

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

ИНТЕРФЕЙС RS-232 В КАНАЛАХ ПОСЛЕДОВАТЕЛЬНОЙ ПЕРЕДАЧИ ДАННЫХ

Методические указания

САМАРА 2004

Составители: В.Г. Иоффе, К.Е. Климентьев

УДК 681.3

Интерфейс RS-232 в каналах последовательной передачи данных: Метод. указания / Самар. гос. аэрокосм. ун-т; Сост. *В. Г. Иоффе, К. Е. Климентьев*. Самара, 2004. 36 с.

Предназначены для студентов, изучающих в рамках специальности 220200 курсы "Интерфейсы АСОИУ", "Системы реального времени" и прочие курсы аналогичной тематики. Также могут быть использованы в курсовом и дипломном проектировании.

Выполнены на кафедре информационных систем и технологий.

Печатается по решению редакционно-издательского совета Самарского государственного аэрокосмического университета

Рецензент доц. Д. Л. Головашкин

Введение

RS-232 представляет собой стандарт для последовательной передачи цифровых данных между двумя устройствами. Он был разработан в 1969 г. рядом крупных промышленных корпораций и опубликован Ассоциацией электронной промышленности США (Electronic Industries Association - EIA) как модификация "С" рекомендуемого стандарта (Recommended Standard - RS) номер 232. В настоящее время этот стандарт широко распространен: в руководящих материалах Международного союза электросвязи ИТУ-Т используются аналогичные стандарты V.24 и V.28, министерство обороны США выпустило его под названием Mil-Std-188C, а в нашей стране он зафиксирован как ГОСТ 18145-81. Стандарт имеет ряд модификаций: модификация "D" RS-232 была принята в 1987 г., а самой последней является модификация "Е", принятая в июле 1991 г. под названием EIA/TIA-232E.

Основанный на этом стандарте интерфейс (в дальнейшем - "интерфейс RS-232") предназначен для подключения аппаратуры, передающей или принимающей данные (АПД, английская аббревиатура - DTE), к оконечной аппаратуре каналов данных (АКД, английская аббревиатура - DCE).

1. Общая характеристика

Основные характеристики интерфейса RS-232:

- поддерживаемая топология соединения - "точка-точка";
- направление передачи - дуплексная и симплексная;
- режим обмена - асинхронный (в последовательных каналах ПЭВМ) и синхронный (реализуется с помощью специальных адаптеров SDLC);
- метод кодирования информации - NRZ;
- используемые сигналы - потенциальные (несимметричные), уровень логического нуля: 3..25В, уровень логической единицы: -3..-25В, в ПЭВМ используются уровни $\pm 12В$;
- скорость передачи информации - от 110 бит/с до 115200 бит/с (см. табл. 1.1);

- количество линий передачи информации и служебных сигналов - до 25 (см. табл. 1.2);
- стандартные разъемы - DB25, DB9 или Mini DIN-8 (см. рис. 1.1).

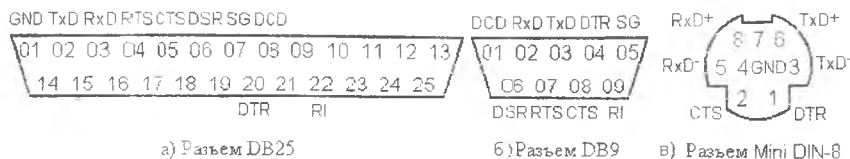


Рис. 1.1. Нумерация контактов на разъемах RS-232

Таблица 1.1

Допустимые скорости передачи

Скорость, бит/с (бод)	Макс. длина экранированного кабеля, м	Макс. длина неэкранированного кабеля, м	Скорость, бит/с (бод)	Макс. длина экранированного кабеля, м	Макс. длина неэкранированного кабеля, м
110	1524	914	4800	304	76
150	1524	914	9600	76	76
300	1524	914	19200	76	76
600	1524	914	38400	76	76
1200	914	914	57600	76	76
2400	304	152	115200	15	15

Таблица 1.2

Линии и контакты RS-232

Сигнал	Номер контакта на DB25	Номер контакта на DB9	Название
FG (GND)	1	---	Основная (или защитная) земля
TD (TxD)	2	3	DTE – посылка данных, DCE – прием
RD (RxD)	3	2	DTE – прием данных, DCE – посылка
RTS	4	7	Запрос передачи
CTS	5	8	Сброс передачи и готовность приема
DSR	6	6	Связь установлена
SG	7	5	Сигнальная земля
DCD	8	1	Обнаружение несущей частоты данных

Сигнал	Номер контакта на DB25	Номер контакта на DB9	Название
---	---	---	Положительное контрольное напряжение
---	---	---	Отрицательное контрольное напряжение
QM	---	---	Режим выравнивания
SDCD	---	---	Обнаружение несущей вторичных данных
SCTS	---	---	Вторичный сброс передачи
STD	---	---	Вторичные передаваемые данные
TC	---	---	Синхронизация передатчика
SRD	---	---	Вторичные принимаемые данные
RC	---	---	Синхронизация приемника
DCR	---	---	Разделенная синхронизация приемника
SRTS	---	---	Вторичный запрос передачи
DTR	20	---	Готовность терминала
SQ	---	---	Качество сигнала
RI	22	---	Индикатор звонка
---	---	---	Селектор скорости данных
TC	---	---	Внешняя синхронизация передатчика
---	---	---	Занятость

Устройства DCE обычно снабжены разъемом типа "male" ("папа"), устройства DTE - разъемом типа "female" ("мама"). Эти разновидности разъемов имеют одинаковую форму и обозначения контактов. Для подключения устройства DCE к DTE (например, модема к компьютеру) используют кабели с "параллельным" расположением линий данных (прил. Д, рис. Д1 и Д2); для подключения DTE к DTE (например, двух компьютеров) используют кабели с перекрестными линиями данных (см. рис. Д3 и Д4). Последний режим называется режимом "нуль-модема". Различные варианты кабелей для соединения приведены в прил. Д.

1.1. Особенности использования сигналов интерфейса

Контакты RxD и TxD соответствуют выходу передатчика и входу приемника. Первоначально на линии передачи данных присутствует высокий уровень сигнала, что соответствует "молчанию" (состояние "BREAK"). Начало передачи данных инициируется передающим

устройством и отмечается появлением "стартового бита", т.е. перепадом уровня с высокого на низкий. Начиная с этого момента передающее устройство посылает, а приемное устройство принимает через фиксированные интервалы времени биты блоков данных, организованных следующим образом (рис. 1.2):

- START - "стартовый бит", отмечающий начало блока данных;
- DATA - от 5 до 8 информационных битов, составляющих блок данных;
- PARITY - контрольный бит (бит четности, бит паритета), позволяющий проконтролировать целостность переданных данных (может отсутствовать);
- STOP - биты, отмечающие конец блока данных ("стоповые" биты); количество битов (1, 1.5 или 2) определяется особенностями линий связи и используемых внешних устройств.



