

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №1
«Последовательный порт»

Выполнил:

Студент группы 950501
Деркач А.В.

Проверил:

Преподаватель
Одинец Д.Н.

Минск, 2021

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.

2. 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

3. 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>
using namespace std;

HANDLE COM_Port_1;
LPCTSTR Port_Name_1 = L"COM1";
HANDLE COM_Port_2;
LPCTSTR Port_Name_2 = L"COM2";

void Read_from_COM()
{
    DWORD Size;
    char Received_Char;
    ReadFile(COM_Port_2, &Received_Char, 1, &Size, 0);
    if (Size > 0)
    {
        cout << Received_Char;
    }
}

int main()
{
    COM_Port_1 = ::CreateFile(Port_Name_1, GENERIC_WRITE, 0, 0,
    OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);
```

```

COM_Port_2 = ::CreateFile(Port_Name_2, GENERIC_READ, 0, 0,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);

if (COM_Port_1 == INVALID_HANDLE_VALUE)
{
    if (GetLastError() == ERROR_FILE_NOT_FOUND)
    {
        cout << "COM-port does not exist!\n";
    }
    cout << "Some other error.\n";
}

DCB Serial_Params = { 0 };
Serial_Params.DCBlength = sizeof(Serial_Params);
if (!GetCommState(COM_Port_1, &Serial_Params))
{
    cout << "Getting state error.\n";
}
Serial_Params.BaudRate = CBR_9600;
Serial_Params.ByteSize = 8;
Serial_Params.StopBits = ONESTOPBIT;
Serial_Params.Parity = NOPARITY;
if (!SetCommState(COM_Port_2, &Serial_Params))
{
    cout << "Error setting serial port state.\n";
}

char data = 'A';
DWORD Size = sizeof(data);
DWORD Bytes_Written;

BOOL Ret = WriteFile(COM_Port_1, &data, Size, &Bytes_Written,
NULL);

cout << Size << " Bytes in string. " << Bytes_Written << "
Bytes sended. " << endl;

Read_from_COM();

return 0;
}

```

3.2. Листинг программы, использующей BIOS прерывание 14h.

```

.model small
.stack 100h

.data

Error_Write db "Write error!",0Dh,0Ah,'$'
Error_Read db "Read error!",0Dh,0Ah,'$'
Information db "Byte sent: $"

```

```

.code

jmp start

Init_COM1 proc
    xor ax,ax
    mov al,10100011b
    mov dx,0
    int 14h
    ret
Init_COM1 endp

IsWrite_COM1 proc
    mov al,'A'
    mov ah,1
    mov dx,0
    int 14h
    test al,80h
    jnz NoWrite
    ret
IsWrite_COM1 endp

NoWrite proc
    mov ah,9
    mov dx,offset Error_Write
    add dx,2
    int 21h
    ret
NoWrite endp

IsRead_COM2 proc
    mov ah,2
    mov dx,1
    int 14h
    test al,80h
    jnz NoRead
    ret
IsRead_COM2 endp

NoRead proc
    mov ah,9
    mov dx,offset Error_Read
    add dx,2
    int 21h
    ret
NoRead 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_COM1
        call IsWrite_COM1
        mov al,'e'
        call IsRead_COM2
        push ax

        mov ah,9
        mov dx,offset Information
        add dx,2
        int 21h

        pop ax
        call Output
        call Exit

end start

```

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

```

.model small
.stack 100h

.data

Error_Write db "Write error!",0Dh,0Ah,'$'
Error_Read db "Read error!",0Dh,0Ah,'$'
Information db "Byte sent: $"
Data_Byte db 'A'
Data_Byte2 db ?

