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:

ИУ5-65Б		**Усынин Ю. А.**
(Группа)	(Подпись, дата)	(Фамилия И.О.)
ИУ5-65Б		**Камалов М. Р.**
(Группа)	(Подпись, дата)	(Фамилия И.О.)
ИУ5-65Б		**Погосян С. Л.**
(Группа)	(Подпись, дата)	(Фамилия И.О.)

Руководитель курсовой работы:

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

Консультант:

(Подпись, дата)	(Фамилия И.О.)

Москва – 2021г.

Содержание

1. Введение

Данная программа, «Local non-Adapter Network UKP», выполненная в рамках курсовой работы по дисциплине «Сетевые технологии в АСОИУ», предназначена для организации обмена файлами в локальной сети, представляющей из себя 2 ПК, соединённых между собой с помощью интерфейса RS232C нуль- модемным кабелем. Программа реализует функцию передачи информации с возможностью докачки после восстановления прерванной связи. Скорость передачи информации и параметры COM-порта заданы по умолчанию. Файл для передачи выбирается из каталога источника отправителя. Имя подкаталога для размещения полученного файла указывается на ПЭВМ- получателе. При передаче файла информация защищается [15,11]-кодом Хемминга.

2. Требования к программе

2.1. Требования к функциональным характеристикам:

К программе предъявляются следующие требования. Программа должна:

- Устанавливать соединение между двумя компьютерами и контролировать его целостность;
- Обеспечивать правильность передачи и приема данных с помощью кодирования пакета по коду Хемминга [15,11];
- Обеспечивать функцию передачи файла из каталога источника;
- Контролировать процессы, связанные с получением, использованием и освобождением различных ресурсов ПЭВМ;
- Обеспечивать функцию докачки файлов, если передача кадров была прервана.

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 принято решение 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 RS-232C является наиболее широко распространенной стандартной последовательной связью между микрокомпьютерами и периферийными устройствами. Интерфейс, который определен стандартом Ассоциации электронной промышленности (EIA), подразумевает наличие оборудования двух видов: терминального DTE и связного DCE.

Терминальное оборудование, например, микрокомпьютер может посылать или принимать данные по последовательному интерфейсу. Оно как бы оканчивает 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 нульмодемного кабеля (Рис. 1).

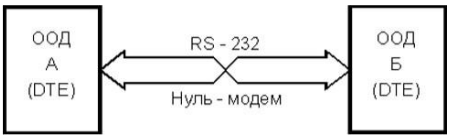

Рисунок 1. Соединение по RS-232C нуль-модемным кабелем

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

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

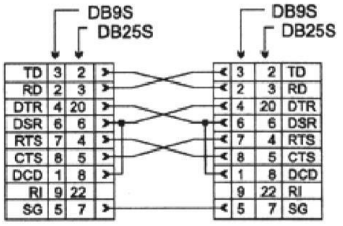

Рис. 2. Полный нуль-модемный кабель.

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**	**Направление 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

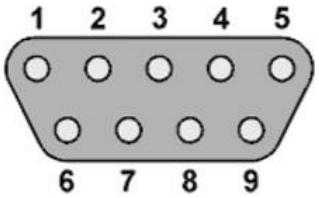

Рисунок 3 – Назначение контактов разъема DB9

Расшифровка работы каждой цепи:

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 сбросом CTS
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

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 двумя 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. Если подтверждение не получено, передающее устройство повторяет передачу кадра на установление соединения до истечения количества попыток (3). Если соединение не устанавливается, выводится сообщение в окне логов.

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

Для передачи информации (файлов, сообщений) предусмотрены специальные типы кадров.

Если один из пользователей желает разорвать соединение - посылается соответствующий кадр, а на компьютере-получателе в окне логов выводится сообщение.

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 [15,11] 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 [15,11]-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 [15,11] с помощью методов класса **Hamming**:

int ErrorDigit (char[] Error) – Возвращает номер разряда с ошибкой;

char[] HammingEncode1511 (char[] ToBeEncoded) – Кодирует 11-битый информационный вектор в 15

char[] HammingDecode1511 (char[] ToBeDecoded) – Формирует из заведомо верного 15-битного закодированного вектора 11-битный информационный

int HammingSindrome1511 (char[] ToBeDecoded) – Вычисляет синдром ошибки

char[] HammingCorrection1511 (char[] code, int number) – Исправляет ошибку в закодированном 15-битном векторе

char[] Decoded (char[] ToBeDecoded) – получая на вход 15-битный закодированный вектор, исправляет в нем ошибки и возвращает 11-битый информационный вектор

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 сообщений 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-порта для канала;
6.1.6. Имя передаваемого файла указывается на передающей ПЭВМ, а имя подкаталога для размещения полученного файла указывается на ПЭВМ-получателе;
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»

