позволяют осуществлять управление сетевыми устройствами практически в любой точке мира.

В настоящее время появилось множество интерфейсных микросхем Ethernet, которые позволяют создавать сетевые устройства на базе различных

Таблица 1. Описание функции SPI_Ethernet_Init

Прототип	void SPI_Ethernet_Init(unsigned char *mac, unsigned char *ip, unsigned char fullDuplex);
Возвращаемое значение	Отсутствует
Описание	MAC-модуль программы. Инициализирует контроллер ENC28J60. Эта функция внутри разделена на две части, чтобы помочь компоновщику, испытывающему недостаток в памяти. Установки контроллера ENC28J60 (параметры, не упоминающиеся здесь, установлены по умолчанию): • стартовый адрес буфера приёмника: 0x0000 • конечный адрес буфера приёмника: 0x19AD • стартовый адрес буфера передатчика: 0x19AE • конечный адрес буфера передатчика: 0x1FFF • RAM-буфер чтения/записи указателей в режиме автоприращения • фильтры приёмника установлены по умолчанию: CRC + MAC Unicast + MAC Broadcast • текущий контроль пауз в полном дуплексном режиме • фреймы заполнены до 60 байт + CRC • максимальный размер пакета установлен равным 1518 • Back-to-Back Inter-Packet Gap: 0x15 в полном дуплексном режиме; 0x12 в полудуплексном режиме • Non-Back-to-Back Inter-Packet Gap: 0x0012 в полном дуплексном режиме; 0x0C12 в полудуплексном режиме • окно коллизий установлено равным 63 в полудуплексном режиме, для адаптации некоторых особенностей ENC28J60 • вывод CLKOUT отключен для уменьшения EMI генерации • полудуплексная петля – перемычка отключена • конфигурация светодиодов: по умолчанию (LEDA – статус подключения, LEDB – активность связи) Параметры: • mac: буфер RAM содержит MAC-адрес • ip: буфер RAM содержит IP-адрес - fullDuplex: переключатель режимов Ethernet. Допустимые значения: 0 (полудуплексный режим) и 1 (полный дуплексный режим)
Требования	Глобальные переменные: • SPI_Ethernet_CS: вывод выборки микросхемы • SPI_Ethernet_CS_Direction: направление вывода выборки микросхемы • SPI_Ethernet_RST: вывод сброса • SPI_Ethernet_RST_Direction: направление вывода сброса Эти переменные должны быть определены перед использованием этой функции. Интерфейс SPI должен быть инициализирован. См. функцию SPI1_Init и SPI1_Init_Advanced
Пример использования функции	<pre>#define SPI_Ethernet_HALFDUPLEX 0 #define SPI_Ethernet_FULLDUPLEX 1 // Назначение выводов управления sfr sbit SPI_Ethernet_Rst at RC0_bit; sfr sbit SPI_Ethernet_CS at RC1_bit; sfr sbit SPI_Ethernet_Rst_Direction at TRISC0_bit; sfr sbit SPI_Ethernet_CS_Direction at TRISC1_bit; // конец назначения выводов управления unsigned char myMacAddr[6] = {0x00, 0x14, 0xA5, 0x76, 0x19, 0x3f}; // мой MAC адрес unsigned char myIpAddr = {192, 168, 1, 60}; // мой IP-адрес SPI1_Init(); SPI_Ethernet_Init(myMacAddr, myIpAddr, SPI_Ethernet_FULLDUPLEX);</pre>

Таблица 2. Описание функции SPI_Ethernet_doPacket

Прототип	unsigned char SPI_Ethernet_doPacket();
Возвращаемое значение	• 0 – успешная обработка пакета (получен нулевой пакет или полученный пакет успешно обработан) • 1 – ошибка приёма или буфер приёмника искажён. Контроллер ENC28J60 должен быть перезапущен • 2 – полученный пакет был адресован не нам (не «наш» IP и не широковещательный адрес IP) • 3 – полученный пакет IP не был IPv4 • 4 – полученный пакет был неизвестного типа для библиотеки
Описание	MAC-модуль программы. Обрабатывает очередной полученный пакет данных, если таковой существует. Пакеты обрабатываются следующим способом: • ARP- и ICMP-запросы получают ответ автоматически • на TCP-запросы вызывается функция SPI_Ethernet_UserTCP, предназначенная для обработки данного типа запроса • на UDP запросы вызывается функция SPI_Ethernet_UserUDP, предназначенная для обработки данного типа запроса Функция SPI_Ethernet_doPacket должна циклически вызываться так часто, насколько это возможно в программе
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	<pre>if (SPI_Ethernet_doPacket() == 0){ // обработка полученных пакетов ... }</pre>