.code

Init_COM1 proc
        mov al,80h

```

```

    mov dx,3FBh
    out dx,al

    mov dx,3F8h
    mov al,00h
    out dx,al
    mov al,0Ch
    mov dx,3F9h
    out dx,al

    mov dx,3FCh
    mov al,00001011b
    out dx,al

    mov dx,3F9h
    mov al,0
    out dx,al
    ret
Init_COM1 endp

IsWrite_COM1 proc
    xor al,al
    mov dx,3FDh
    in al,dx
    test al,10h
    jnz NoWrite
    ret
IsWrite_COM1 endp

NoWrite proc
    mov ah,9
    mov dx,offset Error_Write
    int 21h
    ret
NoWrite endp

IsRead_COM2 proc
    xor al,al
    mov dx,3FDh
    in al,dx
    test al,10b
    jnz NoRead
    ret
IsRead_COM2 endp

NoRead proc
    mov ah,9
    mov dx,offset Error_Read

```

```

        int 21h
        ret
NoRead endp

Send_Byte proc
    mov dx,3F8h
    mov al,Data_Byte
    out dx,al
    ret
Send_Byte endp

Read_Byte proc
    mov dx,3F8h
    in al,dx
    mov Data_Byte2,al
    ret
Read_Byte endp

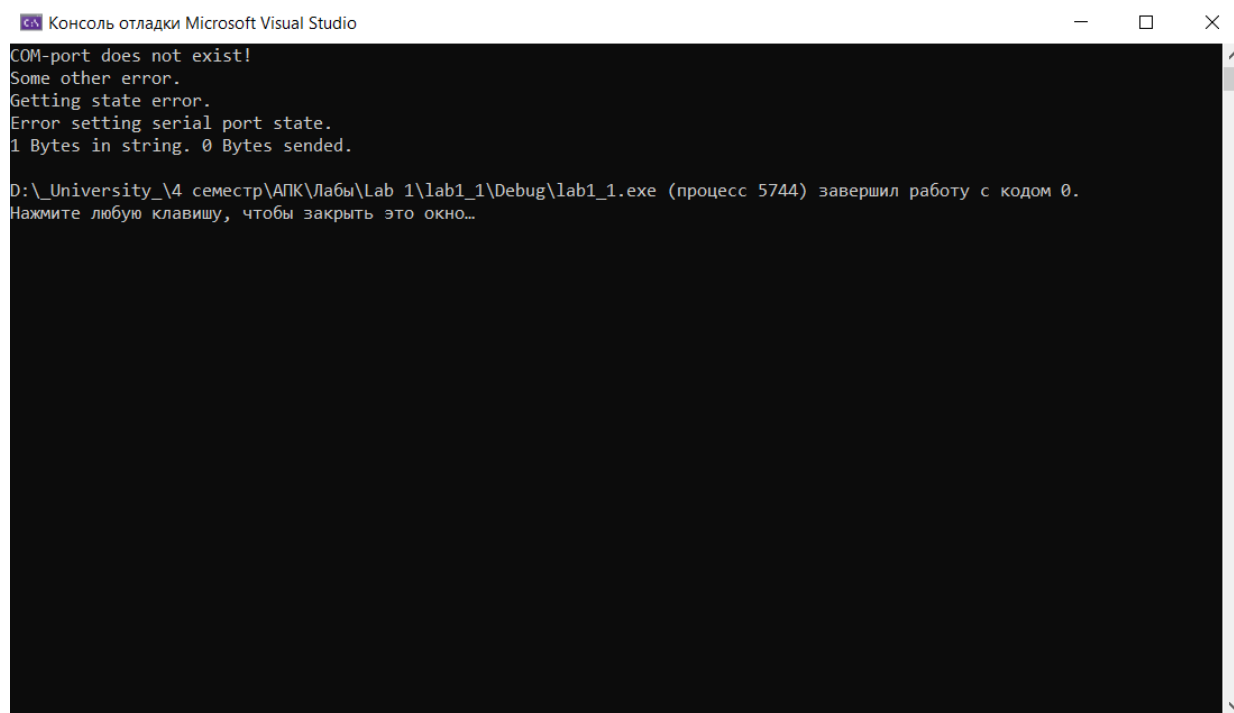
Exit proc
    mov ax,4C00h
    int 21h
    ret
Exit endp

