Белорусский государственный университет информатики и радиоэлектроники

# Кафедра ЭВМ

### Отчет

### по лабораторной работе № 1

«Асинхронная двунаправленная побайтная передача данных»

Проверил:

Марцинкевич В.А.

Выполнил:

студент группы 130501 Блинников В.А

Минск 2023

**Цель работы:** разработать модуль асинхронной побайтной передачи данных, соответствующий физическому уровню модели OSI, на основе последовательных интерфейсов RS-232 и RS-485.

**Краткие теоретические сведения:**

COM-порты (Communication Ports) - это аппаратные интерфейсы на компьютере, которые позволяют подключать устройства и оборудование для последовательной передачи данных. COM-порты предоставляют простой способ для взаимодействия между компьютерами и внешними устройствами, такими как модемы, принтеры, сканеры, датчики, контроллеры и многое другое.

Типы COM-портов: COM-порты могут быть реализованы как физические порты (например, RS-232) или виртуальные порты, создаваемые программно (COM-порты через USB, Bluetooth и др.).

Спецификации передачи данных: COM-порты могут использовать различные стандарты и параметры передачи данных, такие как скорость передачи (бод), биты данных, стоп-биты, паритет и другие.

Последовательная передача данных: Данные передаются по COM-портам последовательно, байт за байтом. Это отличается от параллельной передачи данных, где несколько битов передаются одновременно.

Протоколы и протоколы управления:

Для успешной передачи данных между устройствами через COM-порты часто используются определенные протоколы и протоколы управления (например, протоколы передачи файлов XMODEM, YMODEM, ZMODEM, а также протоколы управления потоком данных).

Синхронизация и управление потоком: Управление потоком данных (например, RTS/CTS и DTR/DSR) позволяет контролировать, когда и какие данные отправляются и принимаются через COM-порт.

Синхронизация данных позволяет обеспечить правильное распознавание начала и конца байтов данных.

Программное взаимодействие: Для работы с COM-портами в программном коде часто используются API (Application Programming Interface) операционной системы, такие как функции WinAPI для Windows или библиотеки для других операционных систем.

Использование COM-портов: COM-порты используются в широком спектре приложений, включая передачу данных между компьютерами, управление периферийными устройствами, сбор данных, автоматизацию и многое другое.

Ошибки и отладка: При работе с COM-портами часто могут возникать проблемы, такие как ошибки передачи данных, проблемы с настройками или конфликты. Отладка и диагностика могут потребовать использования специальных инструментов.

**Листинг кода**

**COM.cpp**

#include "COM.h"

int main()

{

system("chcp 1251 > null");

HANDLE COM\_1 = ::CreateFile(L"COM1", GENERIC\_WRITE, 0, 0, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0);

HANDLE COM\_2 = ::CreateFile(L"COM2", GENERIC\_READ, 0, 0, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0);

SetParams(COM\_1);

SetParams(COM\_2);

int count = 0;

while (1)

{

switch (menu())

{

case '1': system("cls"); WriteCOM(COM\_1, &count); \_getch(); break;

case '2': system("cls"); ReadCOM(COM\_2, &count); \_getch(); break;

case '3': system("cls"); ChangeSpeed(COM\_1); \_getch(); break;

case '4': system("cls"); ChangeSpeed(COM\_2); \_getch(); break;

case '5': system("cls"); GetParams(COM\_1, COM\_2); \_getch(); break;

case '0': system("cls"); cout << "Работа завершена."; return 0;

}

}

return 0;

}

**COM.h**

#include <windows.h>

#include <iostream>

#include "conio.h"

using namespace std;

int menu() //меню

{

system("cls");

cout << "1 - Записать в COM\_1 " << endl

<< "2 - Считать из COM\_2" << endl

<< "3 - Изменить скорость для COM\_1" << endl

<< "4 - Изменить скорость для COM\_2" << endl

<< "5 - Данные COM-портов" << endl

<< "0 - Выход" << endl;

return \_getch();

}

void SetParams(HANDLE Port) //настройка COM-порта

{

DCB Params = { 0 };

if (Port == INVALID\_HANDLE\_VALUE)

{

if (GetLastError() == ERROR\_FILE\_NOT\_FOUND)

cout << "Порт не существует!" << endl;

else cout << "Неизвестная ошибка!" << endl;

}

Params.DCBlength = sizeof(Params);

if (!GetCommState(&Port, &Params)) cout << " Ошибка получения параметров!\n";

Params.BaudRate = 9600;

Params.ByteSize = 8;

Params.StopBits = ONESTOPBIT;

Params.Parity = NOPARITY;

if (!SetCommState(Port, &Params)

cout << "Ошибка настройки!" << endl;