Таблица 3. Описание функции SPI_Ethernet_UserUDP

Прототип	unsigned int SPI_Ethernet_UserUDP(unsigned char *remoteHost, unsigned int remotePort, unsigned int destPort, unsigned int reqLength);
Возвращаемое значение	• 0 – не требуется ответ в запросе • длина области данных ответа UDP, если требуется ответ
Описание	Программный модуль обслуживания UDP-запросов. Вызывается непосредственно библиотекой. Пользователь имеет доступ к UDP-запросам, используя некоторые программные функции SPI_Ethernet_get. Пользователь устанавливает данные в буфере передачи, используя некоторые программные функции SPI_Ethernet_put. Функция должна возвращать длину ответа UDP в байтах или 0, если ничего не требуется передавать. Если не нужно отвечать на запросы UDP, просто определите эту функцию с единственным оператором return(0). Параметры: • remoteHost: IP адрес клиента • remotePort: Порт клиента • destPort: Порт, на который послали запрос • reqLength: Область данных длины запроса UDP Функциональный исходный код приводится в примерах проектов. Код должен быть скорректирован пользователем для формирования необходимого ответа
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	Эта функция непосредственно вызывается библиотекой и не должна вызываться в программе пользователя

микроконтроллеров, однако все они различаются между собой по типу интерфейса с микроконтроллером и способу программирования.

Сдерживающим фактором создания сетевых Ethernet-устройств на базе микроконтроллеров было то, что интерфейс Ethernet является довольно сложным, а ИС контроллеров Ethernet обычно имеют большое количество выводов и ими трудно управлять с помощью микроконтроллера, имеющего небольшой объём оперативной памяти. Разработка программы для микроконтроллера, которая должна осущест-

влять инициализацию интерфейсной микросхемы Ethernet и поддерживать сетевые протоколы, является самым трудоёмким процессом при создании сетевого устройства.

К счастью, существует путь, который упрощает создание сетевого устройства и сводит к минимуму затраты на создание программы. Это стало возможным благодаря компании MicroChip [1], разработавшей микросхему Ethernet-контроллера ENC28J60, и компании MikroElektronika [2], создавшей среду разработки программ mikroC.

Микросхема Ethernet-контроллера ENC28J60 имеет стандартный интерфейс SPI, позволяющий подключить этот контроллер практически к любому микроконтроллеру, а среда разработки mikroC содержит готовые библиотечные функции для контроллера ENC28J60, позволяющие создавать разнообразные Ethernet-устройства. Рассмотрим подробнее эти две составляющие.

АППАРАТНАЯ ЧАСТЬ

Контроллер Ethernet типа ENC28J60 представляет собой микросхему, имеющую 28 выводов и включающую в

реклама

Таблица 4. Описание функции SPI_Ethernet_UserTCP

Прототип	unsigned int SPI_Ethernet_UserTCP(unsigned char *remoteHost, unsigned int remotePort, unsigned int localPort, unsigned int reqLength, char *canClose);
Возвращаемое значение	• 0 – не требуется ответ в запросе • длина области данных ответа TCP/HTTP, если требуется ответ
Описание	Программный модуль обслуживания TCP-запросов. Вызывается непосредственно библиотекой. Пользователь имеет доступ к TCP/HTTP-запросам, используя некоторые программные функции SPI_Ethernet_get. Пользователь устанавливает данные в буфере передачи, используя некоторые программные функции SPI_Ethernet_put. Функция должна возвращать длину ответа TCP/HTTP в байтах или 0, если ничего не требуется передавать. Если не нужно отвечать на запросы TCP/HTTP, просто определите эту функцию с единственным оператором return(0). Параметры: • remoteHost: IP-адрес клиента • remotePort: Порт TCP клиента • localPort: Порт на который послан запрос • destPort: Порт, на который послали запрос • reqLength: Область данных длины запроса TCP/HTTP • canClose: TCP-сокет закрыт Функциональный исходный код приводится в примерах проектов. Код должен быть скорректирован пользователем для формирования необходимого ответа
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	Эта функция непосредственно вызывается библиотекой и не должна вызываться в программе пользователя

Таблица 5. Описание функции SPI_Ethernet_putByte

Прототип	void SPI_Ethernet_putByte(unsigned char v);
Возвращаемое значение	Отсутствует
Описание	Это программный модуль MAC. Он сохраняет один байт по адресу указанному текущим указателем записи ENC28J60 (EWRPT). Параметры: • v: сохраняемый байт
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	<pre>char data; ... SPI_Ethernet_putByte(data); // поместить байт в буфер ENC28J60</pre>

