Московский государственный технический университет имени Н.Э.Баумана.

Факультет ― Информатика и системы управления.

Кафедра ― Системы обработки информации и управления.

|  |
| --- |
| Утверждаю: |
|  |
| " " 2018г. |

Курсовая работа по курсу

Сетевые технологии

«Программа пересылки файлов»

Расчетно-пояснительная записка

(вид документа)

бумага А4

(вид носителя)

16

(количество листов)

Вариант

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛИ: | |
| студенты группы ИУ5-61 | |
| Баглай П.С. |  |
| " " 2018г. | |

Москва 2018 г.

# Оглавление

1. [Физический уровень 3](#_bookmark0)
   1. [Интерфейс RS-232C 3](#_bookmark1)
   2. [Физические параметры интерфейса RS-232C 4](#_bookmark2)
   3. [Асинхронная передача данных 6](#_bookmark3)
   4. [Реализация физического уровня. 9](#_bookmark4)
      1. [Открытие порта 9](#_bookmark5)
      2. [Закрытие порта 10](#_bookmark6)
      3. [Передача и прием данных 10](#_bookmark7)
2. [Канальный уровень 10](#_bookmark8)
   1. [Защита передаваемой информации 10](#_bookmark9)
   2. [Передача данных 12](#_bookmark10)
   3. [Функции кодирования/декодирования. 12](#_bookmark11)
   4. [Форматы кадров 12](#_bookmark12)
3. [Фользовательский уровень 13](#_bookmark13)
   1. [Формы 13](#_bookmark14)

## ФИЗИЧЕСКИЙ УРОВЕНЬ

## Интерфейс RS-232C

Интерфейс RS-232C (официальное название "EIA/TIA–232–E") является наиболее широко распространенной стандартной последовательной связью между микрокомпьютерами и периферийными устройствами. Интерфейс, определенный стандартом Ассоциации электронной промышленности (EIA), подразумевает наличие оборудования двух видов: терминального DTE и связногоDCE.

Терминальное оборудование, например, микрокомпьютер, может посылать или принимать данные по последовательному интерфейсу. Оно как бы оканчивает (terminate) последовательную линию. Связное оборудование - устройства, которые могут упростить передачу данных совместно с терминальным оборудованием. Наглядным примером связного оборудования служит модем (модулятор-демодулятор). Он оказывается соединительным звеном в последовательной цепочке между компьютером и телефонной линией.

Конечной целью подключения является соединение двух устройств DTE.Полная схема соединения включает в себя устройства DCE соединённые с DTE через интерфейс RS-232 и линию удалённой связи. Интерфейс позволяет исключить канал удаленной связи вместе с парой устройств DCE (модемов), соединив устройства непосредственно с помощью нуль-модемного кабеля (Рис.1).

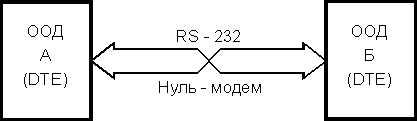


Рис. 1.Соединение по RS-232C нуль-модемным кабелем.

## Физические параметры интерфейса RS-232C

Стандарт RS-232C регламентирует типы применяемых разъемов, что обеспечивает высокий уровень совместимости аппаратуры различных производителей. На аппаратуре DTE (в том числе, и на COM-портах PC) принято устанавливать вилки (male-"папа") DB25-P или DB9-P. Девятиштырьковые разъемы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима.

В случае, когда аппаратура DTE соединяется без модемов ("Короткозамкнутая петля"), то разъемы устройств (вилки) соединяются между собой нуль-модемным кабелем (Zeromodem или Z-modem), имеющим на обоих концах розетки, контакты которых соединяются перекрестно схеме, приведенной на Рис.2.

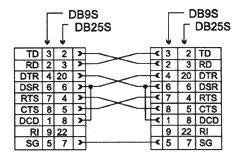


Рис. 2. Полный нуль-модемный кабель.

В таблице 1 приведено назначение контактов разъемов COM-портов (и любой другой аппаратуры DTE). Назначение контактов разъема DB9S (Рис. 3) определено стандартом EIA/TIA-574.