Рис. 1.2. Формат блока данных

Контрольный бит четности представляет собой дополнение количества единичных информационных битов в блоке до четного значения (или, что то же самое, результат побитовой операции "исключающее ИЛИ" над информационными битами). Его значение вычисляется дважды - до передачи блока и после нее. Несовпадение значений бита свидетельствует об искажении переданных данных.

Примечание. В зависимости от установок возможна генерация, передача и проверка инвертированного значения этого бита.

Устройства, связанные через интерфейс RS-232, должны быть строго согласованы по формату блока данных и по скорости передачи. Частоты генераторов приемника и передатчика должны отличаться на малых скоростях передачи не более, чем на 5%, а с увеличением скорости это требование становится еще более жестким.

Сигналы RTS и CTS используются для организации передачи данных из DTE к DCE. Сигнал RTS на выходе передатчика информирует приемник о наличии данных для передачи. Появление

сигнала CTS на входе передатчика сигнализирует о готовности приемника к приему данных.

Сигналы DTR и DSR используются для организации передачи данных из DCE в DTE. Устройство DCE формирует сигнал, поступающий на вход DSR. Если устройство DTE готово к приему, то оно выставляет сигнал DTR. Передача данных начинается по сигналу CTS.

Сигналы RTS, CTS, DTR и DSR целесообразно использовать в том случае, если быстродействия передающего и принимающего устройств существенно различаются. Также устройство DCE может использовать линию RI для сообщения о том, что по каналу связи принимается сигнал вызова. Однако устройство DTE может игнорировать этот сигнал по своему усмотрению.

Сигнал DCD формируется устройством DCE в том случае, если установлено соединение с требуемым качеством связи. Если линия сигнала DCD находится в неактивном состоянии, то это означает либо полное отсутствие данных на входе устройства DCE, либо низкое качество сигнала данных. Прием такого сигнала данных должен быть запрещен.

Защитная линия FG должна быть соединена с корпусом устройства и экраном кабеля. Сигнальная линия SG соединяется с нулевым потенциалом источника питания.

В зависимости от особенностей связываемых по интерфейсу RS-232 устройств могут быть задействованы различные группы сигналов и, как следствие, использованы различные варианты соединительных кабелей. Для автономной отладки обмена данных через RS-232 и тестирования приемного и передающего оборудования могут быть использованы "заглушки" (loopback). Различные варианты "заглушек" также приведены в прил. Д.

1.2. Синхронизация передачи потоков данных

Для управления потоками данных могут использоваться различные варианты протоколов обмена: 1) программный; 2) аппаратный; 3) смешанный.

Программный протокол XON/XOFF предполагает наличие двунаправленного канала передачи данных. Если приемник не готов к приему, то он посылает по своей линии TxD код XOFF. Передатчик, приняв этот код, приостанавливает передачу до тех пор, пока не получит от приемника код XON. При передаче кодов ASCII обычно XOFF кодируется значением 13h, а XON - значением 11h, хотя возможно использование других числовых значений, не встречающихся в передаваемых данных. Также возможна реализация программного протокола обмена с использованием сигналов DTR и DSR, доступ к которым обеспечивается через регистры управления модемом MCR и состояния модема MSR (прил. Г).

Аппаратные протоколы обмена используют управление передатчиком по входу CTS. Если в процессе передачи на линии CTS устанавливается неразрешающий уровень, то завершается передача текущего блока данных и передатчик прекращает работу до сброса сигнала CTS. Управление CTS возможно сигналами RTS или DTR. Аппаратные протоколы требуют использования "полных" вариантов соединительных кабелей (прил. Д).

2. Применение RS-232

Интерфейс RS-232 нашел широкое применение в самых различных областях техники, т.к. его реализация относительно проста, широко поддерживается производителями электронного оборудования и разработчиками программного обеспечения. Являясь внешним интерфейсом, RS-232 используется как канал связи между ПЭВМ и целым рядом автономно функционирующих сложных устройств: аналитических приборов, контрольно-кассовых торговых аппаратов, клинико-диагностического медицинского оборудования, измерительных установок, модемов и т.п.

К недостаткам RS-232 относятся невысокое быстродействие и топология "точка-точка". Поэтому в сложных системах он используется совместно с другими интерфейсами, обеспечивающими высокие скорости обмена и возможность магистральной организации. Обобщенная структура подобного рода систем приведена на рис. 2.1.

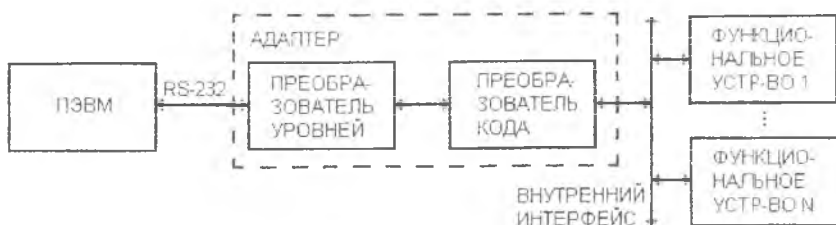


Рис. 2.1. Сопряжение ПЭВМ с функциональными устройствами

Связь с ПЭВМ осуществляется через адаптер, состоящий из преобразователя уровней и преобразователя кода. В зависимости от особенностей системы и функциональных устройств внутренний интерфейс может быть выполнен на базе параллельных или последовательных магистралей.

В качестве параллельных магистралей могут использоваться интерфейсы магистрально-модульных систем PC 104, Compact PCI, VME, MicroPC и т.п. В этом случае адаптер является одним из модулей используемого интерфейса.

При разработке нестандартных магистралей в качестве преобразователей кода можно использовать программируемые адаптеры последовательно интерфейса i8251A, i8250A, i16450, i16550A и др. Однако для этих микросхем требуется процессорное управление.

Возможна реализация преобразователя кода на универсальных сдвиговых регистрах, управление которыми осуществляется аппаратным формированием соответствующих уровней сигналов.

Наиболее целесообразно использование в качестве преобразователей кода однокристальных микроконтроллеров, имеющих асинхронный приемопередатчик и ПЗУ с flash-памятью. Достоинством таких микроконтроллеров является низкая стоимость, малые габариты (возможно использование микросхем с 8 линиями ввода-вывода), низкое энергопотребление.

В качестве преобразователей уровней следует выбирать микросхемы, для работы которых требуется один источник питания (3,5÷5В), а в корпусе содержатся как приемники, так и передатчики, например MAX-232, 235, 222, 242 и другие аналогичные микросхемы.

Адаптер RS-232 является принадлежностью практически всех последовательных магистралей I2C, USB, IEEE-1394, CAN и др. Наиболее простым вариантом расширения возможностей RS-232 является использование токовой петли, интерфейсов RS-422, RS-485.

