Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

ДИСЦИПЛИНА Сетевые технологии в АСОИУ

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

Локальная безадептерная сеть

Выполнили:

ИУ5-63Б		**Богданов Д.А.**
(Группа)	(Подпись, дата)	(Фамилия И.О.)
ИУ5-63Б		**Сёмкин Н.Е.**
(Группа)	(Подпись, дата)	(Фамилия И.О.)
ИУ5-63Б		**Попов М.А.**
(Группа)	(Подпись, дата)	(Фамилия И.О.)

Руководитель курсовой работы:

	Галкин В.А.
(Подпись, дата)	(Фамилия И.О.)

Консультант:

(Подпись, дата)	(Фамилия И.О.)

Москва – 2020г.

Содержание

1. Введение

Данная программа, выполненная в рамках курсовой работы по предмету «Сетевые технологии в АСОИУ», предназначена для организации обмена файловыми текстовыми сообщениями между соединёнными с помощью интерфейса RS232C компьютерами. Программы позволяет обмениваться 3 компьютерам, соединенным через COM-порты, файловыми текстовыми сообщениями, при условии запуска этой программы на трех компьютерах.

2. Требования к программе

2.1. Требования к функциональным характеристикам:

К программе предъявляются следующие требования. Программа должна:

- Устанавливать соединение между тремя компьютерами и контролировать его целостность;
- Обеспечивать правильность передачи и приема данных с помощью кодирования пакета по коду Хемминга [7,4];
- Обеспечивать функцию передачи файла из каталога источника;
- Контролировать процессы, связанные с получением, использованием и освобождением различных ресурсов ПЭВМ;
- Оповещать источник об открытии файла.

При возникновении ошибок обрабатывать их, а в случае необходимости:

- Извещать пользователя своей ПЭВМ,
- Извещать ПЭВМ на другом конце канала.

Программа выполняется под управлением OS Windows 7 и выше.

Было решено выполнить реализацию программы с помощью среды разработки C#.

3. Определение структуры программного продукта

При взаимодействии компьютеров между собой выделаются несколько уровней: нижний уровень должен обеспечивать соединение компьютера со средой передачи, а верхний – обеспечить интерфейс пользователя. Программа разбивается на три уровня: физический, канальный и прикладной (см. Лист 1 «Структурная схема программы»).

- Физический уровень предназначен для сопряжения компьютера со средой передачи.
- Канальный уровень занимается установлением и поддержанием соединения, формированием и проверкой пакетов обмена протоколов верхних модулей.
- Прикладной уровень занимается выполнением задач программы.

4. Физический уровень

4.1. Функции физического уровня

Основными функциями физического уровня являются:

4.1.1. Установление параметров COM-порта;

4.1.2. Установление, поддержание и разъединение физического канала.

4.2. Описание физического уровня

Последовательная передача данных означает, что данные передаются по единственной линии. При этом биты байта данных передаются по очереди с использованием одного провода.

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.

В самом конце передаются один или два стоповых бита **STOP**, завершающих передачу байта.

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 данных. Она также должна быть одинаковой для передатчика и приемника. Измеряется в бодах.

4.2.1. Интерфейс RS-232C

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

Последовательная передача данных означает, что данные передаются по единственной линии. При этом биты байта данных передаются по очереди с использованием одного провода. 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
GND (Signal Ground)	102	7	5
DSR (Data Set Ready)	107	6	6
RTS (Reguest To Send)	105	4	7
CTS (Clear To Send)	106	5	8
RI (Ring Indicator)	125	22	9

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 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.

4.2.2. Нуль-модемный интерфейс

Обмен сигналами между адаптером компьютера и модемом (или 3-м компьютером, присоединенным к исходному посредством 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 выставляет сигнал DTR, который постоянно 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:

1. RI-1 + DSR-1 — DTR-2;
2. DTR-1 — RI-2 + DSR-2;
3. CD-1 — CTS-2 + RTS-2;
4. CTS-1 + RTS-1 — CD-2;
5. RD-1 — TD-1;
6. TD-1 — RD-1;
7. SG-1 — SG-2;

Знак «+» обозначает соединение соответствующих контактов на одной стороне кабеля.