}

void GetParams(HANDLE Port1, HANDLE Port2) //получение параметров COM-порта

{

DCB Params1 = { 0 };

DCB Params2 = { 0 };

if (!GetCommState(Port1, &Params1) || !GetCommState(Port2, &Params2))

{

cout << "Ошибка получения параметров!" << endl;

}

cout << "Скорость COM\_1: " << Params1.BaudRate << endl << "Скорость COM\_2: " << Params2.BaudRate << endl;

}

void WriteCOM(HANDLE Port, int \*count) //запись в COM-порт

{

char data[256] = "";

cout << "Введите строку для передачи: ";

cin >> data;

DWORD size = strlen(data);

DWORD bytes;

BOOL Ret = WriteFile(Port, &data, size, &bytes, NULL);

system("cls");

cout << "Байт записано в COM\_1: " << bytes << "\nДанные: " << data << endl;

\*count += bytes;

}

void ReadCOM(HANDLE port, int \*count) //чтение из COM-порта

{

if (\*count != 0)

{

DWORD size = 0;

char buff;

cout << "Байт считано из COM\_2: " << \*count << "\nДанные: ";

for (int i = 0; i < \*count; i++)

{

ReadFile(port, &buff, 1, &size, NULL);

cout << buff;

}

}

else cout << "Нет данных для считывания";

cout << endl;

\*count = 0;

}

void ChangeSpeed(HANDLE Port) //изменение скорости COM-порта

{

system("cls");

int n;

cout << "Введите скорость: ";

cin >> n;

DCB Params = { 0 };

if (!GetCommState(Port, &Params))

{

cout << "Ошибка получения данных!" << endl;

}

Params.BaudRate = n;

if (!SetCommState(Port, &Params))

{

cout << "Ошибка настройки!" << endl;