В токовой петле электрическим сигналом является не уровень напряжения, а ток в двухпроводной схеме, соединяющей приемник и передатчик. Обычно логической единице соответствует ток (5÷20 мА), а логическому нулю - отсутствие тока. Токовая петля, как правило, использует гальваническую развязку входных цепей с помощью оптронов. Использование токовой петли позволяет увеличить длину линии при фиксированной скорости передачи, но требует применения двух проводов на каждый сигнал.

Лучшими характеристиками обладает дифференциальный интерфейс RS-422 (обеспечивающий работу одного передатчика и 10 приемников на скоростях до 10 Мбит/с) и его магистральный аналог RS-485 (обеспечивающий присоединение до 32 передатчиков и 32 приемников с пропускной способностью канала до 10 Мбит/с).

Серийно выпускаемые адаптеры RS-232 могут быть выполнены в конструктиве используемого параллельного интерфейса (ISA, CompactPCI, PC 104 т.п.) или в виде внешних автономных блоков.

Примером первого типа могут служить выполненные в стандарте ISA платы PCL 740 (одноканальный преобразователь RS-232/422/485-токовая петля), PCL 746+ (четырехканальный адаптер RS-232/422/485) и т.п.

В качестве примера автономных преобразователей можно привести устройства ADAM-4500 (RS-232/422/485), ADAM-4525 (RS-232/CAN) и т.п. Преобразователи ADAM имеют встроенные интеллектуальные средства и операционную систему ROM-DOS, совместимую с Ms-Dos.

В качестве адаптеров связи возможно использование и однокристалльных микроконтроллеров, в составе которых кроме асинхронного последовательного канала имеются порты, поддерживающие другие интерфейсы (I2C, SPI, CAN и т.п.).

Наиболее часто как средство расширения возможностей RS-232 используется RS-485. В простейшем случае согласование можно реализовать последовательным соединением двух преобразователей

уровней RS-232 \leftrightarrow TTL (например, MAX232) и TTL \leftrightarrow RS-485 (например, ADM485 или ADM1485).

Устройствами, совместимыми с интерфейсами RS-232, снабжены практически все вычислительные системы, включая программируемые логические контроллеры, ПЭВМ и мэйнфреймы. ПЭВМ на базе процессоров семейства i80x86 содержат от одного до четырех (обычно два) внешних устройств, поддерживающих "упрощенные" варианты интерфейса RS-232. "Упрощение" заключается в том, что в этих устройствах используются только сигналы DCD, RD, TD, DTR, SG, DSR, RTS, CTS и RI (см. табл. 1.2). Каждое из устройств укомплектовано разъемом типа DB9 или DB25 (см. рис. 1.1), во втором случае часть контактов разъема ни к чему не присоединена. Разъемы обычно размещаются на задней панели компьютера.

Эти устройства носят наименование *асинхронных адаптеров* или *COM-портов* и символически обозначаются как COM1..COM4. Порту COM1 соответствует также символическое обозначение AUX.

Для того чтобы процессор мог реагировать на аппаратные прерывания, вырабатываемые COM-портами, в контроллере прерываний компьютера предусмотрены два уровня приоритетов (табл. 2.1).

Таблица 2.1

Прерывания COM-портов

Порт	Уровень приоритета	Вектор прерывания
COM1 или COM3	IRQ4	0Ch
COM2 или COM4	IRQ3	0Bh

Если в компьютере присутствуют сразу три или четыре COM-порта, то может возникнуть ситуация неоднозначного распознавания источника прерывания. Решение проблемы обычно осуществляется чисто программными методами, например, последовательным опросом источников прерывания.

COM-порты компьютера служат для подключения манипуляторов "мышь" (устройство типа DTE, требуется "перекрестный" кабель), внешних модемов (устройство типа DCE, требуется "параллельный" кабель) и пр. Также они могут быть использованы для организации сетевой связи между двумя компьютерами

(устройствами типа DTE по минимальному варианту "перекрестного" кабеля, см. рис. Д4).

2.1. Связь двух ПЭВМ при помощи оболочки Norton Commander

Необходимым условием успешной связи является идентичность версий программы Norton Commander на обоих связываемых ПЭВМ. Дальнейшее изложение относится к Norton Commander v5.0.

Шаг 1. Запустите программу Norton Commander на обоих связываемых компьютерах. (Примечание. Возможен запуск из DOS-сессии Windows 9X, но не Windows NT).

Шаг 2. Нажмите клавишу F9, выберите ниспадающее меню "Левая" ("Left") или "Правая" ("Right"), а в нем пункт "Связь" ("Link").

Шаг 3. В появившемся окне установите параметры связи:

- номер COM-порта, к которому подсоединен кабель;
- одинаковые режимы работы портов.

Для одного компьютера выберите опцию "Ведущий" ("Master"), а для другого "Ведомый" ("Slave").

Шаг 4. Активизируйте на обоих компьютерах "Связь" ("Линк"). Это надо постараться сделать в течение одной минуты.

Когда связь установлена, то ведущий компьютер управляет функциями ведомого. Можно копировать и перемещать файлы из ведомого компьютера или на ведомый компьютер, переименовывать и удалять файлы, создавать и удалять каталоги, а также изменять атрибуты файлов. Однако нельзя просматривать дерево каталогов ведомого компьютера и запускать находящиеся на нем программы.

2.2. Связь двух ПЭВМ при помощи оболочки Dos Navigator

Запустив программу Dos Navigator, нажмите Alt-F11 на обоих компьютерах. Это надо постараться сделать в течение одной минуты. Когда связь установлена, то оба компьютера обладают равными правами доступа по отношению друг к другу. Можно копировать и перемещать файлы с одного компьютера на другой, переименовывать и удалять файлы, создавать и удалять каталоги, изменять атрибуты файлов, а также просматривать дерево каталогов другого компьютера и запускать находящиеся на нем программы.

2.3.Связь двух ПЭВМ при помощи MS Windows

Режим, обеспечивающий соединение двух компьютеров средствами Windows, носит наименование "Прямого кабельного соединения" ("Direct cable link"). Настройка и использование этого режима в различных версиях Windows имеет свои особенности.

2.3.1. Windows 95, 98 или ME

Необходимо выполнить следующую последовательность действий.

Шаг 1. Убедитесь в наличии следующих компонентов:

- "Программы" - "Стандартные" - "Связь" - "Прямое кабельное соединение";
- "Программы" - "Стандартные" - "Связь" - "Удаленный доступ к сети";
- "Сеть" - "Сетевая плата" - "Контроллер удаленного доступа";
- "Сеть" - "Протокол" - "Microsoft" - "TCP/IP";
- "Сеть" - "Клиент" - "Семейный вход в систему";
- "Сеть" - "Клиент" - "Клиент для сетей Microsoft";
- "Сеть" - "Службы" - "Служба доступа к файлам и принтерам сетей Microsoft".

