

ОГЛАВЛЕНИЕ

Предисловие	7
Беглый обзор микроконтроллеров	8
Микроконтроллеры AVR производства Atmel	8
Микроконтроллеры PIC производства Microchip	11
Глава 1. Протокол RS-232 — основа сетей на базе микроконтроллеров	15
Немного истории	17
Стандартный способ использования RS-232	20
Преобразование напряжений RS-232	23
Глава 2. Реализация протокола RS-232 с использованием микроконтроллера	25
Базовое аппаратное обеспечение RS-232	25
Построение простого микроконтроллерного трансивера RS-232	28
Аппаратное обеспечение интерфейса RS-232	29
DCE на базе микроконтроллера	30
Набор PICkit 1 FLASH Starter Kit производства Microchip	30
Простая реализация микропрограммного обеспечения RS-232	35
Подпрограмма передачи данных RS-232	43
Подпрограмма приема данных RS-232	49
Глава 3. Реализация функций интерфейса RS-232 на языке BASIC	53
BASIC RS-232	53
Глава 4. Формирование аппаратного обеспечения RS-232	59
Дополнительный программный модуль BASIC RS-232	59
Глава 5. Универсальный синхронно-асинхронный приемопередатчик	63
Регистры прерываний USART	67
Реализация протокола RS-232 с использованием Atmel AVR	85
Программирование AVR RS-232	89
Глава 6. Шина I²C... Еще один последовательный протокол	97
Достоинства шины I ² C	99
Шина I ² C	100
Биты ACK и NAK шины I ² C	103
Арбитраж и тактовая синхронизация	104
Адресация шины I ² C	108
Микропрограммное обеспечение шины I ² C	108
Программный код ведущего устройства AVR шины I ² C	110
Программный код ведущего приемника AVR I ² C	115
Программный код ведомого передатчика PIC	117
Микропрограммное обеспечение обмена данными между AVR и PIC по шине I ² C	123
Глава 7. Ethernet	137
Протокол Ethernet	137
Интегральная схема CS8900A-CQ	138
Краткая информация о процедуре сброса CS8900A-CQ	140
Обзор интерфейсов передачи данных CS8900A-CQ	140
Краткая информация о процессе передачи данных микросхемой CS8900A-CQ	140
Краткая информация о процессе приема данных микросхемой CS8900A-CQ	141
Краткая информация о внешней памяти CS8900A-CQ	142

Индикаторы состояний CS8900A-CQ	142
Процессор MAC микросхемы CS8900A-CQ	143
Аппаратное обеспечение Easy Ethernet CS8900A	147
Микроконтроллер PIC16F877	147
Микроконтроллер Microchip PIC18F452	148
Процессор Ethernet микросхемы CS8900A-CQ	148
Питание микросхемы CS8900A-CQ	149
Блок трансформаторов NU1S114-XXX	150
Аппаратное обеспечение Easy Ethernet CS8900A на базе PIC16F877	151
Встроенный интерфейс последовательного программирования ICSP	154
Разработка микропрограммного обеспечения Easy Ethernet CS8900A	157
Настройка микроконтроллера PIC16F877	159
Распределение ресурсов памяти микроконтроллера PIC16F877	162
Прототипы функций	162
Определение переменных	163
Макросы Easy Ethernet CS8900A	170
Набор регистров CS8900A-CQ PacketPage	176
Регистры интерфейса шины CS8900A-CQ	178
Регистр идентификатора продукта — Product Identification Code	178
Регистры состояний и управления CS8900A-CQ	179
Использование регистров	192
Глава 8. Микропрограммное обеспечение CS8900A-CQ	193
Первый шаг	194
Перегрузка CS8900A-CQ	195
Загрузка основных параметров CS8900A-CQ	196
Загрузка индивидуального адреса CS8900A-CQ	198
Включение приемника и передатчика CS8900A-CQ	199
Главный рабочий цикл	201
Кадр под микроскопом	203
ARP-запрос	210
Глава 9. Команда PING и Easy Ethernet CS8900A	225
Глава 10. Протокол UDP и Easy Ethernet CS8900A	243
Приложение UDP Internet Test Panel	245
Глава 11. Протокол TCP и Easy Ethernet CS8900A	261
Физический уровень	263
Канальный уровень	264
Сетевой уровень	264
Транспортный уровень	264
Прикладной уровень	265
Кодирование протокола TCP/IP для устройства Easy Ethernet CS8900A	266
Глава 12. Давайте выполним это вновь	312
Проект Easy Ethernet Whacked	312
Микросхема RTL8019AS	313
Аппаратное обеспечение Easy Ethernet W	322
Микропрограммное обеспечение Easy Ethernet W	324
Инициализация RTL8019AS	326
Устройство Easy Ethernet W в оперативном режиме	344
Передача кадра с использованием устройства Easy Ethernet W	347
Инструменты для дела и игры	351
Глава 13. Применение Easy Ethernet AVR в диалоговом режиме	355
Глава 14. Заключение	365
Приобретение устройств Easy Ethernet	365
Предметный указатель	366

