

Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

**Методические указания
по выполнению курсовой работы по дисциплине
«Сетевые технологии в автоматизированных системах обработки
информации и управления»**

Разработали:

к.т.н., доцент Галкин В.А.

ст. преподаватель Антонов А. И.

Москва - 2026

Аннотация

Данная работа предназначена для обучения студентов основам проектирования и построения простейших локальных сетей на базе интерфейса RS-232. В теоретических разделах авторы подробно изложили материал, необходимый для подготовки студентов к выполнению курсовой работы. Рассмотрены теоретические основы передачи данных, описание протокола RS-232. Описаны функции физического, канального и прикладного уровней. В качестве заданий студентам предлагается применить знания сетевых технологий на практике, разработав систему передачи данных между несколькими устройствами по протоколу RS-232. Студентам необходимо совместно реализовать проект на физическом, канальном и прикладном уровне, обеспечив их взаимодействие между собой, разработать пакет программной документации, а также оформить пояснительные листы.

Содержание

1. Описание функций физического уровня.	3
1.1. Сигналы интерфейса RS-232-C	3
1.2. Нуль-модемный интерфейс.	7
1.2.1. Настройка COM-порта средствами библиотеки javax.comm.	9
1.2.3. Описание класса SerialPort.	9
1.2.4. Класс SerialPortEvent.	11
2. Описание функций канального уровня.	12
2.1. Протокол связи.	12
2.2. Защита передаваемой информации.	12
2.3. Примеры форматов кадров.	13
2.3.1. Информационный I - Кадр.	13
2.3.2. Супервизорный Link - Кадр.	14
2.3.3. Супервизорный Uplink - Кадр.	14
2.3.4. Супервизорный АСК - Кадр.	14
2.3.5. Супервизорный Ret - Кадр.	14
3. Описание функций прикладной уровень.	15
3.1. Пример окна настройки подключения	15
3.2. Пример окна главного меню.	16
3.3. Пример окна личных сообщений.	17
3.4. Пример окна истории сообщений	17
3.5. Взаимодействие с канальным уровнем	18
4. Варианты заданий.	21
5. Литература.	34

1. Описание функций физического уровня.

1.1. Сигналы интерфейса RS-232-C.

Последовательная передача данных означает, что данные передаются по единственной линии. При этом биты байта данных передаются по очереди с использованием одного провода. Для синхронизации группе битов данных обычно предшествует специальный *стартовый бит*, после группы битов следуют *бит проверки на четность* и один или два *стоповых бита*, как показано на рисунке 1. Иногда бит проверки на четность может отсутствовать.

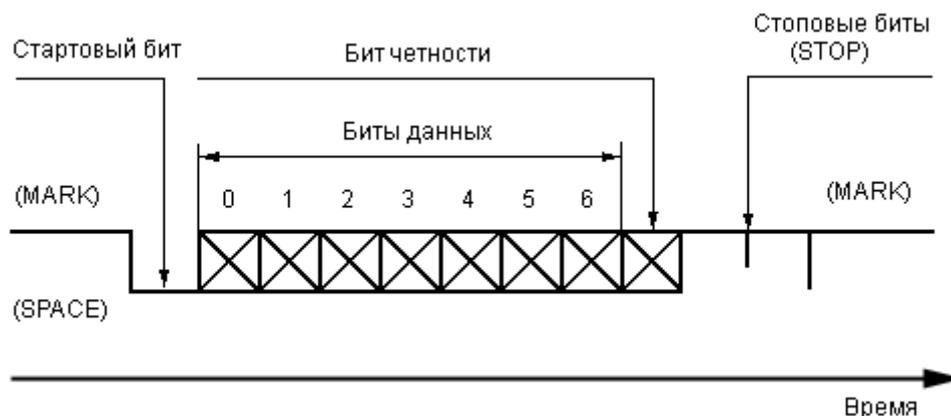


Рисунок 1.

Из рисунка видно, что исходное состояние линии последовательной передачи данных - уровень логической 1. Это состояние линии называют отмеченным — **MARK**. Когда начинается передача данных, уровень линии переходит в 0. Это состояние линии называют пустым — **SPACE**. Если линия находится в таком состоянии больше определенного времени, считается, что линия перешла в состояние разрыва связи — **BREAK**.

Стартовый бит **START** сигнализирует о начале передачи данных. Далее передаются биты данных, вначале младшие, затем старшие.

Контрольный бит формируется на основе правила, которое создается при настройке передающего и принимающего устройства. Контрольный бит может

быть установлен с контролем на четность, нечетность, иметь постоянное значение 1 либо отсутствовать совсем.

Если используется бит четности **P**, то передается и он. Бит четности имеет такое значение, чтобы в пакете битов общее количество единиц (или нулей) было четно или нечетно, в зависимости от установки регистров порта. Этот бит служит для обнаружения ошибок, которые могут возникнуть при передаче данных из-за помех на линии. Приемное устройство заново вычисляет четность данных и сравнивает результат с принятым битом четности. Если четность не совпала, то считается, что данные переданы с ошибкой. Конечно, такой алгоритм не дает стопроцентной гарантии обнаружения ошибок. Так, если при передаче данных изменилось четное число битов, то четность сохраняется, и ошибка не будет обнаружена. Поэтому на практике применяют более сложные методы обнаружения ошибок.

В самом конце передаются один или два стоповых бита **STOP**, завершающих передачу байта. Затем до прихода следующего стартового бита линия снова переходит в состояние **MARK**.

Использование бита четности, стартовых и стоповых битов определяют формат передачи данных. Очевидно, что передатчик и приемник должны использовать один и тот же формат данных, иначе обмен будет невозможен.

Другая важная характеристика — скорость передачи. Она также должна быть одинаковой для передатчика и приемника.

Скорость изменения информативного параметра сигнала обычно измеряется в бодах.

Иногда используется другой термин — биты в секунду (bps). Здесь имеется в виду эффективная скорость передачи данных, без учета служебных битов.