start:
    mov ax,@data
    mov ds,ax
    call Init_COM1
    call IsWrite_COM1
    call Send_Byte
    mov al,2
    call IsRead_COM2
    call Read_Byte
    mov dx,offset Information
    mov ah,9
    int 21h
    mov ah,02h
    mov dl,Data_Byte2
    int 21h
    call Exit

end start

```

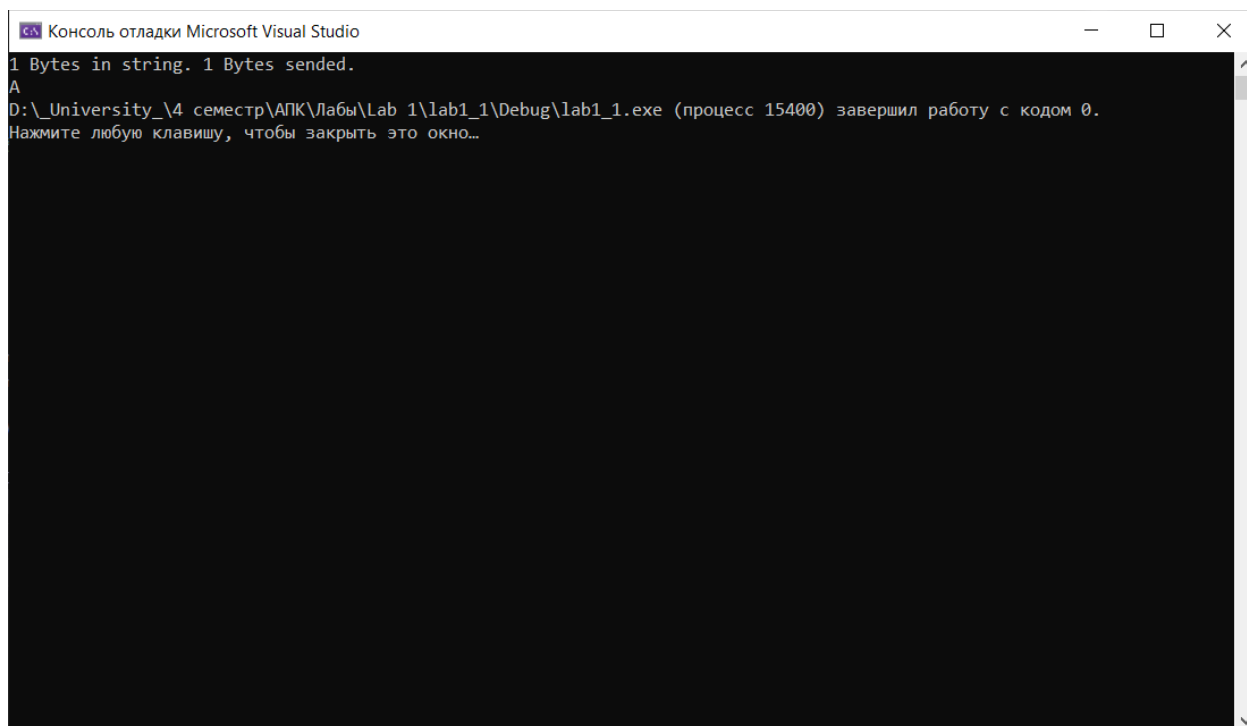
4. Тестирование программ



```
Консоль отладки Microsoft Visual Studio
COM-port does not exist!
Some other error.
Getting state error.
Error setting serial port state.
1 Bytes in string. 0 Bytes sended.