Предисловие

Существует множество философских проблем, которым можно было бы уделить внимание в данной книге. Но автор не претендует на лавры философа или поэта. Все его время посвящено разработке аппаратного обеспечения на базе микроконтроллеров, написанию программного кода, управляющего этим обеспечением, с последующим описанием увлекательных приключений на этом поприще.

Данная книга есть не что иное, как обычный технический документ, содержащий информацию, необходимую для создания успешно работающих сетевых устройств на базе микроконтроллеров. Прежде чем вы перевернете последнюю страницу этой книги, вы узнаете все об интеграции RS-232, I²C и Ethernet в сетевое устройство, предназначенное для организации связи по локальным (LAN) или глобальным (WAN) сетям, или Интернету. Кроме того, к моменту расставания с этой книгой вы будете иметь полное представление о том, как программируются такие устройства.

Сетевые устройства будут создаваться на базе микроконтроллеров Atmel и Microchip. Чтобы обеспечить совместимость, при программировании микроконтроллеров Atmel будет использоваться компилятор ICCAVR Pro С компании ImageCraft, а при программировании микроконтроллеров PIC — компилятор CCS PIC С компании Custom Computer Service. Причем оба компилятора С имеют вполне умеренную стоимость и легко доступны через Интернет. Более того, при необходимости программный код любого проекта, представленного в данной книге, может быть легко перенесен в другие версии языка С.

«Сетевые приключения» начнутся с протокола RS-232, и все полученные сведения о протоколе RS-232 будут в дальнейшем активно использоваться. Другими словами, именно на базе этих сведений будут реализованы шина I²C и интерфейс Ethernet. При этом ни один бит не останется без внимания. Помощь при работе, несомненно, окажут материалы, расположенные на прилагаемом компакт-диске. Кроме того, существует веб-сайт технической поддержки (<http://www.edtp.com>), с помощью которого можно не только получить любую консультацию, но и приобрести компоненты, комплекты и собранные устройства, обсуждаемые в данной книге.

Оба производителя микроконтроллеров, и Atmel, и Microchip, предоставляют специализированные интегрированные средства разработки IDE, которые можно бесплатно загрузить с веб-сайтов Atmel и Microchip. При работе с микроконтроллерами Atmel будет использоваться AVR Studio, при работе с микроконтроллерами Microchip — MPLAB. Дополнительно будут использоваться Microchip MPLAB ICE 2000, Microchip MPLAB ICD 2 и Atmel AVR JTAG ICE. А для демонстрации содержимого передаваемых по сети Ethernet-пакетов будет использоваться Network Associates Sniffer.

Ну, а теперь, когда вы знаете, о чем пойдет речь, можно приступить к созданию сетевых устройств на базе микроконтроллеров.

Беглый обзор микроконтроллеров

Микроконтроллеры AVR производства Atmel

Данные микроконтроллеры обладают уникальными возможностями и высокой работоспособностью в сети. Существует несколько семейств микроконтроллеров AVR, а именно: семейство стандартных микроконтроллеров AVR, маломощных микроконтроллеров AVR, микроконтроллеров tinyAVR и микроконтроллеров ATmega AVR. По ряду причин автор отдает предпочтение семейству ATmega AVR, тем более что многие ранние модели AVR заменяются более быстрыми и более мощными моделями ATmega AVR. Так, например, микроконтроллер ATmega16 заменил ATmega163, а ATmega32 заменил ATmega323. В дополнение к новым возможностям, в новых микроконтроллерах устранены ошибки, обнаруженные в заменяемых моделях.

