Павел Агуров

bhv®

Последовательные интерфейсы ПК

Практика программирования

+ CD-ROM

- Протоколы последовательной связи
- Стандарты RS232/RS485
- Работа в DOS, Windows 98/ME/NT/2000/XP
- Примеры на языке Pascal
- Все уровни доступа
- Поддержка Plug and Play

Павел Агуров

Последовательные интерфейсы ПК

Практика программирования

Санкт-Петербург
«БХВ-Петербург»
2004

УДК 681.3.06
ББК 32.973-018
А23

Агуров П. В.

А23 Последовательные интерфейсы ПК. Практика программирования. — СПб.: БХВ-Петербург, 2004. — 496 с.: ил.

ISBN 5-94157-468-1

Книга представляет собой практическое руководство по программированию последовательных интерфейсов. В первой части книги представлены теоретические сведения о последовательных интерфейсах, во второй — практические примеры и листинги программ на языках Pascal и Delphi, а третья содержит справочные данные, облегчающие поиск необходимой информации. В приложениях приведены дополнительные сведения и ответы на часто задаваемые вопросы. Большое количество практических советов, примеров программ, а также последовательность и простота изложения позволят читателю уверенно овладеть изложенным в книге материалом. Для удобства читателей все исходные коды приводятся на прилагаемом к книге компакт-диске.

Для программистов, занятых в сфере промышленной автоматизации

УДК 681.3.06
ББК 32.973-018

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Алексей Семенов*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Наталия Першакова*
Оформление серии	*Via Design*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 13.02.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 39,99.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-468-1

Содержание

Введение

В книге рассматриваются вопросы программирования, которые по непонятным причинам довольно скудно освещены в доступной литературе, — программирование последовательных коммуникационных устройств. В Интернете и на книжных полках можно найти множество книг, посвященных аппаратной организации последовательной связи. А вот вопросы программирования почему-то ограничиваются описанием битов порта или заголовками функций для работы с портами. Хочется надеяться, что представленная книга заполнит этот пробел.

Почему выбран именно последовательный интерфейс? Простота и низкие аппаратные требования (в сравнении, например, с параллельным интерфейсом) делают его заманчивым методом реализации систем обмена информацией. Видимо по этой причине форумы по программированию заполнены вопросами "Как работать с COM-портом из Windows?", "Как прочитать данные с порта" и т. д. И хотя USB-интерфейс постепенно вытесняет COM, последний остается пока наиболее распространенным, например, в системах промышленной автоматизации.

Операционная система Windows имеет настолько много функций, что начинающий программист просто не знает с чего начать и где искать информацию. Тем более, что функции для работы с портами вовсе не называются `OpenComPort`, а конфигурирование параметров порта ничем не напоминает программирование в DOS. Конечно, MSDN Library (Microsoft Development Network Library) содержит ответы на все эти вопросы, но, во-первых, только на английском языке, и, во-вторых, только на уровне заголовков функций, описания их параметров и скелетов программ. Наша книга содержит и описания, и примеры использования как низкоуровневых функций DOS, так и функций Windows.

Книга построена таким образом, чтобы переход из DOS в Windows и из Windows 98 в Windows 2000/XP стал достаточно простым и минимально трудоемким.

Все исходные коды приведены на прилагаемом компакт-диске.

Для кого эта книга

Мы адресуем книгу нескольким категориям читателей:

- тем, кому необходимо быстро разобраться в принципах работы с последовательными портами и интерфейсами, не углубляясь в дебри теории;

- тем, кого не устраивает стандартный уровень, предоставляемый функциями DOS или Windows, и кому требуется более тонкое программирование порта;
- тем, кому досталась поддержка чужих программ, работающих с оборудованием;
- тем, кому необходимо быстро переделать программу, работающую в старых операционных системах (например, DOS, Windows 95/98) в Windows 2000/XP, а времени на полную переработку нет;
- тем, кто хочет повысить свой профессиональный уровень;
- инженерам, связанным с электроникой, желающим написать программу для работы с оборудованием, не вникая в тонкости программирования;
- ну, и наконец, просто любителям подключать к компьютеру разные устройства и проводить эксперименты.