4.2.3. Асинхронная передача данных

Асинхронный режим передачи является байт-ориентированным (символьно-ориентированным): минимальная пересылаемая единица информации — один байт (один символ) (Рисунок 4). Передача каждого байта начинается со старт-бита, сигнализирующего приемнику о начале посылки, за которым следуют биты данных и, возможно, бит четности. 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 приемник и 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

Рисунок 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 ложным и приемник снова переходит в состояние ожидания. Об этой ошибке приемник может не сообщать.
- Если во время, 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.

4.3. Реализация физического уровня

Пространство имен System.IO.Ports предлагает широкие возможности по настройке COM-порта.

4.3.1. 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. **Port** – объект класса SerialPort, который используется для определения COM-порта.

Port.OpenPort() – функция открытия COM-порта.

После открытия порта производится его сброс. Порт очищается сам при считывании всех байтов с помощью функции **Port.ReadExisting().**

Вызов этой функции позволяет решить две задачи: очистить очереди приема/передачи в драйвере и завершить все находящиеся в ожидании запросы ввода/вывода.

Управление COM-портом осуществляется с помощью методов:

Port.IsConnected() – атрибут, отвечающий за то, открыт порт или нет;

Port.setPortName() – установка имени порта;

4.3.2. Закрытие порта

Закрытие порта осуществляется с помощью функции **Port.ClosePort()**.

4.3.3. Передача и прием данных

Для передачи/приема данных функции выполняются по логике программы с помощью операций записи/чтения из буферов порта.

Функция для передачи данных – **Port.WriteData** (string input, FrameType type);

Функция приема данных – **Port.GetData** (int typeID);

Функция считывания типа кадра с порта при приеме – **Port_DataReceived** (object sender, SerialDataReceivedEventArgs e)

5. Канальный уровень

5.1. Функции канального уровня

На канальном уровне выполняются следующие функции:

5.1.1. Запрос физического соединения;
5.1.2. Управление передачей кадров;
5.1.3. Обеспечение необходимой последовательности блоков данных, передаваемых через межуровневый интерфейс;
5.1.4. Контроль и исправление ошибок;
5.1.5. Запрос на разъединение физического соединения.

5.2. Передача данных

Перед началом передачи данных требуется установить соединение между тремя сторонами, тем самым проверяется доступность приемного устройства и его готовность воспринимать данные. Для этого передающее устройство посылает специальную команду: запрос на соединение, сопровождаемую ответом приемного устройства.

После успешного соединения компьютеры обмениваются кадрами, свидетельствующими об активности соединения.

Для передачи информации (файлов) предусмотрены специальные типы кадров.

5.3. 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.

В данной программе для обеспечения защиты информации используется код Хэмминга [7,4] с кодовым расстояние d = 3.

Число разрешенных кодовых комбинаций для кодов с d = 3 равно $N \leq 2^n(1 + n)^{-1}$. Для кодов Хэмминга выбрано предельное значение разрешенных кодовых комбинаций $N = 2^n*(1 + n)^{-1}$, а число информационных разрядов k определяется как:

$$k = \log[2^n(1 + n)^{-1}] = n - \log(n + 1).$$

Данное уравнение имеет целочисленные решения k=0, 1, 4, 11, 26, ..., которые и определяют соответствующие коды Хэмминга [3,1]-код, [7,4]-код, [15,11]-код и т. д.

Хэмминг предложил размещать проверочные разряды в позициях кодовой комбинации, кратных целой степени двойки, это позволяет по виду синдрома сразу определять ошибочный разряд кода. Рассмотрим алгоритмы кодирования и декодирования на примере [7,4]-кода Хэмминга.

5.3.1. Алгоритм кодирования:

1. Все номера позиций кода нумеруют в двоичной системе счисления, начиная с единицы p-разрядным двоичным числом:

p = ⌈log n ⌉, где ⌈ ⌉ - ближайшее большее целое, n - число разрядов кода.

2. Проверочные разряды размещают в позициях кода, кратных целой степени двойки.
3. Значение с проверочного разряда определяется как сумма по mod2 тех разрядов кода, в номере которых двоичный разряд с i-м весом равен единице.

5.3.2. Алгоритм декодирования

1. Вычисляется синдром ошибки как сумма по mod 2 тех разрядов кода, в номере которых двоичный разряд с i-м весом равен единице.
2. Если синдром равен 0 – значит, кодовая комбинация принята правильно.
3. В ином случае инвертируется соответствующий разряд.