Данная книга не содержит информации о внутренних различиях между AVR и аналогичными микроконтроллерами. Эта информация представлена в спецификациях. Тем не менее, автор считает своим долгом пояснить, почему при создании сетевых устройств он использует именно эти микроконтроллеры. Во-первых, микроконтроллеры семейства ATmega AVR имеют максимальную тактовую частоту 16 МГц. Конечно, ее нельзя назвать высокой, но микроконтроллеры ATmega AVR выполняют большинство инструкций за один такт и способны достигать скорости 16 MIPS (16 миллионов команд в секунду) на частоте 16 МГц.

Как правило, число в названии микроконтроллера ATmega AVR обозначает объем программной флэш-памяти в килобайтах. Так, микроконтроллер ATmega16 содержит 16 Кбайт флэш-памяти, микроконтроллер ATmega32 — 32 Кбайта флэш-памяти, а самая старшая модель микроконтроллеров ATmega AVR, микроконтроллер ATmega128 — 128 Кбайт программной флэш-памяти. Программная флэш-память дополняется памятью SRAM, объем которой определяется объемом программной памяти. Микроконтроллер ATmega16 содержит 1 Кбайт памяти SRAM, микроконтроллер ATmega32 — 2 Кбайт памяти SRAM. Даже микроконтроллер ATmega8, самая младшая модель в семействе ATmega AVR, содержит 1 Кбайт памяти SRAM. Нетрудно догадаться, что микроконтроллеры ATmega64, ATmega128 и ATmega8 содержат соответственно 64 Кбайта, 128 Кбайт и 8 Кбайт флэш-памяти.

Итак, высокое быстродействие и большой объем программной флэш-памяти в сочетании с достаточным объемом SRAM и EEPROM памяти — вот те причины, по которым именно контроллеры AVR семейства ATmega были выбраны автором для реализации сети и сетевого взаимодействия.

Многие дистрибьюторы электронных компонентов поставляют микроконтроллеры AVR производства Atmel по почте. В поставку входит полный набор эле-

ментов, необходимых для организации сети и при этом микроконтроллеры AVR обладают умеренной ценой. Основные особенности микроконтроллеров AVR:

- два 8-битных таймера и один 16-битный таймер позволяют формировать точные задержки;
- встроенный универсальный синхронно-асинхронный приемопередатчик USART выполняет все служебные операции, необходимые для реализации последовательного протокола RS-232;
- двадцать один вектор прерываний охватывает все сетевые компоненты AVR, включая двухпроводной интерфейс (название интерфейса I²C, принятое в Atmel), подсистему SPI и USART.

Универсальный синхронно-асинхронный приемопередатчик AVR

Универсальный синхронно-асинхронный приемопередатчик ATmega AVR USART поддерживает работу в полнодуплексном режиме. Подобно большинству других существующих приемопередатчиков USART, ATmega AVR USART поддерживает 5, 6, 7, 8 или 9 бит данных, плюс 1 или 2 стандартных стоповых бита. Контроллер скорости передачи данных USART микроконтроллера Atmel ATmega AVR является неотъемлемой частью аппаратного обеспечения USART. Типичный приемопередатчик USART компании Atmel схематически представлен на **Рис. 1**.