Даже если вы не относите себя ни к одной из вышеперечисленных категорий, не стоит сразу отставлять книгу на полку. Например, занимаясь не последовательными, а параллельными интерфейсами, вы можете найти необходимую информацию о создании драйверов и INF-файлов для них. Для тех, кого интересуют вопросы программирования "железа", полезным станет описание прямого доступа к портам в Windows 2000/XP и т. п.

Вы не найдете в этой книге ни подробных описаний аппаратной части последовательных интерфейсов, ни рекомендаций по разворачиванию сети на основе RS-485, ни формул для расчета максимальной длины проводника... Зато вы сможете найти листинги и исходные тексты действительно работающих программ с русскими комментариями.

Структура книги

Книга состоит из трех частей. В первой части даются теоретические сведения о последовательных интерфейсах, протоколах, методах коррекции ошибок и т. д. Вторая часть книги — практическая, про нее мы расскажем далее. В третьей части книги приводятся справочные материалы, описание функций, портов, регистров и другие необходимые для работы сведения.

Практически вся вторая часть построена на основе одного примера: передачи данных с одного последовательного порта на другой. В качестве приемника и передатчика можно использовать два компьютера, но это затруднительно как в плане их наличия, так и при отладке программы (если конечно между компьютерами нет локальной сети). В нашей книге мы будем ориентироваться на один компьютер и передавать данные с одного последовательного порта на другой. Все примеры построены по принципу усложнения задачи: от совсем простых примеров в DOS мы дойдем до сложных асинхронных программ в Windows.

Сначала мы будем работать в DOS. Для этого существует несколько причин. Во-первых, эта информация может потребоваться программистам, которым досталась поддержка чужих программ. Примеры будут построены таким образом, что переход в Windows будет достаточно простым делом. Более того, специальные модули позволят обращаться к коммуникационным портам из Windows NT/2000/XP точно так же, как это делается в DOS. Во-вторых, достаточно часто DOS (или его клон) используется в качестве операционной системы промышленных компьютеров. Ну, и наконец, не зная основ, читателю будет сложно ориентироваться в сложных функциях Windows. В DOS мы пройдем полный путь от простейшей программы в несколько строк, использующей сервис BIOS, до программы, обрабатывающей прерывания коммуникационного порта.

Программирование в Windows мы начнем с создания программы, в точности повторяющей структуру программы в DOS. Мы даже заставим работать такую программу в Windows 2000/XP. Затем освоим функции Windows. На этом пути мы познакомимся с синхронной и асинхронной связью, научимся использовать потоки и функции синхронизации. Отдельно рассмотрим функции, которые были добавлены только в Windows 2000/XP.

Программирование в Windows мы завершим созданием программы, демонстрирующей работу Plug and Play. Мы создадим эмулятор устройства, работающий с портом COM1, и заставим Windows найти новое устройство на порту COM2.

В отдельной главе мы рассмотрим интерфейс RS-485, требующий более глубоких знаний о работе коммуникационного порта.

Одна из глав будет посвящена вычислению контрольных сумм и реализации протоколов обмена.

В приложениях вы сможете найти ответы на наиболее часто задаваемые вопросы по программированию коммуникационных портов и некоторые полезные функции.

Краткое описание глав

Главы 1—5 содержат теоретическую часть книги.

- Глава 1 ("Стандарты последовательной связи") дает общие представления об аппаратной части последовательных протоколов, разъемах и о самом стандарте. Специального заострения на аппаратной части не делается. Мы расскажем о многих RS-протоколах, принципах последовательной связи, кратко опишем разъемы и кабели, а также дадим представление о ресурсах и функциях компьютера.
- В главе 2 ("Протоколы") рассматриваются основные методы организации последовательной связи с точки зрения программных протоколов обмена.

Мы расскажем о текстовых и бинарных протоколах, о методах предотвращения и обнаружения потери данных.

- В главе 3 ("Последовательные протоколы IBM PC") мы расскажем об использовании последовательной связи в персональном компьютере. Мы не будем рассматривать аппаратные реализации интерфейсов, а только лишь протоколы обмена устройств с компьютером. Эта глава будет полезна не только тем, кто интересуется работой устройств компьютера, но и тем, кто думает о создании своих протоколов обмена.
- В главе 4 ("Промышленные последовательные протоколы") рассмотрена возможность использования последовательных протоколов обмена в промышленных условиях. Мы познакомимся с протоколами MODBUS, ProfiBus и CAN. Опять же, мы будем рассматривать только программную составляющую протоколов, оставляя аппаратную реализацию за рамками книги.
- В главе 5 ("COM-порты и Plug and Play") мы расскажем о работе системы автоматического обнаружения устройств, дадим некоторые сведения о формате INF-файла, необходимого для установки драйвера устройства, и полностью опишем алгоритм работы Plug and Play для последовательных портов.