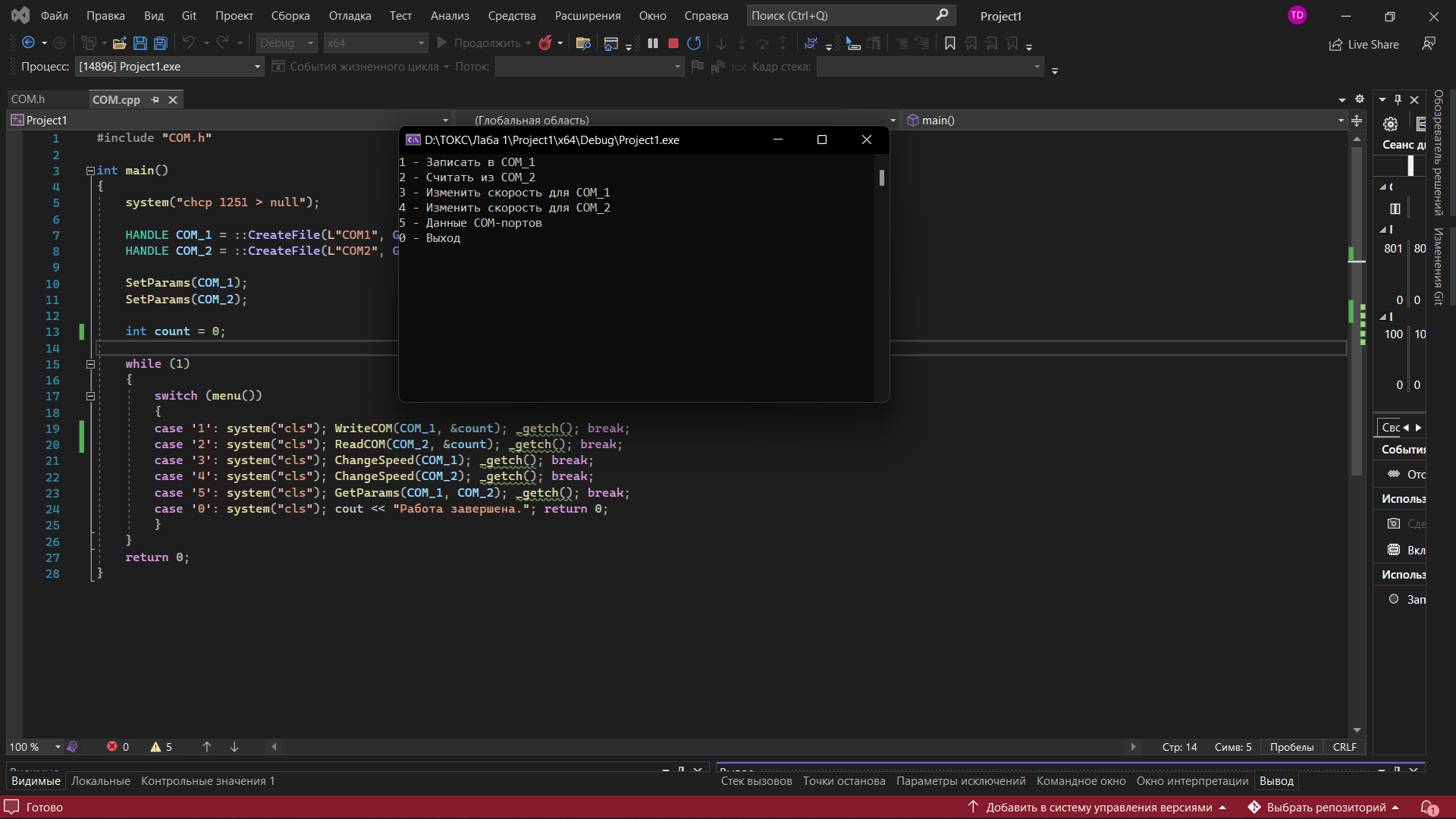
}

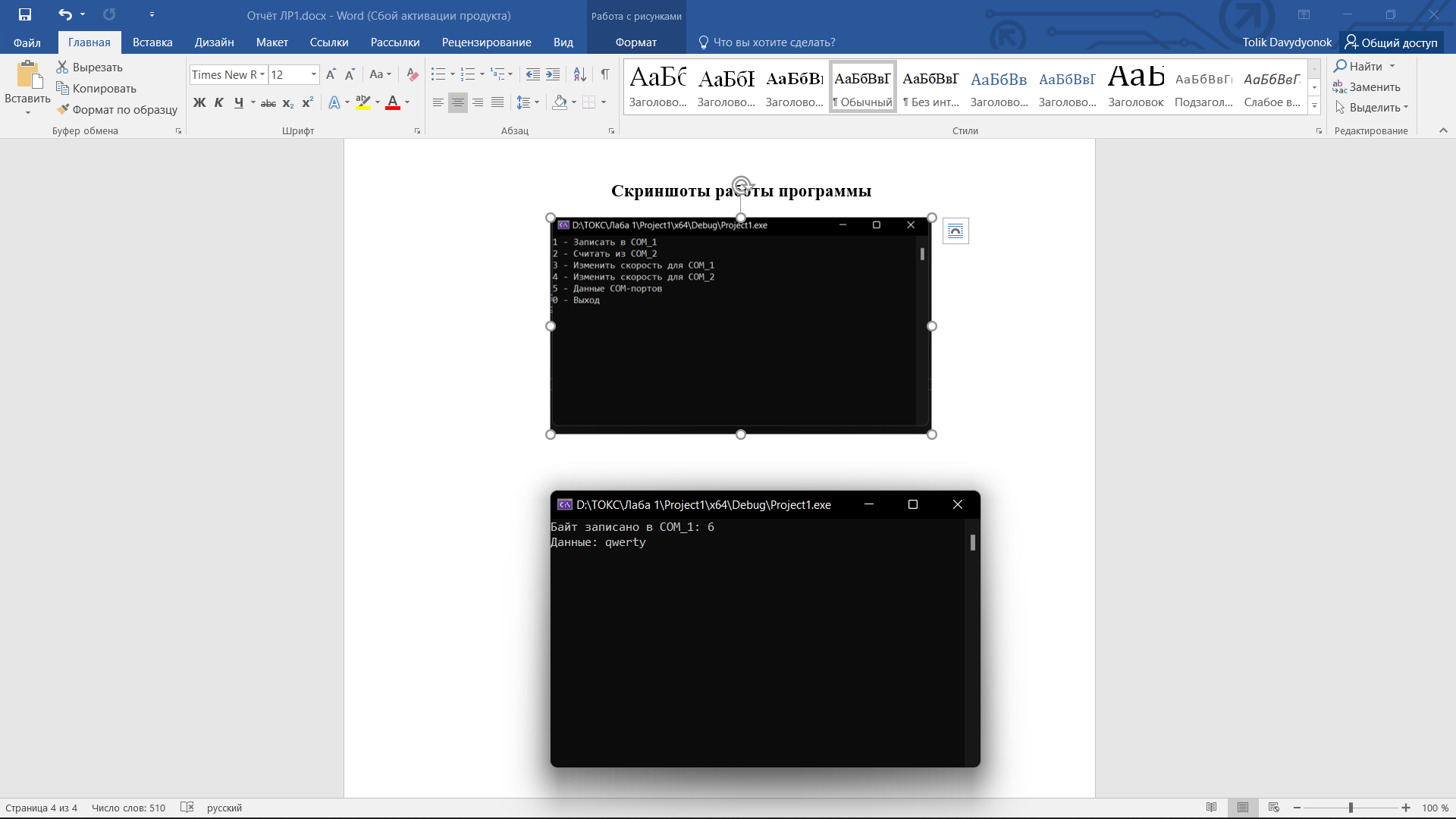
system("cls");

