Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №1

«Последовательный порт»

Выполнил: Проверил:

Студент группы 250505 Преподаватель

Зорин А.Ю. Одинец Д.Н.

Минск 2024

1. Постановка задачи

Разработать программный модуль реализации процедуры передачи (приёма) байта информации через последовательный интерфейс.

Программа должна демонстрировать программное взаимодействие с последовательным интерфейсом с использованием следующих механизмов:

1. Прямое взаимодействие с портами ввода-вывода (write, read)
2. Использование BIOS прерывания 14h
3. Работа с COM-портом через регистры как с устройствами ввода-вывода.
4. Алгоритм

Программа состоит из нескольких подпрограмм (частей программы), представляющих собой некоторые функции. К ним относятся функции:

* Инициализация порта
* Запись байта информации в порт
* Чтение байта информации из порта
* Вывод результата на экран

1. Листинг программы

Далее приведены листинги программ, реализующие различные механизмы передачи (приёма) информации через последовательный интерфейс.

3.1. Листинг программы, взаимодействующей с портами ввода-вывода.

#include <windows.h>

#include <iostream>

HANDLE COM1;

HANDLE COM2;

void writeToCOM1()

{

char data = 'A';

DWORD size = sizeof(data);

DWORD writtenBytes;

// имя буфер размер записи переменная записи

WriteFile(COM1, &data, size, &writtenBytes, nullptr);

std::cout << writtenBytes << " Bytes sended." << std::endl;

}

void readFromCOM2()

{

DWORD size;

char data;

// имя буфер размер чтения переменная чтения

ReadFile(COM2, &data, 1, &size, nullptr);

std::cout << size << " Bytes received." << std::endl;

if (size > 0)

{

std::cout << data;

}

}

int main()

{

//открытие COM-портов

// имя режим доступа совм.доступ защита автосоздание файла //шаблон для создания

COM1 = ::CreateFile("COM1", GENERIC\_WRITE, 0, nullptr, OPEN\_EXISTING,0, nullptr);

COM2 = ::CreateFile("COM2", GENERIC\_READ, 0, nullptr, OPEN\_EXISTING, 0, nullptr);

if (COM1 == INVALID\_HANDLE\_VALUE)

{

std::cout<<"Opening COM error!"<<std::endl;

return 0;

}

DCB settings; //основные настройки COM-порта

settings.DCBlength = sizeof(settings);

if (!GetCommState(COM1, &settings))

{

std::cout << "Getting state error.\n";

}

settings.BaudRate = CBR\_9600; //speed

settings.ByteSize = 8; //amount bits

settings.StopBits = ONESTOPBIT; //amount stopbits

settings.Parity = NOPARITY;

if (!SetCommState(COM2, &settings)) //установка конфигурации

{

std::cout << "Error setting serial port state.\n";

}

writeToCOM1();

readFromCOM2();

return 0;

}

3.2. Листинг программы, использующей BIOS прерывание 14h.

.model small

.stack 100h

.data

writeError db "Write error!",0Dh,0Ah,'$'

readError db "Read error!",0Dh,0Ah,'$'

.code

jmp start

init proc

xor ax,ax

mov al,10100011b ;connection params: speed and amount of bits (101 - 2400, 11 -8)

mov dx,0 ;port num - COM1

int 14h

ret

init endp

checkToWrite proc

mov al,'A' ;symbol

mov ah,1 ;write

mov dx,0 ;COM1

int 14h ;return LSD+symbol in ax

test al,80h ;like AND but changes flag ZF

jnz errorWrite ;no FIFO

ret

checkToWrite endp

errorWrite proc

mov ah,9

mov dx,offset writeError

add dx,2

int 21h

ret

errorWrite endp

checkToRead proc

mov ah,2

mov dx,1 ;COM2

int 14h ;return LSR+symbol in ax

test al,80h

jnz errorRead

ret

checkToRead endp

errorRead proc

mov ah,9

mov dx,offset readError

add dx,2

int 21h

ret

errorRead endp

Output proc

mov ah,02h

mov dl,al

int 21h

ret

Output endp

Exit proc

mov ax,4C00h

int 21h

ret

Exit endp

start:

call init

call checkToWrite

call checkToRead

call Output

call Exit

end start

3.3. Листинг программы, работающей с COM-портами через регистры как с устройствами ввода-вывода.

#include <windows.h>

#include <conio.h>

#include <stdio.h>

#include <dos.h>

#include <stdlib.h>

int Com\_Init(int port);

void com\_outchar(char chr, int base\_port);

char com\_inchar(int base\_port);

int check\_snd(int base\_port);

int Com\_Init(int port)

{

char base\_port = port;

unsigned int div = 12; //9600

unsigned int regst;

regst = inp(base\_port + 0x03); // LCR

outp(base\_port + 0x03, regst | 0x80); // DLAB=1

outp(base\_port + 0x01, (div >> 8) & 0x00ff);

outp(base\_port, div & 0x00ff);

outp(base\_port + 0x03, regst & 0x7f); //DLAB=0

outp(base\_port + 0x01, 0x00);

outp(base\_port + 0x03, 11); // Установка параметров передачи: проверка четности, 6 бит, 2 стоповых бита

outp(base\_port + 0x04, 0x00); // Установка уровней DTR и RTS в 0

return 1;

};

void com\_outchar(char chr, int base\_port)

{

char regst = inp(base\_port + 0x04); // MCR

outp(base\_port + 0x04, regst | 0x02); // RTS on

outp(base\_port, chr);

delay(100);

while (!check\_snd(base\_port)); // Ожидание доставки

outp(base\_port + 0x04, regst & 0xfd); // Очистка RTS

}

char com\_inchar(int base\_port)

{

return inp(base\_port);

}

int check\_snd(int base\_port)

{

unsigned char regst;

regst = inp(base\_port + 0x05); // LSR

return ((regst & 0x20) >> 5) == 0x01; // check 5 bit

int main()

{

int port1 = 0x3f8;

int port2 = 0x2f8;

char inf;

if (!Com\_Init(port1))

printf("COM initialization error\n");

printf("Previous (COM2): %c\n", com\_inchar(port2));

if (check\_snd(port1) == 1)

printf("Ready to write\n");

else

printf("Port not ready... broken?\n");

printf("Enter character: ");]

rewind(stdin); ]

inf = getchar();

com\_outchar(inf, port1);

printf("Current (COM2): %c\n", com\_inchar(port2));

return 0;

}

1. Тестирование программ

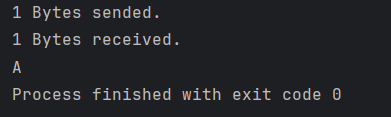


Рисунок 4.1 – Результат работы программы, взаимодействующей с портами ввода-вывода.



Рисунок 4.2 – Результат работы программы, использующей BIOS прерывание 14h.

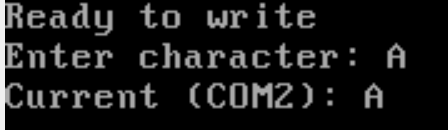


Рисунок 4.3 – Результат работы программы, работающей с COM-портами через регистры как с устройствами ввода-вывода.

1. Заключение

В ходе лабораторной удалось передать 1 байт информации через последовательный порт с использованием различных механизмов.

Для эмуляции COM портов использовался Virtual Serial Port Driver, для эмуляции DOS - Oracle Virtual Box.