Шаг 2. Убедитесь, что активизированы следующие опции:

- "Сеть" - "Доступ к файлам и принтерам" - "Файлы этого компьютера можно сделать общими";
- "Сеть" - "Доступ к файлам и принтерам" - "Принтеры этого компьютера можно сделать общими".

Шаг 3. Разрешите совместный доступ для требуемых ресурсов компьютера (дисков, каталогов и пр.), щелкнув на их пиктограмме правой клавишей мыши и активизировав опцию "Свойства" - "Доступ" - "Общий ресурс" в одном из следующих режимов:

- "Только чтение";
- "Доступ с паролем";
- "Полный доступ".

Шаг 4. Запустите на каждом из двух связываемых компьютеров "Пуск" - "Программы" - "Стандартные" - "Связь" - "Прямое кабельное соединение". Если прямое кабельное соединение выполняется впервые, то необходимо выбрать режим "Ведущий" или "Ведомый" и выполнить настройку на конкретный COM-порт.

Шаг 5. Выполните на ведущем компьютере поиск ведомого: "Пуск" - "Найти" - "Компьютер" - "Найти".

После установки прямого кабельного соединения ведущему компьютеру становятся доступны ресурсы ведомого (файлы, диски, принтеры, доступ к локальной сети и пр.).

2.3.2.Windows NT

Следующая процедура позволяет настроить прямое кабельное соединение в Windows NT 4.0.

Шаг 1. В "Панели управления" щелкните по пиктограмме "Модем". Вручную из списка "Standard modem types" ("Типы стандартных модемов") выберите и установите устройство "Dialup Networking Serial Cable between 2PCs" ("Нуль-модемный кабель для последовательных портов").

Шаг 2. В "Панели управления" щелкните по пиктограмме "Network" ("Сеть"), переместитесь на вкладку "Adapters" ("Адаптеры") и выберите устройство "MS Loopback Adapter". Затем на запрос типа фрейма введите "802.3" и укажите требуемый порт (COM1 или COM2).

Шаг 3. На ведомом компьютере запустите службу удаленного доступа (RAS - Remote Access Service), и ведомый компьютер перейдет в состояние ожидания подключений. На ведущем компьютере потребуется создать новое соединение с использованием установленного устройства.

2.3.3.Windows 2000 или XP

Прямое кабельное соединение в операционных системах этой группы настраивается следующим образом.

Шаг 1. Загрузите операционную систему с правами администратора и создайте новое подключение, нажав "Пуск" - "Настройка" - "Сеть и удаленный доступ к сети" - "Создание нового подключения" - "Прямое подключение к другому компьютеру через последовательный, параллельный или инфракрасный порт", выбрав порт и назначив режим работы - ведущий или ведомый.

Шаг 2. Ведомый компьютер автоматически переходит в состояние ожидания подключений сразу после создания нового подключения и перезагрузки. На ведущем компьютере потребуется запустить прямое кабельное соединение с использованием

созданного на предыдущем шаге подключения: "Пуск"- "Настройка"- "Сеть и удаленный доступ к сети" и т.п.

2.3.4. Связь Windows 9X с Windows NT/2000/XP.

При попытке установить прямое кабельное соединение между компьютерами с операционными системами двух разных классов часто возникает ситуация несоответствия протоколов аутентификации (протоколов проверки паролей).

Если компьютер, работающий под управлением Windows NT/2000/XP, является ведущим, то для согласования протоколов щелкните правой кнопкой мыши по пиктограмме прямого кабельного соединения и в панели "Свойства" выберите закладку "Безопасность". Далее выберите "Дополнительные (особые параметры)", нажмите кнопку "Настройка" и установите на этой панели параметры в соответствии с рис. 2.2.

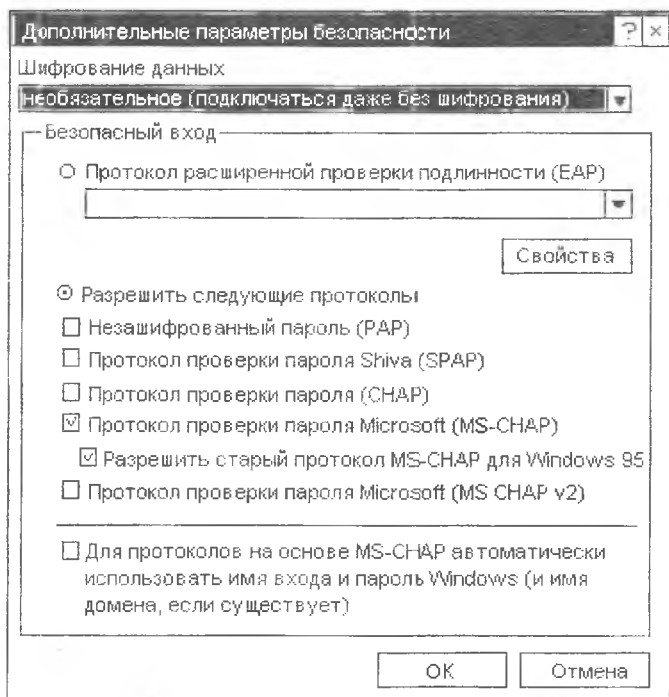


Рис. 2.2. Настройка протокола аутентификации в Windows 2000

Если компьютер, работающий под управлением Windows NT/2000/XP, является ведомым, то щелкните правой кнопкой мыши по пиктограмме "Входящие подключения", в открывшемся контекстном меню выберите пункт "Свойства" и перейдите к закладке "Пользователи". Теперь необходимо поставить галочки против идентификаторов тех пользователей, которым разрешен удаленный вход в систему. Удостоверьтесь также, что сброшена галочка "Все пользователи должны держать в секрете свои пароли и данные" и установлена галочка "Всегда разрешать подключение без пароля устройствам с прямым соединением, таким, как карманные компьютеры (Palmtop PC)".

Еще одной особенностью прямого кабельного соединения компьютеров с Windows 2000 является ошибка, приводящая к разрыву связи через одну минуту отсутствия обмена. Эта ошибка исправляется установкой пакета обновления Service Pack 2 или запуском командного файла, производящего выполнение команды ping <IP-адрес> в бесконечном цикле.

3. Программирование устройств на основе RS-232

3.1. Программирование COM-портов средствами процедур BIOS

В любой ПЭВМ на базе процессоров Intel обязательно присутствует BIOS (базовая система ввода-вывода) - стандартный набор программных процедур, предназначенных для обслуживания внешних устройств, таких как жесткие и гибкие диски, клавиатура, порты параллельной и последовательной передачи данных и т.п. Интерфейс доступа к этим процедурам стандартизован и основан на использовании программных прерываний с передачей параметров через регистры процессора. Для программирования COM-портов используется программное прерывание 14h. Код операции передается через 8-битовый регистр AH (см. табл. 3.2). Код номера порта задается в регистре DX: 0-COM1, 1-COM2 и т.д.