Интерфейс RS232C описывает несимметричный интерфейс, работающий в режиме последовательного обмена двоичными данными. Интерфейс поддерживает как асинхронный, так и синхронный режимы работы.

Интерфейс называется несимметричным, если для всех цепей обмена интерфейса используется один общий возвратный провод — сигнальная «земля».

Интерфейсы 25-ти (DB25) или 9-ти (DB9) контактный разъем.

Наименование сигнала	Цепь	Номер контакта	
		DB25P	DB9S
DCD (Data Carrier Detect)	109	8	1
RD (Receive Data)	104	3	2
TD (Transmit Data)	103	2	3
DTR (Data Terminal Ready)	108	20	4
	102	7	5
GND (Signal Ground)	107	6	6
DSR (Data Set Ready)	105	4	7
RTS (Request To Send)	106	5	8
CTS (Clear To Send)	125	22	9
RI (Ring Indicator)			

В интерфейсе реализован биполярный потенциальный код на линиях между DTE и DCE. Напряжения сигналов в цепях обмена симметричны по отношению к уровню сигнальной «земли» и составляют не менее +3В для двоичного нуля и не более -3В для двоичной единицы.

Входы TD и RD используются устройствами DTE и DCE по-разному. DTE использует вход TD для передачи данных, а вход RD для приема данных. И наоборот, устройство DCE использует вход TD для приема, а вход RD для

передачи данных. Поэтому для соединения двух DTE необходимо перекрестное соединение линий TD и RD в нуль-модемном кабеле.

Рассмотрим самый низкий уровень управления связью - подтверждение связи.

В начале сеанса связи компьютер (DTE) должен удостовериться, что модем (DCE) находится в рабочем состоянии. Для этой цели компьютер путем открытия COM-порта подает сигнал по линии DTR. В ответ модем подает сигнал по линии DSR. Затем, после вызова абонента, модем подает сигнал по линии DCD, чтобы сообщить компьютеру, что он произвел соединение с удаленной системой.

Более высокий уровень используется для управления потоком (скоростью обмена данными) и также реализуется аппаратно. Этот уровень необходим для того, чтобы предотвратить передачу большего числа данных, чем то, которое может быть обработано принимающей системой.

В полудуплексных соединениях DTE подает сигнал RTS, когда оно желает передать данные. DCE отвечает сигналом по линии CTS, когда оно готово, и DTE начинает передачу данных. До тех пор, пока оба сигнала RTS и CTS не примут активное состояние, только DCE может передавать данные. Иногда для соединения двух устройств DTE эти линии (RTS и CTS) соединяются вместе на каждом конце кабеля. В результате получаем то, что другое устройство всегда готово для получения данных (если при большой скорости передачи принимающее устройство не успевает принимать и обрабатывать данные, возможна потеря данных).

Для решения всех этих проблем для соединения двух устройств типа DTE используется специальный нуль-модемный кабель.

1.2. Нуль-модемный интерфейс.

Обмен сигналами между адаптером компьютера (DTE) и модемом (DCE) (или 2-м компьютером, присоединенным к исходному посредством нуль-

модемного кабеля стандарта RS-232C) строится по стандартному сценарию, в котором каждый сигнал генерируется сторонами лишь после наступления определенных условий. Такая процедура обмена информацией называется запрос/ответным режимом, или “рукопожатием” (**handshaking**). Большинство из приведенных в таблице сигналов как раз и нужны для аппаратной реализации “рукопожатия” между адаптером и модемом.

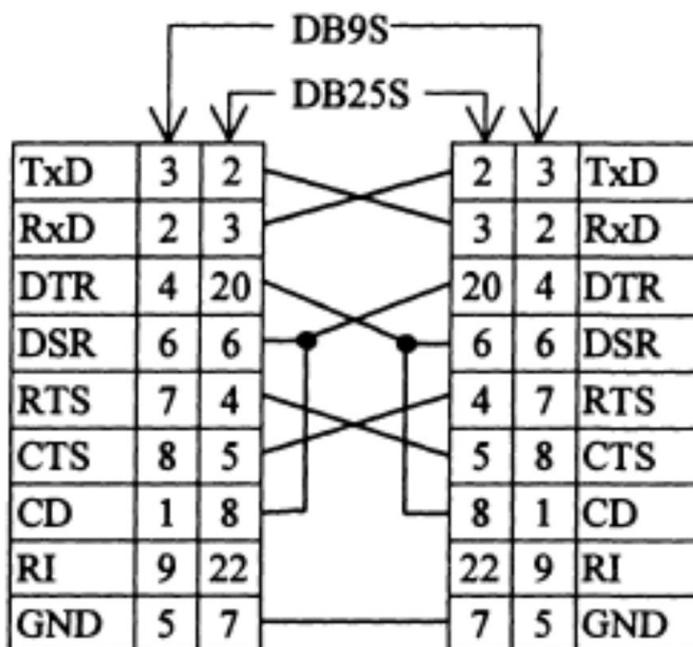
Обмен сигналами между сторонами интерфейса **RS-232C** выглядит так:

1. компьютер после включения питания и открытия COM-порта выставляет сигнал **DTR**, который удерживается активным. Если модем включен в электросеть и исправен, он отвечает компьютеру сигналом **DSR**. Этот сигнал служит подтверждением того, что **DTR** принят, и информирует компьютер о готовности модема к приему информации;
2. если компьютер получил сигнал **DSR** и хочет передать данные, он выставляет сигнал **RTS**;
3. если модем готов принимать данные, он отвечает сигналом **CTS**. Он служит для компьютера подтверждением того, что **RTS** получен модемом и модем готов принять данные от компьютера. С этого момента адаптер может бит за битом передавать информацию по линии **TD**;
4. получив байт данных, модем может сбросить свой сигнал **CTS**, информируя компьютер о необходимости “притормозить” передачу следующего байта, например, из-за переполнения внутреннего буфера; программа компьютера, обнаружив сброс **CTS**, прекращает передачу данных, ожидая повторного появления **CTS**.

