

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Організація комп’ютерних мереж»**

**на тему: «Програмування зв’язку між двома ПК через послідовний порт»**

Виконав:

Студент гр. ПЗ2011

Кулик С.В.

Прийняв:

Івченко Ю. М.

Дніпро, 2023

**Тема:** Програмування зв’язку між двома ПК через послідовний порт.

**Мета:** Вивчити принципи побудови послідовного порту і його використання для зв’язку між двома персональними комп’ютерами (ПК). Отримати практичні навички програмування зв’язку між двома ПК.

# Порядок виконання роботи

1.1. Ознайомитися з принципами побудови послідовного порту і його використання

для організації зв’язку між двома ПК для обміну інформацією.

1.2. Розробити програму для обміну інформацією між двома ПК з використанням

переривань INT 14h функцій BIOS.

1.3. Провести налагодження програми.

1.4. Виконати програму.

# Короткий опис відомостей про асинхронний послідовний адаптер

Практично кожний комп’ютер обладнаний хоча б одним асинхронним послідовним адаптером. Як правило, він реалізований у вигляді окремої плати або розташований прямо на материнській платі комп’ютера. Його іноді називають асинхронним адаптером RS-232-C або портом RS-232-C. Асинхронний адаптер звичайно містить декілька COM-портів, через які до комп’ютера можна підключити зовнішні пристрої.

Кожному COM-порту відповідає декілька регістрів, через які програма має до нього доступ, і певна лінія IRQ для сигналізування комп’ютеру про зміну стану порту. На етапі ініціалізації модулі BIOS присвоюють кожному COM-порту унікальний номер. Наприклад, комп’ютер може мати чотири порти COM1 – COM4.

Для нас найбільш цікаво використання COM-портів для обміну даними між двома комп’ютерами через модеми і нуль-модеми. Якщо комп’ютери розташовані досить близько, тоді їх можна зв’язати за допомогою спеціального кабеля, який називається нуль-модемом. Цей кабель підключається до COM-портів обох комп’ютерів і дозволяє організувати обмін даними з високими швидкостями – 115000 біт/с. Іноді для визначення швидкості передачі інформації використовують термін – символи за секунду (cps). Він більш реально відображає швидкість передачі даних, оскільки не враховує службові біти, які не несуть корисної інформації. Наприклад, фактична швидкість передачі інформації для модема 2400 біт/с становить всього 240 символів за секунду, замість 300 (оскільки на кожний байт корисної інформації передається два службових біта). Але це значення приблизне. Оскільки при передачі даних можуть використовуватися протоколи корекції помилок і стиснення переданої інформації. Тому реальна швидкість передачі інформації залежить від якості телефонного каналу, алгоритму стиснення, самої інформації, а також багатьох інших факторів.

Зовні кожний COM-порт асинхронного послідовного адаптера представляється особистим роз’ємом. Існує два стандарти на роз’єми COM-порту: DB25 і DB9, які мають відповідно 25 і 9 контактів. Але вони передають однакові сигнали.

**Текст програми**

//Організація комп’ютерних мереж

//Лабораторна №1

//Виконала студентка групи 931(ПЗ2011) Пазика Катерина

//Програму для обміну інформацією між двома ПК з використанням віртуальних послідовних портів

#include <iostream>

#include <windows.h>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

HANDLE h\_сom1, h\_сom2;

char buffer[10];

DWORD bytesRead, bytesWritten;

COMMTIMEOUTS timeouts = { 0 };

// Відкриття портів

h\_сom1 = CreateFile(L"COM1", GENERIC\_READ | GENERIC\_WRITE, 0, NULL, OPEN\_EXISTING, 0, NULL);

h\_сom2 = CreateFile(L"COM2", GENERIC\_READ | GENERIC\_WRITE, 0, NULL, OPEN\_EXISTING, 0, NULL);

// Перевірка на успішне відкриття портів

if (h\_сom1 == INVALID\_HANDLE\_VALUE || h\_сom2 == INVALID\_HANDLE\_VALUE)

{

cout << "Error opening ports.\n";

return 1;

}

// Налаштування параметрів портів

DCB dcbSerialParams = { 0 };

dcbSerialParams.DCBlength = sizeof(dcbSerialParams);

if (!GetCommState(h\_сom1, &dcbSerialParams) || !GetCommState(h\_сom2, &dcbSerialParams))

{

cout << "Error getting port parameters.\n";

CloseHandle(h\_сom1);

CloseHandle(h\_сom2);

return 1;

}

dcbSerialParams.BaudRate = CBR\_9600;

dcbSerialParams.ByteSize = 8;

dcbSerialParams.StopBits = ONESTOPBIT;

dcbSerialParams.Parity = NOPARITY;

if (!SetCommState(h\_сom1, &dcbSerialParams) || !SetCommState(h\_сom2, &dcbSerialParams))

{

cout << "Error in configuring port settings.\n";

CloseHandle(h\_сom1);

CloseHandle(h\_сom2);

return 1;