Функции прерывания 14h

Функция	Содержимое регистров при вызове прерывания	Значения, возвращаемые в регистрах
Инициализация порта	АН=00h; AL – параметры инициализации порта (см. табл. 3.3) DX - номер порта	АН - состояние порта (см. табл. 3.4) AL – состояние модема (см. табл. 3.5)
Передача байта	АН=01h AL - данные DX - номер порта (0-3)	АН – бит 7 сброшен в 0 при успехе и установлен в 1 при ошибке; биты 0-6 - состояние порта
Прием байта	АН - 02h DX - номер порта (0-3)	АН – состояние порта AL – принятые данные
Получение состояния порта	АН=03h DX- номер порта (0-3)	АН – состояние порта AL – состояние модема

Таблица 3.3

Параметры инициализации порта

Биты	Назначение
0-1	Длина блока данных: 00 – 5 битов, 01 – 6 битов, 10 – 7 битов, 11 – 8 битов
2	Количество стоповых битов: 0 – 1 бит, 1 – 2 бита
3-4	Тип проверки на четность: x0 – проверка отсутствует, 01 – проверка на нечетность, 11 – проверка на четность
5-7	Скорость передачи данных: 000 – 110 бод, 001 – 150 бод, 010 – 300 бод, 011-600 бод, 100 – 1200 бод, 101 – 2400 бод, 110 – 4800 бод, 111 – 9600 бод

Таблица 3.4

Состояние порта

Бит	Назначение (если соответствующий бит установлен в 1)
0	В буфере чтения есть / появились новые данные
1	Произошла ошибка переполнения (часть непрочитанных данных утеряна)
2	Произошла ошибка четности (прочитанные данные искажены)
3	Произошла ошибка синхронизации (разрушен формат кадра)
4	Обнаружено состояние BREAK (разрыв связи)
5	Буферный регистр передатчика пуст
6	Сдвиговый регистр передатчика пуст
7	Тайм-аут

Состояние модема

Бит	Назначение (если соответствующий бит установлен в 1)
0	Линия CTS изменила свое состояние
1	Линия DSR изменила свое состояние
2	Линия RI изменила свое состояние
3	Линия DCD изменила свое состояние
4	Сброс для передачи (состояние линии CTS)
5	Запрос для передачи (состояние линии DSR)
6	Обнаружен звонок (состояние линии RI)
7	Обнаружена несущая (состояние линии DCD)

Основным недостатком методов программирования COM-портов средствами процедур BIOS является невозможность использования скоростей передачи данных выше 9600 бод и режима фиксации четности. Кроме того, выполнение процедур ROM BIOS возможно только при работе процессора в реальном режиме.

3.2. Прямое программирование контроллера COM-портов

Наиболее полно использовать возможности передачи данных через COM-порты ПЭВМ позволяет прямое программирование их контроллера, осуществляемое через порты ввода-вывода. По умолчанию за каждым COM-портом в адресном пространстве ввода-вывода закреплена группа из 8 портов с последовательно расположенными адресами (см. табл. 3.6 и 3.7).

Таблица 3.6

Базовые адреса портов ввода-вывода

COM-порт	Начальный адрес области (BASE)
COM1	3F8h
COM2	2F8h
COM3	3E8h
COM4	2E8h

Регистры контроллера

Относительный адрес	Назначение
BASE	Регистр данных передатчика и приемника
BASE+1	Регистр управления прерываниями
BASE+2	Регистр идентификации прерываний
BASE+3	Управляющий регистр
BASE+4	Регистр управления модемом
BASE+5	Регистр состояния линии
BASE+6	Регистр состояния модема
BASE+7	Регистр расширения

3.2.1. Инициализация COM-порта

Как отмечалось выше, для организации асинхронного обмена данными между двумя устройствами RS-232 они должны поддерживать один и тот же режим передачи данных, характеризующийся одинаковой скоростью передачи-приема и одинаковым форматом блока данных.

По умолчанию для COM-портов ПЭВМ устанавливаются следующие параметры: скорость передачи - 2400 бод; проверка на четность не выполняется; количество стоповых битов - 1; длина передаваемого символа - 8 битов. Для того чтобы изменить эти параметры, необходимо выполнить следующие действия.

Шаг 1 - сброс COM-порта в исходное состояние. Это действие не является обязательным и рекомендуется только в тех случаях, если COM-порт ранее был уже проинициализирован какой-нибудь другой программой (например, драйвером мыши). Необходимо записать 0 в *регистр разрешения прерываний* с адресом BASE+1 (запретив тем самым прерывания) и *регистр идентификации прерываний* с адресом BASE+2 (запретив тем самым буферизацию). Затем следует однократно прочитать регистры с адресами BASE+3, BASE+4 и BASE+5, эта операция сбросит возможно оставшиеся от предыдущей работы с устройством битовые признаки.

Шаг 2 - установка скорости передачи данных. Скорость передачи данных кодируется в соответствии с табл. 3.8. Младший байт этого значения помещается по адресу BASE, старший - по адресу BASE+1, причем предварительно необходимо установить особый

режим доступа к этим регистрам, записав в порт $BASE+3$ значение с единицей в старшем бите.

Шаг 3 - установка формата блока данных. Формат блока данных кодируется отдельными битами и группами битов регистра управления, расположенного по адресу $BASE+3$ (см. табл. Г3). Этот регистр доступен как для чтения (если необходимо определить текущий формат блока данных), так и для записи (если необходимо этот формат изменить). Старшим битом этого регистра кодируется способ доступа к регистрам $BASE$ и $BASE+1$, поэтому при установке формата блока данных этот бит должен быть равен 0.

Таблица 3.8

Кодировка скорости передачи

Скорость, бит/с	16-ричный код	Скорость, бит/с	16-ричный код	Скорость, бит/с	16-ричный код
110	410	1200	60	19200	6
150	300	2400	30	38400	3
300	180	4800	18	57600	2
600	00	9600	0C	115200	1

3.2.2. Обмен данными при помощи опроса битов готовности

Передача данных. Перед записью символа в регистр данных COM-порта, расположенный по адресу $BASE$, необходимо убедиться, что он готов для передачи. Признаком готовности является установленный в единицу бит 5 регистра состояния линии с адресом $BASE+5$ (см. табл. Г5).

Прием данных. Перед чтением символа из регистра данных COM-порта, расположенного по адресу $BASE$, необходимо убедиться, что этот символ получен и готов для считывания. Признаком готовности является установленный в единицу бит 0 регистра состояния линии с адресом $BASE+5$ (см. табл. Г5).

Достоинством описанных способов передачи данных является простота реализации, недостатком - необходимость постоянного опроса битов готовности. Пример приведен в прил. А.