5.4. Функции кодирования/декодирования

Кодирование и декодирование данных в программе осуществляется кодом Хемминга [7,4] с помощью методов класса **Hamming**:

int ErrorDigit (byte Error) – Возвращает ошибочную цифру;

byte [] HammingEncode74 (byte ToBeEncoded) – Кодирует один информационный байт в 2 закодированных

string HammingDecode74 (byte ToBeDecoded) – Формирует из информационного байта массив для декодирования

byte HammingSimptome74 (byte ToBeDecoded) – Вычисляет синдром ошибки

byte HammingCorrection74 (byte code, int number) – Исправляет ошибку

byte Decode (byte [] OneEncodedByteInTwoBytes) – Декодирование 2 байтов в один информационный

5.5. Типы кадров

Кадры, передаваемые с помощью функций канального уровня, имеют различное назначение.

MSG-кадр. Тип 0. Кадр, содержащий сообщение пользователя. Содержит заголовок кадра, сообщение пользователя.

ACK-кадр. Тип 1. Кадр, посылающийся для подтверждения согласия на прием файла. Содержит заголовок и сообщение.

FILEOK-кадр. Тип 2. Кадр, содержащий информацию о передаваемом файле. Содержит заголовок кадра, размер пересылаемого файла.

FRAME-кадр. Тип 3. Кадр, подтверждающий доставку информационного кадра получателю. Содержит заголовок кадра, номер доставленного информационного кадра.

FILE-кадр. Тип 4. Информационный кадр. Содержит тип кадра, тип пересылаемого файла, размер файла, общее количество кадров файла, номер текущего кадра и закодированную информацию файла.

ERR_FILE-кадр. Тип 5. Кадр, свидетельствующий об ошибке. Содержит информацию о типе кадра.

6. Прикладной уровень

Функции прикладного уровня обеспечивают интерфейс программы с пользователем через систему форм и меню. Прикладной уровень предоставляет нижнему уровню информацию о пути до пересылаемого файла.

На данном уровне обеспечивается вывод принятых и отправленных файлов в окно диалога пользователей.

6.1. Функции прикладного уровня

6.1.1. Интерфейс с пользователем через систему меню;
6.1.2. Выбор файла;
6.1.3. Отправка файла;
6.1.4. Установка режима работы;
6.1.5. Установка номера COM-портов для каналов;
6.1.6. Имя передаваемого файла указывается на передающей ПЭВМ;
6.1.7. Уведомления об ошибках и установлении соединения.

Пользовательский интерфейс выполнен в среде Visual Studio.

6.2. Оконные формы

6.2.1. Окно «Form1»