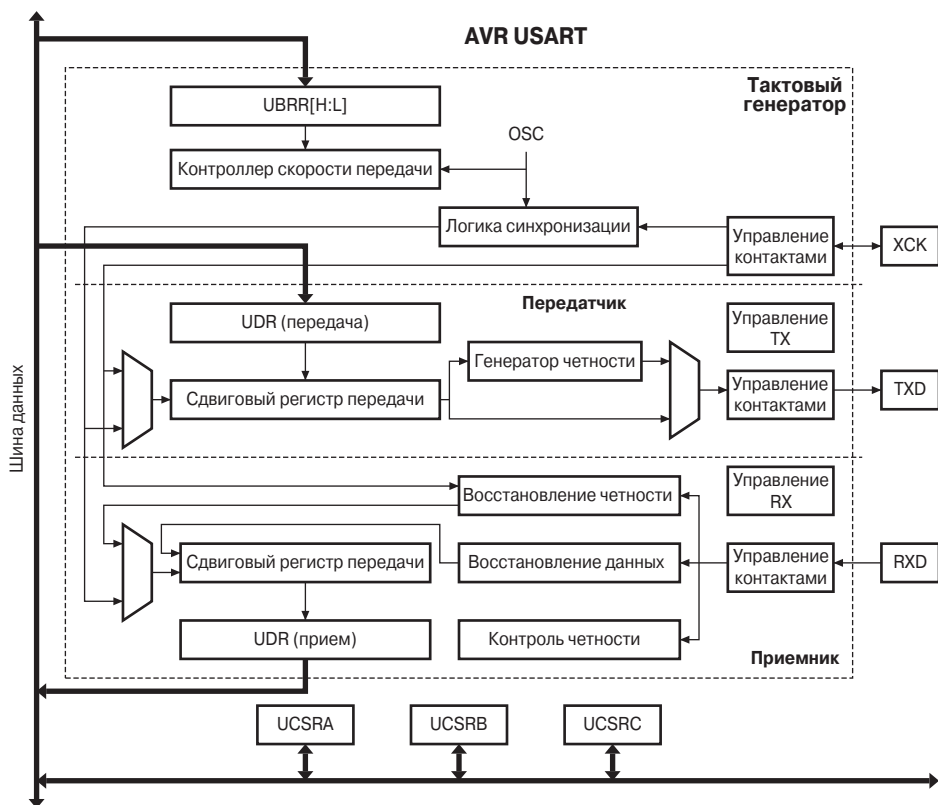


Рис. 1. Схема типичного универсального синхронно-асинхронного приемопередатчика компании Atmel.

Все модели микроконтроллеров ATmega AVR содержат приемопередатчик USART, микроконтроллер ATmega128 содержит два приемопередатчика USART.

Универсальные синхронно-асинхронные приемопередатчики Atmel USART позволяют микроконтроллерам ATmega AVR работать в режиме MPCM (Multi-processor Communication Mode — многопроцессорный режим связи). Данный режим использует адресацию для связывания нескольких процессоров с использованием одной и той же последовательной шины. Режим MPCM использует 9-символьный формат кадра и парадигму ведущий/ведомый.

Двухпроводной последовательный интерфейс

Фирма Atmel интерфейс I²C обозначает как TWI (Two-wire Interface — двухпроводной интерфейс). Отличаясь только названием, интерфейс TWI выглядит и работает аналогично интерфейсу I²C. Стандартная 7-битная адресная схема I²C позволяет подключать к двухпроводной шине и адресовать до 128 устройств. Работа в ведущем и в ведомом режимах поддерживается на скоростях до 400 кГц. Для облегчения борьбы с ложными срабатываниями из-за шумов, TWI-модуль Atmel имеет схему подавления шумов. TWI может также использоваться для активизации AVR.

Программирование ATmega AVR

Загрузка кода в микроконтроллер ATmega AVR не является сложной задачей. Существует много способов ее решения. Так, для загрузки может использоваться модуль программирования AVR ISP (In-System Programmer — программатор системы), подключаемый к последовательному порту персонального компьютера и стоящий менее 40\$. Или программирование микроконтроллера AVR может выполняться с использованием макетной платы STK500. Микроконтроллеры ATmega AVR, содержащие 16 Кбайт и более программной памяти, также поддерживают интерфейс JTAG, который может использоваться для программирования программной флэш-памяти ATmega AVR. Вне зависимости от выбранного способа загрузки кода в ATmega AVR, AVR Studio поддерживает все перечисленные устройства. AVR Studio производства Atmel представляет собой интерфейсное программное обеспечение IDE, выполняемое на персональном компьютере.

Интегрированная среда разработки AVR Studio используется автором данной книги вместе с компилятором ICCAVR Pro C производства ImageCraft. Компилятор ICCAVR Pro C представляет собой компилятор языка программирования C для микроконтроллеров ATmega AVR, полностью поддерживающий стандарт ANSI. Генератор кодов и поддержка вычислительных возможностей микроконтроллеров AVR — вот те причины, по которым автор выбрал компилятор ICCAVR Pro C.

Эмуляция ATmega AVR

Эмуляция ATmega AVR выполняется с использованием интегрированной среды разработки AVR Studio. AVR Studio предоставляет интерфейс ко многим устройствам эмуляции AVR. В данной книге в качестве устройства эмуляции

используется внутрисхемный эмулятор AVR JTAG ICE. Внутрисхемный эмулятор AVR JTAG ICE подключается к встроенному модулю отладки микроконтроллера AVR. Модуль OCD (On-Chip Debugger — встроенный отладчик) микроконтроллеров ATmega AVR устраняет необходимость в использовании специального эмулирующего устройства.