3.2.3. Обмен данными с использованием прерываний

Для установки режима работы СОМ-порта с использованием прерываний необходимо в процессе инициализации записать требуемое значение в регистр с адресом $BASE+i$ (см. табл. Г1). Кроме того, необходимо размаскировать требуемые прерывания в регистре маски системного контроллера прерываний с адресом $21h$ (бит 3 отвечает за $IRQ3$, а бит 4 - за $IRQ4$; нулевое значение бита означает разрешение соответствующего прерывания). Также необходимо предусмотреть наличие специальной процедуры обработки возникающих прерываний. Эта процедура должна завершаться посылкой значения $20h$ в управляющий регистр системного контроллера прерываний с адресом $20h$.

Пример простой программы, принимающей символы из СОМ-порта в режиме прерывания, приведен в прил. Б.

Поскольку при обмене данными через СОМ-порт возможны различные причины возникновения прерывания, то для их определения возможно использование *регистра идентификации прерывания* с адресом $BASE+2$ (см. табл. Г2): бит 0 сигнализирует о возникновении какого-либо прерывания, а биты 1-2 несут информацию о причине его возникновения.

Также может оказаться целесообразным управление буферизацией данных, т.е. процессом накопления посылаемых и принимаемых символов в промежуточном буфере. Управление производится через биты 0-2 и 6-7 *регистра идентификации прерывания* с адресом $BASE+2$ (см. табл. Г2). Установка ненулевой длины буфера вызывает возникновение прерывания не сразу при передаче или приеме символа, но лишь при заполнении буфера.

Примечание. При использовании "заглушек" (см. рис. Д5 и Д6) эта особенность может быть использована для генерации циклических прерываний в ПЭВМ с нужным периодом.

3.3. Программирование СОМ-портов в MS Windows

Прямое программирование контроллеров внешних устройств в операционных системах семейства MS Windows сопряжено с трудностями, поскольку непосредственный доступ пользова-

тельских программ к портам ввода-вывода противоречит требованиям безопасности и надежности системы. Эта проблема решается обращением к специализированным компонентам операционной системы - к *драйверам виртуальных устройств*. Для стандартных внешних устройств ПЭВМ (COM-портов, LPT-портов, таймера, часов реального времени и т.п.) доступ к драйверам выполняется через функции WIN32 API и поэтому средствами любой системы программирования - Borland Delphi, Borland C/C++ Builder, MS Visual Basic, MS Visual C/C++ и пр.

3.3.1. Открытие порта.

Необходимо перед первым обращением к COM-порту и выполняется при помощи универсальной API-функции `CreateFile()`:

```
HANDLE CreateFile(  
LPCTSTR lpFileName, // Имя файла или устройства  
DWORD dwDesiredAccess, // Тип доступа  
DWORD dwShareMode, // Права разделяемого доступа  
LPSECURITY_ATTRIBUTES lpSecurityAttributes, // Атрибуты защиты  
DWORD dwCreationDisposition, // Режимы обращения  
DWORD dwFlagsAndAttributes, // Атрибуты файла или устройства  
HANDLE hTemplateFile // Описатель шаблона );
```

Рекомендуемые значения параметров функции:

- `dwDesiredAccess` - `GENERIC_READ|GENERIC_WRITE` с числовым значением `C0000000h`;
- `dwShareMode` - `0`;
- `lpSecurityAttributes` - `NULL`;
- `dwCreationDisposition` - `OPEN_EXISTING` с числовым значением `3`;
- `dwFlagsAndAttributes` - `FILE_FLAG_OVERLAPPED` с числовым значением `40000000h`;
- `hTemplateFile` - `NULL`.

Функция `CreateFile()` при успешной инициализации возвращает уникальный идентификатор открытого порта, который в дальнейшем необходимо использовать в качестве параметра для операций настройки порта и ввода-вывода.

3.3.2. Настройка параметров передачи.

Это действие выполняется при помощи функций **SetCommState()** (установка формата кадра и скоростей передачи) и **SetCommTimeouts()** (установка временных параметров процесса передачи).

```
BOOL SetCommState(  
HANDLE hCommDev, // Идентификатор порта  
LPDCB lpdcB // Адрес блока параметров);
```

Параметры настройки передачи данных передаются функции **SetCommState()** в области памяти со структурой, соответствующей типу DCB. Для того чтобы заполнить эту область памяти корректными значениями, можно использовать сервисную функцию **BuildCommDCB()**:

```
BOOL BuildCommDCB(  
LPCTSTR lpDef, // Идентификатор порта  
LPDCB lpDCB // Адрес блока параметров );
```

Этой функции параметры настройки передаются в виде текстовой строки, например:

```
DCB d;  
d.DCBlength = sizeof(DCB); // Размер структуры  
BuildCommDCB ("baud=9600 parity=N data=8 stop=1", &d);
```

3.3.3. Операции ввода-вывода.

Осуществляются стандартными функциями **WriteFile()** и **ReadFile ()**:

```
BOOL WriteFile(  
HANDLE hFile, // Идентификатор файла или порта  
LPCVOID lpBuffer, // Адрес буфера данных  
DWORD nNumberOfBytesToWrite, // Количество байтов для записи  
LPDWORD lpNumberOfBytesWritten, // Реальное количество байтов  
LPOVERLAPPED lpOverlapped // Буфер перекрытия );
```

```

BOOL ReadFile(
HANDLE hFile, // Идентификатор файла или порта
LPVOID lpBuffer, // Адрес буфера данных
DWORD nNumberOfBytesToRead, // Количество данных для чтения
LPDWORD lpNumberOfBytesRead, // Реальное количество байтов
LPOVERLAPPED lpOverlapped // Буфер перекрытия );

```

В общем случае рекомендуемое значение для параметра **lpOverlapped** есть **NULL**.

По умолчанию функции **WriteFile()** и **ReadFile()** ожидают завершения операций ввода-вывода бесконечное время. Установить конечное время тайм-аута позволяет функция **SetCommTimeouts()**:

```

BOOL SetCommTimeouts(
HANDLE hCommDev, // Идентификатор файла или порта
LPCOMMTIMEOUTS lpctmo // Адрес блока параметров );

```

Параметры настройки динамических параметров передачи передаются функции **SetCommTimeouts()** в области памяти, структура которой соответствует типу **COMMTIMEOUTS**.

3.3.4. Закрывание порта.

Необходимо после окончания работы с COM-портом. Выполняется при помощи функции **CloseHandle()**:

```

BOOL CloseHandle(
HANDLE hObject // Идентификатор открытого файла или порта);

```

3.3.5. Прочие операции.

Они могут понадобиться при программировании COM-портов в среде Windows, можно найти в справочнике по функциям Win32 API. Среди них отметим:

- **SetupComm()** - устанавливает размеры внутреннего буфера ввода-вывода для драйвера COM-порта;
- **PurgeComm()** - очищает буфер ввода-вывода от "мусора";
- **FlushFileBuffers()** - инициирует немедленную передачу содержимого буфера в устройство.