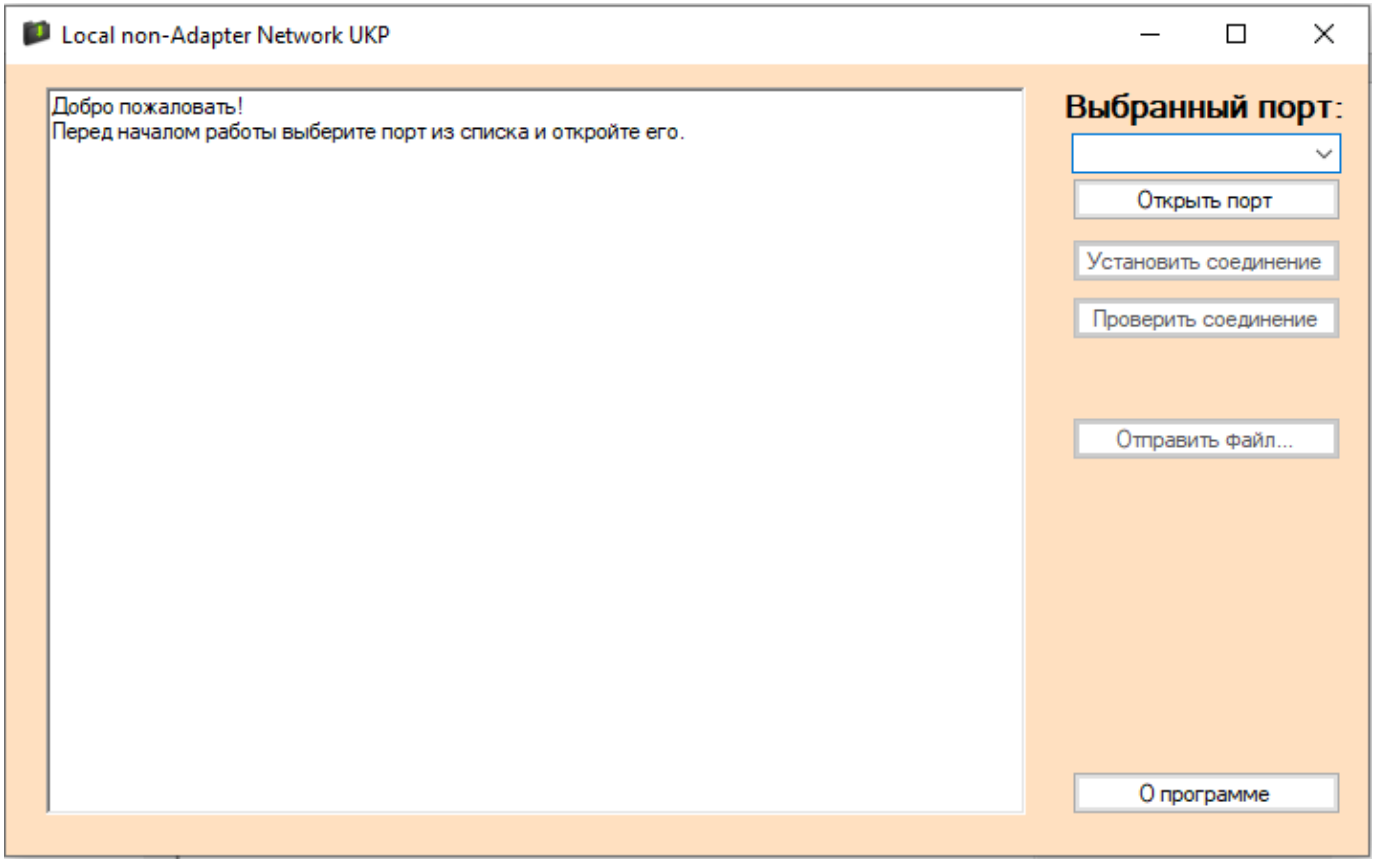

Рисунок 5 – Стартовое окно «Form1»

Здесь реализованы следующие возможности:

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

6.2.2. Функции окна «Form1»