Микроконтроллеры PIC производства Microchip

Большинство микроконтроллеров PIC содержат все необходимое для создания сетевых устройств. Старшие модели микроконтроллеров PIC содержат аппаратно реализованные модули UARTS и USARTS, реализующие синхронную и асинхронную передачу данных в рамках протокола RS-232. Для младших моделей микроконтроллеров PIC, не содержащих аппаратных реализаций модулей UART или USART, модуль UART может быть реализован программно.

При работе в сети синхронизация имеет первостепенное значение. Старшие модели микроконтроллеров PIC могут содержать до трех встроенных таймеров. Даже компактная 8-выводная модель микроконтроллера PIC12F675 содержит 8-битный и 16-битный таймеры. Таймеры могут использоваться для генерации миллисекундных или микросекундных задержек, или для отсчета времени суток.

К счастью, компилятор CCS C и реализованные в компиляторе процедуры, специфические для микроконтроллеров PIC, значительно облегчают создание работающего приложения RS-232 PIC. Кроме этого, компилятор CCS C содержит средства поддержки интерфейса I²C. Микроконтроллеры семейства Microchip PIC содержат также значительный объем флэш-памяти для размещения программного кода, сверхоперативной памяти SRAM и хранилища данных пользователя. Большой объем памяти SRAM приобретает особое значение при создании буферных областей для сетевых приложений, управляемых прерываниями.

Проектирование и изготовление аппаратного и программного обеспечения будет выполняться с использованием микроконтроллеров PIC производства Microchip, содержащих флэш-память. Микроконтроллеры, содержащие флэш-память, выбраны автором ввиду их невысокой стоимости и легкой доступности. Кроме этого, для реализации всех стандартных операций над этими микроконтроллерами не требуется дополнительное оборудование. Так, наличие флэш-памяти позволяет отказаться от устройства ультрафиолетового стирания памяти EPROM. Поскольку флэш-компоненты могут программироваться и перепрограммироваться с использованием интерфейса ICSP (In-Circuit Serial Programming — встроенный последовательный интерфейс программирования), для разработки требуется меньшее количество компонентов, так как в процессе отладки программного кода отсутствует необходимость помещать компоненты в устройство ультрафиолетового стирания.

Для управления работой сетевого устройства выбран микроконтроллер PIC16F877 — старшая модель семейства PIC16F87X. Микроконтроллер PIC16F877 может работать на частоте 20 МГц, с командным циклом 200 наносекунд. Микроконтроллер содержит 8К слов программной флэш-памяти и 368 байт памяти SRAM или памяти данных. В случае необходимости для хранения конс-

тант или другой информации, которая должна быть сохранена даже при отключении питания устройства, может использоваться блок энергонезависимой памяти EEPROM объемом 256 байт. Поскольку нашей целью является реализация последовательного порта RS-232 с использованием микроконтроллера PIC, вы убедитесь, насколько важны прерывания при работе микроконтроллеров PIC. Микроконтроллер PIC16F877 может прерываться 15 различными способами.

В сетевых приложениях также очень важны линии ввода/вывода. Линии ввода/вывода требуются не только для выполнения сетевых задач, но и для мониторинга напряжения, подключения и отключения внешних устройств и т. д. Например, простое приложение драйвера микроконтроллера устройства Ethernet использует, по меньшей мере, 16 линий ввода/вывода. Микроконтроллер PIC16F877 содержит 33 линии ввода/вывода, часть из которых может использоваться для контроля — операции, которая выполняется микроконтроллером лучше всего.

Микроконтроллер PIC16F877 обладает развитыми функциональными возможностями, позволяющими использовать его вне задач организации сетей. Однако автор, прежде всего, заинтересован предоставить вам информацию об использовании PIC16F877 при создании сетевого устройства.

Итак, приступим к изучению одного из самых любимых автором данной книги сетевого модуля, модуля PIC16F877 USART.

Модуль PIC16F877 USART

USART — это сокращение Universal Synchronous/Asynchronous Receiver/Transmitter (универсальный синхронно-асинхронный приемопередатчик). В микроконтроллерах PIC модуль USART часто называется SCI или Serial Communications Interface (последовательный связной интерфейс). Вы, вероятно, слышали термин UART (Universal Asynchronous Receiver/Transmitter — универсальный асинхронный приемопередатчик) на протяжении многих лет, в течение которых в последовательных портах персональных компьютеров использовались только интегральные схемы. И сегодня некоторые современные микроконтроллеры используют модули UART вместо USART.