}

// Налаштування таймаутів

timeouts.ReadIntervalTimeout = 50;

timeouts.ReadTotalTimeoutConstant = 50;

timeouts.ReadTotalTimeoutMultiplier = 10;

timeouts.WriteTotalTimeoutConstant = 50;

timeouts.WriteTotalTimeoutMultiplier = 10;

if (!SetCommTimeouts(h\_сom1, &timeouts) || !SetCommTimeouts(h\_сom2, &timeouts))

{

cout << "Error setting timeouts.\n";

CloseHandle(h\_сom1);

CloseHandle(h\_сom2);

return 1;

}

// Вивід меню

int menu;

do {

cout << "\nSelect the menu option:\n";

cout << "1. Send data from port COM1 to port COM2\n";

cout << "2. Send data from port COM2 to port COM1\n";

cout << "3. Get data from the COM1 port\n";

cout << "4. Get data from the COM2 port\n";

cout << "5. Exit\n";

//cout << "Ваш вибір: ";

cin >> menu;

switch (menu) {

case 1:

// Отримання даних

cout << "Enter the data to send from port COM1 to port COM2: ";

cin >> buffer;

// Відправка даних з порту COM1 на порт COM2

if (!WriteFile(h\_сom1, buffer, strlen(buffer), &bytesWritten, NULL))

{

cout << "Error sending data from the COM1 port.\n";

break;

}

cout << "Data has been successfully sent\n";

break;

case 2:

// Отримання даних

cout << "Enter the data to send from port COM2 to port COM1: ";

cin >> buffer;

// Відправка даних з порту COM2 на порт COM1

if (!WriteFile(h\_сom2, buffer, strlen(buffer), &bytesWritten, NULL))

{

cout << "Error sending data from the COM2 port.\n";

break;

}

cout << "Data has been successfully sent\n";

break;

case 3:

// Отримання даних з порту COM1

memset(buffer, 0, sizeof(buffer));

if (!ReadFile(h\_сom1, buffer, sizeof(buffer) - 1, &bytesRead, NULL))

{

cout << "Error receiving data from the COM1 port.\n";

break;

}

cout << "Data received from the COM1 port: " << buffer << endl;

break;

case 4:

// Отримання даних з порту COM2

memset(buffer, 0, sizeof(buffer));

if (!ReadFile(h\_сom2, buffer, sizeof(buffer) - 1, &bytesRead, NULL))

{

cout << "Error receiving data from the COM2 port.\n";

break;

}

cout << "Data received from the COM2 port: " << buffer << endl;

break;

case 5:

// Вихід з програми

break;

default:

// Невірний вибір опції

cout << "Wrong value! Try again.\n";

break;

}

} while (menu != 5);

// Закриття портів

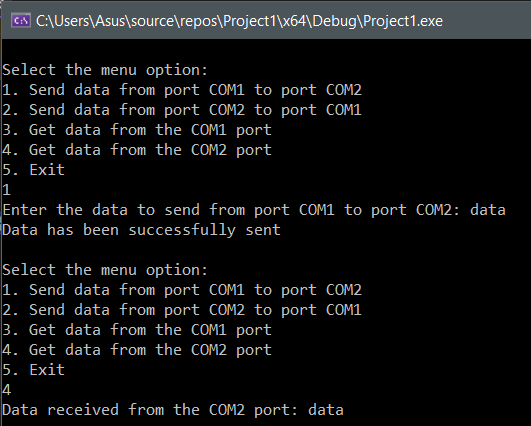
CloseHandle(h\_сom1);

CloseHandle(h\_сom2);

return 0;

}

**Результати роботи програми**

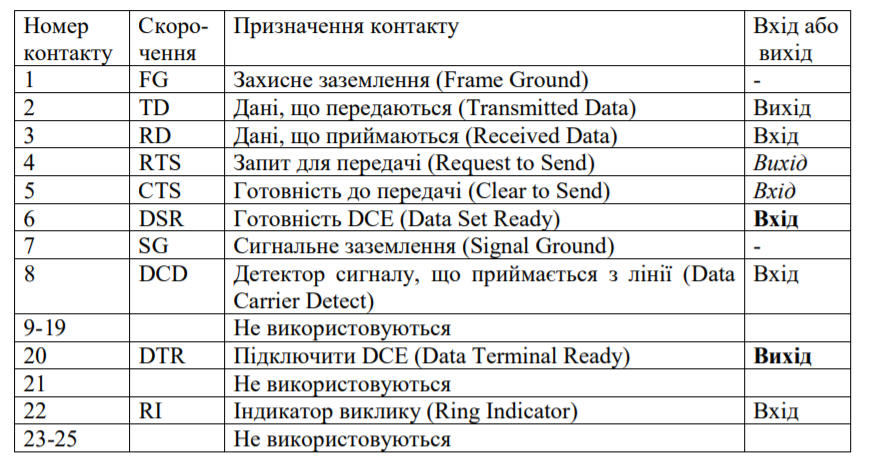
****

**Висновок**

У результаті виконання лабораторної роботи набуто навичок програмування зв’язку між двома ПК через послідовний порт, також відомий як COM-порт. Послідовний порт - це інтерфейс, який дозволяє обмінюватися даними між комп'ютером та зовнішнім пристроєм через послідовну (посимвольну) передачу. Для підключення двох ПК за допомогою послідовного порту можна використовувати спеціальний кабель - Null-modem або crossover-кабель. Для програмування зв'язку між ПК через послідовний порт можна використовувати мову програмування, таку як С++ та використовувати стандартні бібліотеки для роботи з послідовним портом.