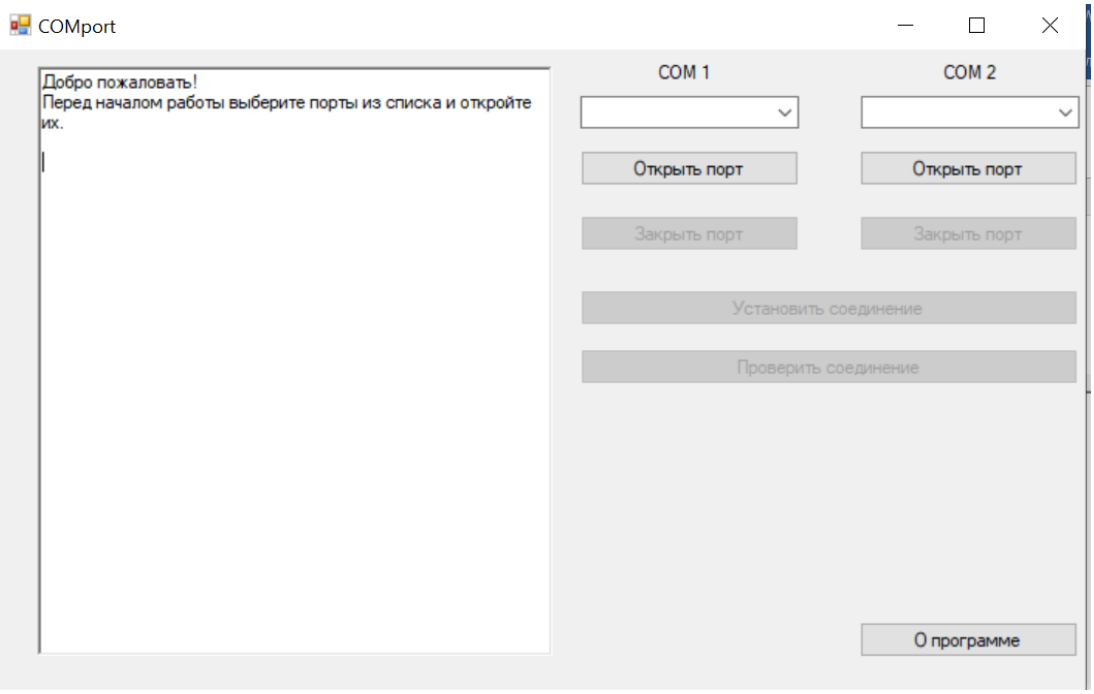

Рисунок 5 – Стартовое окно «Form1» для первого компьютера

Здесь реализованы следующие возможности:

- Отображение текущей истории.
- Открытие/закрытие портов.
- Присоединение
- Закрытие соединения
- Мониторинг активности соединения
- Отправка файла

Функции окна «Form1»

При нажатии на выпадающий список появляется возможность выбора COM-порта компьютера. (Рисунок 6)

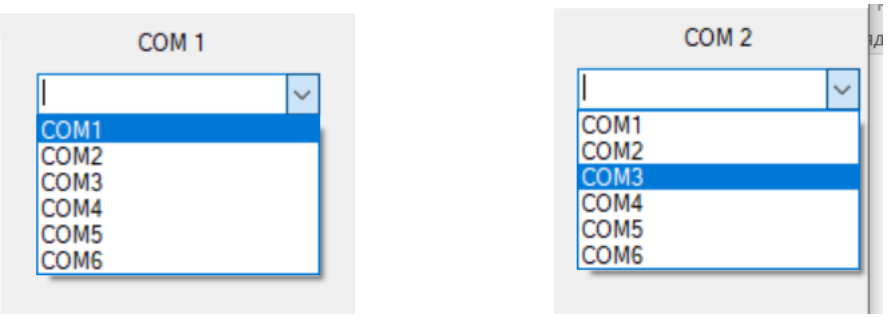

Рисунок 6 – Выбор COM-портов

При нажатии на кнопку «Открыть порт» происходит открытие порта. При этом в поле появляется соответствующая запись. Кнопки отправки файла и выбора соединения становятся активными. (Рисунок 7)