Главы 6—17 содержат практическую часть книги.

- Глава 6 ("Использование сервисов BIOS") начинает практическую часть с самой простой программы последовательной связи. Использование сервисов BIOS позволяет написать программу последовательного обмена всего из нескольких строк. К сожалению, такая программа будет очень ограничена: сервис BIOS не позволяет использовать микросхему UART на полную мощность. Эта глава будет интересна тем, кому приходится применять чужие модули последовательной связи, использующие сервисы BIOS. Кроме того, промышленные компьютеры с клоном DOS обычно предоставляют именно такой способ работы с внешними устройствами.
- В главе 7 ("Прямое программирование портов") избавиться от ограничений BIOS нам поможет прямое обращение к портам. Эти сведения будут полезны не только программистам, работающим в DOS, но и тем, кто создает свой драйвер для Windows.
- Использование прямого доступа не избавляет нас от неприятной необходимости постоянного опроса порта. В главе 8 ("Использование обработки прерываний") мы научимся обрабатывать прерывания последовательного порта и на этом завершим программирование в режиме DOS. Сведения этой главы пригодятся программистам, использующим промышленные контроллеры с установленным DOS.
- Глава 9 ("Переход в Windows") демонстрирует переход из DOS в Windows. Приведенные здесь программы еще не используют функций Windows, но

уже работают под управлением операционных систем Microsoft. Эта глава будет полезна тем, кто хочет перенести свои программы из DOS в Windows с минимальными изменениями. Вы сможете работать с портами в Windows 2000/XP так же как в DOS.

- В главе 10 ("Использование функций Windows") мы, наконец, доберемся до непосредственного использования функций Windows. Так же как в DOS мы начнем с простейшей программы обмена, но уже используя функции Windows.
- Глава 11 ("Использование потоков") по смыслу похожая на главу 8 ("Использование обработки прерываний"), рассказывает о том, как разгрузить основной процесс от постоянного опроса порта в Windows.
- Отдельный поток, созданный в главе 11, к сожалению, не избавляет от необходимости постоянного опроса порта, а значит, лишней загрузки процессора. В главе 12 ("Функции асинхронного доступа и события") используются асинхронные функции работы с портом совместно с функциями событий, которые позволяют нам создать поток, "просыпающийся" только в момент появления новых данных. Как мы покажем в нашем примере, загрузка процессора при этом будет практически незаметна.
- В главе 13 ("Специальные коммуникационные функции") мы опишем функции, которые не требовались в предыдущих главах, но потребуются в следующих. Кроме того, будут описаны функции, которые появились только в Windows 2000.
- До сих пор мы занимались приемом и передачей только одного байта. Для реальной программы этого маловато. В главе 14 ("Реализация протоколов обмена") мы создадим специальные компоненты, которые позволят нам разделить транспортные функции последовательного порта и функции работы с конкретным протоколом обмена.
- В главах теоретической части мы описывали протоколы обмена, такие как MODBUS, протоколы модемов XModem, YModem и другие, а в главе 15 ("Вычисление контрольных сумм") приведем примеры вычисления контрольных сумм, применяемых в этих протоколах.
- Несмотря на договоренность не рассматривать аппаратных реализаций протоколов, интерфейс RS-485 заслуживает отдельной главы 16 ("Интерфейс RS-485"), т. к. его программная реализация требует дополнительных знаний и функций.
- Конечно, описание протокола Plug and Play, которое мы привели в главе 5, достаточно интересно. Но ведь всегда гораздо интереснее попробовать самому! Программа из главы 17 ("Работа Plug and Play") будет эмулировать присутствие некоторого устройства, подключенного к порту COM2 с помощью обработки сообщений порта COM1. Также мы напишем INF-файл, устанавливающий "драйверы" нашего устройства.

Главы 18—23 содержат справочную часть книги.