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

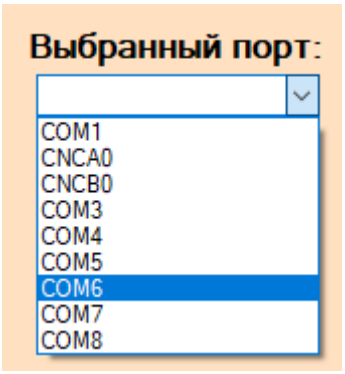

Рисунок 6 – Выбор COM-порта

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

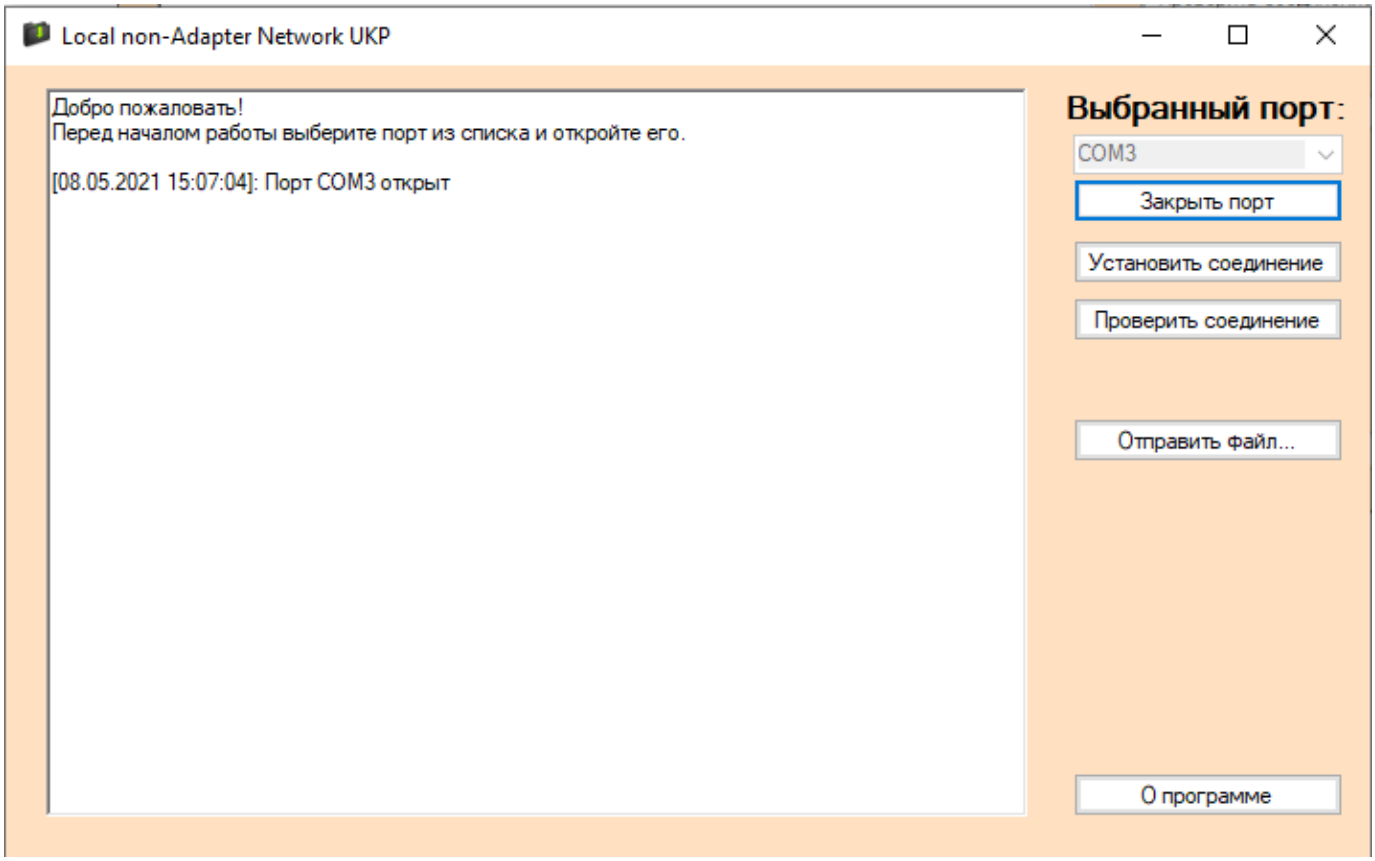

Рисунок 7 – Открытие COM-порта

При нажатии на кнопку «Проверить соединение» возможно проверить текущее состояние соединения (Рисунок 8). Важно, что для установления соединения должны быть открыты оба порта! На Рисунке 9 показано сообщение при обоих открытых портах.