Таблица 6. Описание функции SPI_Ethernet_putBytes

Прототип	void SPI_Ethernet_putBytes(unsigned char *ptr, unsigned int n);
Возвращаемое значение	Отсутствует
Описание	Программный модуль MAC. Сохраняет несколько байт в оперативной памяти ENC28J60, начиная с адреса, указанного текущим указателем записи ENC28J60 (EWRPT). Параметры: • ptr: буфер ОЗУ, содержащий байты, которые нужно записать в память ENC28J60 • n: количество байтов, которые нужно записать
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	<pre>char *buffer = "mikroElektronika"; ... SPI_Ethernet_putBytes(buffer, 16); // поместить массив из ОЗУ в буфер ENC28J60</pre>

Таблица 7. Описание функции SPI_Ethernet_putConstBytes

Прототип	void SPI_Ethernet_putConstBytes(const unsigned char *ptr, unsigned int n);
Возвращаемое значение	Отсутствует
Описание	Программный модуль MAC. Сохраняет несколько байт констант в оперативной памяти ENC28J60, начиная с адреса, указанного текущим указателем записи ENC28J60 (EWRPT). Параметры: • ptr: буфер, содержащий байты константы, которые нужно записать в память ENC28J60 • n: количество байтов, которые нужно записать
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	<pre>const char *buffer = "mikroElektronika"; ... SPI_Ethernet_putConstBytes(buffer, 16); // поместить массив констант в буфер ENC28J60</pre>

Таблица 8. Описание функции SPI_Ethernet_putString

Прототип	unsigned int SPI_Ethernet_putString(unsigned char *ptr);
Возвращаемое значение	Количество байтов, записанных в ОЗУ ENC28J60
Описание	Программный модуль MAC. Сохраняет целую строку (исключая завершающий ноль) в ОЗУ ENC28J60, начиная с текущего указателя записи ENC28J60 (EWRPT). Параметры: • ptr: строка, которая должна быть записана в ОЗУ ENC28J60
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	<pre>char *buffer = "mikroElektronika"; ... SPI_Ethernet_putString(buffer); // поместить строку в буфер ENC28J60</pre>

Таблица 9. Описание функции SPI_Ethernet_putConstString

Прототип	unsigned int SPI_Ethernet_putConstString(const unsigned char *ptr);
Возвращаемое значение	Количество байтов, записанных в ОЗУ ENC28J60
Описание	Программный модуль MAC. Сохраняет строковую константу (исключая завершающий ноль) в ОЗУ ENC28J60, начиная с текущего указателя записи ENC28J60 (EWRPT). Параметры: • ptr: строковая константа, записываемая в ОЗУ ENC28J60
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	const char *buffer = "mikroElektronika"; ... SPI_Ethernet_putConstString(buffer); // поместить строковую константу в буфер ENC28J60

Таблица 10. Описание функции SPI_Ethernet_getByte

Прототип	unsigned char SPI_Ethernet_getByte();
Возвращаемое значение	Байт, прочитанный из ОЗУ ENC28J60
Описание	Программный модуль MAC. Считывает байт по адресу указанному текущим указателем чтения ENC28J60 (ERDPT)
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	char buffer; ... buffer = SPI_Ethernet_getByte(); // прочитайте байт из буфера ENC28J60

Таблица 11. Описание функции SPI_Ethernet_getBytes

Прототип	void SPI_Ethernet_getBytes(unsigned char *ptr, unsigned int addr, unsigned int n);
Возвращаемое значение	Отсутствует
Описание	Программный модуль MAC. Считывает несколько байт из оперативной памяти RAM ENC28J60, начиная с заданного адреса. Если величина 0xFFFF задана как параметр адреса, чтение начнётся с текущего значения указателя чтения ENC28J60 (ERDPT). Параметры: • ptr: буфер для хранения байтов прочитанных из ОЗУ ENC28J60. • addr: стартовый адрес ОЗУ ENC28J60. Корректное значение адреса от 0 до 8192. • n: количество байтов, которые нужно прочитать
Требования	Контроллер Ethernet должен быть инициализирован. См. функцию SPI_Ethernet_Init
Пример использования функции	char buffer[16]; ... SPI_Ethernet_getBytes(buffer, 0x100, 16); // читать 16 байтов, начинающихся с адреса 0x100

себя буфер приёмопередатчика, MAC-адрес и интерфейс физического уровня РНУ, необходимые для поддерж-

ки интерфейса Ethernet. Микросхема ENC28J60 является автономным Ethernet-контроллером, полностью соответ-

ствующим спецификации IEEE 802.3 и поддерживающим обмен данными в сетях Ethernet типа 10BASE-T по витой