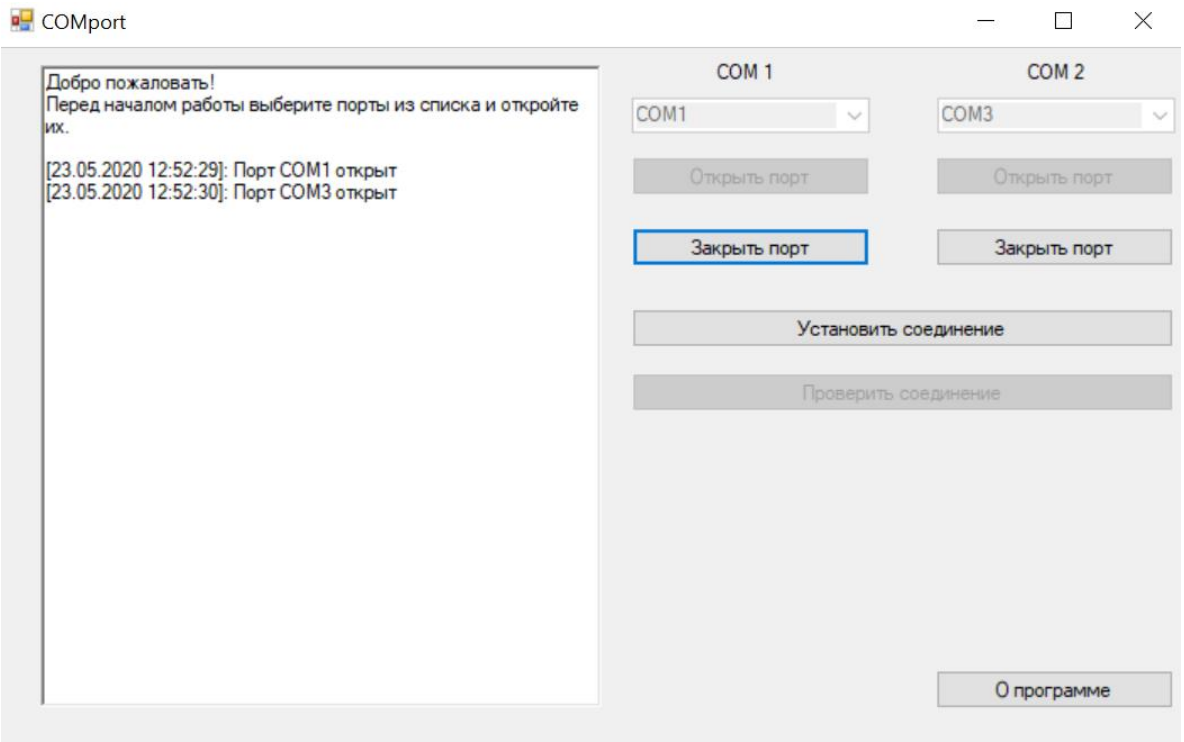

Рисунок 7 – Открытие COM-портов

Так как соединение ещё не установлено, проверить его невозможно. Важно, что для установления соединения должны быть открыты оба порта! На Рисунке 8 нажата кнопка, при условии, что оба порта открыты «Установить соединение», показано сообщение.

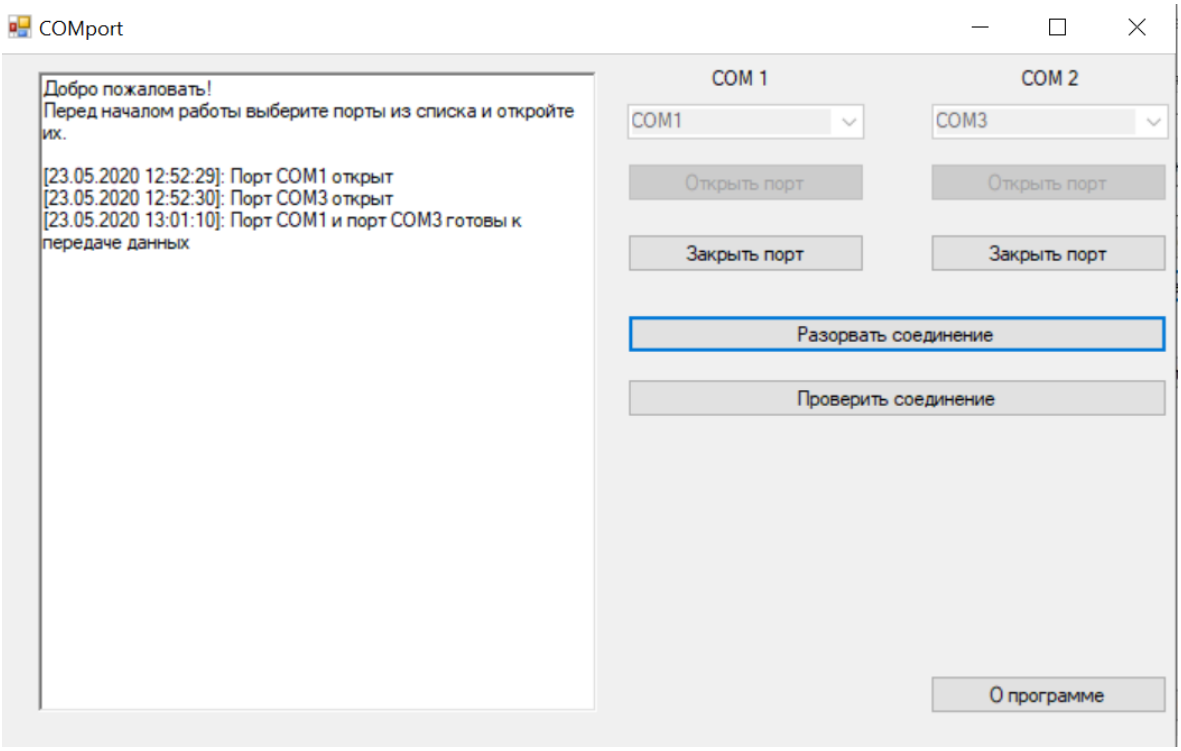

Рисунок 8 – Проверка соединения с парой открытых портов

При нажатии на кнопку «Проверить соединение» возможно проверить текущее состояние соединения. «Соединение установлено», при условии, что порты всех компьютеров открыты, и везде нажата кнопка «Установить соединение». И сразу появляются кнопки отправить на определённую ЭВМ (Рисунок 9).