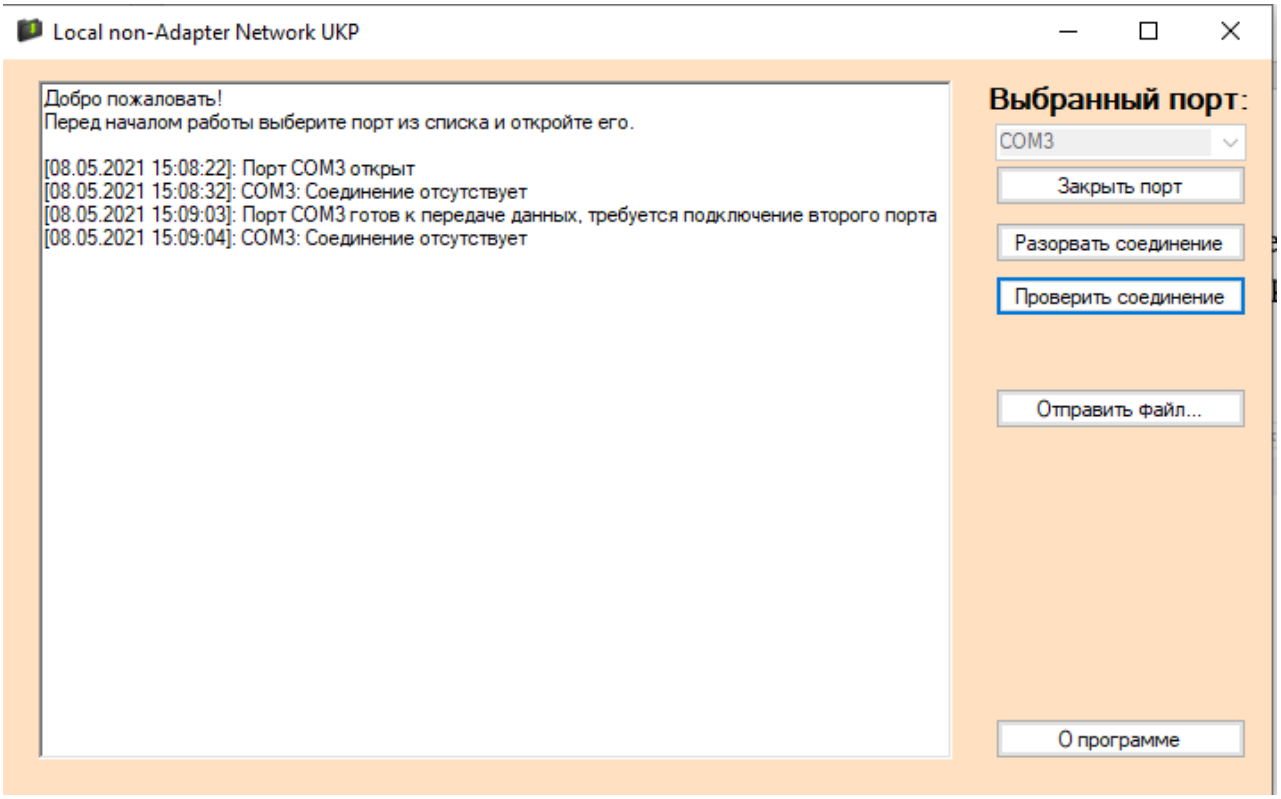

Рисунок 8 – Проверка соединения с одним открытым портом

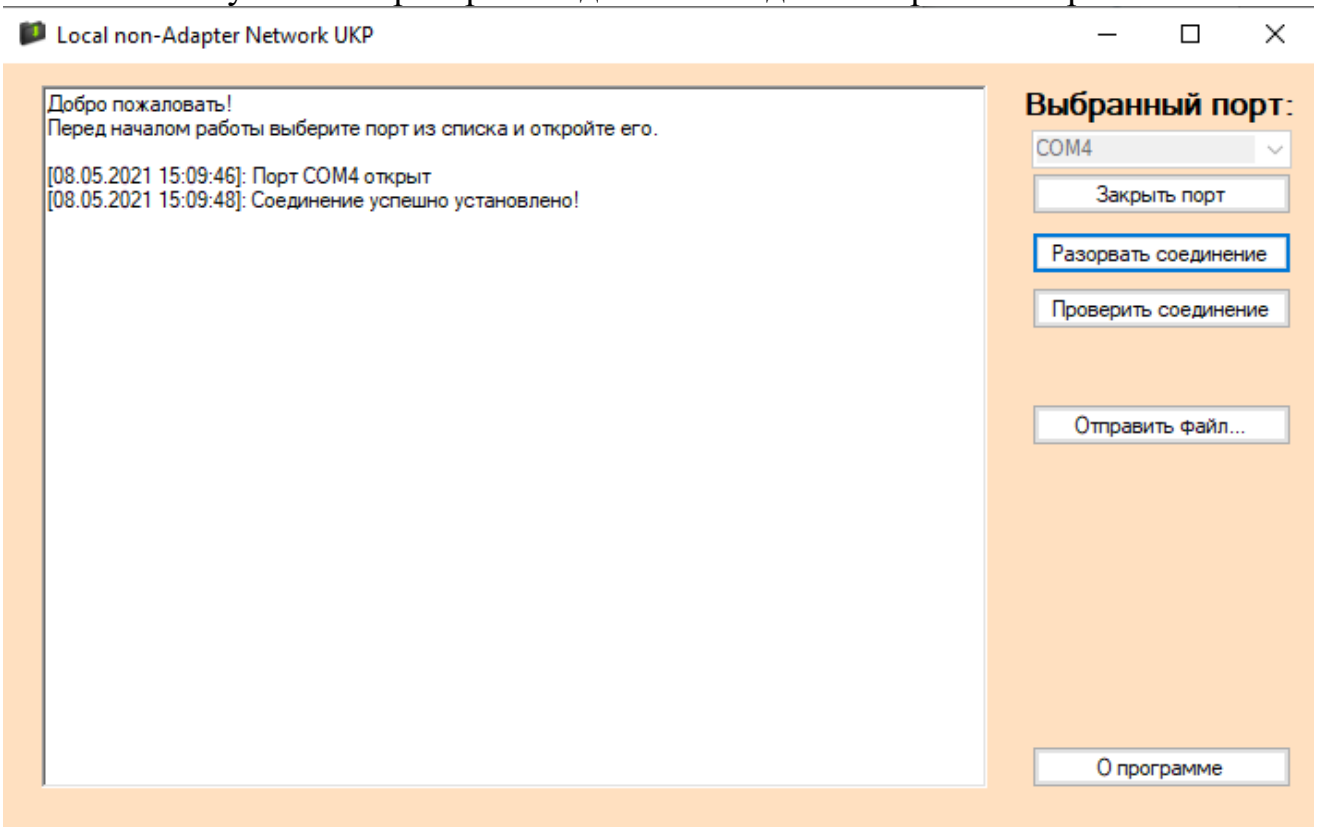

Рисунок 9 – Проверка соединения с парой открытых портов

Реализована возможность выбора и отправки файлов через приложение Проводник. При нажатии на кнопку «Отправить файл…», открывается окно выбора файла. (Рисунок 10)