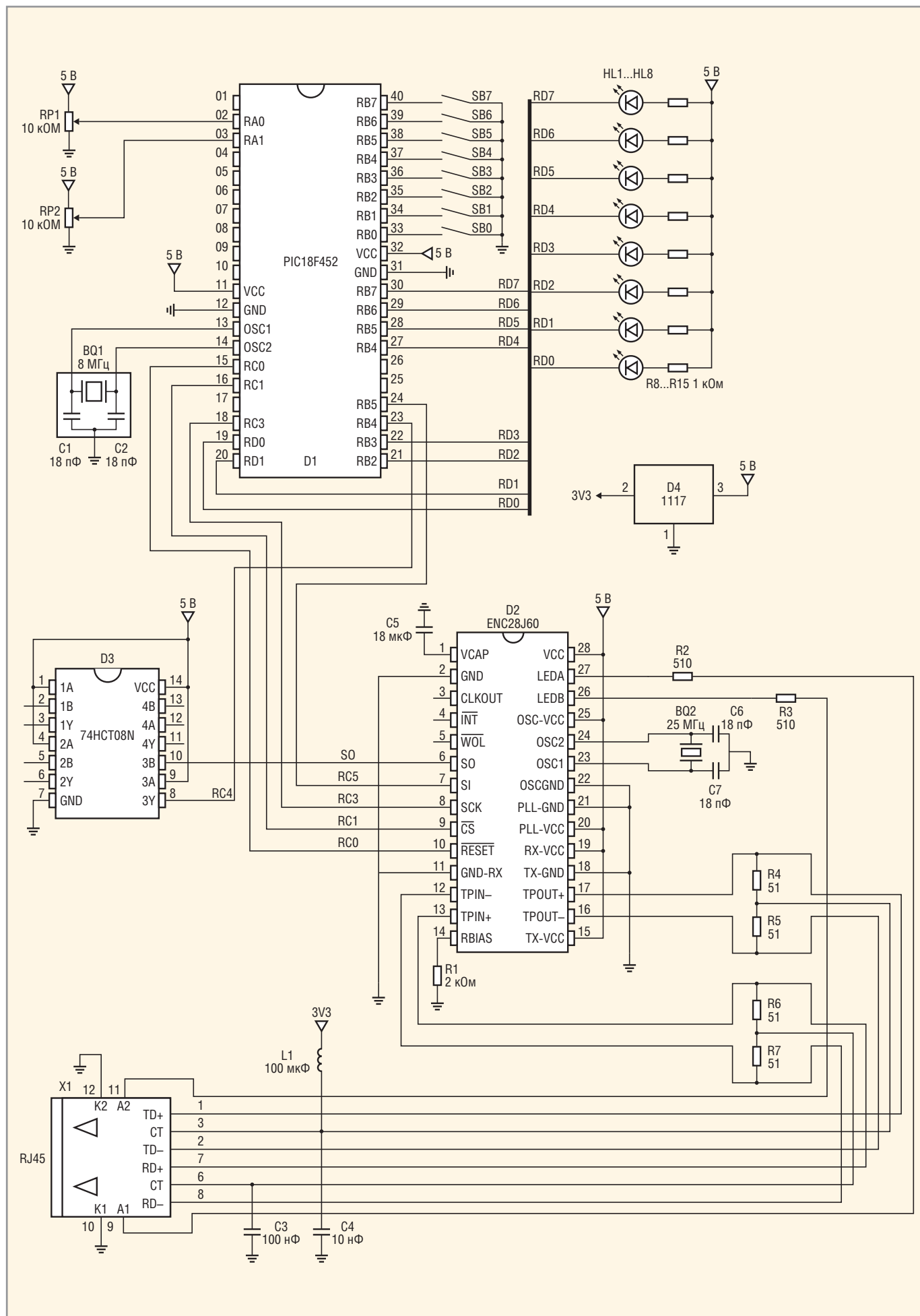


Рис. 1. Схема сетевого устройства на базе микроконтроллера PIC

паре. Ниже приведены основные характеристики Ethernet-контроллера ENC28J60:

- поддержка дуплексного и полудуплексного режима;
- программируемый повтор передачи при возникновении коллизии;
- автоматическая генерация контрольной суммы;
- автоматическое отбрасывание ошибочных пакетов;
- семь источников прерываний, объединённых в два выходных сигнала;
- совместимость со стандартом IEEE 802.3;
- поддержка уровней MAC и PHY;
- поддержка порта 10BASE-T с автоматическим определением полярности;
- скорость обмена данными по интерфейсу SPI до 10 Мбит/с;
- тактовая частота 25 МГц;
- вывод тактовой частоты с предварительным внутренним делением;
- напряжение питания от 3,14 до 3,45 В;
- TTL-толерантные входные сигналы;
- промышленный и коммерческий температурные диапазоны работы;
- два программируемых светодиода для отображения режима работы;
- корпус на 28 выводов типа SPDIP, SSOP, SOIC и QFN.