Таблица 1**.** Разъемы и сигналы интерфейса RS-232C.

|  |  |  |  |
| --- | --- | --- | --- |
| **Обозначение цепи** | **Контакт разъема** | **Направление** | **Название цепи** |
| **RS232** | **DB9S** | **Вход/Выход** |  |
| PG | - | - | Protect Ground – Защитная земля |
| TD | 3 | Выход | Transmit Data – Передаваемые данные |
| RD | 2 | Вход | Receive Data – Принимаемые данные |
| RTS | 7 | Выход | Request To Send - Запрос на передачу |
| CTS | 8 | Вход | Clear To Send - Готовность модема к приему данных  для передачи |
| DSR | 6 | Вход | Data Set Ready - Готовность модема к работе |
| SG | 5 | - | Signal Ground - Схемная земля |
| DCD | 1 | Вход | Data Carrier Detect – Несущая обнаружена |
| DTR | 4 | Выход | Data Terminal Ready - Готовность терминала (PC) к  работе |
| RI | 9 | Вход | Ring Indicator – Индикатор вызова |

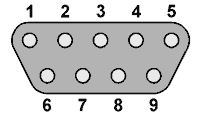


Рис. 3. Назначение контактов разъема DB9.

1. Установкой DTR компьютер указывает на желание использовать модем.
2. Установкой DSR модем сигнализирует о своей готовности и установлении соединения.
3. Сигналом RTS компьютер запрашивает разрешение на передачу и заявляет о своей готовности принимать данные от модема.
4. Сигналом CTS модем уведомляет о своей готовности к приему данных от компьютера и передаче их в линию.
5. Снятием CTS модем сигнализирует о невозможности дальнейшего приема (например, буфер заполнен) — компьютер должен приостановить передачу данных.
6. Сигналом CTS модем разрешает компьютеру продолжить передачу (в буфере появилось место).
7. Снятие RTS может означать как заполнение буфера компьютера (модем должен приостановить передачу данных в компьютер), так и отсутствие данных для передачи в модем. Обычно в этом случае модем прекращает пересылку данных в компьютер.
8. Модем подтверждает снятие RTS сбросомCTS.
9. Компьютер повторно устанавливает RTS для возобновления передачи.
10. Модем подтверждает готовность к этим действиям.
11. Компьютер указывает на завершение обмена.
12. Модем отвечает подтверждением.
13. Компьютер снимает DTR, что обычно является сигналом на разрыв соединения (повесить трубку).
14. Модем сбросом DSR сигнализирует о разрыве соединения.

## Асинхронная передача данных

Асинхронный режим передачи является байт-ориентированным (символьно-ориентированным): минимальная пересылаемая единица информации — один байт (один символ). Формат посылки байта иллюстрирует Рис.4. Передача каждого байта начинается со старт-бита, сигнализирующего приемнику о начале посылки, за которым следуют биты данных и, возможно, бит четности (Parity). Завершает посылку стоп-бит, гарантирующий паузу между посылками. Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определенное значение, обеспечивает простой механизм синхронизации приемника по сигналу от передатчика. Подразумевается, что приемники передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик генерирует внутренние стробы, по которым приемник фиксирует последующие принимаемые биты. В идеале стробы располагаются в середине битовых интервалов, что позволяет принимать данные и при незначительном рассогласовании скоростей приемника и передатчика. Очевидно, что при передаче 8 бит данных, одного контрольного и одного стоп-бита предельно допустимое рассогласование скоростей, при котором данные будут распознаны верно, не может превышать 5 %. С учетом фазовых искажений и дискретности работы внутреннего счетчика синхронизации реально допустимо меньшее отклонение частот. Чем меньше коэффициент деления опорной частоты внутреннего генератора (чем выше частота передачи), тем больше погрешность привязки стробов к середине битового интервала, и требования к согласованности частот становятся более строгие. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала. Взаимодействие этих факторов приводит к повышению требований к согласованности частот приемника и передатчика с ростом частоты обмена.



Рис. 4. Формат асинхронной передачи RS-232C.

