

Утверждаю:

Галкин В.А. " __ " _____ 2020г.

**Курсовая работа по курсу
Сетевые технологии в АСОИУ
«Программа передачи файлов»**

Расчетно-пояснительная записка
(вид документа)

бумага А4
(вид носителя)
14
(количество листов)

Вариант 13

ИСПОЛНИТЕЛИ:

студенты группы ИУ5-64 и
ИУ5-62

Тарасов В.Ю. _____

Морозенков О.Н. _____

" __ " _____ 2020 г.

Москва - 2020

Оглавление

1. Введение.....	3
2. Требования к программе.....	3
3. Физический уровень	3
3.1 Интерфейс RS-232C	3
3.2 Асинхронная передача данных	6
3.3 Реализация физического уровня	8
3.3.1 Открытие COM-порта.....	8
3.3.2 Закрытие COM-порта.....	9
3.3.3 Передача данных	9
4. Канальный уровень	11
4.1 Защита передаваемой информации	11
4.2 Передача данных	12
4.3 Функции кодирования/декодирования	12
4.4 Типы кадров.....	12
5. Пользовательский уровень	13

1. Введение

Данная программа, выполненная в рамках курсовой работы по предмету “Сетевые технологии” предназначена для пересылки двоичных файлов между соединенными с помощью интерфейса RS-232C компьютерами.

2. Требования к программе

Программное изделие выполняется на языке Javascript под управлением Windows. Для работы программы требуются 2 ПЭВМ типа IBM PC AT (/XT), соединенные нуль-модемным кабелем через интерфейс RS-232C, либо 1 PC с эмуляцией 6 COM-портов.

3. Физический уровень

3.1 Интерфейс 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"), 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.

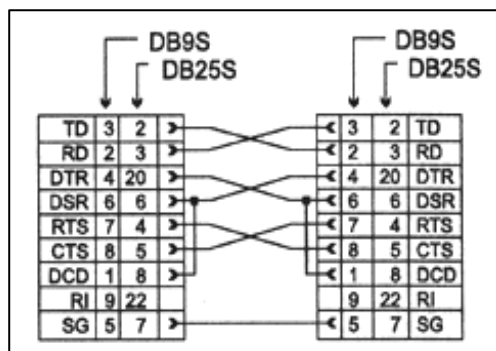


Рис. 2. Полный нуль-модемный кабель.

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	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

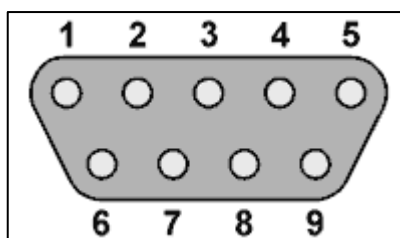


Рис. 3. Назначение контактов разъема DB9.

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 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.

3.2 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 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. 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 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.

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 двоичном 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-битные форматы 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.

3.3 Реализация физического уровня

3.3.1 Открытие COM-порта

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

SerialPort.open() – функция открытия COM-порта.

Установка параметров COM-порта осуществляется в конструкторе **SerialPort**:

dataBits – устанавливается число информационных битов;

parity – осуществляется проверка четности;

ComPort.PortName – имя порта;

stopBits – число стоповых битов;

ComPort.ReadBufferSize – устанавливается размер буфера чтения;

ComPort.WriteBufferSize – устанавливается размер буфера записи;

ComPort.Handshake – протокол установления связи;

Мониторинг физического соединения осуществляется функцией **SerialPort.get()**

3.3.2 Заккрытие COM-порта

Заккрытие порта осуществляется с помощью функции **SerialPort.close()**.