Модуль PIC16F877 USART позволяет избежать многих проблем при организации связи с другими последовательными устройствами. Так, скорость передачи данных модулем PIC16F877 USART регулируется встроенным контроллером, что позволяет отказаться от написания специальной процедуры. Модуль USART, как и модуль UART, не требует написания процедур, выполняющих поиск стартовых битов в поступающих данных или определяющих интервалы времени между битами поступающих данных. Вся работа выполняется модулем USART. Модуль USART позволяет связываться с другими последовательными устройствами в дуплексном или в полудуплексном режиме. Дуплексный режим позволяет двум последовательным устройствам в процессе обмена информацией одновременно получать и передавать данные. Полудуплексный режим позволяет в каждый момент времени одному последовательному устройству только передавать данные, а другому — только принимать данные.

Модуль PIC16F877 MSSP

MSSP, или Master Synchronous Serial Port (главный синхронизирующий последовательный порт), представляет собой самостоятельную коммуникационную подсистему микропроцессора PIC16F877. MSSP используется приложениями I²C в качестве последовательного интерфейса. Как и в случае модуля USART, в работе модуля MSSP используются регистры и биты индикации состояния.

Шина I²C использует шесть регистров MSSP для управления, определения статуса и буферизации. Две линии ввода/вывода (I/O) PIC16F877 предназначены для использования шиной I²C: RC3 — для SCL (синхронизация), RC4 — для SDA (данные). Аналогично функции синхронизации USART, шина I²C работает в конфигурации ведущий/ведомый.

Шина I²C — изобретение корпорации Philips, представляющее собой элегантное решение проблемы взаимодействия интегральных микросхем в телевизорах и стереоприемниках. Интерфейс I²C будет подробно рассмотрен в данной книге, так как он используется в микроконтроллерах PIC. Благодаря Philips, нас окружают сотни различных устройств разных производителей, использующих предоставляемые шиной I²C возможности.

Микроконтроллер PIC16F877 наилучшим образом подходит для использования в нетрадиционных сетевых решениях. Используя прецизионные таймеры PIC16F877, мы можем объединять разрабатываемые протоколы и организовывать обмен битами между устройствами. Например, автор данной книги разработал приложение, использующее PIC для синхронизации данных, передаваемых и получаемых через вывод параллельного порта персонального компьютера. Кроме этого, ранние версии микроконтроллеров PIC16C5X, содержащие 18 линий ввода/вывода, не включали в свой состав модули UART или USART. Поэтому автор был вынужден создать программную эмуляцию модуля UART, который в настоящее время реализуется аппаратным способом. Как вы сможете убедиться далее, программная реализация модуля UART все еще остается необходимым элементом набора программных инструментов при разработке сетевых и коммуникационных приложений с использованием компактных, не содержащих модуля USART, микроконтроллеров PIC, имеющих 8 линий ввода/вывода.

Микроконтроллер PIC18F452

В данной книге написание программного кода, управляющего сетевым обменом, будет рассматриваться на примере микроконтроллера PIC18F452. Микроконтроллер PIC18F452 полностью совместим по контактам с микроконтроллером PIC16F877. Микроконтроллер PIC18F452 содержит 16 Кбайт встроенной программной памяти, дополненной 1.5 Кбайт памяти SRAM. Перечисленные особенности позволяют использовать микроконтроллер PIC18F452 при создании приложений Ethernet LAN. Помимо увеличенного объема внутренней памяти, микроконтроллер PIC18F452 может работать вдвое быстрее микроконтроллера PIC16F877 (тактовая частота микроконтроллера PIC18F452 — 40 МГц). Все коммуникационные устройства микроконтроллера PIC18F452 работают аналогично коммуникационным устройствам микроконтроллера PIC16F877, для генерации кода может использоваться компилятор CCS PIC C Compiler.