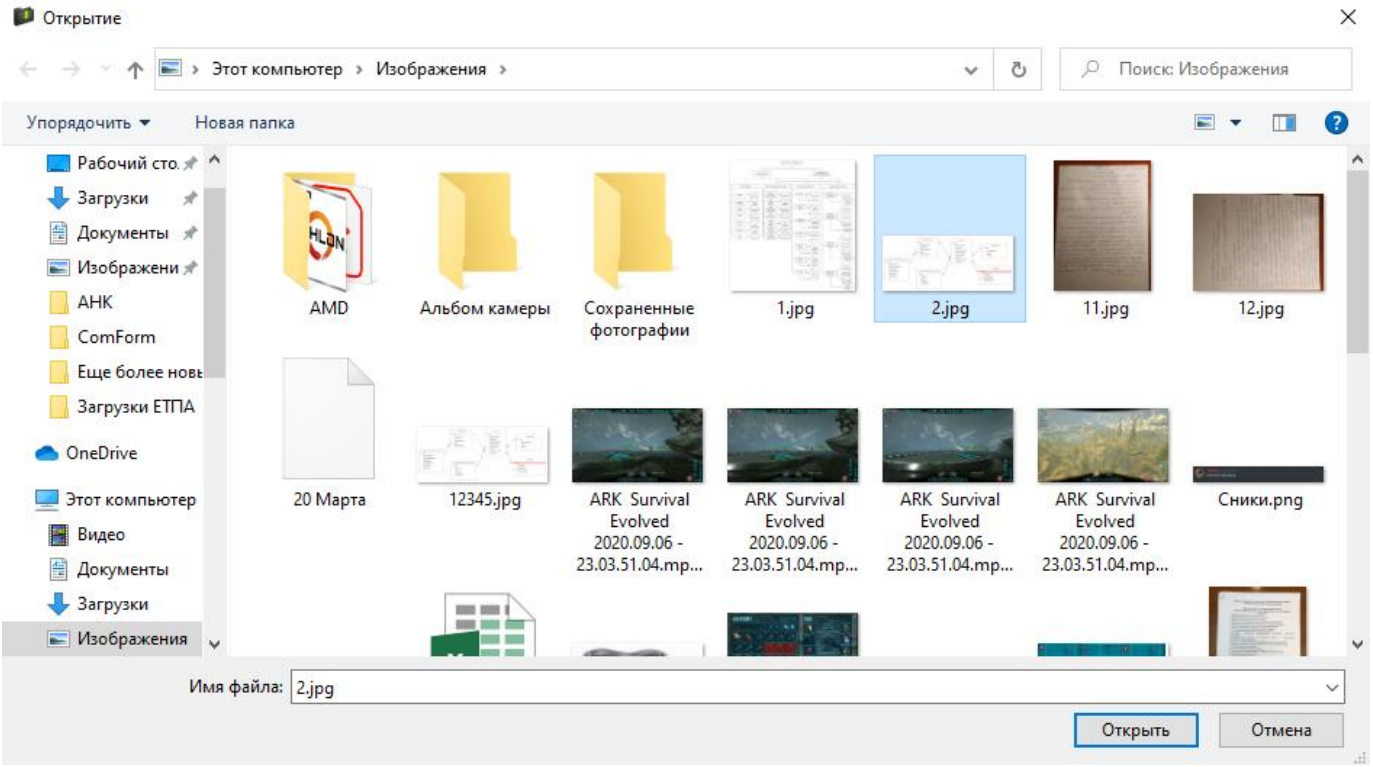

Рисунок 10 – Выбор файла

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

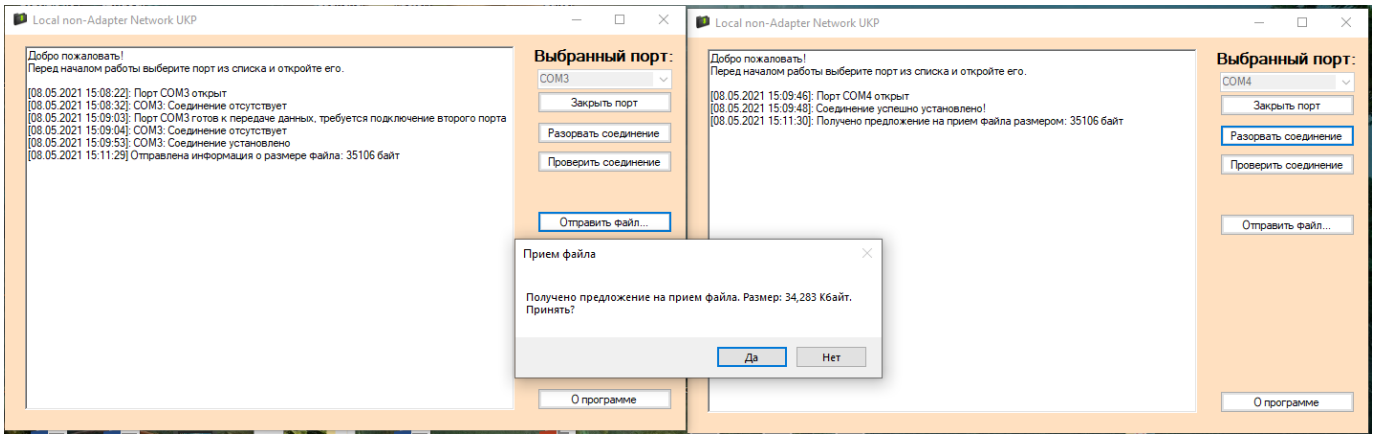

Рисунок 11 – Получение файла

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

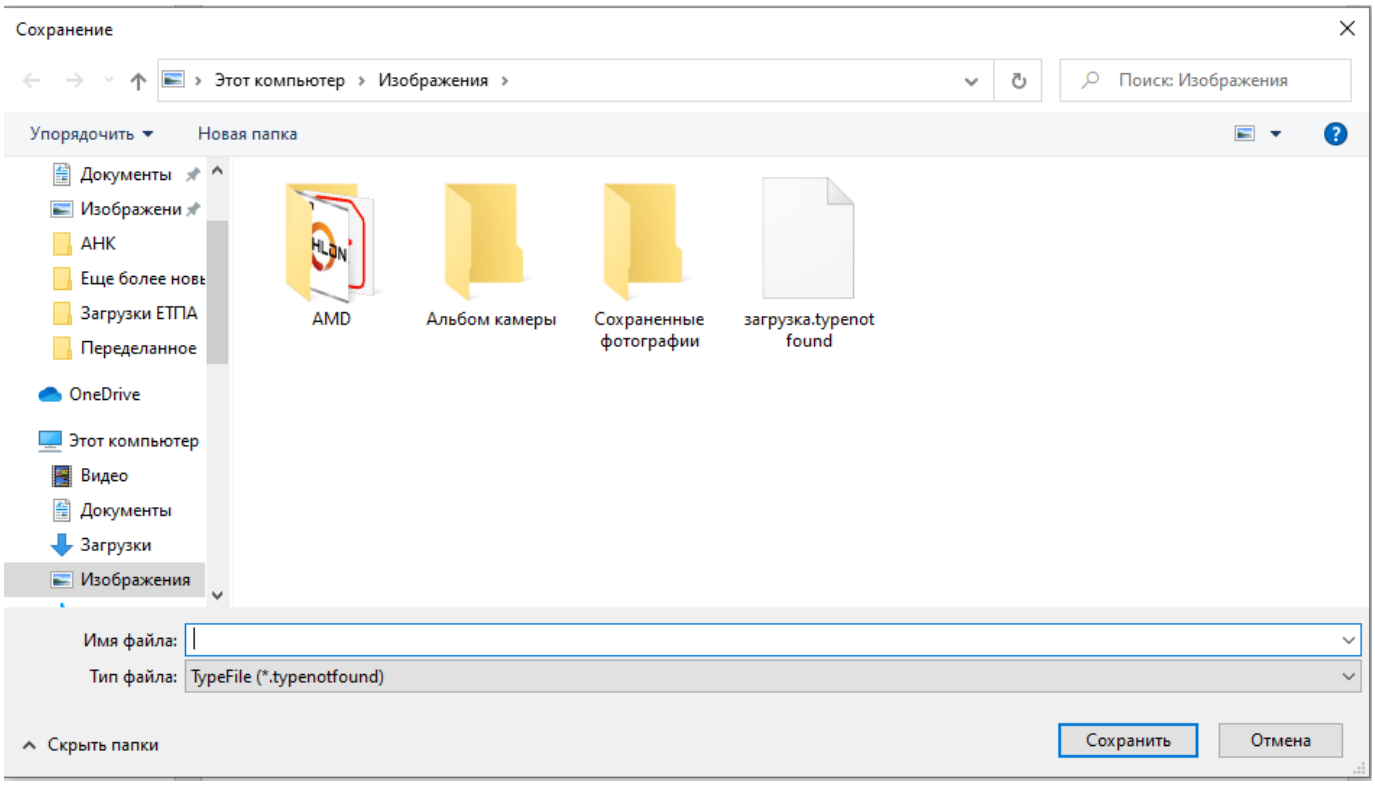