Формат асинхронной посылки позволяет выявлять возможные ошибки передачи.

* Если принят перепад, сигнализирующий о начале посылки, а по стробу старт-бита зафиксирован уровень логической единицы, старт-бит считается ложными приемник снова переходит в состояние ожидания. Об этой ошибке приемник может не сообщать.
* Если вовремя, отведенное под стоп-бит, обнаружен уровень логического нуля, фиксируется ошибка стоп-бита.
* Если применяется контроль четности, то после посылки бит данных передается контрольный бит. Этот бит дополняет количество единичных бит данных до четного или нечетного в зависимости от принятого соглашения. Прием байта с неверным значением контрольного бита приводит к фиксации ошибки.
* Контроль формата позволяет обнаруживать, обрыв линии: как правило, при обрыве приемник ―видит логический нуль, который сначала трактуется как старт-бит и нулевые биты данных, но потом срабатывает контроль стоп-бита.

Для асинхронного режима принят ряд стандартных скоростей обмена: 50, 75,110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 и 115200 бит/с. Иногда вместо единицы измерения ― бит/с используют ― бод(baud), но при рассмотрении двоичных передаваемых сигналов это некорректно. В бодах принято измерять частоту изменения состояния линии, а при недвоичном способе кодирования (широко применяемом в современных модемах) в канале связи скорости передачи бит(бит/с) и изменения сигнала (бод) могут отличаться в несколько раз.

Количество бит данных может составлять 5, 6, 7 или 8 (5- и 6-битныеформаты распространены незначительно). Количество стоп-бит может быть 1, 1,5 или 2(―полтора бита‖ означает только длительность стопового интервала).

В данной работе используются следующие параметры передачи данных:

* Число информационных бит – 8;
* Количество стоповых бит – 1;
* Бит паритета (контроля четности) – не используется;
* Скорость обмена – 9600.

## Реализация физического уровня

## Открытие порта

В ОС Windows доступ к COM-портам предоставляется посредством файловых интерфейсов. Для работы с портом - функции пространства имён **System.IO.Ports** из библиотеки классов .NET FRAMEWORK.

* **ComPort –** объект класса SerialPort, который используется для определения COM-порта.
* **ComPort.Open() –** функция открытия COM-порта.

Установка параметров COM-порта осуществляется с помощью функций:

* **ComPort.DataBits()** – устанавливается число информационных битов;
* **ComPort.Parity()** – осуществляется проверка четности;
* **ComPort.PortName()** – имя порта;
* **ComPort.StopBits()** - число стоповых битов;
* **ComPort.ReadBufferSize()** – устанавливается размер буфера чтения;
* **ComPort.WriteBufferSize()** - устанавливается размер буфера записи;
* **ComPort.Handshake()** - протокол установления связи;

Далее производится настройка тайм-аутов чтения и записи порта с помощью функций:

* **ComPort.ReadTimeout ()** – таймаут чтения;
* **ComPort.WriteTimeout()** – таймаут записи;

Тайм-ауты необходимы для правильной работы функций чтения из порта и записи в порт.

## Закрытие порта

Закрытие порта осуществляется с помощью функции **ComPort.Close()**.

## Передача и прием данных

Для передачи/приема данных нет специальных функций - выполняются по логике программы с помощью операций записи/чтения из буферов порта.

Функция для передачи данных - **ComPort.Write (byte[] InfBytes, intoffset, intcount);**

Функция приема данных – **byte ComPort.ReadByte();**

Функция считывания символов с порта при приеме – **byte ComPort.ReadByte().**

## КАНАЛЬНЫЙ УРОВЕНЬ

## Защита передаваемой информации

При передаче данных по линиям, входящим в коммутируемую сеть, чаще всего возникают ошибки, обусловленные электрическими помехами. Эти помехи в свою очередь могут вызвать ошибки в цепочке или пакете последовательных битов.

Для обнаружения ошибок применяют разнообразные корректирующие коды. Например, линейный код, код Хемминга, циклический код, логический код 4B/5B. Для практической реализации циклических кодов могут быть использованы схемы с регистрами сдвига и обратными связями, или таблицы (в программе используется полином X3+X+1). В рамках данной курсовой работы необходима реализация алгоритма кодирования циклическим кодом.