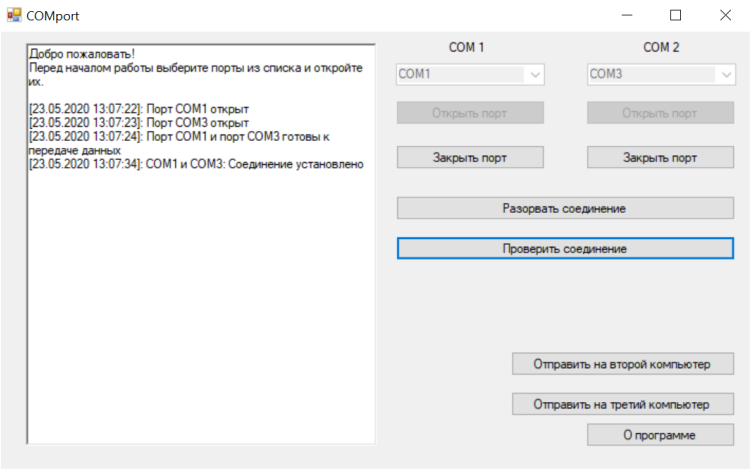

Рисунок 9 – Проверка соединения с двумя открытыми портами

Реализована возможность выбора и отправки файлов через приложение Проводник. При нажатии на кнопку для отправления на нужный ПК, открывается окно выбора файла. (Рисунок 10)