Микроконтроллер PIC12F675

Иногда приходится наблюдать забавную картину: водитель малолитражки, выжав из машины все возможное, несмотря на все свистки и гудки, не может обогнать мощный автомобиль. Подобные ассоциации возникают относительно небольшого микроконтроллера PIC12F675, имеющего 8 выводов. Данный микроконтроллер компактен как физически, так и логически. Микроконтроллер PIC12F675 содержит только 1К слов программируемой флэш-памяти и 64 байта памяти SRAM. Только 6 выводов микроконтроллера отвечают за ввод/вывод информации, но при этом он содержит два таймера, аналого-цифровой преобразователь и компаратор. Как и «большие ребята», микроконтроллер PIC12F675 содержит память EEPROM, емкость которой, однако, не превышает 128 байт. В результате изощренного кодирования на базе микроконтроллера PIC12F675 нами будет реализована вполне конкурентоспособная версия интерфейса RS-232.

Программирование микроконтроллеров PIC

Обсуждаемые в данной книге микроконтроллеры PIC построены с использованием флэш-памяти и программируются с помощью интерфейса ICSP (In-Circuit Serial Programming — встроенный последовательный интерфейс программирования). Поскольку данная книга посвящена организации связи и созданию сетевых устройств на базе микроконтроллеров, для программирования микроконтроллеров PIC автор предлагает не использовать аппаратное или программное обеспечение марки «сделано в гараже». Автор будет придерживаться программного обеспечения и программаторов производства компании Microchip. Вы можете выполнять программирование флэш-памяти микроконтроллеров PIC с использованием Microchip MPLAB ICD 2 (In-Circuit Debugger — встроенный отладчик) или Microchip PRO MATE II.

Эмуляция микроконтроллеров PIC

Microchip MPLAB ICD 2 и MPLAB ICE 2000 будут использоваться для отладки и отображения микропрограммного кода PIC, представленного в данной книге. Для демонстрации кода, внутренних регистров и областей памяти автор будет использовать Microchip MPLAB ICE 2000, систему эмуляции микроконтроллеров PIC. Аналогично компилятору CCS C Compiler и программатору Microchip PRO MATE II, MPLAB ICE 2000 и Microchip MPLAB ICD 2 изначально поддерживаются Microchip MPLAB. Объединение Microchip PRO MATE II, Microchip MPLAB ICE 2000, Microchip MPLAB ICD 2 и CCS PIC C Compiler позволяет автору продемонстрировать работу микропрограммного кода и памяти микроконтроллера PIC и всего разрабатываемого устройства с использованием только одного экрана MPLAB IDE.

ГЛАВА 1

ПРОТОКОЛ RS-232 — ОСНОВА СЕТЕЙ НА БАЗЕ МИКРОКОНТРОЛЛЕРОВ

Поскольку знание методов обработки данных с использованием протокола RS-232 облегчает реализацию сложных коммуникационных протоколов, изучение организации сетей начнем с протокола RS-232. Вы по достоинству оцените протокол RS-232 на этапе разработки собственных проектов.

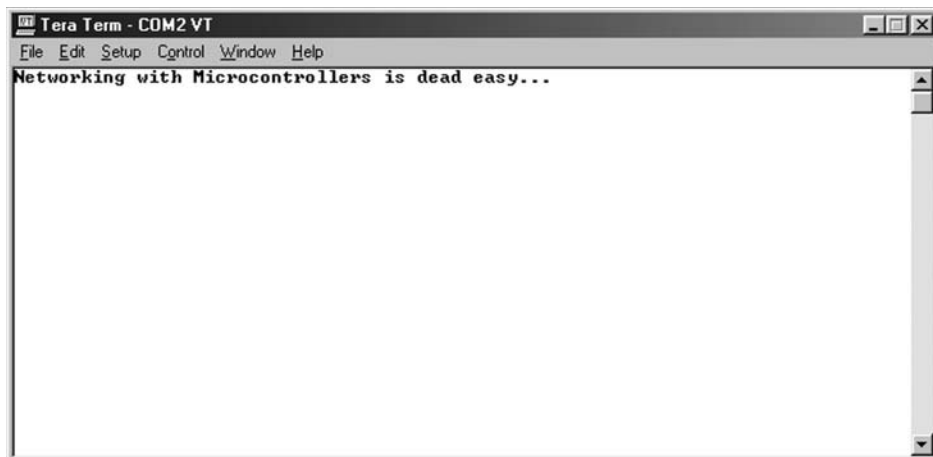


Рис. 1.1. Окно эмулятора терминала. Реализация обмена информацией по протоколу RS-232 с использованием микроконтроллера — чрезвычайно эффективное решение. Если вы продолжаете чтение данной книги, то убедитесь, что знание методов реализации простого протокола RS-232 с использованием микроконтроллера может облегчить разработку и отладку более сложных проектов.