- Глава 18 ("Сервис BIOS INT14H/INT21H") — справочник функций BIOS.
- Глава 19 ("Порты IBM PC") — справочник коммуникационных портов и их значений. С этими сведениями мы пользовались в главах 7—9.
- Глава 20 ("Структуры Windows для работы с COM-портами") — справочник структур Windows и описания полей этих структур.
- Глава 21 ("Функции Windows для работы с COM-портами") — справочник функций Windows.
- Глава 22 ("Функции `DeviceIoControl` для последовательного порта") — описание кодов прямого доступа к драйверу последовательного порта.
- Глава 23 ("AT-команды и регистры модемов") — краткий справочник основных AT-команд и регистров модемов.

Программные требования

Все программы мы будем реализовывать на языках Borland Pascal 7.0 и Borland Delphi 6. Поскольку мы не будем использовать никаких специфических функций, присущих именно этим версиям языков, все примеры могут быть скомпилированы в других версиях практически без модификации.

Версия Windows может быть любой — от Windows 98 до Windows 2000/XP. Конечно, примеры работы с функциями, которые появились только в Windows 2000, не будут работать в Windows 98, но обратной совместимости мы будем придерживаться: даже прямое программирование портов станет доступным в Windows 2000 с помощью специального модуля.

Все примеры, приведенные в книге, тестировались в версиях Windows 98 SE, Windows 2000 Workstation и Windows 2000 Advanced Server.

Для компиляции драйверов, приведенных в книге, потребуется MS Visual Studio и Windows 2000 DDK. На компакт-диске содержатся уже готовые модули драйверов, так что эту часть можно пропустить.

О программном коде

Книга содержит полные исходные коды всех программ, однако многие листинги содержат только изменения кода относительно предыдущего листинга. Такое сокращение позволяет не только экономить место, но и улучшить понимание кода, делая акцент только на новой функциональности. Код на компакт-диске содержит полные тексты без сокращений.

В программах для DOS мы не будем приводить код функций работы со строками. Код этих функций можно найти в приложениях или на компакт-диске к книге. В программах на Delphi мы не приводим код самого

проекта (DPR-файл) и код формы (DFM-файлы). Все исходные коды можно найти на компакт-диске. Еще раз повторим, что такие сокращения не означают отсутствие возможности скомпилировать и попробовать приведенные примеры на своем компьютере, а преследуют цель экономии места. Все необходимые исходные коды и модули находятся на компакт-диске.

Обозначения

При написании чисел мы будем придерживаться следующих правил:

- шестнадцатеричные числа будут иметь префикс "$", например "$45";
- шестнадцатеричные числа могут иметь префикс "0x" или постфикс "H", если того требует контекст изложения или формат строки, например "INT 14H";
- битовые последовательности заключены в угловые скобки, например "<0010>".

Аппаратные требования

Достаточно обычного домашнего компьютера, на котором компиляция программы в Delphi занимает приемлемое для вас время.

Для тестирования последовательной связи необходимо иметь либо два компьютера, либо один компьютер с двумя последовательными портами. В последнем случае нам потребуется PS/2- или USB-мышь, т. к. оба последовательных порта мы займем для экспериментов.

Для соединения портов между собой нам потребуется нуль-модемный кабель, желательно полностью распаянный, но для большей части программ достаточно трехпроводного кабеля.

Благодарности

В первую очередь автор хочет поблагодарить заместителя главного редактора Евгения Рыбакова. Без его советов и критики эта книга никогда бы не была создана. Автор благодарит всех друзей и родных, которые терпели его в процессе написания книги, а также коллектив издательства "БХВ-Петербург". Отдельная благодарность Вадиму Шарикову (компания Bixel, **http://bixel.ru**), дизайнеру сайта PvaSoft.

Обратная связь

Обо всех ошибках или пожеланиях сообщайте по адресу **books@pvasoft.com**.

ЧАСТЬ I

Протоколы и интерфейсы

Глава 1

Стандарты последовательной связи

Стандарт есть легкоусвояемая форма,
маскирующая присутствие или отсутствие содержания[1].

Глава дает общие представления об аппаратной части последовательных протоколов, разъемах и о самом стандарте. Специального заострения внимания на аппаратной части не делается.

1.1. Стандарты последовательной связи

Последовательный интерфейс используется для связи двух устройств между собой. Данные в одну сторону передаются по одному проводу с помощью последовательности битов.