Модем может передать данные в компьютер, когда он обнаружит несущую в линии и выставит сигнал — **DCD**. Программа компьютера, принимающая данные, обнаружив этот сигнал, читает приемный регистр, в который сдвиговый регистр “собрал” биты, принятые по линии приема данных **RD**. Когда для связи используются только приведенные в таблице данные, компьютер не может попросить модем “повременить” с передачей следующего байта. Как следствие, существует опасность переопределения помещенного

ранее в приемном регистре байта данных вновь “собранным” байтом. Поэтому при приеме информации компьютер должен очень быстро освобождать приемный регистр адаптера. В полном наборе сигналов **RS-232C** есть линии, которые могут аппаратно “приостановить” модем.

Нуль-модемный интерфейс характерен для прямой связи компьютеров на небольшом расстоянии (длина кабеля до 15 метров). Для нормальной работы двух непосредственно соединенных компьютеров нуль-модемный кабель должен выполнять следующие соединения:



При таком варианте нуль-модемного кабеля открытие COM-портов на обоих компьютерах создает все необходимые условия для установления соединения, перечисленные выше в п.п.1- 4.

1.2.1. Настройка COM-порта средствами библиотеки `java.comm`.

Прикладная библиотека `java.comm` предлагает широкие возможности по настройке COM-порта. Подробное описание библиотеки <http://java.sun.com/products/javacomm/reference/api/index.html>.

1.2.3. Описание класса SerialPort.

Класс SerialPort дает возможность управления последовательными портами компьютера. Он определяет минимальную функциональность для работы с ними.

Поля класса:

DATABITS_5 – 5 бит данных.

DATABITS_6 - 6 бит данных.

DATABITS_7 - 7 бит данных.

DATABITS_8 - 8 бит данных.

FLOWCONTROL_NONE – отключить управление обменом данными.

FLOWCONTROL_RTSCTS_IN – RTS/CTS управление обменом данными.

FLOWCONTROL_RTSCTS_OUT – RTS/CTS управление обменом данными.

FLOWCONTROL_XONXOFF_IN – XON/XOFF управление обменом данными.

FLOWCONTROL_XONXOFF_OUT – XON/XOFF управление обменом данными.

PARITY_EVEN - Дополнение до четности.

PARITY_MARK - Бит четности всегда 1.

PARITY_NONE - Бит четности отсутствует.

PARITY_ODD – Дополнение до нечетности.

PARITY_SPACE – Бит четности всегда 0.

STOPBITS_1 - один стоп-бит.

STOPBITS_1_5 - полтора стоп-бита.

STOPBITS_2 – два стоп-бита.

Методы класса:

addEventListener() – добавляет обработчик события для Com-порта.

getBaudRate() – возвращает установленную скорость передачи.

getStopBits() – возвращает число стоп-битов.

getDataBits() – возвращает число бит данных.

getFlowControlMode() – возвращает режим управления обменом данными.

getParity() – возвращает установленный контроль четности.

isCD() – возвращает значение CD бита порта.

isCTS() – возвращает значение CTS бита порта.

isDSR() – возвращает значение DSR бита порта.

isDTR() – возвращает значение DTR бита порта.

isRI() – возвращает значение RI бита порта.

isRTS() – возвращает значение RTS бита порта.

notifyOnBreakInterrupt() – уведомляет об прерывании разрыва линии.

notifyOnCarrierDetect() – уведомляет об изменении бита CD.

notifyOnDataAvailable() – уведомляет об приходе байт в буфер COM-порта.

notifyOnDSR() – уведомляет об изменении бита DSR.

notifyOnFramingError() – уведомляет об ошибке FE .

notifyOnOutputEmpty() – уведомляет о пустоте выходного буфера.

notifyOnOverrunError() – уведомляет об ошибке OE.

notifyOnParityError() – уведомляет об ошибке PE.

notifyOnRingIndicator() – уведомляет об изменении бита RI.

removeEventListener() – удаляет обработчик события.

setSerialPortParameters() – устанавливает параметры COM-порта

Методы, наследуемые от класса **CommPort**:

getOutputStream() – возвращает указатель на входной буфер.

getInputStream() - возвращает указатель на выходной буфер.

open() – открывает COM-порт.

close() – закрывает COM-порт.

1.2.4. Класс **SerialPortEvent**.

Этот класс определяет возможные события, происходящие на COM-порте.

Поля класса:

BI – сигнал BI.

CD – сигнал CD.

CTS – сигнал CTS.

DSR – сигнал DSR.

FE – сигнал ошибки FE.

OE – сигнал ошибки OE.

PE – сигнал ошибки PE.

RI – сигнал RI.

OUTPUT_BUFFER_EMPTY – сигнал отсутствия данных в выходном буфере.

DATA_AVAILABLE – сигнал наличия данных в входном буфере.

Методы класса:

getEventType() – возвращает тип события.

getNewValue() - возвращает текущее значение измененного сигнала

getOldValue() – возвращает предыдущее значение измененного сигнала

2. Описание функций канального уровня.

На канальном уровне выполняются следующие функции:

1. Запрос логического соединения;
2. Управление передачей кадров;
3. Обеспечение необходимой последовательности блоков данных, передаваемых через межуровневый интерфейс;
4. Контроль и обработка ошибок;
5. Проверка целостности логического соединения;
6. Посылка подтверждения.
7. Запрос на разъединение логического соединения.

2.1. Протокол связи.

В основном протокол содержит набор соглашений или правил, которого должны придерживаться обе стороны связи для обеспечения получения и корректной интерпретации информации, передаваемой между двумя сторонами. Таким образом, помимо управления ошибками и потоком протокол связи регулирует также такие вопросы, как формат передаваемых данных — число битов на каждый элемент и тип используемой схемы кодирования, тип и порядок сообщений, подлежащих обмену для обеспечения (свободной от ошибок и дубликатов) передачи информации между двумя взаимодействующими сторонами.

Перед началом передачи данных требуется установить соединение между двумя сторонами, тем самым проверяется доступность приемного устройства и его готовность воспринимать данные. Для этого передающее устройство посылает специальную команду: запрос на соединение и ожидает ее приема с другого СОМ-порта. Все компьютеры, находящиеся в сети и готовые к установке соединения, ретранслируют чужие запросы на соединение.