Пример программирования COM-портов в среде Windows приведен в прил. В.

Рекомендуемая литература

- 1 Фролов А.В., Фролов Г.В. Программирование модемов. - М.: Диалог-МИФИ, 1994. - 240 с.
- 2 Новиков Ю.В., Калашников О.А., Гуляев С.Э. Разработка устройств сопряжения для персонального компьютера типа IBM PC. - М.: ЭКОМ, 2002. - 224 с.
- 3 Касперски К. Укрощение Интернета. - М.: СОЛОН-Р, 2002. - 288 с.

Приложение А. Библиотека процедур, использующих опрос битов готовности

base=3F8h

; Чтение байта. Результат возвращается в регистре AL
rd_com:

```
mov     cx, 0FFFFh; Кол-во попыток
@1:    mov     dx, base+5
        in     al, dx
        test  al, 00000001b ; Порт готов?
        jnz  @2
        loop @1
        stc                    ; Признак неудачи
        ret
```

```
@2:    mov     dx, base
        in     al, dx ; Чтение байта
        cld                    ; Признак успеха
        ret
```

; Запись байта. Данные передаются через регистр AH
wr_com:

```
mov     cx, 0FFFFh ; Кол-во попыток
@3:    mov     dx, base+5
        in     al, dx
        test  al, 00100000b ; Порт готов?
        jnz  @4
        loop @3
        stc                    ; Признак неудачи
        ret
```

```
@4:    mov     al, ah
        mov     dx, base
        out    dx, al ; Запись байта
        cld                    ; Признак успеха
        ret
```

Приложение Б. Программа, использующая прерывания

```
(* -- Прием данных из COM1 с использованием прерываний -- *)
*) uses Dos;

const COM1=$3F8;
var c : byte; p : pointer;

(* ----- Процедура обработки прерывания 0Ch ----- *)
procedure getbyt; interrupt;
begin
  c := port[COM1];
  ...
  port[$20] := $20;
end;

begin
(* ----- Перехват прерывания 0Ch ----- *)
  getintvec($0C, p);
  setintvec($0C, @getbyt);
(* ----- Сброс порта в исходное состояние ----- *)
  port[COM1+1]:=0; (* Запрет прерываний *)
  port[COM1+2]:=0; (* Запрет буферизации *)
  c := port[COM1+3];
  c := port[COM1+4];
  c := port[COM1+5];
(* ----- Установка скорости обмена ----- *)
  port[COM1+3] := $80;
  port[COM1+0] := $C0; (* Младший байт *)
  port[COM1+1] := 0; (* Старший байт *)
(* ----- Установка формата блока данных ----- *)
  port[COM1+3] := 3; (* d=8, s=1, p=no *)
(* -- Разрешение прерывания при получении символа -- *)
  port[COM1+1]:=1;
(* -- Размаскирование IRQ4 в контроллере прерываний -- *)
  c := port[$21];
  c := c and $EF;
```

```

port[$21] := c;
(* ----- Цикл обработки данных ----- *)
while not FINISH do begin
  ...
end;
(* - Восстановление старого обработчика прерывания 0Ch
-- *)
setintvec($0C, p1);
end.

```

Приложение В. Пример программирования COM-порта в среде Windows

```

/* --- Тестирование портов при помощи "заглушки" --- */
#include <windows.h>
#include <stdio.h>
#include <mem.h>

HANDLE p; DCB d; COMMTIMEOUTS c;
char ch, buf[]={"Samara State Aerospace University"};
int q, i=0; unsigned long n;

main() {
/* ----- Открыть устройство ----- */
p=CreateFile("COM2", GENERIC_READ|GENERIC_WRITE,
            0, NULL, OPEN_EXISTING, 0, NULL);
/* ----- Установить формат кадра ----- */
d.DCBlength=sizeof(DCB); d.fNull=TRUE;
BuildCommDCB("baud=1200 parity=N data=8 stop=1",&d);
SetCommState(p, &d);
/* --- Установить тайм-ауты (в миллисекундах) --- */
setmem( &c, sizeof(c), 0); /* Обнуление остальных полей */
c.ReadIntervalTimeout=100; /* Тайм-аут для чтения */
SetCommTimeouts(p, &c);
/* ----- Передать/принять строку данных ----- */
while (buf[i]) {
WriteFile(p, &(buf[i+1]), 1, &n, NULL);
q=ReadFile(p, &ch, 1, &n, NULL);
}
}

```

```

    putchar (ch);
}
/* ----- Закрыть устройство ----- */
CloseHandle(p);

```

Приложение Г. Регистры контроллера COM-портов

Таблица Г1

Регистр управления прерываниями (BASE+1)

Бит	Назначение
0	1 – разрешение прерывания при готовности принимаемых данных
1	1 – разрешение прерывания после успешной передачи символа
2	1 – разрешение передачи при разрыве связи или при ошибке
3	1 – разрешение прерывания при изменении одного из сигналов: CTS, DSR, RI, DCD
4-7	Не используются

Таблица Г2

Регистр идентификации прерываний и управления буферизацией (BASE+2)

Бит	Назначение
0	При чтении регистра: 1 – отсутствие прерываний. При записи в регистр: 1 – разрешение буферизации.
1-2	При чтении регистра – причина возникшего прерывания: 00 – произошло прерывание при изменении состояния одного из сигналов: CTS, DSR, RI, DCD; 01 – произошла успешная передача символа; 10 – данные приняты и доступны для чтения; 11 – произошло прерывание при переполнении приемника, при ошибке четности, при ошибке синхронизации или при разрыве связи. При записи в регистр – сброс содержимого буферов: 01 – приемного буфера; 10 – передающего буфера; 11 – обоих буферов.
3	При чтении: 1 – буфер не полон, но в нем есть данные; При записи: 1 – разрешение использования ПДП (в ПЭВМ не реализовано)
4-5	Не используются
6-7	При записи в регистр – управление длиной буфера: 00 – 1 байт; 01 – 4 байта; 10 – 8 байтов; 11 – 14 байтов.

Регистр управления (BASE+3)

Бит	Назначение
0-1	Длина передаваемого символа в битах: 00 - 5 битов; 01 - 6 битов; 10 - 7 битов; 11 - 8 битов.
2	Количество стоповых битов: 0 - 1 бит; 1 - 2 бита.
3-4	Режим контроля четности: 00 - не выполняется; 01 - проверка на нечетность; 11 - проверка на четность.
5	При установке этого бита в единицу бит четности всегда равен 0, если установлен режим "проверка на четность", и 1, если установлен режим "проверка на нечетность"
6	Иницирует вывод строки нулей как признака разрыва связи
7	Управление режимом доступа к регистрам данных (BASE+0) и управления прерываниями (BASE+1): 0 - регистры используются как обычно; 1 - регистры используются для задания скорости передачи данных.