Мы будем рассматривать подключение одного или нескольких устройств к компьютеру, т. к. такая комбинация наиболее часто используется в реальной жизни. Естественно, при подключении нескольких устройств к компьютеру обмен производится только с одним из этих устройств.

Для соединения со стороны компьютера используется интерфейс, называемый *COM-порт* (COMmunication port, коммуникационный порт). Этот порт обеспечивает асинхронный обмен и реализуется на микросхемах универсальных асинхронных приемопередатчиков (UATR, Universal Asynchronous Receiver Transmitter), совместимых с семейством i8250. Хотя стандарт RS-232C предусматривает и асинхронный, и синхронный режимы обмена, COM-порт компьютера поддерживает только асинхронный режим. Для реализации синхронного обмена применяются специальные адаптеры, например, SDLC или V.35.

---

[1] Все афоризмы взяты с юмористического сайта **http://www.anekdot.ru**. Часть эпиграфов принадлежит перу Стаса Янковского **http://syy.narod.ru/wordsd.htm**.

Далее мы опишем основные протоколы последовательной связи, но сначала несколько замечаний.

Слово "протокол" вносит некоторую путаницу. С одной стороны, RS-232 и RS-485 называют протоколами, а с другой, MODBUS, ZModem и CAN — тоже протоколы. Какой же из них настоящий? Дело в том, что последовательная связь состоит из двух составляющих — аппаратной и программной.

Стандарты RS-232, RS-485 и другие описывают аппаратную часть: разъемы, назначение сигналов, уровни напряжений и т. п. Вторая составляющая — программная реализация протоколов, т. е. договоренность о правилах передачи данных. В нашей книге мы не будем подробно рассматривать аппаратные реализации всех интерфейсов. Исключение составит описание разъемов, сигналов и соединительного кабеля, необходимого нам для работы. А вот токовая петля, тонкости организации сетей RS-485 и подробное описание различий между RS-протоколами останутся за рамками книги. Интерфейс RS-485 удостоится отдельного рассмотрения из-за необходимости дополнительных программных решений при его реализации.

Вторую составляющую протокола — программную реализацию — мы будем рассматривать в *гл. 2*, а создание программ отложим до второй части книги.

И еще одно замечание. Все стандарты имеют двойные, а то и тройные названия. Двойственность названий связана с тем, что Международный союз электросвязи ITU-T использует аналогичные стандарты с названиями V.xx. И, вообще говоря, два названия это еще не предел. Так, стандарту RS-232 соответствует ISO 2110, Министерство обороны США выпустило практически идентичный стандарт Mil-Std-188C, а в нашей стране подобный стандарт введен ГОСТом 18145-81.

Для краткости мы будем использовать два специальных обозначения:

- *DTE* (Data Terminal Equipment) — оконечное оборудование, принимающее или передающее данные. В качестве DTE может выступать компьютер, принтер, плоттер или другое периферийное оборудование;
- *DCE* (Data Communications Equipment) — аппаратура канала данных. Функция DCE состоит в обеспечении возможности передачи информации между двумя или большим числом DTE. Для этого DCE должно обеспечить соединение с DTE, с одной стороны, и с каналом передачи — с другой. Роль DCE чаще всего выполняет модем, например, как показано на рис. 1.1.

Устройства DTE могут быть соединены напрямую, без DCE, с помощью *нуль-модемного кабеля*, как показано на рис. 1.2. Разводка такого кабеля показана на рис. 1.3.