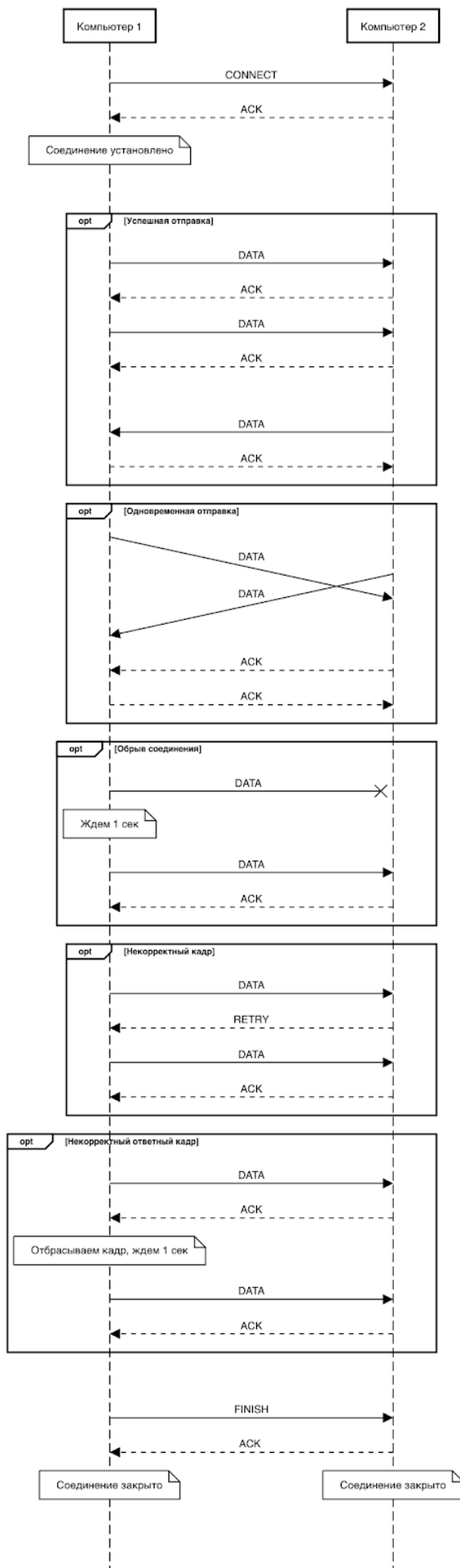
3.3.3 Передача данных

Для передачи и приема данных нет специальных функций – эти операции выполняются по логике программы с помощью операций записи и чтения из буферов порта.

Функция для передачи данных - **SerialPort.write (buff ArrayBuffer);**

Функция считывания данных - **SerialPort.read();**

3.3.4 Диаграмма работы физического уровня



4. Канальный уровень

4.1 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.

В данной программе передаваемая информация защищена циклическим кодом [7,4].

Этапы кодирования циклическим кодом:

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

Этапы декодирования 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 принятого полинома на порождающий полином (если $S(x) = 0$, ошибки нет или она не обнаружена).

$r(x)=V(x)+e(x)$, где $e(x)$ – вектор ошибки;

$e(x)=V(x)+q(x)*g(x)+S(x)$ или $e(x)=[m(x)+q(x)]*g(x)+s(x)$, т.е. синдром ошибки $s(x)$ есть остаток от деления вектора ошибки на порождающий полином.

Функция декодирующего устройства заключается в оценке полинома вектора ошибки $e(x)$ по синдрому $s(x)$.

4.2 Передача данных

В программе ведущей станцией автоматически назначается станция, инициирующая передачу. Компьютер-передатчик после подготовки текстового сообщения для передачи передаёт компьютеру-приёмнику информационные кадры (структуру кадра см. на листе «Блоки 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, дожидаясь прихода положительной квитанции после отсылки каждого блока. В случае ошибки передачи компьютер-приёмник передаёт компьютеру-передатчику отрицательную квитанцию, и передатчик повторяет передачу.

После успешной передачи сообщения программа возвращается в исходное состояние.

4.3 Функции кодирования/декодирования

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

encode(data ArrayBuffer) – кодирование;

decode(data ArrayBuffer) – декодирование;

check(byte data) – Проверка правильности полученного кода.

4.4 Типы кадров

Типы кадров:

DATA – информационный;

CONNECT – стартовый кадр начала подключения;

FINISH – кадр окончания подключения;

ACK – квитанция с подтверждением об успешной передаче;

RETRY – отрицательная квитанция;

5. Пользовательский уровень

Скриншот главной формы приложения приведен на рисунке 5.

