Белорусский государственный университет информатики и радиоэлектроники

# Кафедра ЭВМ

### Отчет

### по лабораторной работе № 3

### «Асинхронная пакетная передача данных»

Проверил:

Марцинкевич В.А.

Выполнил:

студент группы 130501

Гнездилов А.М

Минск 2022

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

**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 - 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, &COM\_2); \_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.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"

#include <string>

#include <stack>

#include <cmath>

using namespace std;

class COM

{

private:

char name[5];

HANDLE handle;

public:

COM(string str);

void SetParams(); //настройка COM-порта

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

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

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

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

};

**COM.cpp**

#include "COM.h"

#define n 8

#define iter 10

string hemming(string str)

{

int c = 0, i = 0, h = 0;

while (i < str.length())

{

if (str[i] == '1') c++;

i += 2;

}

if (c % 2 != 0) str.replace(0, 1, "1");

if (c % 2 == 0) str.replace(0, 1, "0");

i = 1;

c = 0;

while (i < str.length())

{

if (str[i] == '1') c++;

if (h == 2) {

i += 3;

h = 0;

}

else {

i++;

h++;

}

}

if (c % 2 != 0) str.replace(1, 1, "1");

if (c % 2 == 0) str.replace(1, 1, "0");

i = 3;

c = 0;

h = 0;

while (i < str.length())

{

if (str[i] == '1') c++;

if (h == 4) {

i += 3;

h = 0;

}

else {

i++;

h++;

}

}

if (c % 2 != 0) str.replace(3, 1, "1");

if (c % 2 == 0) str.replace(3, 1, "0");

i = 7;

c = 0;

h = 0;

while (i < str.length())

{

if (str[i] == '1') c++;

if (h == 7) {

i += 3;

h = 0;

}

else {

i++;

h++;

}

}

if (c % 2 != 0) str.replace(7, 1, "1");

if (c % 2 == 0) str.replace(7, 1, "0");

return str;

}

string code(string data)

{

string ret;

for (int i = 0; i < data.length(); i++)

{

stack<int> ST;

int count = 0;

string str;

int a, t, d;

t = 0;

d = 1;

a = data[i];

cout << data[i] << " - " << a << " - ";

while (a)

{

t += (a % 2) \* d;

a = a / 2;

d = d \* 10;

}

do

{

ST.push(t % 10);

t /= 10;

count++;

} while (t);

count = n - count;

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

str += '0';

while (!ST.empty())

{

t = ST.top();

ST.pop();

char s = { char(t + '0') };

str += s;

}

cout << str << " - ";

str.insert(0, "0");

str.insert(1, "0");

str.insert(3, "0");

str.insert(7, "0");

str = hemming(str);

cout << str << endl;

ret += str;

}

\_getch();

system("cls");

return ret;

}

string decode(string data)

{

string ret;

string str;

int s = n + 4;

for (int i = 0; i < data.length() / s; i++)

{

for (int j = i \* s; j < i \* s + s; j++) str += data[j];

cout << str << " - ";

str.erase(7, 1);

str.erase(3, 1);

str.erase(1, 1);

str.erase(0, 1);

cout << str << " - ";

int st = 0;

int ch = 0;

for (int j = str.length() - 1; j >= 0; j--)

{

if (str[j] == '1') ch += pow(2, st);

st++;

}

cout << ch << " - ";

char a = ch;

cout << a << endl;

ret += a;

str.clear();

}

\_getch();

system("cls");

return ret;

}

string error(string data)

{

int s = n + 4;

for (int i = 0; i < data.length() / s; i++)

{

string str;

if (data[i \* s + iter] == '0') str = "1";

if (data[i \* s + iter] == '1') str = "0";

data.replace(i \* s + iter, 1, str);

}

return data;

}

string deerror(string data)

{

string ret, buf1, buf2, str, er1, er2;

int s = n + 4, c = 0, e = 0;

for (int i = 0; i < data.length() / s; i++)

{

for (int j = i \* s; j < i \* s + s; j++)

{

buf1 += data[j];

}

buf2 = buf1;

buf1.replace(0, 1, "0");

buf1.replace(1, 1, "0");

buf1.replace(3, 1, "0");

buf1.replace(7, 1, "0");

buf1 = hemming(buf1);

if (buf1 != buf2)

{

c = 0;

for (int i = 0; i < s; i++)

{

if (buf1[i] != buf2[i]) c += i;

}

er1 = buf2;

if(c != 0) er1.replace(c, 1, "\*");

er2 += er1;

if (buf1[c] == '1') str = "0";

if (buf1[c] == '0') str = "1";

buf1.replace(c, 1, str);

}

ret += buf1;

buf1.clear();

}

if (!er2.empty()) cout << "Ошибка!" << endl << er2 << endl;

\_getch();

system("cls");

return ret;

}

COM::COM(string str)

{

name[0] = 'C';

name[1] = 'O';

name[2] = 'M';

name[3] = '\_';

if (str == "READ")

{

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

name[4] = '2';

}

if (str == "WRITE")

{

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

name[4] = '1';

}

}

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;

}

void COM::WriteCOM(int\* flag, COM\* COM)

{

char data1[256] = "";

char data2[256] = "";

string str;

string a, b;

cout << "Введите строку для передачи: ";

cin >> data1;

system("cls");

char s[4] = { char(strlen(data1) + '0'), '\0' };

str += COM->name;

str += name;

b = code(data1);

a = error(b);

str += a;

str += s;

str += '~';

for (int i = 0; i < str.length(); i++)

data2[i] = str[i];

DWORD size = strlen(data2);

DWORD bytes;

BOOL Ret = WriteFile(handle, &data2, size, &bytes, NULL);

system("cls");

cout << "Байт записано в COM\_1: " << s << endl << "Данные: " << data1 << " [" << b << "]" << endl;

\*flag = 1;

}

void COM::ReadCOM(int\* flag)

{

if (\*flag != 0)

{

DWORD size = 0;

char buff;

string data;

int fl = 0;

do

{

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

if (buff != '~')

data += buff;

} while (buff != '~');

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

string str;

string sour;

string pour;

for (int i = 0; i < 5; i++)

pour += data[i];

for (int i = 5; i < 10; i++)

sour += data[i];

for (int i = 10; i < data.length() - 1; i++)

str += data[i];

string info = decode(deerror(str));

cout << "Приемник: " << pour << endl;

cout << "Источник: " << sour << endl;

cout << "Данные: " << str << " [" << info << "]" << endl;

cout << "Контрольная сумма: " << data[data.length() - 1] << endl;

}

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

\*flag = 0;

}

void COM::ChangeSpeed()

{

system("cls");

int s;

cout << "Введите скорость: ";

cin >> s;

DCB Params = { 0 };

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

Params.BaudRate = s;

if (!SetCommState(handle, &Params)) cout << "Ошибка настройки!" << endl;

system("cls");

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

}

**Скриншоты работы программы**