Большинство функциональных блоков в микросхеме реализовано аппаратно. К ним относится фильтр входящих пакетов, вычислитель контрольной суммы, внутренний канал DMA и т.д. Более детально с микросхемой Ethernet-контроллера ENC28J60 можно ознакомиться в [3].

ПРОГРАММНАЯ ЧАСТЬ

Среда разработки mikroC была подробно рассмотрена в [4]. Она включает в себя библиотеку для программной поддержки Ethernet-контроллера ENC28J60 и совместима с любым микроконтроллером PIC, имеющим встроенный интерфейс SPI и не менее 4 Кб постоянной памяти. Рекомендуется настроить тактовую частоту для интерфейса SPI от 8 до 10 МГц. Если использовать более низкую скорость тактирования, могут быть пропущены некоторые запросы.

Библиотека SPI Ethernet поддерживает следующие протоколы и запросы:

- протокол IPv4;
- запросы ARP;
- запросы эхо ICMP;
- запросы UDP;
- запросы TCP (без стека и без пакетной реконструкции);

- клиент ARP с кэшированием;
- клиент DNS;
- клиент UDP;
- клиент DHCP.

Пакетная фрагментация библиотекой не поддерживается.

Библиотека включает в себя следующие функции для семейств микроконтроллеров PIC16 и PIC18:

- SPI_Ethernet_Init;
- SPI_Ethernet_Enable;
- SPI_Ethernet_Disable;
- SPI_Ethernet_doPacket;
- SPI_Ethernet_putByte;

- SPI_Ethernet_putBytes;
 - SPI_Ethernet_putString;
 - SPI_Ethernet_putConstString;
 - SPI_Ethernet_putConstBytes;
 - SPI_Ethernet_getByte;
 - SPI_Ethernet_getBytes;
 - SPI_Ethernet_UserTCP;
 - SPI_Ethernet_UserUDP;
- и функции только для семейства PIC18:
- SPI_Ethernet_getIpAddress;
 - SPI_Ethernet_getGwIpAddress;
 - SPI_Ethernet_getDnsIpAddress;
 - SPI_Ethernet_getIpMask;

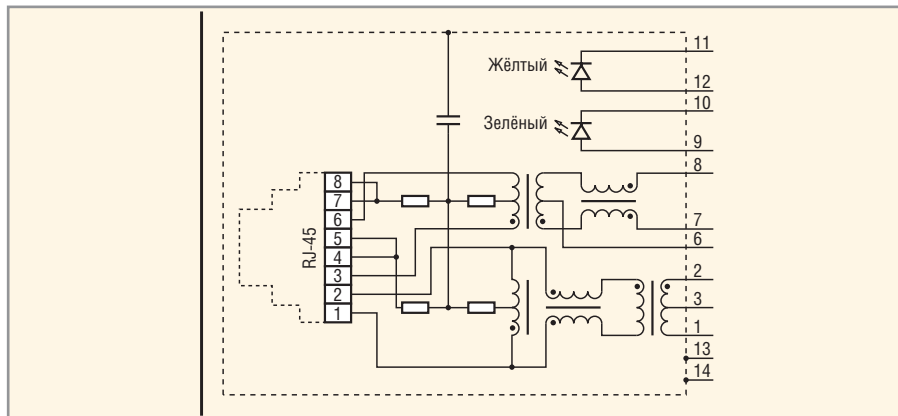


Рис. 2. Схема согласующего модуля Ethernet для витой пары

PIC + ENC28J60 Mini Web Server

[Reload](#)

ADC		PORTB		PORTD	
AN2	688	BUTTON #0	ON	LED #0	OFF Toggle
AN3	712	BUTTON #1	OFF	LED #1	ON Toggle
		BUTTON #2	ON	LED #2	OFF Toggle
		BUTTON #3	OFF	LED #3	ON Toggle
		BUTTON #4	ON	LED #4	OFF Toggle
		BUTTON #5	OFF	LED #5	ON Toggle
		BUTTON #6	ON	LED #6	OFF Toggle
		BUTTON #7	OFF	LED #7	ON Toggle

This is HTTP request # 1

Рис. 3. Основная интернет-страница микроконтроллера

- SPI_Ethernet_confNetwork;
- SPI_Ethernet_arpResolve;
- SPI_Ethernet_sendUDP;
- SPI_Ethernet_dnsResolve;
- SPI_Ethernet_initDHCP;
- SPI_Ethernet_doDHCPLeaseTime;
- SPI_Ethernet_renewDHCP.

Функции, необходимые для создания сетевого устройства с поддержкой протоколов UDP и TCP/IP, описаны в таблицах 1 – 11 соответственно.