Рис. 1.1. Полная схема соединения по RS-232

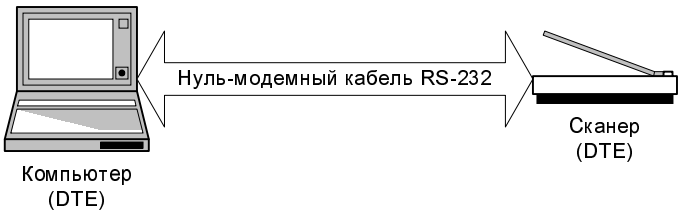

Рис. 1.2. Соединение по RS-232 нуль-модемным кабелем

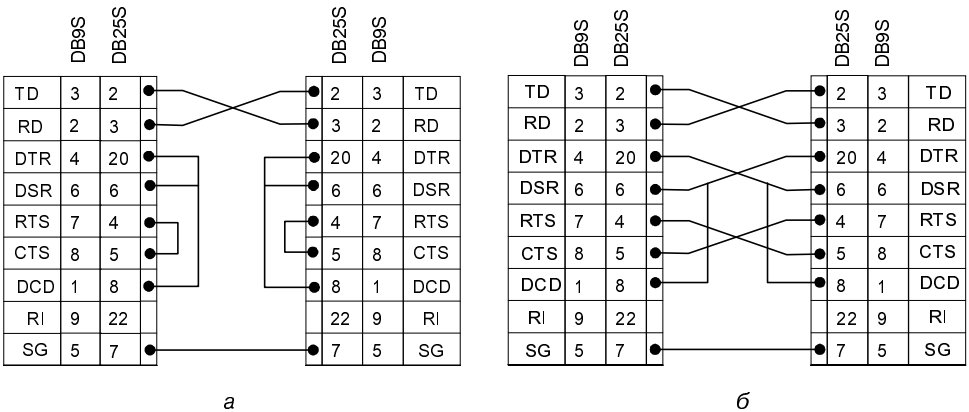

Рис. 1.3. Нуль-модемный кабель:
а — минимальный; б — полный

1.1.1. Протокол RS-232

Из-за простоты и низких аппаратных требований (в сравнении, например, с параллельным интерфейсом), последовательные интерфейсы активно используются в электронной промышленности. В настоящее время наиболее распространенным является стандарт, разработанный Ассоциацией промышленных средств связи (TIA, Telecommunication Industry Association, **http://www.tiaonline.org**) и Ассоциацией электронной промышленности (EIA,

Electronic Industries Alliance, **http://www.eia.org**) "EIA/TIA-232-E", более известный под названием "RS-232".

Стандарт RS-232 (его официальное название "Interface Between Data Terminal Equipment and Data Circuit-Termination Equipment Employing Serial Binary Data Interchange") предназначен для подключения аппаратуры, передающей или принимающей данные, к оконечной аппаратуре каналов данных. Стандарт описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъемов.

Интерфейс RS-232 используется и во многих устройствах обычного персонального компьютера, начиная с "мыши" и модема до ключей аппаратной защиты. И хотя уже все компьютеры имеют интерфейс USB, интерфейс RS-232 еще жив и активно применяется.

Согласно стандарту RS-232, сигнал (последовательность битов) передается напряжением. Передатчик и приемник являются *несимметричными*: сигнал передается относительно общего провода (в отличие от симметричной передачи протокола RS-485 или RS-422). В табл. 1.1 приведены границы напряжений для сигналов приемника и передатчика. Логическому нулю на входе приемника соответствует диапазон +3...+12 В, а логической единице — диапазон −12...−3 В. Диапазон −3...+3 В — зона нечувствительности, обеспечивающая гистерезис приемника (передатчика). Уровни сигнала на выходах должны быть в диапазоне −12...−5 В для представления логической единицы и +5...+12 В для представления логического нуля.

Вообще говоря, стандарт RS-232 состоит из трех частей. Первая часть, стандарт RS-232C, была принята в 1969 году и содержит описание электрических цепей и сигналов несимметричной последовательной связи. Вторая часть, стандарт RS-232D, принята в 1987 году и определяет дополнительные линии тестирования, а также формально описывает разъем DB-25. Третья часть, RS-232E, принята в 1991 году.

Таблица 1.1. *Границы напряжений COM-порта (стандарт RS-232)*

	Диапазон напряжения входа приемника	**Диапазон напряжения выхода передатчика**	**Состояние управляющего сигнала**	**Состояние линии данных**
Логический 0	от –12 до –3 В	от –12 до –5 В	ON	MARX
Логическая 1	от +3 до +12 В	от +5 до +12 В	OFF	SPACE

Важно!

Интерфейс не обеспечивает гальванической развязки устройств. Подключение и отключение интерфейсных кабелей устройств с независимым питанием должно производиться при отключенном питании.

Один из самых неприятных недостатков стандарта RS-232 — плохая помехозащищенность и, соответственно, короткие линии передачи. Естественно, были созданы стандарты, решающие эти проблемы.