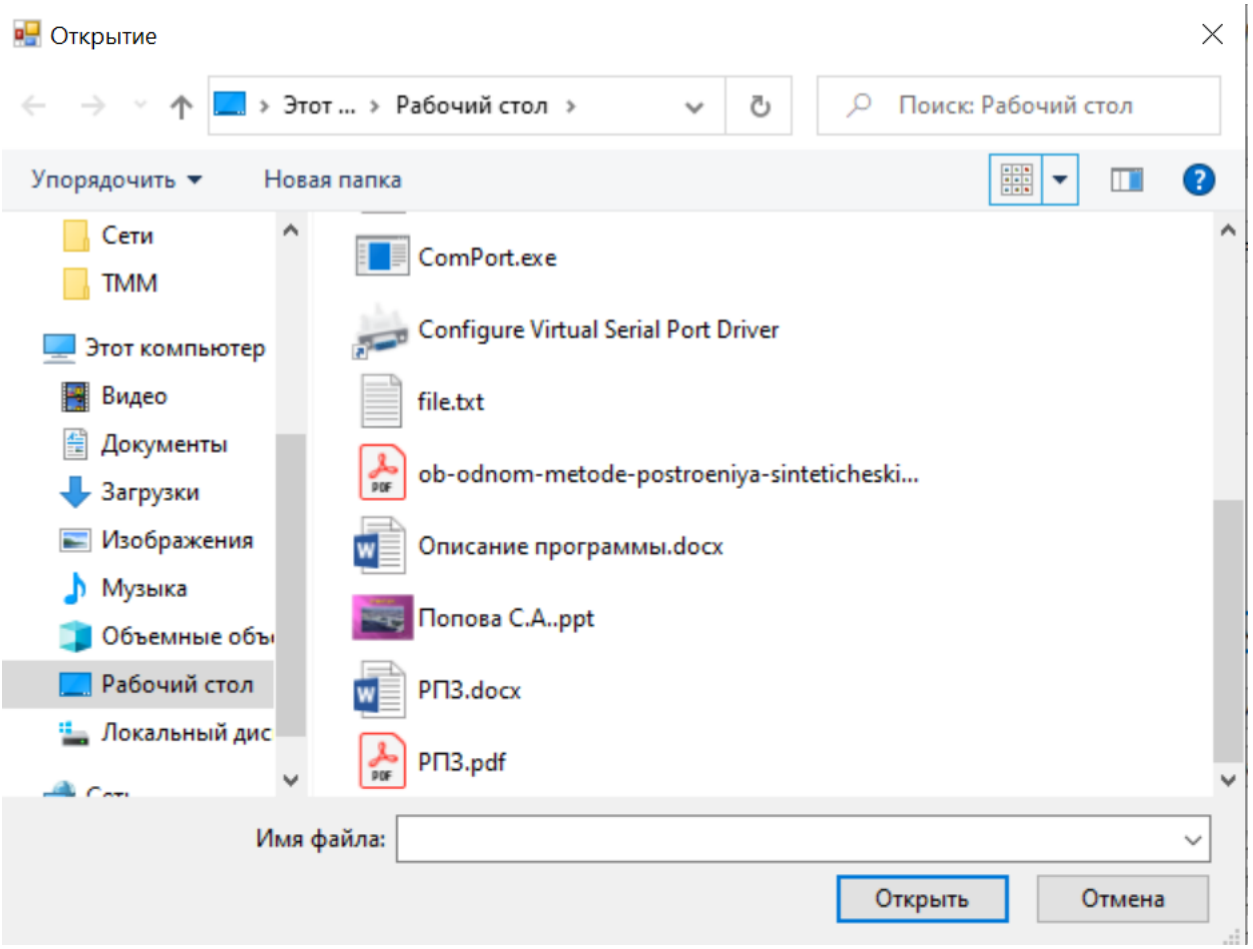

Рисунок 10 – Выбор файла

Если пользователь-отправитель нажимает кнопку «Открыть», то пользователю-отправителю выводится сообщение-уведомление об отправке файла в поле, а у пользователя-получателя выводится сообщение о размере передаваемого файла. У получателя есть возможность принять файл или отказаться. (Рисунок 11)

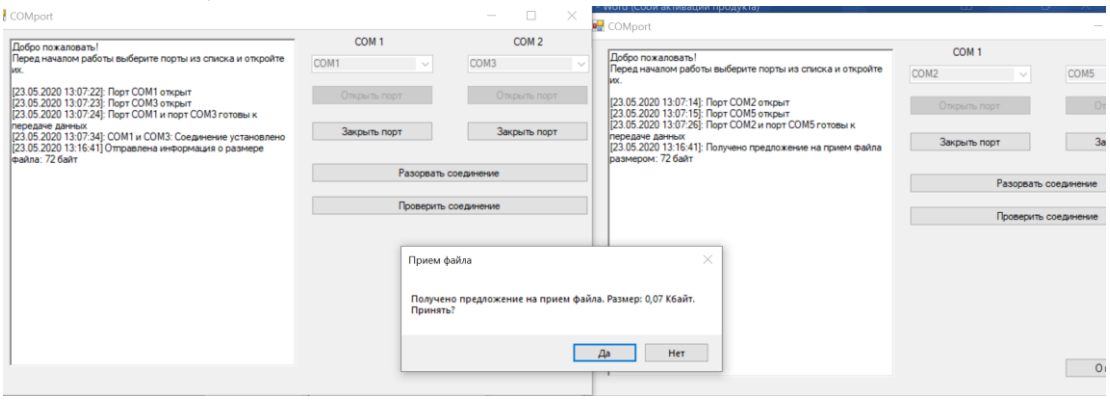

Рисунок 11 – Получение файла

У пользователя-получателя при подтверждении получения файла начинается загрузка файла, и после загрузки файла открывается окно выбора каталога для сохранения файла. (Рисунок 12)

