

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



УДУНТ ННІ ДІТ

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.ПЗ1911
Сафонов Д.Є.
Прийняв:
Івченко Ю.М.

Дніпро, 2022

Тема. Програмування зв'язку між двома ПК через послідовний порт.

Мета.

1. Вивчити принципи побудови послідовного порту і його використання для зв'язку між двома персональними комп'ютерами (ПК).
2. Отримати практичні навички програмування зв'язку між двома ПК.

Порядок виконання роботи.

1. Ознайомитися з принципами побудови послідовного порту і його використання для організації зв'язку між двома ПК для обміну інформацією.
2. Розробити програму для обміну інформацією між двома ПК з використанням переривань INT 14h функцій BIOS.
3. Провести налагодження програми.
4. Виконати програму.

Короткий опис відомостей про асинхронний послідовний адаптер

Практично кожний комп'ютер обладнаний хоча б одним асинхронним послідовним адаптером. 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 сигналізуванню комп'ютеру про зміну стану порту. На етапі ініціалізації модулі BIOS присвоюють кожному COM-порту унікальний номер.

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 кабелю, який називається нуль-модемом. Цей кабель підключається до COM-портів обох комп'ютерів і дозволяє організувати обмін даними з високими швидкостями – 115000 біт/с.

Важливою характеристикою є швидкість передачі даних. Вона вимірюється в бітах за секунду (біт/с) або в бодах. Швидкість в бодах визначається числом змін сигналу, який передається модемом по телефонній лінії, які відбулися за одну секунду. Різні способи модуляції можуть кодувати в одному значенні сигналу декілька біт. Терміни “біт/с” і “бод” рівнозначні, коли говорять про передачу даних через COM-порт.

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.

Інтерфейс RS-232-C визначає обмін між пристроями двох типів: DTE (Data Terminal Equipment – термінальний пристрій) і DCE (Data Communication Equipment – пристрій зв'язку). Частіше комп'ютер є термінальним пристроєм. Модеми і принтери завжди є пристроями зв'язку.

На етапі ініціалізації системи модуль процедури початкової загрузки BIOS тестує асинхронні порти RS-232-C і ініціалізує їх. Їх базові адреси записуються в області даних BIOS, починаючи з адреси 0040:0000h. Проглянути наявність COM-портів на комп'ютері можна за допомогою програми DEBUG.EXE (команда d40:0). Адаптери звичайно мають базову адресу і займають діапазон адрес відповідно:

- COM1 – 3F8h (3F8h-3FFh),
- COM2 – 2F8h (2F8h-2FFh),
- COM3 – 3E8h (3E8h-3EFh),
- COM4 – 2E8h (2E8h-2EFh).

Порти асинхронного адаптера можуть виробляти переривання:

- COM1, COM3 – IRQ4 (відповідає INT 0Ch);
- COM2, COM4 – IRQ3 (відповідає INT 0Bh).

Функції переривання 14h

Усі функції мають два обов'язкових параметри на вході:

- АН = номер функції;
- DX = номер порту:
 - 0 – COM1,
 - 1 – COM2,
 - 2 – COM3,
 - 3 – COM4;

Також усі функції мають на виході АН = стан порту асинхронного адаптера.

Призначення функції	AL на виході	На вході	На виході	
Ініціалізація(00h)	стан модему	AL = параметри ініціалізації (див. Нижче)		
передача байта(01h)	зберігається	AL = байт, який передається	сталася помилка, якщо	біт 7 регістра АН встановлений в 1
отримання байта(02h)	прийнятий байт			регістр АН не дорівнює 0,
Стан порту асинхронного адаптера(03h)	стан модему			

параметри ініціалізації (х – стан біта байдуже будь-який):

Біти	Зміст								
D1, D0	Довжина слова в бітах								
	значення	00		01		10		11	
	довжина	5 біт		6 біт		7 біт		8 біт	
D2	Кількість стопових бітів: <ul style="list-style-type: none">• 0 – 1 біт,• 1 – 2 біт.								
D4, D3	Парність: <ul style="list-style-type: none">• x0 – контроль на парність не використовується,• 01 – контроль на непарність,• 11 – контроль на парність.								
D7-D5	Швидкість передачі даних								
	значення	000	001	010	011	100	101	110	111
	Швидкість(біт/с)	100	150	300	600	1200	2400	4800	9600

байт	стан порту асинхронного адаптера	стан модему
D0	Тайм-аут, якщо встановлений цей біт, інші біти не мають значення	Лінія CTS змінила стан
D1	Регістр зсуву передавача пустий	Лінія DSR змінила стан
D2	Буферний регістр передавача пустий	Лінія RI змінила стан
D3	Виявлено стан BREAK	Лінія DCD змінила стан
D4	Помилка синхронізації	Стан лінії CTS
D5	Помилка парності	Стан лінії DSR
D6	Помилка переповнення вхідного регістра	Стан лінії RI
D7	Дані готові	Стан лінії DCD