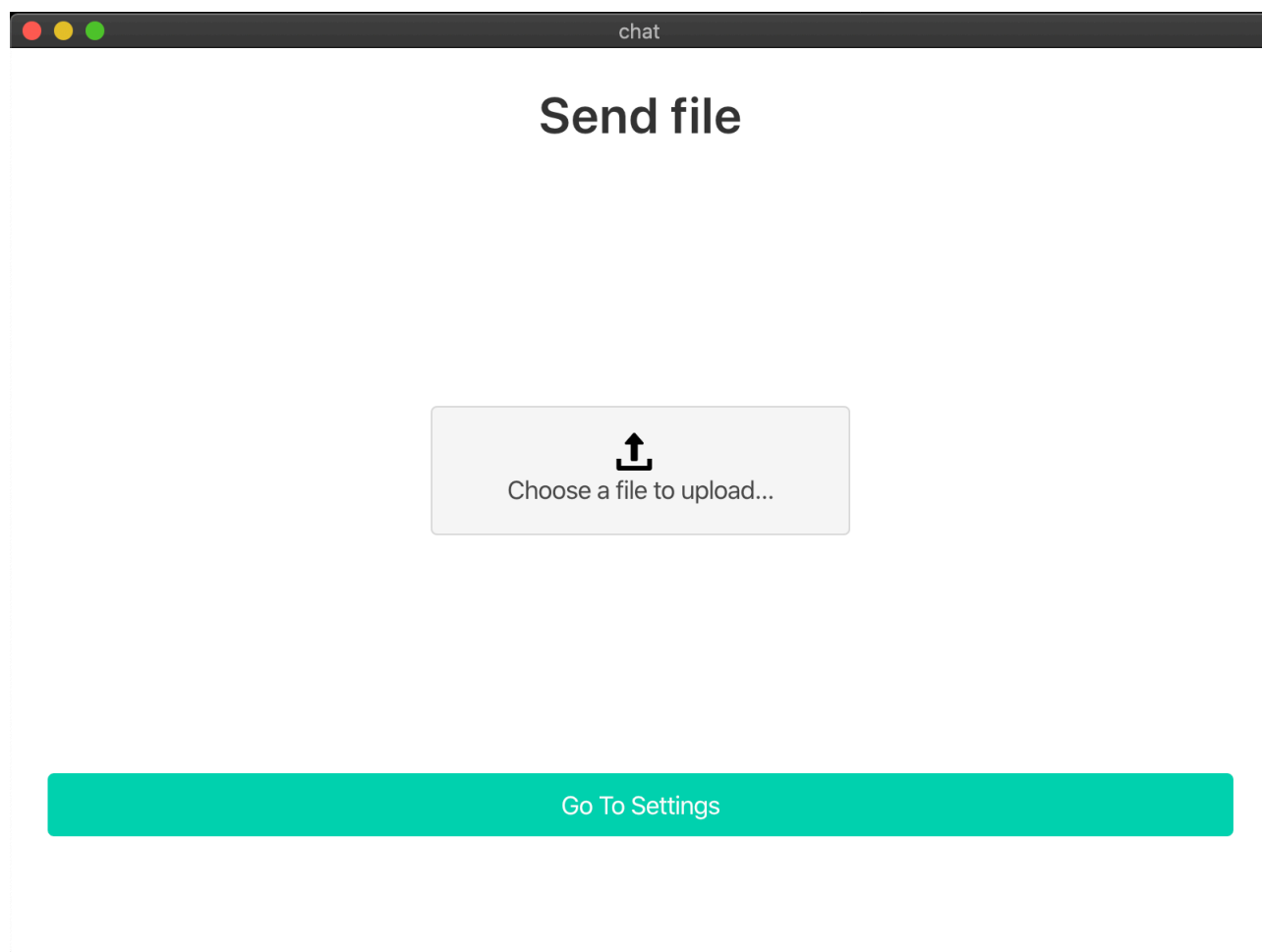


Рис. 5. Главная страница.

На данной странице есть оповещение, о состоянии соединения. Если оно будет потеряно придет оповещение.

