Белорусский государственный университет информатики и радиоэлектроники

# Кафедра ЭВМ

### Отчет

### по лабораторной работе № 2

«Асинхронная пакетная передача данных»

Проверил:

Марцинкевич В.А.

Выполнил:

студенты группы 130501

Гнездилов А.М.

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:**

Алгоритмы стаффинга – это методы сжатия данных, которые позволяют уменьшить размер передаваемой информации, удаляя из них избыточную или ненужную информацию. Используются в области сетевых технологий для сжатия данных, чтобы уменьшить объем трафика и улучшить эффективность передачи данных.

Принцип алгоритма

1. Идентификация повторяющихся или однотипных данных в исходных данных
2. Замена этих данных более короткими символами, которые у получателя будут раскодированы
3. Создание таблицы или словаря, которая содержит соответствия между оригинальными данными и их заменами.
4. Передача этой таблицы получателю.

Это методы обработки данных, которые используются для обеспечения корректной передачи данных по сетям или каналам связи. Позволяют избежать конфликтов между данными и управляющими символами, которые могли бы вызвать ошибки.

* Бит-стаффинг – применяется на физическом уровне передачи данных (последовательные порты). Определенные биты вставляются в данные перед их отправкой, чтобы избежать появления последовательно идущих битов, совпадающих с управляющими символами.
* Байт-стаффинг – применяется на более высоких уровнях коммуникации (сетевой уровень).

Стоп-бит и старт-бит - это элементы последовательной передачи данных, которые используются для синхронизации и разграничений байтов данных внутри кадра данных.

* Старт-бит – это первый бит в кадре данных. Всегда имеет значение «0» и служит для определения начала передачи данных.
* Стоп-бит – это биты, которые следуют за битами данных и служат для определения конца кадра данных. Используется один стоп-бит, но могут и два для дополнительной надежности.

**Листинг кода**

**Main.cpp**

#include "COM.h"

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 - Изменить старт-бит" << endl

<< "6 - Изменить стоп-бит" << endl

<< "7 - Данные COM-портов" << endl

<< "0 - Выход" << endl;

return \_getch();

}

int main()

{

system("chcp 1251 > null");

COM COM\_1("WRITE");

COM COM\_2("READ");

COM\_1.SetParams();

COM\_2.SetParams();

int flag = 0;

while (1)

{

switch (menu())

{

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

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

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

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

case '5': system("cls"); COM\_1.ChangeStartBit(&COM\_2); \_getch(); break;

case '6': system("cls"); COM\_2.ChangeStopBit(&COM\_1); \_getch(); break;

case '7': system("cls"); COM\_1.GetParams(); COM\_2.GetParams(); \_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;

class COM

{

private:

string name;

HANDLE handle;

char stop, start;

public:

COM(string str);

void SetParams(); //настройка COM-порта

void GetParams(); //получение параметров COM-порта

void WriteCOM(int \*flag); //запись в COM-порт

void ReadCOM(int\* flag); //чтение из COM-порта

void ChangeSpeed(); //изменение скорости COM-порта

void ChangeStopBit(COM \*COM); //изменение стоп-бита COM-порта

void ChangeStartBit(COM \*COM); //изменение старт-бита COM-порта

};

**COM.cpp**

#include "COM.h"

COM :: COM(string str)

{

start = '0';

stop = '1';

if (str == "READ")

{

name = "COM\_2";

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

}

if (str == "WRITE")

{

name = "COM\_1";

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

}

}

void COM :: SetParams()

{

DCB Params = { 0 };

if (handle == INVALID\_HANDLE\_VALUE) { cout << "Ошибка!" << endl; return; }

Params.DCBlength = sizeof(Params);

if (!GetCommState(handle, &Params)) {

cout << "Ошибка получения параметров COM-порта!\n";

return;

}

Params.BaudRate = 9600;

Params.fAbortOnError = TRUE;

Params.ByteSize = 8;

Params.StopBits = ONESTOPBIT;

Params.Parity = NOPARITY;

if (!SetCommState(handle, &Params)) {

cout << "Ошибка настройки параметров COM-порта!" << endl; return; }

}

void COM :: GetParams()

{

DCB Params = { 0 };

if (!GetCommState(handle, &Params)) {

cout << "Ошибка получения параметров COM-порта!" << endl; return; }

cout << "Параметры " << name << ":" << endl

<< "Скорость: " << Params.BaudRate << endl

<< "Старт-бит: " << start << endl

<< "Стоп-бит: " << stop << endl << endl;

}

void COM::WriteCOM(int\* flag)

{

char data[256] = "";

cout << "Введите строку для передачи: ";

cin >> data;

system("cls");

cout << "Записано в COM\_1: " << data << endl;

for (int i = strlen(data); i > 0; i--) data[i] = data[i - 1];

data[0] = start;

data[strlen(data)] = stop;