РЕАЛИЗАЦИЯ

Рассмотрим реализацию Ethernet-устройств на базе микроконтроллеров PIC, AVR и 8051. Сетевое устройство Ethernet в каждом случае будет представлять собой мини-сервер, содержащий HTML-страницу и способный обрабатывать сетевые запросы.

Схема сетевого устройства на базе микроконтроллера PIC представлена на рисунке 1. На ней показаны все элементы и цепи, необходимые для правильного функционирования устройства.

Как видно из схемы, связь сетевого устройства с сетью Ethernet осуществляется через согласующий элемент X1, который представляет собой небольшой модуль, объединяющий восьмиконтактный соединитель RJ45, развязывающие трансформаторы, фильтрующие дроссели, нагрузочные сопротивления, развязывающий конденсатор и два светодиода. На рисунке 2 представлена схема модуля.

Такие согласующие модули применяются практически во всех современных компьютерах. Они существенно экономят место на печатной плате сетевого устройства благодаря своей компактности. Два светодиода, входящие в их состав, обеспечивают индикацию несущей и активности обмена по сети.

Функционирование мини-сервера, содержащего HTML-страницу, возлагается на программу, которая будет храниться и выполняться в микроконтроллере PIC. Исходный код этой программы приведен в листинге, раз-

мещённом в дополнительных материалах к статье на сайте журнала. Программа транслируется с помощью среды разработки mikroC PRO for PIC; полученный в результате трансляции HEX-файл прошивается в микроконтроллер PIC18F452.

Пример программы демонстрирует использование библиотечных функций Ethernet. Сетевое устройство с помощью данной программы выполняет следующие функции:

- отвечает на запросы ARP и ICMP;
- отвечает на запросы UDP по любому порту и возвращает адрес IP-клиента с номером порта, а также принятую строку символов, преобразованных в верхний регистр;
- отвечает на запросы HTTP по порту 80, обслуживая метод GET с параметрами: / – возвращает основную страницу с тегами HTML; /s – возвращает статус устройства в виде текстовой строки; /t0 – /t7 – переключает разряд 0 – 7 порта D.

Для всех других запросов – возвращает основную страницу с тегами HTML.

Главный модуль программы осуществляет циклический вызов функции SPI_Ethernet_doPacket, которая обеспечивает обработку всех запросов по сети Ethernet. Из кода программы (см. листинг) видно, что гипертекстовая страница, состоящая из тегов HTML, содержится в статических строковых переменных indexPage и indexPage2 и передаётся источнику запросов с помощью функции обработки TCP-запросов SPI_Ethernet_UserTCP.

При обращении к данному устройству, подключенному к сети Ethernet, по адресу 192.168.20.60 с помощью любого интернет-браузера в ответ будет получена основная интернет-страница с тегами HTML, внешний вид которой представлен на рисунке 3.

Эта страница HTML отображает состояние двух каналов АЦП, положение кнопок, считываемых портом PORTB, и состояние индикаторов, управляемых через порт PORTD. Обновление страницы происходит каждые три секунды. Изменяя уровни сигналов на входах АЦП и нажимая кнопки, подключенные к порту PORTB микроконтроллера PIC, можно видеть изменение их состояния на странице. Для включения или отключения светодиодов, подключенных к порту PORTD микроконтроллера,