Также необходимо информировать пользователя о неисправностях в физическом канале, поэтому для поддержания логического соединения необходимо предусмотреть специальный кадр, который непрерывно будет посылаться с одного компьютера на другой, сигнализируя тем самым, что логическое соединение активно. В протоколе этот кадр и кадр запроса на соединение может быть один и тот же.

2.2. Защита передаваемой информации.

При передаче данных по линиям могут возникать ошибки, вызванные электрическими помехами, связанными, например, с шумами, порожденными коммутирующими элементами сети. Эти помехи могут вызвать множество ошибок в цепочке последовательных битов. Контроль ошибок либо

совершается функциями COM-порта (это методы контроля ошибок из-за увеличения темпа работы (overrun error), ошибок четности (parity error) и ошибок кадрирования (faming error)), если это определено заданием, либо применением циклического кода. Исправление ошибок необходимо реализовать средствами канального уровня методом ARQ.

2.3. Примеры форматов кадров.

Все кадры могут иметь одинаковую структуру, представленную ниже:

Стартовый байт	Адрес получателя	Адрес отправителя	Тип кадра	Длина поля данных*	Данные*	Стоповый байт
1 байт	1 байт	1 байт	1 байт	1 байт	N байт	1 байт

Примечание: поля отмеченные * - не обязательны и зависят от значения блока команда.

Стартовый и стоповый байт – служат для определения начала и конца кадра.

Для них принято значение 0xFF.

Адрес получателя – байт, содержащий адрес получателя кадра. Существует широковещательный адрес (0x7F).

Адрес отправителя – байт, содержащий адрес отправителя кадра.

Адреса в системе назначаются динамически и лежат в промежутке от 0x01 до 0x7E.

Тип кадра – байт, содержащий код типа кадра.

Длина поля данных – не обязательное поле, содержит длину поля **Данные**.

Данные – не обязательное поле, содержит какие-либо данные, передаваемые в кадре.

2.3.1. Информационный I - Кадр.

Информационный кадр. Может служить для передачи коротких сообщений между компьютерами, включенными в кольцо. Поля **Длина поля данных** и **Данные** – присутствуют. Поле **Данные** содержит текст передаваемого короткого сообщения.

2.3.2. Супервизорный Link - Кадр.

Кадр установки соединения. Может служить для установки логического соединения типа «кольцо». В поле **Адрес получателя** – обязательно широковещательный адрес, т.к. используется в системе в начальный момент, когда адреса всех компьютеров сети не определены. Поля **Длина поля данных** и **Данные** – присутствуют. Поле **Данные** содержит адреса и текстовые псевдонимы всех пользователей сети.

2.3.3. Супервизорный Uplink - Кадр.

Кадр разрыва соединения. Может служить для разрыва логического соединения типа «кольцо». В поле **Адрес получателя** – обязательно широковещательный адрес. Поля **Длина поля данных** и **Данные** – отсутствуют.

2.3.4. Супервизорный АСК - Кадр.

Кадр подтверждения безошибочного приема тестового сообщения. Может служить для контроля передачи текстовых сообщений. В поле **Адрес получателя** – не должно быть широковещательного адреса. Поля **Длина поля данных** и **Данные** – отсутствуют.

2.3.5. Супервизорный Ret - Кадр.

Кадр запроса повторения последнего отправленного кадра. Может служить для исправления возникших ошибок в ходе передачи кадров. В поле **Адрес получателя** – не должно быть широковещательного адреса. Поля **Длина поля данных** и **Данные** – отсутствуют.

3. Описание функций прикладного уровня.

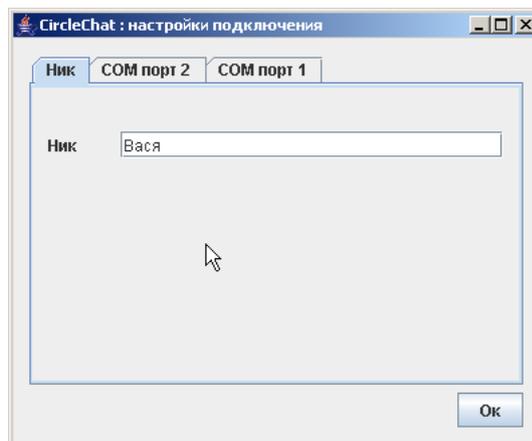
Функции прикладного уровня обеспечивают интерфейс программы с пользователем. Форма взаимодействия с пользователем определяется техническим заданием. Например, прикладной уровень предоставляет каналному уровню текстовое сообщение для передачи по кольцу..

На данном уровне производится взаимодействие с пользователем, например: ввод\вывод сообщений, управление подключением, выбор файла для передачи, работа с историей сообщений и т.п..

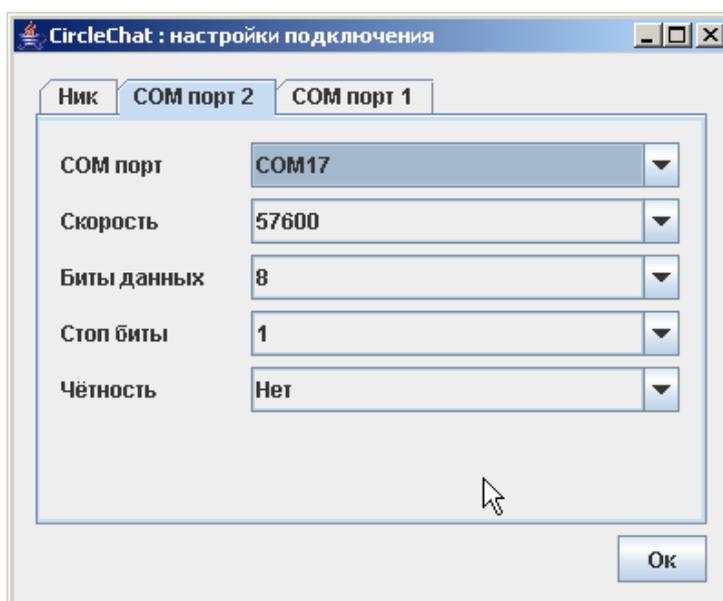
Пользовательский интерфейс может быть выполнен, например, в среде NetBeans с использованием библиотеки Java javax.swing. Интерфейс программы, как правило, выполняется многооконным. При запуске программы должно появиться окно настроек, где пользователю предлагается ввести имя и настроить COM порты. После выбора настроек появляется главное окно, в котором расположены элементы управления, позволяющие выполнять установление и разрыв соединения, обеспечивающие работу с историей и отправку\приём общих сообщений, выбор необходимого файла и т.п.