1.1.2. Протокол RS-422A

Стандарт RS-422A (другое название ITU-T V.11) определяет электрические характеристики симметричного цифрового интерфейса. Он предусматривает работу на более высоких скоростях (до 10 Мбит/с) и больших расстояниях (до 1000 м) в интерфейсе DTE-DCE. Для его практической реализации, в отличие от RS-232, требуются два физических провода на каждый сигнал. Реализация симметричных цепей обеспечивает наилучшие выходные характеристики.

Подобно V.28, данный стандарт является простым описанием электрических характеристик интерфейса и не определяет параметры сигналов, типы разъемов и протоколы управления передачей данных. Для линий интерфейсов RS-422A и RS-423A могут быть использованы различные проводники (или пары проводников) одного и того же кабеля.

Стандарт RS-422A был разработан совместно с RS-423A и позволяет размещать линии этих интерфейсов в одном кабеле. Он несовместим с RS-232, и взаимодействие между RS-422A и RS-232 может быть обеспечено только при помощи специального интерфейсного конвертера.

Практически это полнодуплексный протокол RS-485. Прием и передача идут по двум отдельным парам проводов, причем на каждой паре может быть только по одному передатчику.

1.1.3. Протокол RS-423A

Стандарт RS-423A (другое название V.6) определяет электрические характеристики несимметричного цифрового интерфейса. "Несимметричность" означает, что данный стандарт, подобно RS-232, для каждой линии интерфейса использует только один провод. При этом для всех линий используется единый общий провод.

Как и RS-422A, этот стандарт не определяет сигналы, конфигурацию выводов или типы разъемов. Он содержит только описание электрических характеристик интерфейса. Стандарт RS-423A предусматривает максимальную скорость передачи 100 Кбит/с.

1.1.4. Протокол RS-485

Полное название этого стандарта TIA/EIA-485A (аналогичный стандарт описывается в ISO 8482). Он включает в себя 17 страниц и определяет требования к электрическим характеристикам формирователей и приемников.

В нем ничего не говорится ни о качестве сигнала, ни о методах доступа к линиям связи, ни о протоколе обмена, ни о технологической схеме (назначении контактов и т. п.). "Приложение А" этого стандарта описывает принципы работы устройств на основе RS-485.

В дополнение к описанию стандарта был опубликован документ TSB89, называющийся "Application Guidelines for TIA/EIA-485A". TSB89 состоит из 23 страниц и посвящается разъяснению, как применять устройства, определяемые стандартом TIA/EIA-485A для физических сетей.

Подробно этот интерфейс будет обсуждаться в *гл. 16*.

1.1.5. Протокол RS-499

Стандарт RS-449, в отличие от RS-422A и RS-423A, содержит информацию о параметрах сигналов, типах разъемов, расположении контактов и т. п. В этом отношении RS-449 является дополнением к стандартам RS-422A и RS-423A. Стандарту RS-449 соответствует международный стандарт V.36.

Комбинация RS-449, RS-422A и (или) RS-423A первоначально предназначалась для замены RS-232. Однако этого не произошло, хотя данные стандарты нашли достаточно широкое применение в качестве высокоскоростного интерфейса DTE-DCE.

Стандарт RS-449 определяет 30 сигналов интерфейса, большинство из которых имеют эквиваленты в RS-232, но добавлены и новые. Обозначения большинства сигналов были изменены во избежание путаницы.

Десять сигналов RS-449 определены как линии первой категории. Эта группа сигналов включает в себя все основные сигналы данных и синхронизации, такие как "Передаваемые данные", "Принимаемые данные", "Синхронизация терминала". Скорость передачи сигналов первой категории существенно зависит от длины кабеля. Для линий этой категории на скоростях до 20 Кбит/с могут использоваться стандарты RS-422A либо RS-423A; на скоростях выше 20 Кбит/с (до 2 Мбит/с) — только RS-422A.

Оставшиеся 20 линий классифицируются как линии второй категории и используются стандартом RS-423A. К этой категории относятся такие управляющие линии, как "Качество сигнала", "Выбор скорости передачи" и др.

Стандарт RS-449 определяет тип разъема и, в отличие от RS-232, распределение контактов разъема. Используемые разъемы имеют 37 контактов для прямого и 9 контактов для обратного канала.

1.1.6. Протокол RS-562

Протокол RS-562 представляет собой низковольтную реализацию RS-232 с максимальным уровнем сигнала 3,5 В.