На этой странице так же есть кнопка выбора файла для отправки. После нажатия на кнопку открывается, нативный File Explorer, в котором удобно можно выбрать файл для загрузки.

Сразу после выбора файла, вам придет оповещение, отправлен ли файл. Или нет. В случае успешной отправки файла, вы увидите “File was sent”.

Когда программа определит, что вам пришел новый файл, то вы сразу увидите форму сохранения файла, в которой вам нужно прописать полный путь, где следует сохранить новый файл. И после этого нажать кнопку “Done”.

Если сохранение файла пройдет успешно, вы увидите оповещение “File was saved”, иначе вы увидите сообщение об ошибке.

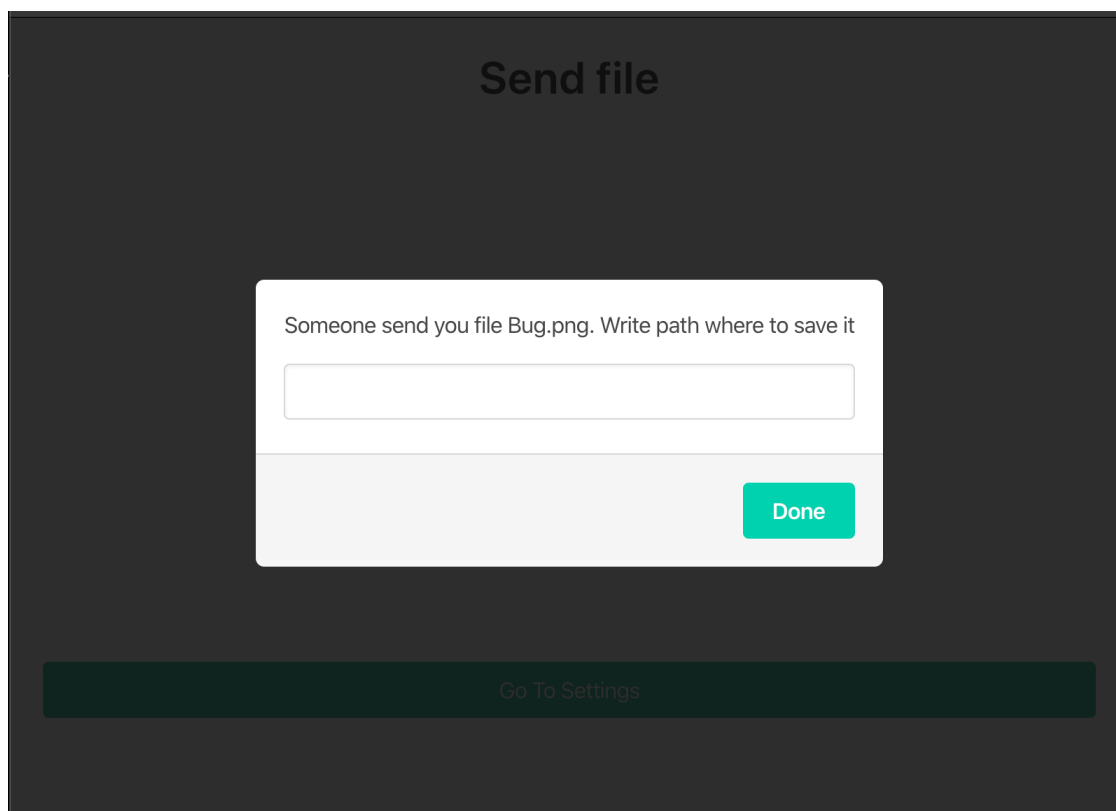


Рис. 6. Форма сохранения файла.

Чтобы изменить настройки подключения, достаточно нажать на большую зеленую кнопку “Go To Settings”, которая откроет странице настроек порта.

The image shows a window titled "Settings" with a standard macOS-style title bar (red, yellow, and green buttons). The window content is centered and contains the following elements:

- A large heading "Settings".
- A label "COM-port" above a text input field containing the text "COM1".
- A label "Speed" above a text input field containing the text "10".
- A green rectangular button labeled "Submit" positioned below the "Speed" input field.

Рис. 7. Страница настроек.