3.1. Пример окна настройки подключения

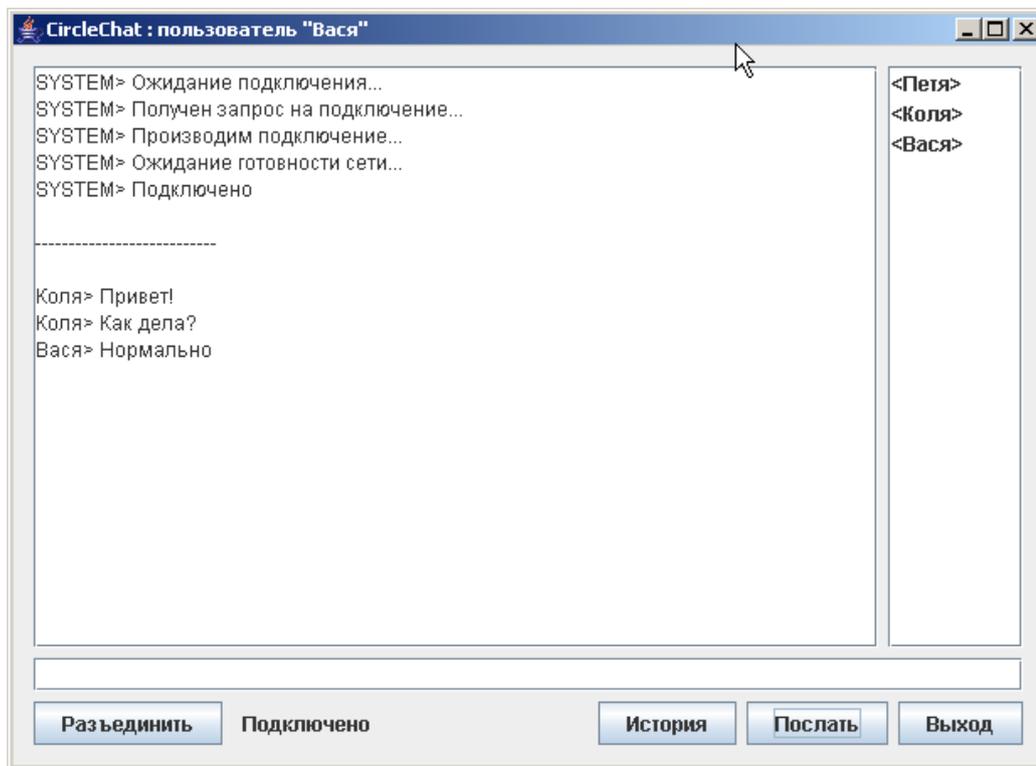
Вкладка "Ник" – позволяет ввести имя пользователя



Вкладка "COM порт N" – позволяет установить COM порт для связи и настроить его параметры.



3.2. Пример окна главного меню.



Список слева – имена подключённых пользователей. Окно в центре – область, в которую выводятся сообщения пользователей и системные сообщения (с пометкой "SYSTEM>").

По нажатию кнопки "Послать" или клавиши Enter на строке ввода производится отправка сообщения. Сообщение дублируется в собственном окне сообщений.

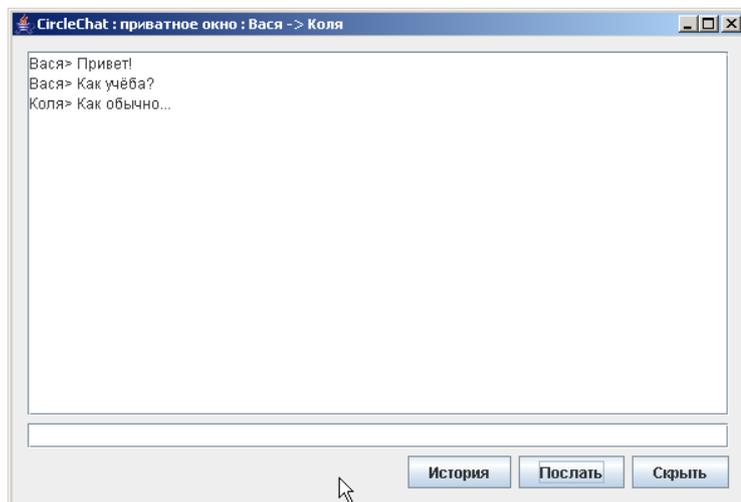
По нажатию кнопки "История" появляется окно истории сообщений.

По нажатию кнопки "Разъединить" производится отключение от сети с разрывом кольца.

Строка в центре внизу – состояние связи.

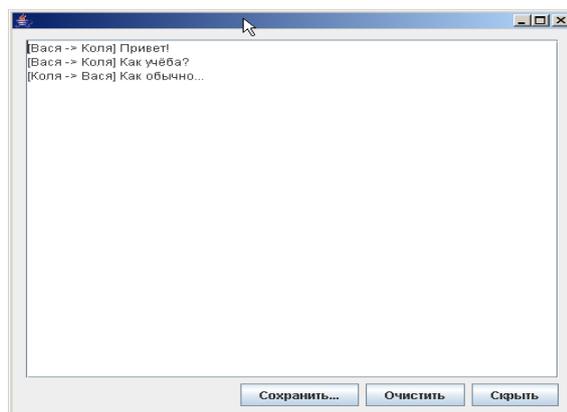
При двойном клике на имени пользователя появляется окно личных сообщений.