Таблица Г4

Регистр управления модемом MCR (BASE+4)

Бит	Назначение
0	Состояние линии DTR
1	Состояние линии RTS
2	Состояние линии OUT1 (запасной)
3	Состояние линии OUT2 (запасной)
4	Запуск самодиагностики порта с использованием "заглушки"
5	0
6	0
7	0

Таблица Г5

Регистр состояния линии (BASE+5)

Бит	Назначение
0	1 - получен и готов для чтения новый символ
1	1 - произошла ошибка переполнения входного буфера, данные частично утеряны
2	1 - произошла ошибка четности, полученные данные искажены
3	1 - произошла ошибка синхронизации, режимы передачи и приема не совпадают
4	1 - обнаружен запрос на прерывание
5	1 - порт готов для передачи очередного символа

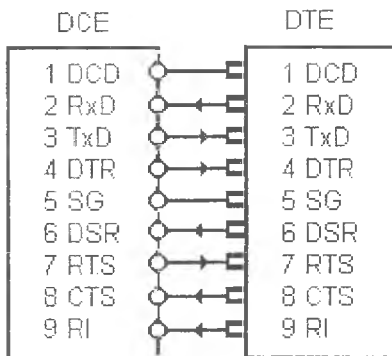
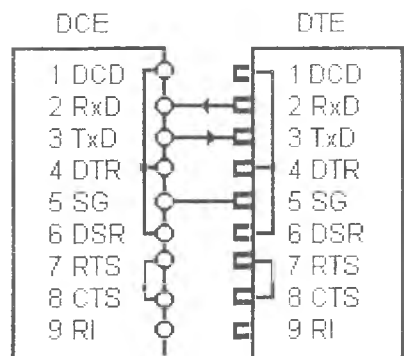
Бит	Назначение
6	0 - идет побитная передача символа; 1 - передача символа завершена
7	1 - тайм-аут, устройство не связано с компьютером

Таблица Г6

Регистр состояния модема MSR (BASE+6)

Бит	Назначение
0	Линия CTS изменила состояние
1	Линия DTS изменила состояние
2	Линия RI изменила состояние
3	Линия DCD изменила состояние
4	Состояние линии CTS (разрешение на посылку)
5	Состояние линии DSR (данные готовы)
6	Состояние линии RI (звонок)
7	Состояние линии DCD (нсущая)

Приложение Д. Варианты распайки кабелей и заглушек

Рис. Д1 Полный кабель
DTE-DCE ("параллельный")Рис. Д2 Минимальный кабель
DTE-DCE ("параллельный")

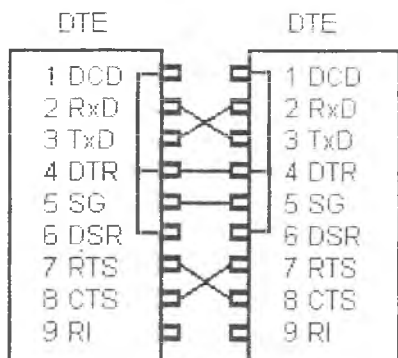


Рис. Д3 Полный нуль-модемный ("перекрестный") кабель DTE-DTE

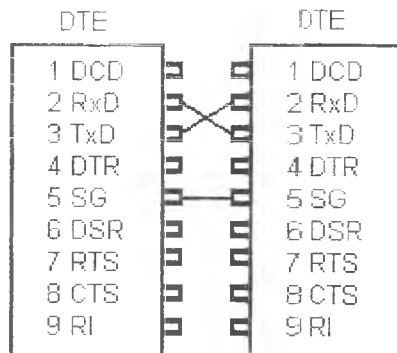


Рис. Д4 Минимальный нуль-модемный ("перекрестный") кабель DTE-DTE

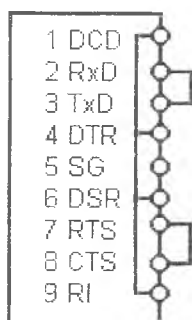


Рис. Д5 Полная заглушка

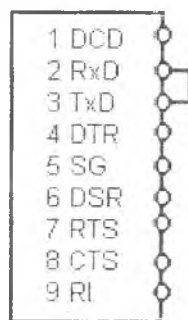


Рис. Д6 Минимальная заглушка

Содержание

Введение.....	3
1. Общая характеристика.....	3
1.1. Особенности использования сигналов интерфейса.....	5
1.2. Синхронизация передачи потоков данных.....	7
2. Применение RS-232.....	8
2.1. Связь двух ПЭВМ при помощи оболочки Norton Commander.....	12
2.2. Связь двух ПЭВМ при помощи оболочки Dos Navigator.....	12
2.3. Связь двух ПЭВМ при помощи MS Windows.....	13
2.3.1. Windows 95, 98 или ME.....	13
2.3.2. Windows NT.....	14
2.3.3. Windows 2000 или XP.....	14
2.3.4. Связь Windows 9X с Windows NT/2000/XP.....	15
3. Программирование устройств на основе RS-232.....	16
3.1. Программирование COM-портов средствами процедур BIOS.....	16
3.2. Прямое программирование контроллера COM-портов.....	18
3.2.1. Инициализация COM-порта.....	19
3.2.2. Обмен данными при помощи «спроса битов готовности».....	20
3.2.3. Обмен данными с использованием прерываний.....	21
3.3. Программирование COM-портов в MS Windows.....	21
3.3.1. Открытие порта.....	22
3.3.2. Настройка параметров передачи.....	23
3.3.3. Операции ввода-вывода.....	23
3.3.4. Закрытие порта.....	24
3.3.5. Прочие операции.....	24
Рекомендуемая литература.....	25
Приложение А. Библиотека процедур, использующих опрос битов готовности.....	26
Приложение Б. Программа, использующая прерывания.....	27
Приложение В. Пример программирования COM-порта в среде Windows.....	28

Приложение Г. Регистры контроллера СОМ-портов.....	29
Приложение Д. Варианты распайки кабелей и заглушек.....	31

Учебное издание

ИНТЕРФЕЙС RS-232 В КАНАЛАХ ПОСЛЕДОВАТЕЛЬНОЙ ПЕРЕДАЧИ ДАнных

Методические указания

Составители: *Владислав Германович Иоффе*
Константин Евгеньевич Климентьев

Редактор Т. К. Крестина
Компьютерная верстка О. А. Ананьев

Подписано в печать 16.11.04. Формат 60×84 1/16.

Бумага офсетная. Печать офсетная.

Усл. печл. 2,0. Усл. кр.-отг. 2,12. Уч.-издл 2,25.

Тираж 100 экз. Заказ № . Арт. С-52/2004.

Самарский государственный аэрокосмический
университет. 443086 Самара, Московское шоссе. 34.

РИО Самарского государственного аэрокосмического
университета. 443086 Самара, Московское шоссе, 34.