**Контрольні питання**

1. Нуль-модемне з'єднання - це тип з'єднання двох комп'ютерів або інших пристроїв без використання мережевого комутатора або маршрутизатора. В такому з'єднанні для передачі даних використовується спеціальний кабель з перехресними з'єднаннями на кінцях, який змушує сигнали передаватися з одного комп'ютера на інший напрямок відносно кожного проводу. Схема нуль-модемного з'єднання складається з роз'ємів DB-9 або DB-25, які підключаються до вхідно-вихідних портів на комп'ютерах або пристроях, і кабелю, в якому проводи перетинаються, щоб забезпечити правильну передачу сигналів. Стандартний нуль-модемний кабель зазвичай має довжину від 1 до 5 метрів. Максимальна відстань між комп'ютерами залежить від типу кабелю і швидкості передачі даних. Наприклад, для передачі даних зі швидкістю 115200 біт/с через RS-232, максимальна відстань між комп'ютерами з обох сторін кабелю не повинна перевищувати 15 метрів. Максимальна відстань - до 1,5 км, але швидкість падає до 300 біт / с.
2. Мікросхема UART (Universal Asynchronous Receiver/Transmitter) - це пристрій, що забезпечує зв'язок між комп'ютером та зовнішнім пристроєм за допомогою передачі даних у вигляді послідовних бітів в асинхронному режимі. Основні принципи роботи мікросхеми UART полягають у перетворенні вхідної послідовності даних (байтів) в паралельний формат для подальшої обробки комп'ютером і в перетворенні паралельної вихідної послідовності даних в послідовний формат для передачі на зовнішній пристрій. UART отримує та передає дані за допомогою визначеного протоколу передачі даних, який містить в собі стандартні налаштування, такі як швидкість передачі (біт/сек), біти даних (7 або 8), біти стопу (1 або 2) та біт парності (немає, парний або непарний). Ця мікросхема призначена для забезпечення зв'язку між мікроконтролером або комп'ютером та зовнішнім пристроєм за допомогою послідовного інтерфейсу.



1. Кількість com-портів на ПК може бути різною і залежить від типу ПК та його конфігурації. Зазвичай на стандартних настільних комп'ютерах доступно два com-порти - COM1 та COM2.

Один зі способів визначити кількість портів: ***Панель керування \ Система і безпека -> Система -> Диспетчер пристроїв -> пункт Порти (COM і LPT).***

1. Основні одиниці виміру швидкості передачі даних включають

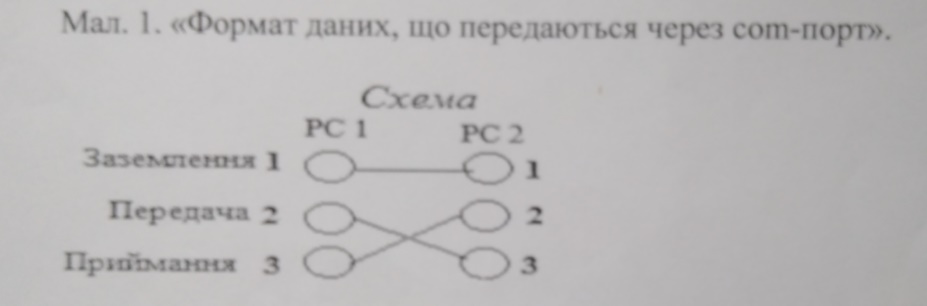
- біти в секунду (біт/с) - вимірює кількість бітів, що передаються за одну секунду;

- бод (bps або baud) - вимірює кількість символів, які передаються за секунду. Один символ може складатися з декількох бітів, тому бод і біт/с не завжди співпадають. Наприклад, для передачі одного символу ASCII-коду, використовується 8 біт (байт), тому передача зі швидкістю 9600 бод відповідає швидкості приблизно 960 байт/с.

- символи в секунду (символ/с) - вимірює кількість символів, які передаються за секунду. Ця одиниця виміру використовується зазвичай для вимірювання швидкості передачі даних в мережах зв'язку.

У загальному, обрання одиниці виміру залежить від контексту та типу пристрою, що використовується для передачі даних.

1. Формат переданих даних в послідовному порту: це 2 регістри: буферний та зсувний. Байт передається на буферний регістр, а мікросхема автоматично копіює в зсувний та по 1 біту передає в кабель.



1. Ініціалізація com-порту - це процес налаштування вхідно-вихідних параметрів послідовного порту для взаємодії з підключеним пристроєм. Ініціалізація включає встановлення швидкості передачі даних, формату даних та інших параметрів, необхідних для правильної передачі даних. Ініціалізацію com-порту можна виконати за допомогою спеціального програмного забезпечення або вбудованої програми BIOS на ПК.