3.3. Пример окна личных сообщений.



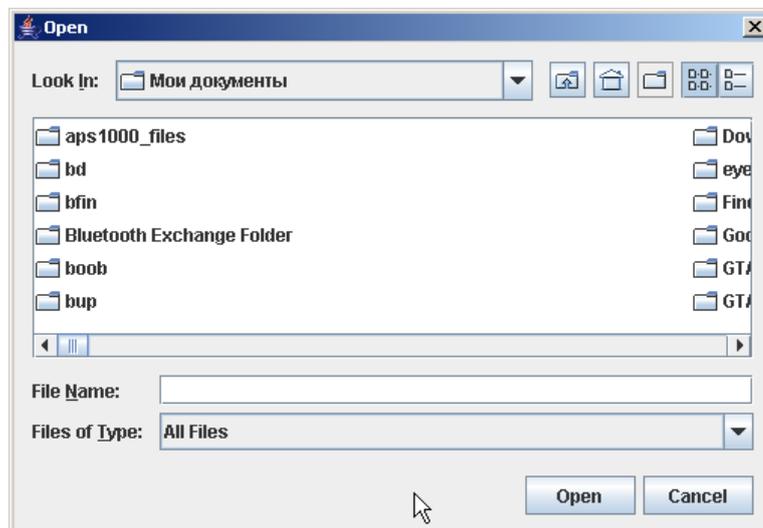
Кнопка "Скрыть" скрывает окно. Открыть его можно двойным кликом по имени собеседника.

3.4. Пример окна истории сообщений



По нажатию кнопки "Очистить" производится удаление истории.

По нажатию кнопки «Сохранить» пользователю предоставляется выбор файла для сохранения:



И производится запись истории в этот файл.

По нажатию кнопки "Скрыть" окно истории скрывается. Повторным нажатием кнопки "История" его можно показать вновь.

3.5. Взаимодействие с канальным уровнем

Взаимодействие с канальным уровнем происходит путём передачи\приёма сообщений через три очереди сообщений: очередь входящих сообщений, очередь исходящих сообщений и системная очередь.

В очередях сообщений хранятся пакеты сообщений. Через системную очередь передаются системные пакеты.

Формат пакета сообщения

1. Имя пользователя получателя	2. Имя пользователя отправителя	3. Текст сообщения
Строка	Строка	Строка

Поле "Имя пользователя получателя" хранит имя пользователя, которому адресовано сообщение.

Поле "Имя пользователя отправителя" хранит имя пользователя отправителя сообщения.

Поле "Текст сообщения" содержит передаваемое текстовое сообщение.

Формат системного пакета

1. Идентификатор события
Число

В поле "Идентификатор события" передаётся номер системного события.

Оно может принимать следующие значения:

Номер события	Описание
NO_ACK	Сообщение не доставлено
DISCONNECT	Связь прекращена
DISRUPTION	Разрыв целостности кольца
CONNECT	Связь установлена
CONNECT_REQUEST	Запрос установки связи
DISCONNECT_REQUEST	Запрос разрыва связи
ACK	Сообщение доставлено

4. Варианты заданий.

Группа ИУ5-61Б

Вариант 1:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи текстовых файлов. Скорость обмена и параметры СОМ-порта выбираются пользователями ПК. Имя передаваемого файла задается источником. При передаче файла защитить передаваемую информацию циклическим [7,4]-кодом.

Вариант 2:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в ненаправленное кольцо через интерфейсы RS232C (СОМ1 и СОМ2) , и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи выбирает пользователь одного из ПК и рассылает по сети. Для контроля ошибки использовать средства коммуникационного порта СОМ1 и СОМ2.

Вариант 3:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи двоичных файлов. Скорость обмена и параметры СОМ-порта выбираются пользователем на каждой стороне. Файл выбирается из каталога на стороне источника получателем. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Вариант 4:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в направленное кольцо через интерфейсы RS232 C (СОМ1 и СОМ2) , и реализующей функции адресной и широковещательной передачи коротких

сообщений. Параметры передачи заданы по умолчанию. Для контроля ошибки использовать циклический [7,4]-код.

Вариант 5:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию одновременной двунаправленной передачи файлов. Скорость обмена и параметры СОМ-порта выбирает пользователь каждого ПК. Имя передаваемого файла выбирается из каталога источника отправителем. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Вариант 6:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи коротких сообщений и файлов. Скорость обмена и параметры СОМ-порта заданы по умолчанию. Имя передаваемого файла выбирается из каталога источника ведущей станцией. При передаче файла защитить передаваемую информацию циклическим [7,4]-кодом.

Вариант 7:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через интерфейс RS232C , и реализующей функцию передачи текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах. Скорость обмена и параметры СОМ-порта выбирает пользователь одного из ПК. Передаваемую информацию защитить [7,4]-кодом Хэмминга.

Группа ИУ5-62Б

Вариант 8:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в направленное маркерное кольцо через интерфейсы RS232C (COM1 и COM2) , и

реализующей функции адресной передачи текстовых сообщений в конверте. Уведомлять отправителя о вскрытии конверта получателем. Параметры обмена заданы по умолчанию. Для контроля ошибки использовать циклический [7,4] - код.

Вариант 9:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из N ПК, соединенных нульмодемно в направленное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей функции передачи текстов диалога любой пары абонентов. Параметры обмена заданы по умолчанию. Для контроля ошибки использовать [7,4]-код Хэмминга.

Вариант 10:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из N ПК, соединенных нульмодемно в направленное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей при передаче текстовых сообщений основные функции электронной почты:

- почтовый ящик (на жестком диске);
- подтверждение доставки при вскрытии конверта;
- переадресация;
- ответ.

Параметры обмена заданы по умолчанию. Для контроля ошибки использовать циклический [7,4]-код.