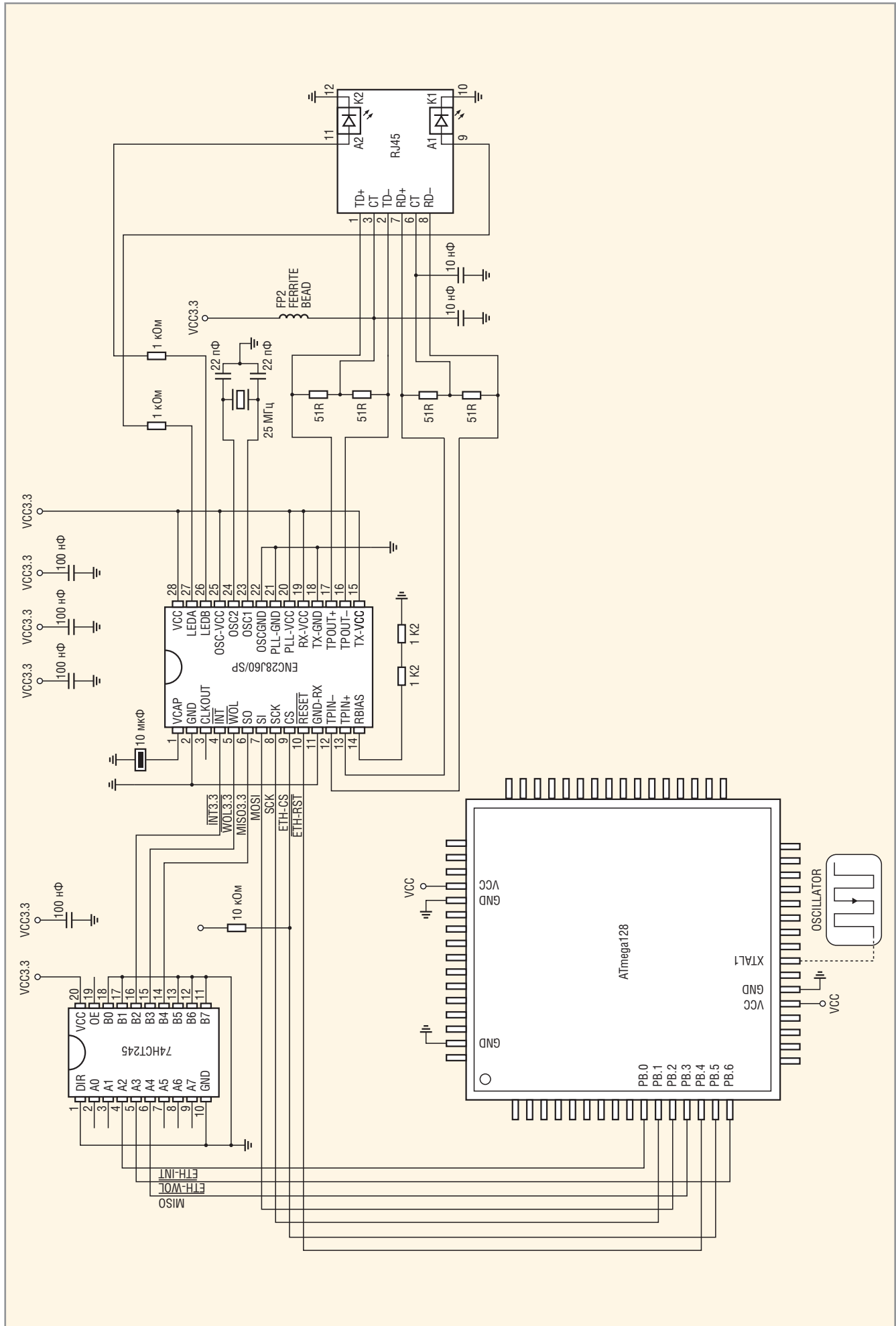


Рис. 4. Схема сетевого устройства на базе микроконтроллера AVR

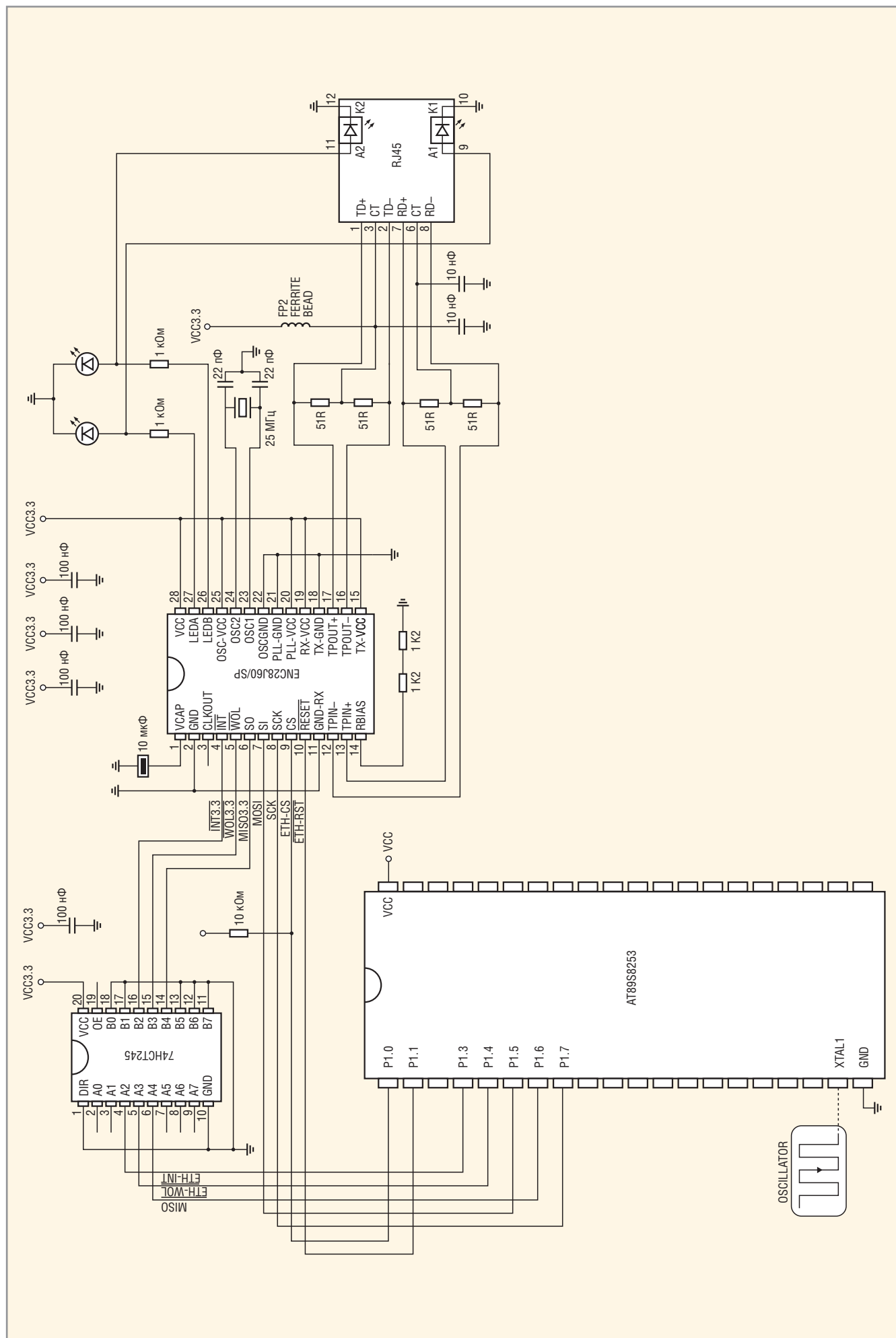


Рис. 5. Схема сетевого устройства на базе микроконтроллера 18051

лера PIC, необходимо с помощью мышки щёлкнуть по соответствующему элементу Toggle (Переключить) на странице. Таким образом, можно осуществлять контроль и управление элементами, подключенными к соответствующим портам микроконтроллера PIC. Обновление страницы можно выполнять принудительно с помощью кнопки Reload. Под таблицей выводится статистика запросов, поступивших на мини-сервер.

Тестирование обработки запросов UDP сетевым устройством можно производить из самой среды разработки mikroC с помощью встроенного терминала UDP. Введите произвольные символы в поле передачи и отправьте их сетевому устройству, в ответ вы должны получить те же символы с изменённым регистром, т.е. вместо строчных символов – прописные, и наоборот.

Проверку работоспособности сетевого устройства можно выполнить с помощью утилиты ping.exe, входящей в состав операционной системы Windows. В нашем случае достаточно ввести в командной строке и выполнить команду ping 192.168.20.60. В ответ на

эту команду операционная система выдаст сообщение о количестве отправленных, полученных и потерянных пакетов. Если сетевое устройство исправно, потерь быть не должно.