DWORD size = strlen(data);

DWORD bytes;

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

\*flag = 1;;

}

void COM :: ReadCOM(int \*flag)

{

if (\*flag != 0)

{

DWORD size = 0;

char buff;

int fl = 0;

cout << "Считано из COM\_2: ";

do

{

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

if (buff != stop && fl == 1) cout << buff;

if (buff == start && fl == 1) cout << "[!]";

fl = 1;

} while (buff != stop);

PurgeComm(handle, PURGE\_RXABORT | PURGE\_TXABORT | PURGE\_RXCLEAR | PURGE\_TXCLEAR);

}

else cout << "Нет данных для считывания!";

cout << endl;

\*flag = 0;

}

void COM :: ChangeSpeed()

{

system("cls");

int n;

cout << "Введите скорость: ";

cin >> n;

DCB Params = { 0 };

if (!GetCommState(handle, &Params)) cout << "Ошибка получения данных!" << endl;

Params.BaudRate = n;

if (!SetCommState(handle, &Params))

{

cout << "Ошибка настройки!" << endl;

}

system("cls");

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

}

void COM :: ChangeStopBit(COM \*COM)

{

system("cls");

cout << "Введите стоп-бит: ";

cin >> stop;

COM->stop = stop;

system("cls");

cout << "Установлен стоп-бит: " << stop << endl;

}

void COM :: ChangeStartBit(COM \*COM)

{

system("cls");

cout << "Введите старт-бит: ";

cin >> start;

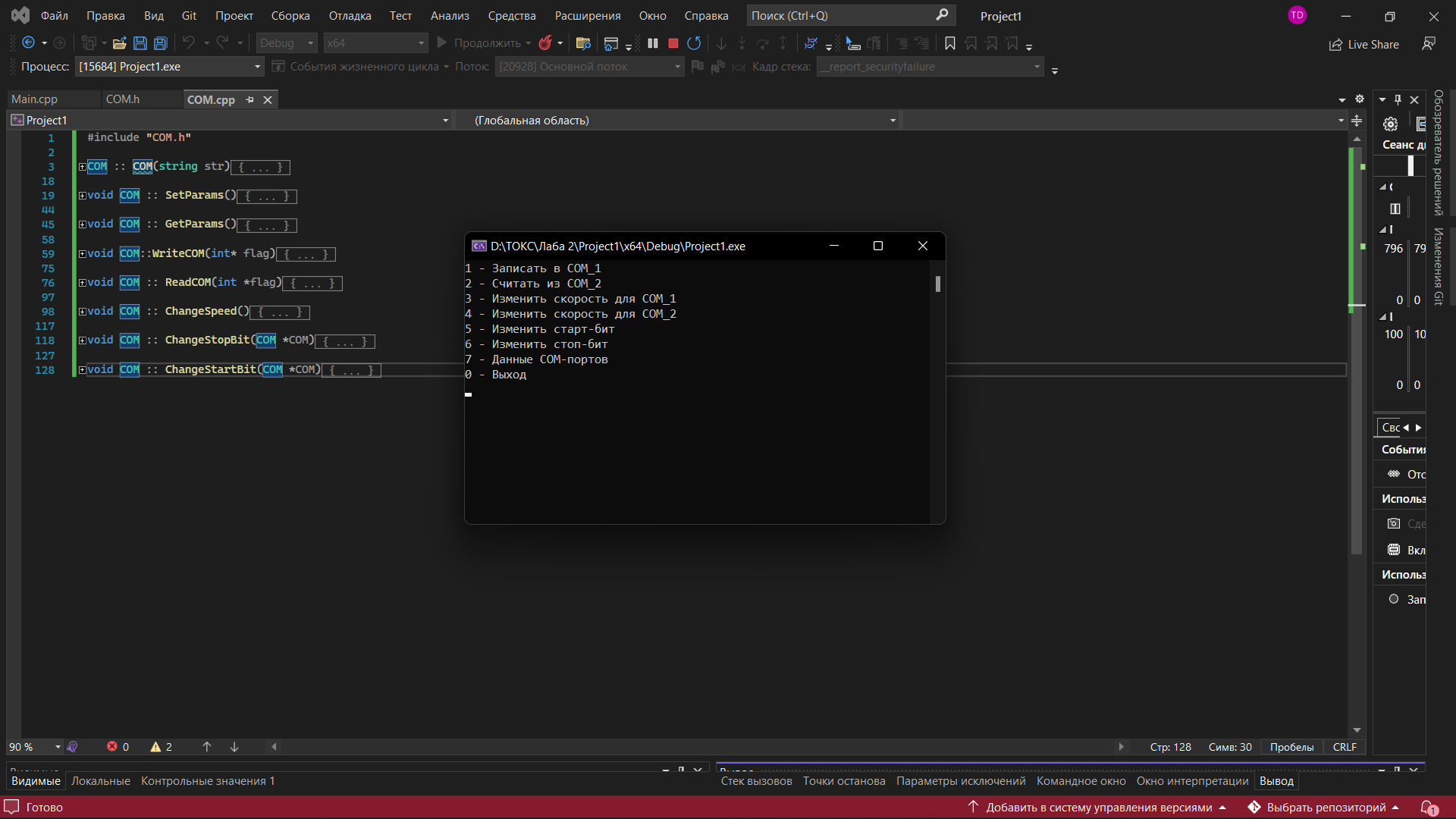
COM->start = start;