Рисунок 12 – Сохранение файла

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

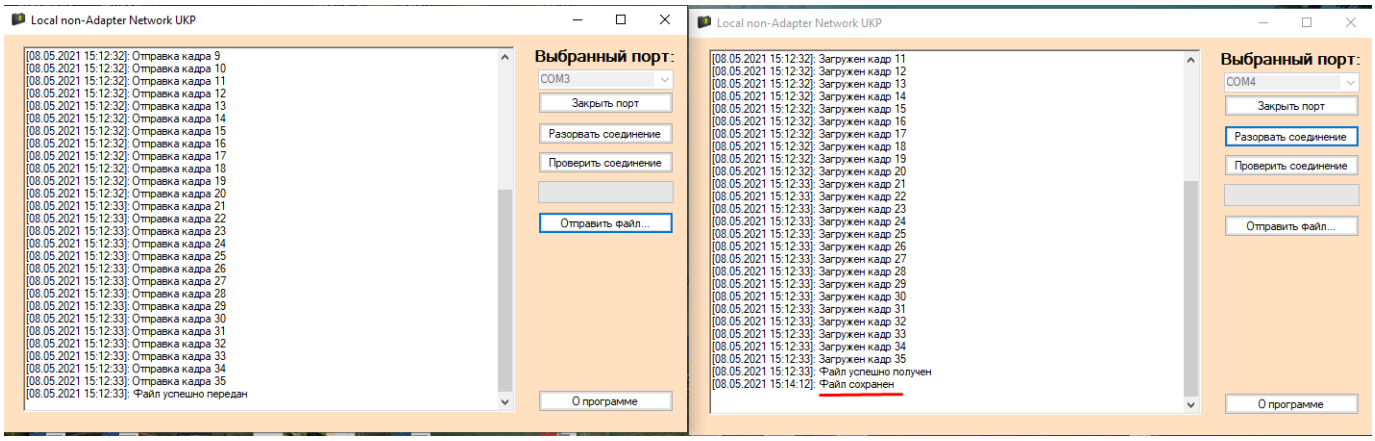

Рисунок 13 – Уведомление о сохранении файла

Если получатель откажется от получения файла, то загрузка файла производиться не будет, и в лог выведется соответствующее сообщение. (Рисунок 14)