Адрес 192.168.20.60 присвоен в программе переменной myIpAddress и при необходимости может быть изменён, с последующей перетрансляцией программы и перепрограммированием микроконтроллера. Кроме смены адреса, в программу можно внести и другие необходимые изменения. Например, можно расширить список выполняемых команд или дополнить таблицу получаемых от контроллера данных, т.е. можно запрограммировать полученное сетевое устройство на выполнение любых действий и операций.

Схемы аналогичных сетевых устройств на базе микроконтроллера AVR и 8051 представлены на рисунках 4 и 5 соответственно. Проекты с исходными кодами программ и готовые прошивки в виде HEX-файлов для этих устройств можно загрузить с интернет-страницы журнала. Состав и назначение этих программ очень схожи с описанной выше программой для

микроконтроллера PIC, а используемые в программах библиотечные функции для Ethernet имеют такие же названия.

Естественно, что транслировать программы для сетевых устройств на основе данных микроконтроллеров необходимо в среде разработки, соответствующей микроконтроллеру. Для микроконтроллеров AVR подходит среда разработки mikroC PRO for AVR, а для микроконтроллеров серии 8051 – среда mikroC PRO for 8051.

ЗАКЛЮЧЕНИЕ

В последнее время появилось множество новых микроконтроллеров, интегрированных с интерфейсом Ethernet, что позволяет быстро создавать «умные» и компактные сетевые устройства.

ЛИТЕРАТУРА

1. <http://www.mikroe.com>.
2. <http://www.microchip.com>.
3. <http://www.microchip.com/downloads/en/DeviceDoc/39662b.pdf>.
4. *Вальна О.* Современная среда разработки mikroC для программирования микроконтроллеров на языке высокого уровня Си. Современная электроника. 2010. № 6. С. 64. ©