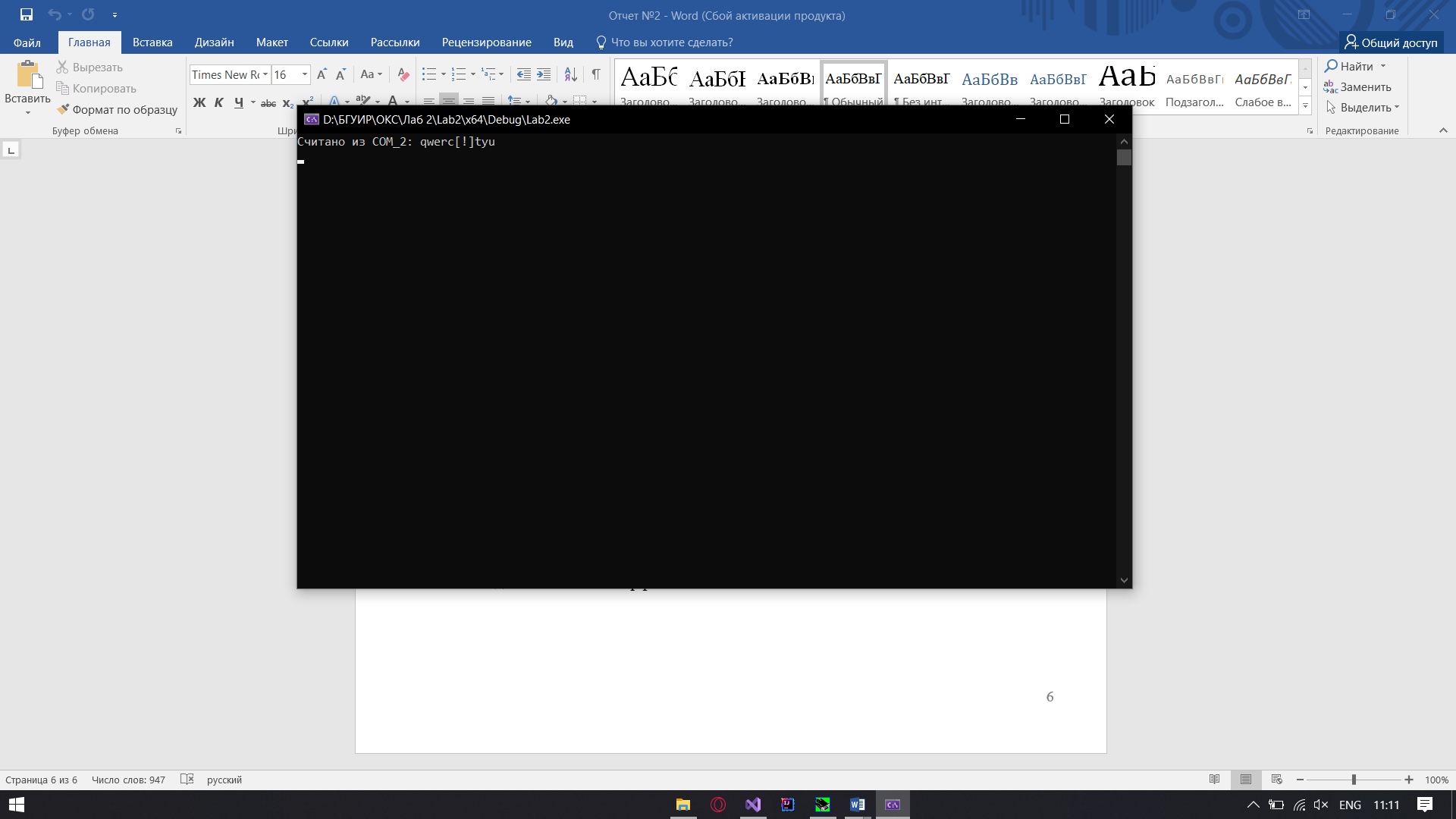
system("cls");

cout << "Установлен старт-бит: " << start << 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.