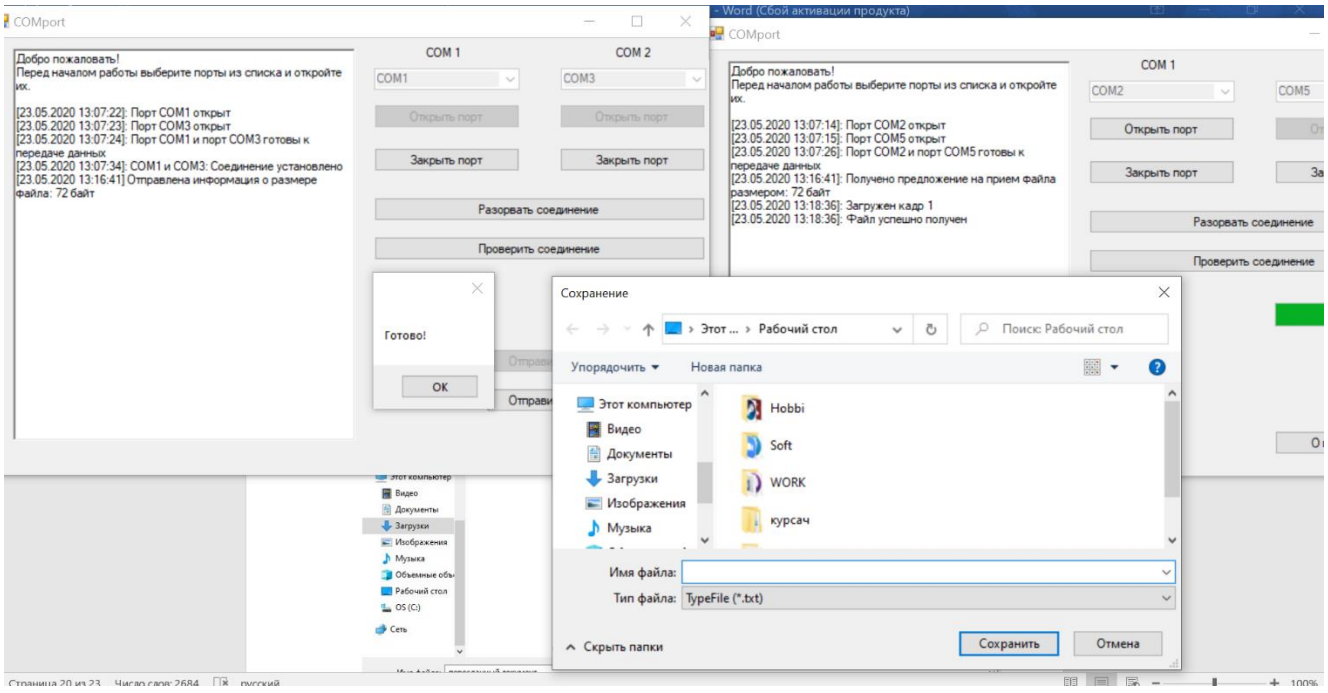

Рисунок 12 – Сохранение файла

Если получатель нажал на кнопку «Сохранить», в лог выведется соответствующее сообщение. (Рисунок 13)

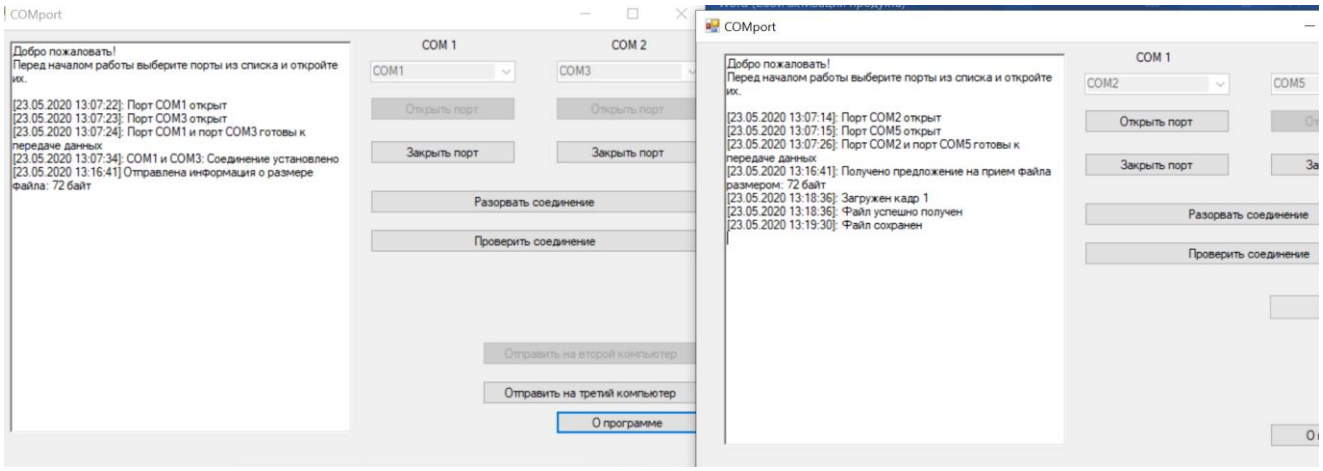

Рисунок 13 – Уведомление о сохранении файла

Если получатель откажется от получения файла, то загрузка файла производиться не будет.

Также возможно производить такие же манипуляции и с другими парами ПК.