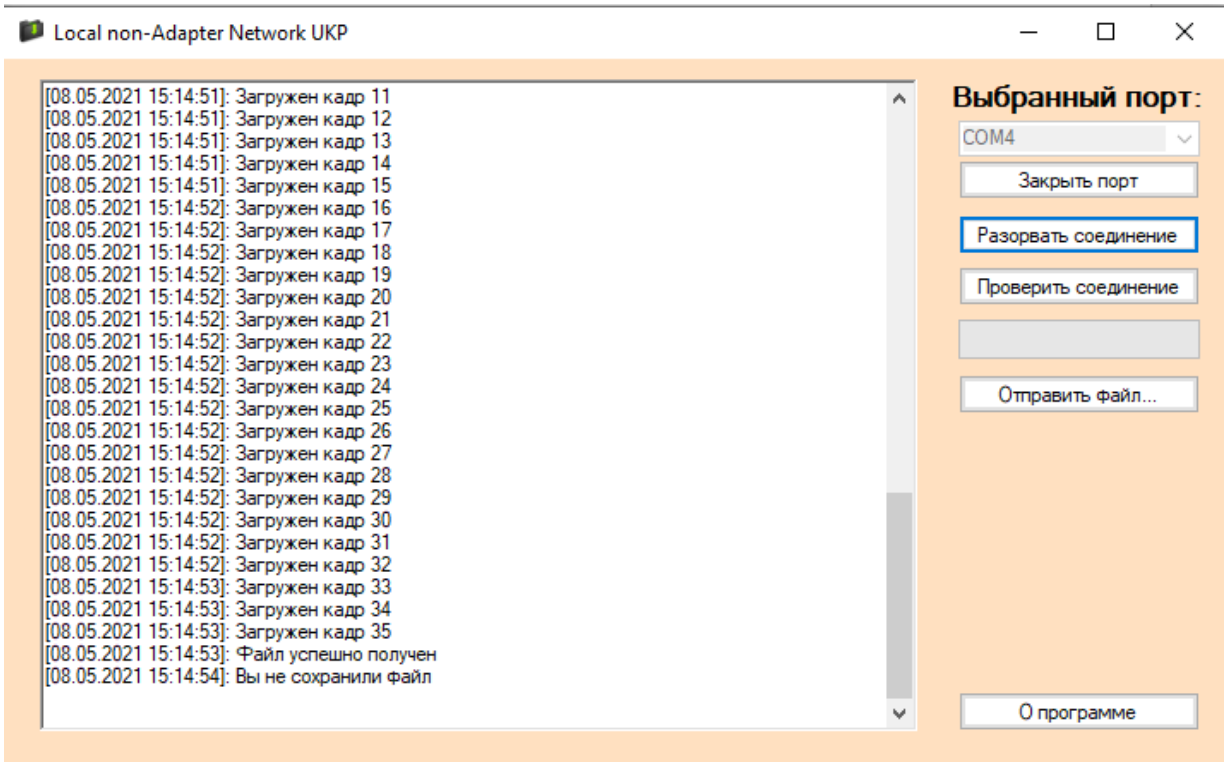

Рисунок 14 – Уведомление об отказе от получения файла

Во время загрузки файла может оборваться соединение между портами. В таком случае у пользователя-отправителя возникает окно с предложением докачки файла при восстановлении соединения. (Рисунок 15)

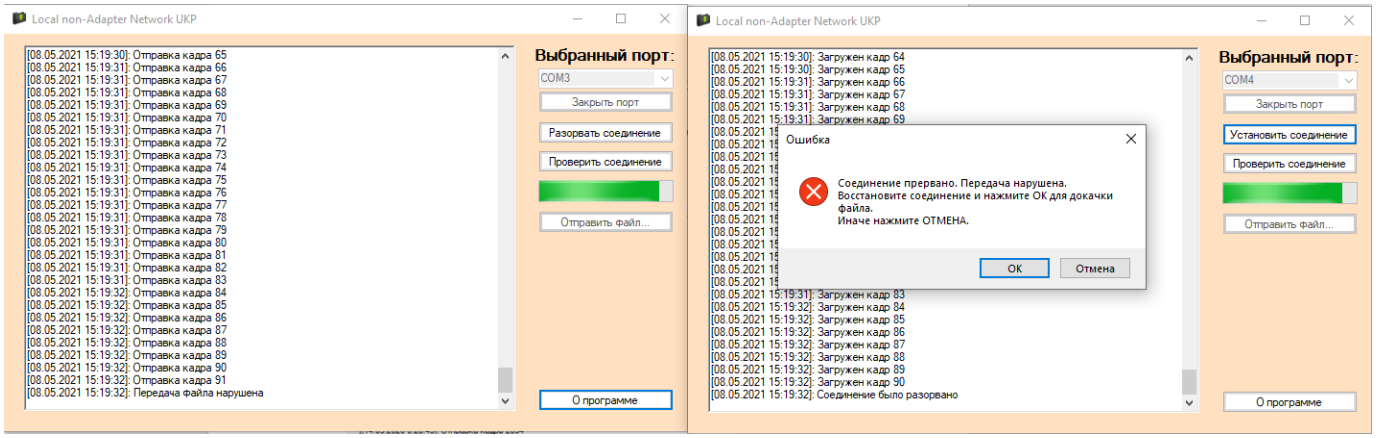

Рисунок 15 – Предложение о докачке файла при обрыве соединения

Если соединение восстановлено и нажата кнопка «ОК» возобновляется отправка файла. (Рисунок 16) Если пользователь-отправитель нажимает кнопку «ОК» без установленного соединения между парой портов, то окно с предложением докачки возникает вновь.