Вариант 11:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи текстовых файлов. Обеспечить просмотр содержимого файла при приеме и управление на прикладном уровне темпом передачи. Скорость обмена и параметры COM-порта заданы по умолчанию. Файл задается источником. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Вариант 12:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в кольцо через интерфейсы RS232C (COM1 и COM2) , и реализующей функции адресной и широковещательной передачи коротких сообщений с возможностью прикрепления файла. Параметры передачи выбираются пользователем ПК. Для контроля ошибки использовать [7,4]-код Хэмминга.

Вариант 13:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи двоичных файлов. Скорость обмена и параметры СОМ-портов выбирает пользователь одного ПК. Имя передаваемого файла также задается ведущей станцией. При передаче файла защитить передаваемую информацию циклическим [7,4]-кодом.

Вариант 14:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в ненаправленное кольцо через интерфейсы RS232C (COM1 и COM2) , и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи определяет ведущая станция. Для контроля ошибки использовать [7,4]-код Хэмминга.

Вариант 15:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3 ПК, соединенных нульмодемно в ненаправленное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей при передаче текстовых сообщений основные функции электронной почты:

- почтовый ящик (на жестком диске);
- подтверждение доставки при вскрытии конверта;
- переадресация;
- рассылка всем;
- ответ.

Параметры обмена заданы по умолчанию. Для контроля ошибки использовать циклический [7,4]-код.

Группа ИУ5-63Б

Вариант 16:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи файлов и коротких сообщений. Скорость обмена и параметры СОМ-порта выбираются пользователем. Имя передаваемого файла выбирается из каталога источника получателем. При передаче файла защитить передаваемую информацию циклическим [7,4]-кодом.

Вариант 17:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи файлов с возможностью докачки после восстановления прерванной связи. Скорость обмена и параметры СОМ-порта заданы по умолчанию. Файл выбирает из каталога источника отправитель. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Вариант 18:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно через интерфейс RS232C в ненаправленное кольцо, и реализующей функцию передачи писем абонентов. Папки для входящих и исходящих писем отображать в разных окнах. Оповещать источник о вскрытии письма. Скорость обмена и параметры СОМ-порта выбирают пользователи ПК. Передаваемую информацию защитить циклическим [7,4]-кодом.

Вариант 19:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в

направленное маркерное кольцо через интерфейсы RS232C (COM1 и COM2) , и реализующей функции адресной передачи файловых текстовых сообщений. Оповещать источник об открытии файла. Параметры обмена заданы по умолчанию. Для контроля ошибки использовать [7,4]-код Хэмминга.

Вариант 20:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из N ПК, соединенных нульмодемно в направленные маркерное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей функции передачи текстов диалога любой пары абонентов. Параметры обмена устанавливает ведущая станция. Для контроля ошибки использовать циклический [7,4]-код.

Вариант 21:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из N ПК, соединенных нульмодемно в направленные кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей при адресной и групповой передаче текстовых сообщений основные функции электронной почты:

- почтовый ящик;
- адресная книга;
- подтверждение вскрытия конверта;
- переадресация;
- ответ.

Параметры обмена заданы по умолчанию. Для контроля ошибки использовать [7,4]-код Хэмминга.

Вариант 22:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию одновременной двунаправленной передачи файлов. Скорость обмена и параметры COM-порта выбирает пользователь каждого ПК. Имя передаваемого файла выбирается из

каталога источника получателем. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Вариант 23:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию одновременной передачи файлов с возможностью «докачки» после восстановления прерванной связи. Скорость обмена и параметры СОМ-порта заданы по умолчанию. Файл выбирает из каталога источника отправитель. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Группа ИУ5-64Б

Вариант 24:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию одновременной двунаправленной передачи файлов. Скорость обмена и параметры СОМ-порта выбирает пользователь одного из ПК. Имя передаваемого файла выбирается из своего каталога отправителем. При передаче файла защитить передаваемую информацию [15,11]-кодом Хэмминга.

Вариант 25:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи коротких сообщений и файлов. Скорость обмена и параметры СОМ-порта заданы по умолчанию. Имя передаваемого файла выбирается из каталога источника ведущей станцией. При передаче файла защитить передаваемую информацию циклическим [15,11]-кодом.

Вариант 26:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через

интерфейс RS232C , и реализующей функцию передачи текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах. Скорость обмена и параметры СОМ-порта выбирает пользователь одного из ПК. Передаваемую информацию защитить [15,11]-кодом Хэмминга.

Вариант 27:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи текстовых файлов. Скорость обмена и параметры СОМ-порта выбираются пользователями ПК. Имя передаваемого файла задается источником. Каталоги для сохранения полученных файлов задают получатели. При передаче файла защитить передаваемую информацию циклическим [15,11]-кодом.

Вариант 28:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в ненаправленное кольцо через интерфейсы RS232C (СОМ1 и СОМ2) , и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи выбирает пользователь одного из ПК и рассылает по сети. Для контроля ошибки использовать код Хэмминга [7,4].

Вариант 29:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи двоичных файлов. Скорость обмена и параметры СОМ-порта выбираются пользователем на каждой стороне. Файл выбирается из каталога на стороне источника получателем. При передаче файла защитить передаваемую информацию [15,11]-кодом Хэмминга.

Группа ИУ5-66Б

Вариант 30:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в направленное кольцо через интерфейсы RS232 C (COM1 и COM2) , и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи заданы по умолчанию. Для контроля ошибки использовать циклический [15,11]-код.

Вариант 31:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из N ПК, соединенных нульмодемно в направленное маркерное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей функции передачи текстов диалога любой пары абонентов. Параметры обмена устанавливает ведущая станция. Для контроля ошибки использовать циклический [15,11]-код.

Вариант 32:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из N ПК, соединенных нульмодемно в направленное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей при адресной и групповой передаче текстовых сообщений основные функции электронной почты:

- почтовый ящик;
- адресная книга;
- подтверждение вскрытия конверта;
- переадресация;
- ответ.

Параметры обмена заданы по умолчанию. Для контроля ошибки использовать [15,11]-код Хэмминга.

Вариант 33:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи файлов с

возможностью докачки после восстановления прерванной связи. Скорость обмена и параметры СОМ-порта заданы по умолчанию. Файл выбирает из каталога источника отправитель. При передаче файла защитить передаваемую информацию [15,11]-кодом Хэмминга.