Текст програми

```
// Safonov Danylo pz1911(931)
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

short portNumber() {
    short pt;
    clrscr();
    printf("Port number: ");
    scanf("%d", &pt);
    return pt - 1; // when passing it as argument to interruption it is 0
}

// params
//      D0 D1 |          D2 |          D3 D4 |          D5 D6 D7
// word len(bits) | stop len(bits) | parity check | transfer speed
void initialize(char params) {
    char res;
    short pt = portNumber();

    _asm {
        push ax
        push dx

        mov ah, 00h
        mov dx, pt
        mov al, params
        INT 14h
        mov res, ah

        pop dx
        pop ax
    }

    cout << "\n\nPort's state:\n" << res << endl;
    int value = (int)res;
    for (int i = 7; i >= 0; i--) {
        cout << (1 << i) & value;
    }
    getch();
}

void send(char* str) {
    char res, s;
    short pt = portNumber();
    do {
        do {
            s = *str;
            _asm {
                push ax
                push dx

                mov ah, 01h
                mov dx, pt
                mov al, s
                INT 14h
            }
        } while (s != '\0');
        str++;
    } while (str != NULL);
}
```

```

        mov res, ah

        pop dx
        pop ax
    }
    } while (res & 0x80); // 10000000 binary => error
} while (*(str++));
cout << "Send finished \n";
}

void recieve(char* str) {
    char err;
    char data;
    short pt = portNumber();

    do {
        do {
            _asm {
                push ax
                push dx

                mov ah, 02h
                mov dx, pt
                INT 14h
                mov err, ah
                mov data, al

                pop dx
                pop ax
            }
        } while (err);
        str[i] = data;
    } while (str[i++]);
}

int main() {
    clrscr();
    bool exit = false;
    while (!exit) {
        cout << "Menu \n"
            << "1.Initilize \port \n"
            << "2.Send message \n"
            << "3.Recieve message \n"
            << "4.Exit\n\n Make your choice=>> ";

        int option;
        cin >> option;
        switch (option) {
            case 1: {
                int speed;
                cout << "Choose speed\n"
                    << "1.100 b/s\n" // 192 = 11000000
                    << "2.150 b/s\n" // 193 = 11000001
                    << "3.300 b/s\n" // 194 = 11000010
                    << "4.600 b/s\n" // 195 = 11000011
                    << "5.1200 b/s\n" // 196 = 11000100
                    << "6.2400 b/s\n" // 197 = 11000101
                    << "7.4800 b/s\n" // 198 = 11000110
                    << "8.9600 b/s\n" // 199 = 11000111
                    << "Your choise - ";
            }
        }
    }
}

```

```

        cin >> speed;
        initialize(speed + 191);
        break;
    }
    case 2: {
        char str[256];
        cout << "Sended message: ";
        gets(str);
        send(str);
        getch();// wait for newline
        break;
    }
    case 3: {
        char str[256];
        recieve(str);
        cout << "Recieved string: " << str;
        getch();// wait for newline
        break;
    }
    default: {
        exit = true;
        break;
    }
}
}
return 0;
}

```

Висновок

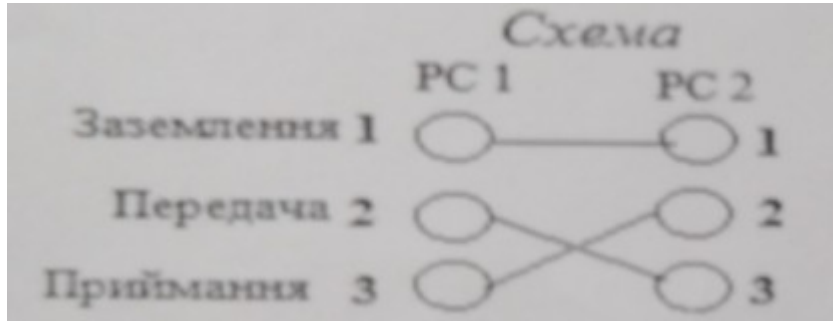
В ході виконання лабораторної роботи були розглянуті принципи побудови послідовного порту і його використання для зв'язку між двома персональними комп'ютерами для обміну інформацією, була розроблена програма для обміну інформацією між двома персональними комп'ютерами з використанням переривання 14h.

Відповіді на контрольні питання

1. Що таке нуль-модемне з'єднання.

Нуль-модемне з'єднання — спосіб з'єднання двох пристроїв кабелем RS-232

Схема.



Стандартна і максимальна відстань між комп'ютерами (відповідні їм швидкості) при передачі по нуль-модемному кабелю.

Стандартна відстань: 15.24м(50 футів), 115кбіт/с

Максимальна відстань: 1524м(5000 футів), 100біт/с

2. Призначення і принципи функціонування мікросхеми UART.