D:\_University_\4 семестр\АПК\Лабы\Lab 1\lab1_1\Debug\lab1_1.exe (процесс 5744) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рисунок 4.1 – Результат работы программы, взаимодействующей с портами ввода-вывода, при включенной эмуляции COM-портов.



```
Консоль отладки Microsoft Visual Studio
1 Bytes in string. 1 Bytes sended.
А
D:\_University_\4 семестр\АПК\Лабы\Lab 1\lab1_1\Debug\lab1_1.exe (процесс 15400) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рисунок 4.2 – Результат работы программы, взаимодействующей с портами ввода-вывода, при включенной эмуляции COM-портов.

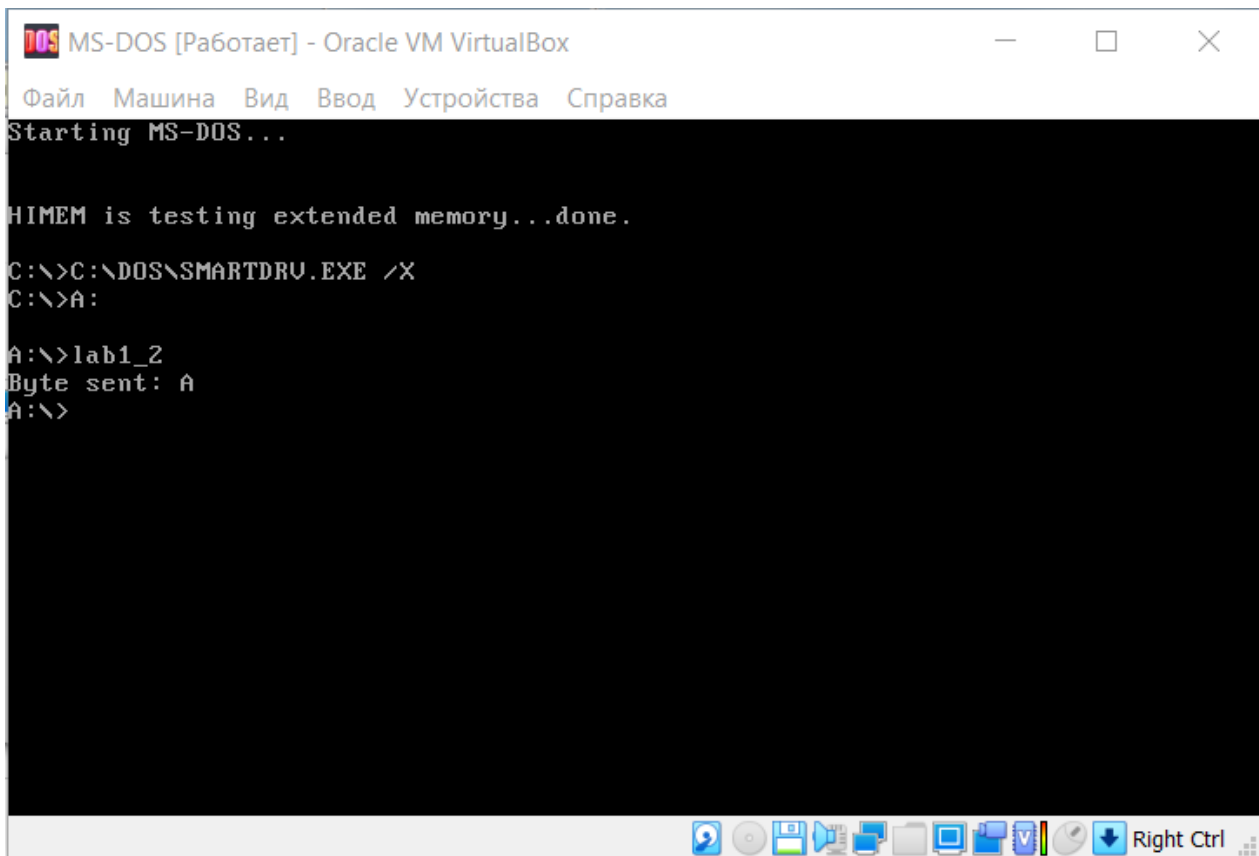


Рисунок 4.3 – Результат работы программы, использующей BIOS прерывание 14h.

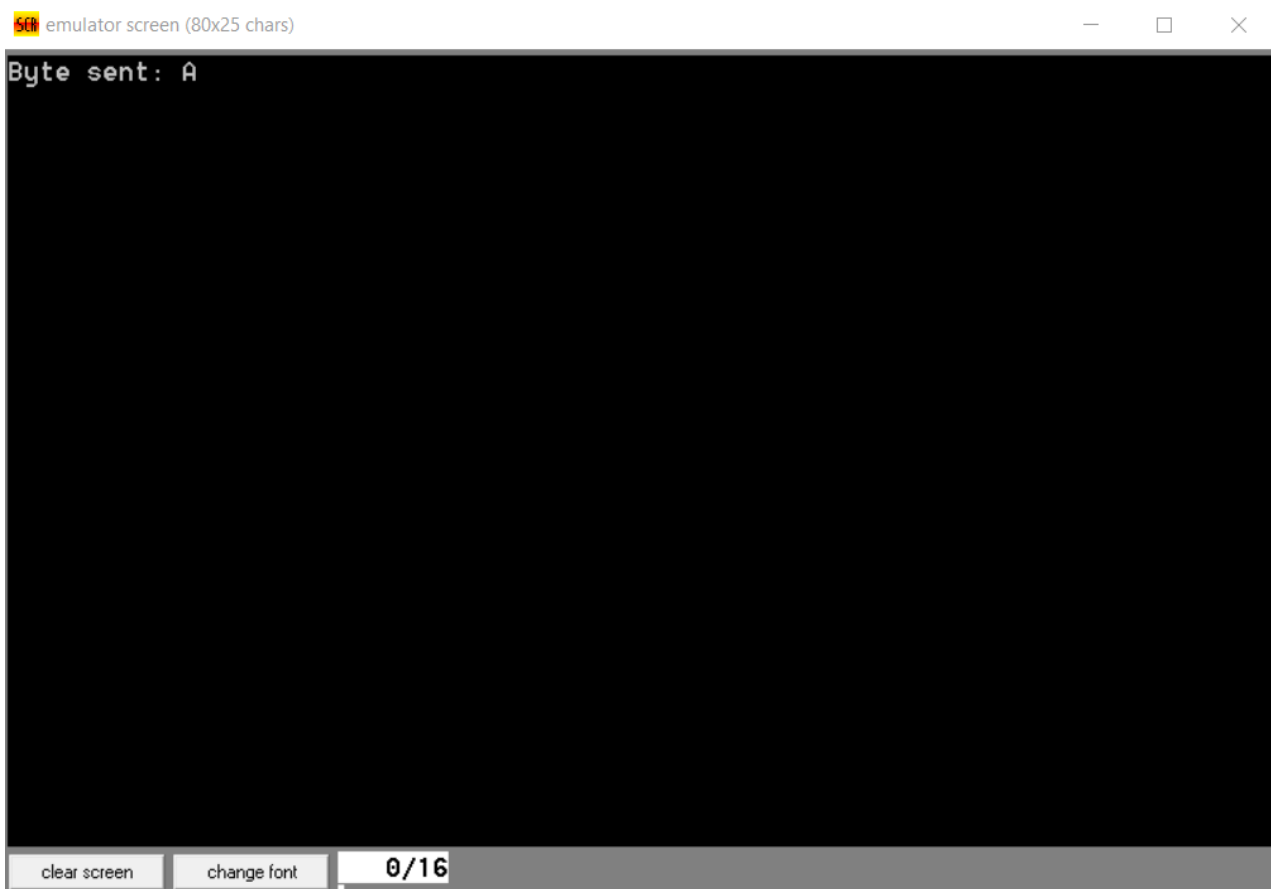


Рисунок 4.4 – Результат работы программы, работающей с СОМ-портами через регистры как с устройствами ввода-вывода.

5. 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 используется Oracle Virtual Box на хосте 64-ех разрядной Windows 10.