Рассмотрим алгоритм циклического кода:

1. Задана информационная последовательность m(x). Умножить заданный полином степени(k-1) на х(n-k), т.е. сдвинуть в сторону старших разрядов на(n-k); где n=r+k, r - степень образующего полинома, k - число информационных разрядов данной последовательности;
2. Получить остаток от деления полинома х(n-k)\*m(x) на g(x) – образующий полином. Степень остатка ≤ n-k-1.
3. Объединить остаток р(х) и исходный полином х(n-k)\*m(x) для получения кодового слова; p(x)@ х(n-k)\*m(x), где @ -конкатенация;

Декодирование циклического кода:

V(x) - передаваемый кодовый полином; r(x) -принятый;

r(x)=g(x)\*q(x)+S(x), где q(x) - частное, S(x) - остаток от деления переданного полинома на порождающий полином;

S(x)=S0+S1\*x+...+S(n-k-1)\*x(n-k-1) - синдром ошибки (если S(x)=0, ошибки нет или она не обнаружена).

r(x)=V(x)+e(x), где e(x) – вектор ошибки; e(x)=V(x)+q(x)\*g(x)+S(x)=[m(x)+q(x)]\*g(x)+S(x)

Задача декодирующего устройства - оценка вектора g(x) по синдрому ошибки.

По значению синдрома ошибки определяется бит, в котором была обнаружена ошибка:

|  |  |
| --- | --- |
| Синдром | Номер бита |
| 0001 | 0 |
| 0010 | 1 |
| 0011 | 2 |
| 0100 | 3 |
| 0101 | 4 |
| 0110 | 5 |
| 0111 | 6 |
| 1000 | 7 |
| 1001 | 8 |
| 1010 | 9 |
| 1011 | 10 |
| 1100 | 11 |
| 1101 | 12 |
| 1110 | 13 |
| 1111 | 14 |

## Передача данных

Компьютер – передатчик после выбора пользователем файла для передачи передаёт компьютеру-приёмнику информационные кадры, содержащие имя файла и его длину (структуру кадра см. на листе «Структура протокольных блоков данных»). Компьютер– приёмник принимает эти кадры и компьютеру – передатчику отправляет положительную квитанцию на каждый из этих кадров. В случае ошибки передачи компьютер –приёмник передаёт компьютеру – передатчику отрицательную квитанцию и передатчик повторяет передачу.

В случае прихода положительной квитанций на получение заголовка файла и согласия на прием, компьютер-передатчик начинает поблочную передачу файлов, дожидаясь прихода положительной квитанции после отсылки каждого блока. В случае ошибки передачи компьютер – приёмник передаёт компьютеру – передатчику отрицательную квитанцию.

После успешной передачи файлов программа возвращается в исходное состояние.

## Функции кодирования/декодирования

Кодирование и декодирование данных в программе осуществляется циклическим кодом с помощью функций:

Кодирование - **byte[] Kodir(byte[] Ish, int k, stringPorPolinom);**

Декодирование - **byte[] Dekodir(string Ish, int k, stringPorPolinom);**

Проверка правильности полученного кода - **bool Proverka(string Ish, int k,string PorPolinom).**

В этих функциях **int k** - число кодируемых за раз битов.

## Форматы кадров

Типы кадров:

* I - информационный;
* A - квитанция с подтверждением об успешной передаче;
* R – отрицательная квитанция;
* Y - пользователь согласен и готов принимать файл; N - пользователь отклонил прием файла;
* E– конец передачи.

### Информационный кадр.

Для передачи информационных кадров:

[стартовый байт][тип кадра][длина информационной части(в байтах)][Информационные байты][стоповый байт] - все поля, кроме информационного - по одному байту

### Управляющий кадр.

Структура управляющего кадра:

[стартовый байт][тип кадра][стоповый байт]

Передаем заголовок, затем ждем: если получаем Y - кадр, то передаем сам файл, если получаем N - кадр - ничего не делаем. E– кадр указывает на то, что передан весь файл.