Информация, представленная в окне эмулятора терминала на **Рис. 1.1**, была сгенерирована автором с использованием очень простого микропрограммного обеспечения и простейшего, стандартного микроконтроллера стоимостью два доллара. Для передачи информации в окно терминала была организована простейшая сеть. 8-битный микроконтроллер, не содержащий аппаратно реализованного универсального синхронно-асинхронного приемопередатчика USART (Universal Synchronous/Asynchronous Receiver/Transmitter), передает ASCII-символы на микросхему преобразователя RS-232. Последовательный кабель соединяет микросхему преобразователя RS-232 с последовательным портом персо-

нального компьютера. Таким образом, ASCII-символы передаются микропрограммным обеспечением микроконтроллера через линию ввода/вывода микроконтроллера на интегральную схему преобразователя RS-232, далее, через последовательный кабель — на схему USART/RS-232 персонального компьютера и, наконец, в окно эмулятора терминала, представленное на **Рис. 1.1**.

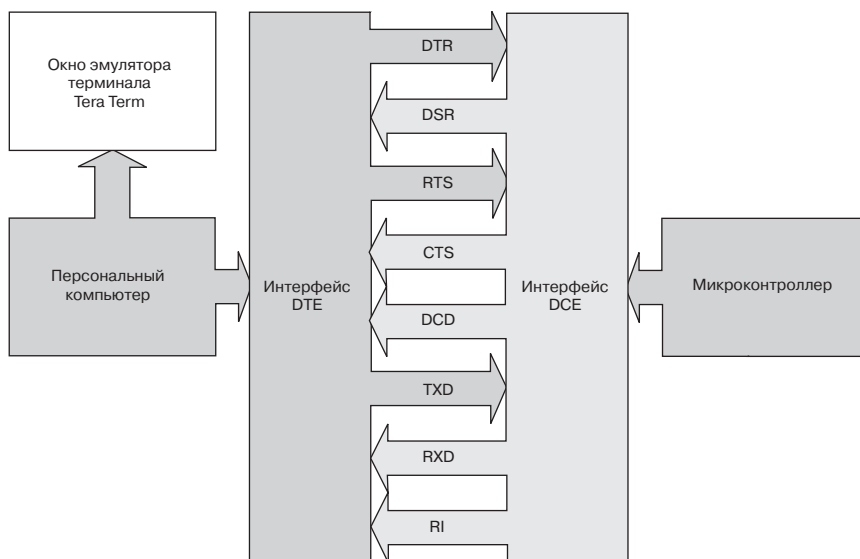


Рис. 1.2. Простейшая форма организации сети на базе микроконтроллеров. Интерфейсы DTE и DCE обычно включают в свой состав специализированные схемы преобразования напряжения, преобразующие уровни напряжения RS-232 в уровни напряжения, совместимые с компьютерным оборудованием, подключенным к каналам линий связи. Простейшая форма протокола RS-232 использует только сигналы TXD, RXD и землю.

Итак, только что была описана одна из простейших форм организации сетей на базе микроконтроллеров. Как можно видеть на **Рис. 1.2**, протокол RS-232 служит для двусторонней передачи данных между терминальным оборудованием (Terminal Equipment, DTE) и оконечным оборудованием передачи данных (Data Communications Equipment, DCE).

Примером устройства DTE может служить последовательный порт персонального компьютера. В нормальном состоянии DTE-интерфейс персонального компьютера выдает сигналы DTR (Data Terminal Ready — сигнал готовности терминала к передаче данных) и RTS (Request To Send — сигнал готовности данных к передаче). Сигналы DTR и RTS называются сигналами управления модемом. Стандартный интерфейс DCE-устройства отвечает на сигнал DTR сигналом DSR (Data Set Ready — сигнал готовности). На сигнал RTS последовательного порта персонального компьютера (DTE-устройства), DCE отвечает сигналом CTS (Clear To Send — сигнал готовности, показывающий, что DTE может продолжать передачу данных). Стандартный внешний модем, подключенный к последовательному порту персонального компьютера, представляет собой прекрасный пример устройства DCE.