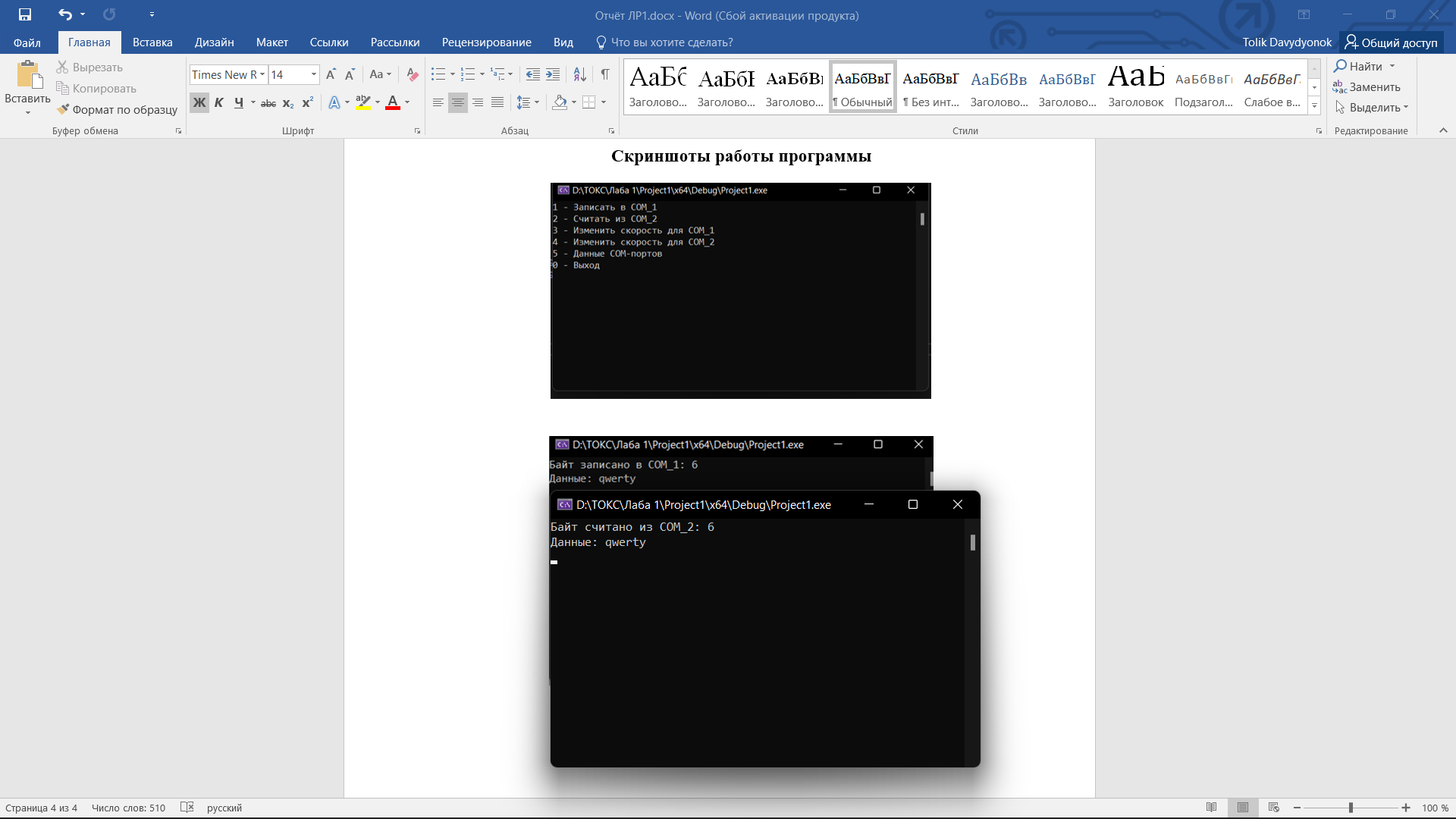
cout << "Установлена скорость: " << n << endl;

}

**Скриншоты работы программы**







**Вывод:** разработали модуль асинхронной побайтной передачи данных, соответствующий физическому уровню модели OSI, на основе последовательных интерфейсов RS-232 и RS-485.