1.1.7. Протокол V.24

Рекомендация V.24 содержит описание линий и набора сигналов обмена между DTE и DCE. В RS-232 используются другие обозначения линий, однако линии интерфейса RS-232 и рекомендации V.24 выполняют совершенно одинаковые функции. V.24 определяет большее количество линий, чем RS-232, поскольку стандарт V.24 используется и в других интерфейсах. В этом смысле RS-232 является подмножеством V.24. Рекомендация V.24 не определяет электрические характеристики или другие физические аспекты реализации, такие как тип разъема, расположение контактов, длина кабеля и скорость обмена. Технические вопросы реализации интерфейса подробно изложены в стандарте V.28.

1.1.8. Протокол V.28

Рекомендация V.28 определяет только электрические характеристики интерфейса V.24, обеспечивающего работу по несимметричным двухполярным линиям обмена на скоростях до 20 Кбит/с. К таким характеристикам относятся уровни используемых сигналов, емкостное сопротивление и т. д. Данная рекомендация не содержит требований к длине кабеля, типу разъемов и расположению их контактов. Поэтому рекомендация V.28 может рассматриваться как подмножество стандарта RS-232.

1.1.9. Протокол V.35

Стандарт V.35 появился в начале 1980-х годов как спецификация интерфейса между устройствами доступа к сети (мультиплексором, модемом или др.) и высокоскоростной сетью с коммутацией пакетов. Первоначально эта спецификация использовалась для подключения групповых модемов (модемных пулов) к коммутационному устройству.

Рекомендация V.35 определяет синхронный интерфейс для работы по аналоговым широкополосным каналам с полосой пропускания 60—108 кГц и скоростью передачи до 48 Кбит/с.

В приложении к стандарту определяется вид электрического соединения, обеспечивающего высокоскоростной последовательный интерфейс между мультиплексором и коммутационным оборудованием сети.

Модемы, использующие стандарт V.35, не получили широкого распространения на рынке, но интерфейс в качестве высокоскоростной замены RS-232 прижился. В спецификации стандарта не был определен тип электрического разъема, но фирма IBM в свое время стала выпускать совместимые с V.35 большие прямоугольные разъемы с массивными прижимными винтами. Получилось очень надежное соединение. Остальные производители коммутационной техники стали повторять конструкцию соединителя IBM, который и стал стандартом де-факто, и был принят в качестве рекомендации ISO 2593.

1.1.10. Протокол X.21

Стандарт X.21 впервые был опубликован в 1972 году. Он определяет физические характеристики и процедуры управления для интерфейса DTE-DCE в режиме синхронной передачи данных и может применяться как в сетях с коммутацией каналов, так и в сетях на выделенных линиях. Стандарт предусматривает дуплексную работу при условии, что устройства связаны друг с другом реальными, а не виртуальными цифровыми линиями связи. Функциональные процедуры X.21 формализованы в виде диаграмм состояний, рассмотрение которых выходит за рамки данной книги.

Рекомендация ITU-T X.21 определяет формат передаваемых символов, которые представляются в коде MTK-5 (Международный телеграфный код № 5). Данный интерфейс рассчитан на сквозную цифровую передачу. В нем процесс установления соединения и разъединения полностью автоматизирован при помощи набора сигналов о состоянии соединения и о его неисправностях. В ходе передачи данных через интерфейс могут передаваться любые последовательности битов.

Создатели этого стандарта стремились максимально упростить его. Так, соединение DTE с DCE требует существенно меньшего числа сигнальных линий, чем аналогичное соединение для интерфейса RS-232.

1.1.11. Рекомендация X.21bis

Рекомендация X.21bis была разработана для обеспечения возможности подключения к сетям передачи данных общего пользования тех пользователей, которые используют для этого аналоговые выделенные или коммутируемые каналы и имеют синхронные модемы, работающие согласно рекомендациям серии V.

1.1.12. Краткое сравнение RS-протоколов

В табл. 1.2 мы привели краткое сравнение 4 наиболее распространенных RS-протоколов. Более подробную информацию можно найти в соответствующей литературе.

Таблица 1.2. *Сравнение последовательных интерфейсов*

	RS-232CV.24	RS-422AV.11	RS-423AV.6	RS-485
Максимальная длина кабеля, м	15	1200	1200	1200
Число приемников	1	10	10	32
Число передатчиков	1	1	1	32