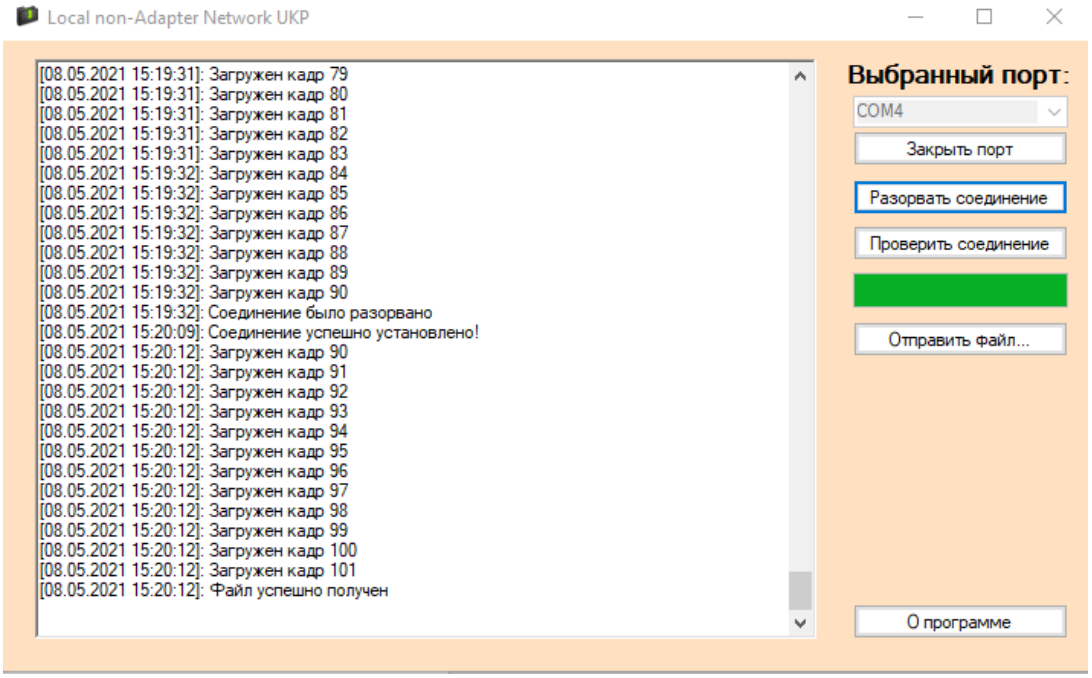

Рисунок 16 – Докачка файла

Если нажата кнопка «Отмена», то отправка файла прекращается. (Рисунок 17)

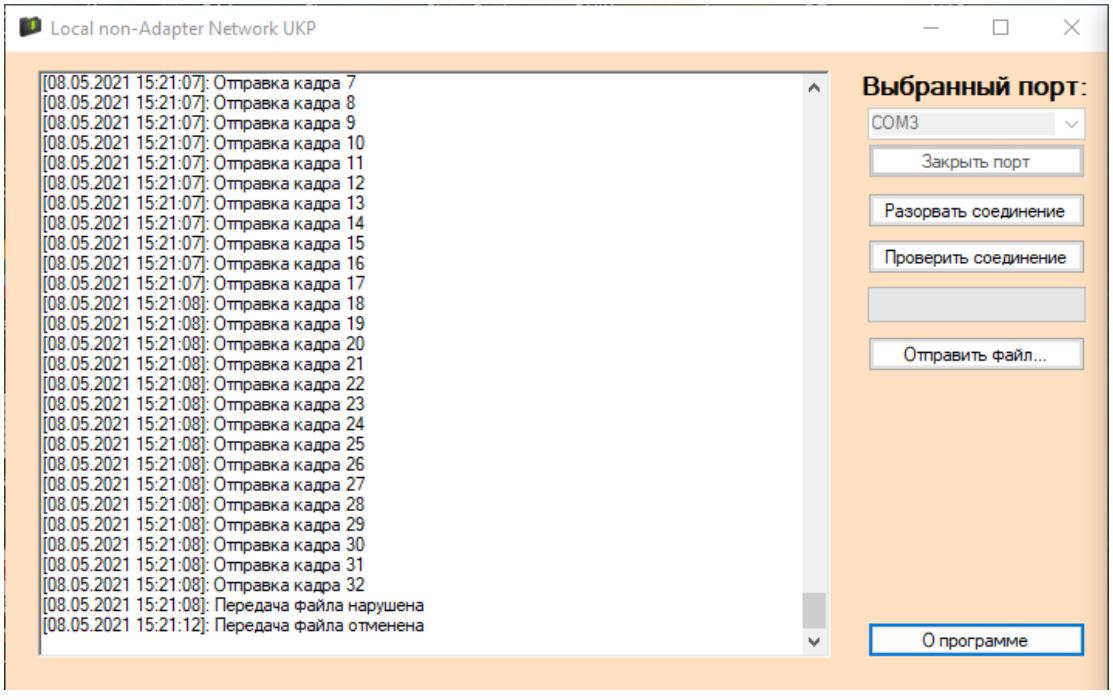

Рисунок 17 – Отмена докачки файла