Вариант 34:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно через интерфейс RS232C в ненаправленное кольцо, и реализующей функцию передачи писем абонентов. Папки для входящих и исходящих писем отображать в разных окнах. Оповещать источник о вскрытии письма. Скорость обмена и параметры СОМ-порта выбирают пользователи ПК. Передаваемую информацию защитить циклическим [15,11]-кодом.

Вариант 35:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в ненаправленное кольцо через интерфейсы RS232C (COM1 и COM2), и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи выбирает пользователь одного из ПК и рассылает по сети. Для контроля ошибки использовать код Хэмминга [15,11].

Вариант 36:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через интерфейс RS232C, и реализующей функцию передачи текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах и сохранять в журнале на каждой стороне. Скорость обмена и параметры СОМ-порта выбирает пользователь одного из ПК. Передаваемую информацию защитить [15,11]-кодом Хэмминга.

Вариант 37:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи файлов с

возможностью «докачки» после восстановления прерванной связи. Скорость обмена и параметры СОМ-порта задает пользователь одного из ПК. Файл выбирает из своего каталога отправитель. При передаче файла защитить передаваемую информацию [15,11]-кодом Хэмминга.

Группа РТ5-61Б

Вариант 38:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи текстовых файлов. Скорость обмена и параметры СОМ-порта выбираются пользователями ПК. Имя передаваемого файла задается источником. При передаче файла защитить передаваемую информацию циклическим [15,11]-кодом.

Вариант 39:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в ненаправленное кольцо через интерфейсы RS232C (COM1 и COM2) , и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи выбирает пользователь одного из ПК и рассылает по сети. Для контроля ошибки использовать возможности кода Хэмминга [7,4].

Вариант 40:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи двоичных файлов. Скорость обмена и параметры СОМ-порта выбираются пользователем на каждой стороне. Файл выбирается из каталога на стороне источника получателем. При передаче файла защитить передаваемую информацию [15,11]-кодом Хэмминга.

Вариант 41:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 3-х ПК, соединенных нульмодемно в направленное кольцо через интерфейсы RS232 C (COM1 и COM2) , и реализующей функции адресной и широковещательной передачи коротких сообщений. Параметры передачи заданы по умолчанию. Для контроля ошибки использовать циклический [7,4]-код.

Вариант 42:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию одновременной двунаправленной передачи файлов с возможностью «докачки» после разрыва соединения. Скорость обмена и параметры COM-порта выбирает пользователь каждого ПК. Имя передаваемого файла выбирается из каталога источника отправителем. При передаче файла защитить передаваемую информацию [7,4]-кодом Хэмминга.

Вариант 43:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных через интерфейс RS232C нульмодемным кабелем, и реализующей функцию передачи коротких сообщений и файлов. Скорость обмена и параметры COM-порта заданы по умолчанию. Имя передаваемого файла выбирается из каталога источника ведущей станцией. При передаче файла защитить передаваемую информацию циклическим [15,11]-кодом.

Вариант 44:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через интерфейс RS232C , и реализующей функцию передачи текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах. Скорость обмена и параметры COM-порта выбирает пользователь одного из ПК. Передаваемую информацию защитить [15,11]-кодом Хэмминга.

Группа ИУ5Ц-81Б

Вариант 45:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через интерфейс RS232C , и реализующей функцию передачи текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах. Скорость обмена и параметры СОМ-порта выбирает пользователь одного из ПК. Передаваемую информацию защитить [7,4]-кодом Хэмминга.

Группа ИУ5Ц-82Б

Вариант 46:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через интерфейс RS232C, и реализующей функцию передачи текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах. Скорость обмена и параметры СОМ-порта выбирают пользователи каждого ПК. Передаваемую информацию защитить [7,4]-кодом Хэмминга.

Группа ИУ5Ц-83Б

Вариант 47:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через интерфейс RS232C, и реализующей функцию передачи файлов и текста диалога абонентов. Принимаемый и передаваемый тексты отображать в разных окнах. Скорость обмена и параметры СОМ-порта выбирают пользователи каждого ПК. Передаваемую информацию защитить [7,4]-кодом Хэмминга.

Группа ИУ5Ц-84Б

Вариант 48:

Разработать протоколы взаимодействия объектов до прикладного уровня локальной сети, состоящей из 2-х ПК, соединенных нульмодемно через

интерфейс RS232C, и реализующей функцию передачи файлов. Имя файла выбирает передатчик из своего каталога. При сохранении принимаемого файла предусмотреть возможность его переименования. Скорость обмена и параметры СОМ-порта выбирают пользователи каждого ПК. Передаваемую информацию защитить циклическим [7,4]-кодом.

К защите предъявляются:

1. Расчетно-пояснительная записка, включающая в приложении комплект технической документации на программный продукт, содержащий:

Приложение 1 - Техническое задание;
Приложение 2 - Описание программы;
Приложение 3 - Руководство пользователя;
Приложение 4 - Программа и методика испытаний;
Приложение 5 - Графическая часть на 9-12 листах

формата А4:

- Структурная схема программы.
- Структура протокольных блоков данных.
- Структурные схемы основных процедур взаимодействия объектов по разработанным протоколам.
- Временные диаграммы работы протоколов.
- Граф диалога пользователя.
- Алгоритмы программ.

2. Папка с технической и программной документацией в формате:

<группа>_<Вариант>_КР_СТ_в_АСОИУ.zip.

5. Литература.

1. Галкин В.А., Григорьев Ю.А. Телекоммуникации и сети: Учеб. пособие для вузов. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2003.
2. Арнольд Кен, Гослинг Джеймс. Язык программирования Java / Пер. с англ. Е. Матвеев .- СПб. : Питер , 1997 .- 304 с.
3. Java™ 2 Platform Standard Edition 5.0 API Specification. Режим доступа: <http://java.sun.com/j2se/1.5.0/docs/api/index.html> (дата обращения 01.12.2013).