UART(Universal asynchronous receiver-transmitter) - універсальний асинхронний приймач-передавач, використовується для передачі даних паралельно та послідовно за стандартом RS-232.

Існує два стандарти на роз'єми COM-порту: DB25 і DB9, які мають відповідно 25 і 9 контактів.

Але вони передають однакові сигнали. Тільки два контакти цих роз'ємів використовуються для передачі і прийому даних. Інші передають різноманітні допоміжні і управляючі сигнали.

Інтерфейс RS-232-C визначає обмін між пристроями двох типів: DTE (Data Terminal Equipment – термінальний пристрій) і DCE (Data Communication Equipment – пристрій зв'язку). Частіше комп'ютер є термінальним пристроєм. Модеми і принтери завжди є пристроями зв'язку. Для з'єднання двох термінальних пристроїв, наприклад, двох комп'ютерів, як мінімум, необхідне перехресне з'єднання ліній TR і RD, що і реалізоване примітивним нуль-модемом. Однак у більшості випадків цього недостатньо, оскільки для приладів DTE і DCE функції, які виконують лінії DSR, DTR, DCD, CTS і RTS, асиметричні.

3. Визначити кількість com-портів на ПК.

На етапі ініціалізації модулі BIOS присвоюють кожному COM-порту унікальний номер. Наприклад, комп'ютер може мати чотири порти COM1 – COM4.

Проглянути наявність COM-портів на комп'ютері можна за допомогою програми DEBUG.EXE (команда d40:0). Адреси COM-портів є свого роду стандартом. Адаптери звичайно мають базову адресу і займають діапазон адрес відповідно:

- COM1 – 3F8h (3F8h-3FFh),
- COM2 – 2F8h (2F8h-2FFh),
- COM3 – 3E8h (3E8h-3EFh),
- COM4 – 2E8h (2E8h-2EFh).

4. Швидкість передачі даних: біт/с, бод, символ/с.

біт/с — кількість бітів переданих за секунду.

Бод — кількість змін сигналу за секунду.

Різні способи модуляції можуть кодувати в одному значенні сигналу декілька біт. Тому значення швидкості передачі даних модемом у бодах може відрізнятися від значення швидкості в бітах за секунду. Наприклад, для модемів, які відповідають рекомендації CCITT V.32 і працюють зі швидкістю 4 9600біт/с, використовується комплексний метод модуляції, в якому інформація кодується одночасною зміною фази і амплітуди. Це дозволяє закодувати в кожному значенні сигналу чотири біти корисної інформації. Таким чином, модем, який працює зі швидкістю 9600 біт/с, передає дані зі швидкістю 2400 бод. Терміни “біт/с” і “бод” рівнозначні, коли говорять про передачу даних через COM-порт.

символ/с — кількість переданих символів за секунду, без урахування службових бітів(наприклад при використанні коду Хеммінга, тощо)

5. Формат переданих даних.

Повідомлення обрамляється стартовим бітом, який повідомляє отримувач про надходження нового повідомлення, далі йдуть біти повідомлення, опціональний біт парності та стопові біти.

6. Ініціалізація com-порту

Для ініціалізації портів використовується функція 00h переривання 14h. Перед викликом потрібно заповнити вхідні регістри:

- AH — номер функції(00h)
- DX — номер порту:
 - 0 – COM1,
 - 1 – COM2,
 - 2 – COM3,
 - 3 – COM4;
- AL — параметри ініціалізації (див. Нижче)

параметри ініціалізації (x – стан біта байдуже будь-який):

Біти	Зміст								
D1, D0	Довжина слова в бітах								
	значення	00		01		10		11	
	довжина	5 біт		6 біт		7 біт		8 біт	
D2	Кількість стопових бітів: <ul style="list-style-type: none">• 0 – 1 біт,• 1 – 2 біт.								
D4, D3	Парність: <ul style="list-style-type: none">• x0 – контроль на парність не використовується,• 01 – контроль на непарність,• 11 – контроль на парність.								
D7-D5	Швидкість передачі даних								
	значення	000	001	010	011	100	101	110	111
	Швидкість(біт/с)	100	150	300	600	1200	2400	4800	9600

На виході маємо:

- AH – стан порту
- AL – стан модему

байт	стан порту асинхронного адаптера	стан модему
D0	Тайм-аут, якщо встановлений цей біт, інші біти не мають значення	Лінія CTS змінила стан
D1	Регістр зсуву передавача пустий	Лінія DSR змінила стан
D2	Буферний регістр передавача пустий	Лінія RI змінила стан
D3	Виявлено стан BREAK	Лінія DCD змінила стан
D4	Помилка синхронізації	Стан лінії CTS
D5	Помилка парності	Стан лінії DSR
D6	Помилка переповнення вхідного регістра	Стан лінії RI
D7	Дані готові	Стан лінії DCD