## ПОЛЬЗОВАТЕЛЬСКИЙ УРОВЕНЬ

## Формы

Главная форма:

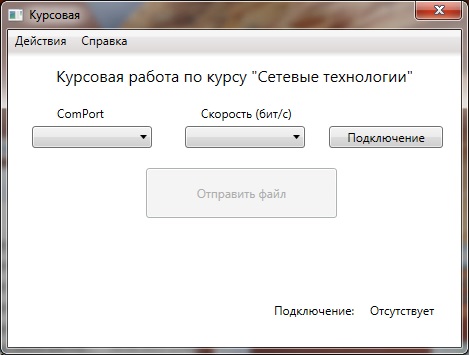


Рис. 5. Главная форма.

На главной форма присутствуют следующие элементы:

1. Надписи
   1. «Курсовая работа по курсу «Сетевые технологии»
   2. Надпись состояния подключения («Подключение: Отсутствует»,

«Подключение: Активно»)

* 1. Надписи «ComPort» и «Скорость (бит/с)»

1. Кнопка «Подключение» и параметры, для подключения
2. Кнопка «Отправить файл» для выбора и отправки файла
3. Меню
   1. Действия
      1. Отправить файл – открывает диалог для выбора отправляемого файла
      2. Закрыть – закрывает окно программы
   2. Справка
      1. О программе – выдает диалоговое сообщение с назначением программы и кратким описанием ее применения:

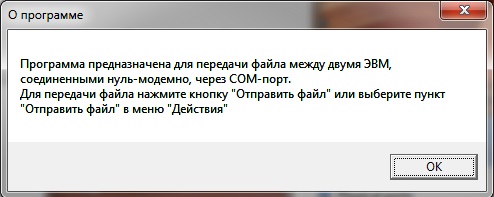


Рис. 6. Справка о программе.

После успешного подключения на главном окне, кнопка отправки файла станет активной. Окно будет выглядеть следующим образом:

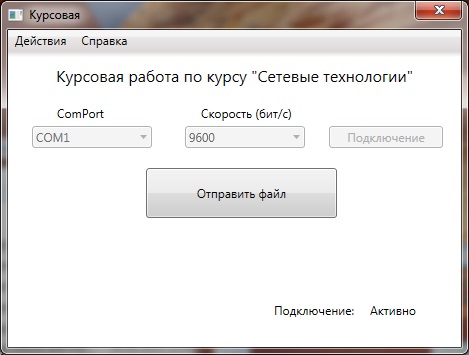


Рис. 8. Успешное подключение.

Во время передачи файла на принимающей машине появляется надпись: «Идет прием файла. Подождите…»:

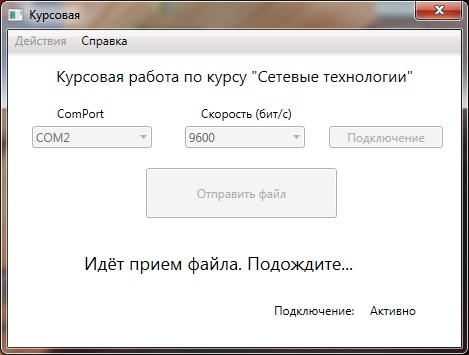


Рис. 9. Приём файла, принимающий ПК.

Во время передачи файла на передающей машине появляется полоса заполнения показывающая ход выполнения передачи:

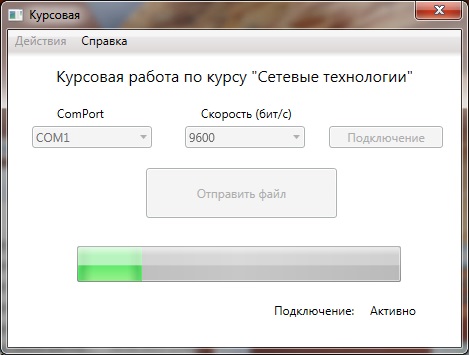


Рис. 10. Отправка файла, передающий ПК.