

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №1
«Последовательный порт»

Выполнил:
Студент группы 050503
Деруго Д.В.

Проверил:
Преподаватель
Одинец Д.Н.

Минск, 2022

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 COM1;
LPCTSTR Port_Name_1 = L"COM2";
HANDLE COM2;
LPCTSTR Port_Name_2 = L"COM3";

int main() {

    // Initialization COM1, COM2
    COM1 = ::CreateFile(Port_Name_1, GENERIC_WRITE, 0, 0, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);
    COM2 = ::CreateFile(Port_Name_2, GENERIC_READ, 0, 0, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);

    if (COM1 == INVALID_HANDLE_VALUE) {
        if (GetLastError() == ERROR_FILE_NOT_FOUND) {
            cout << "Can't find COM pair\n";
        }
        cout << "Some other error\n";
    }

    // Set parameters
```

```

DCB Serial_Params = { 0 };
Serial_Params.DCBlength = sizeof(Serial_Params);
if (!GetCommState(COM1, &Serial_Params)) {
    cout << "Getting state error\n";
}
Serial_Params.BaudRate = CBR_9600; // asm -- 2400
Serial_Params.ByteSize = 8;
Serial_Params.StopBits = ONESTOPBIT;
Serial_Params.Parity = NOPARITY;
if (!SetCommState(COM2, &Serial_Params)) {
    cout << "Getting state error\n";
}

char data = '0';
DWORD Size = sizeof(data);
DWORD Bytes_Written;

BOOL Ret = WriteFile(COM1, &data, Size, &Bytes_Written, NULL);

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

char Received_Char;
ReadFile(COM2, &Received_Char, 1, &Size, 0);
if (Size > 0) {
    cout << Received_Char;
}

return 0;
}

```

3.2. Листинг программы, использующей BIOS прерывание 14h.

```

.model small
.stack 256

.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          ; clear ax
    mov al,00100111b   ; baud 150, stop-bit 1, 7 bit word
    mov dx,0           ; port-transmitter number
    int 14h            ; read
    ret

```

```
Init_COM1 endp
```

```
IsWrite_COM1 proc    ; verification
    mov al,'2'        ; set symbol
    mov ah,1          ;
    mov dx,0          ; port-transmitter number
    int 14h
    test al,80h        ; 10000000b mean signal is detected
    jnz NoWrite        ; no signal!
    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 dx,1          ; port-recivier number
    int 14h
    test al,80h        ; 10000000b mean signal is detected
    jnz NoRead
    ret
IsRead_COM2 endp
```

```
NoRead proc
    mov ah,09h
    mov dx,offset Error_Read
    add dx,2
    int 21h
    ret
NoRead endp
```

```
Output proc
    mov ah,02h
    mov dl,al          ; transmitted byte in 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, '2'
    call IsRead_COM2
    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 '3'
Data_Byte2 db ?

.code

Init_COM1 proc
; set 7=1 to 3FB -- 3F8, 3F9 able to control speed
    mov al, 80h ; 7bit = 1
    mov dx, 3FBh ; LCR number
    out dx, al ; set setting

; set frequency 9600 baud
    mov dx, 3F8h ; COM1 number
    mov al, 00h ;
    out dx, al ; set
    mov al, 0Ch ;
    mov dx, 3F9h ; setting data
    out dx, al ; transfer frequency

; set in Modem Control Register required bytes
; 0 - DTR, 1 - RTS, 3 - OUT2
    mov dx, 3FCh ; MCR number
```

```

    mov al,00001011b
    out dx,al

    mov dx,3F9h ; 3FB 7b=1, 3F9 calls interrupts
    mov al,00h ; 0b & 1b = 0, no interrupts
    out dx,al ; set
    ret
Init_COM1 endp

IsWrite_COM1 proc
    xor al,al
    mov dx,3FDh ; portout function
    in al,dx
    test al,10h ; check 5 = 1 setted
    jnz NoWrite ; bit = 0, some error in port
    ret
IsWrite_COM1 endp

NoWrite proc
    mov ah,09h
    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 ; check bit setted
    jnz NoRead ; bit = 0, some error in port
    ret
IsRead_COM2 endp

NoRead proc
    mov ah,09h
    mov dx,offset Error_Read
    int 21h
    ret
NoRead endp

Send_Byte proc
    mov dx,3F8h
    mov al,Data_Byte
    out dx,al ; load data in COM1-port
    ret
Send_Byte endp

```

```

Read_Byte proc
    mov dx,3F8h
    in al,dx
    mov Data_Byte2,al ; load byte from COM1-port
    ret
Read_Byte endp

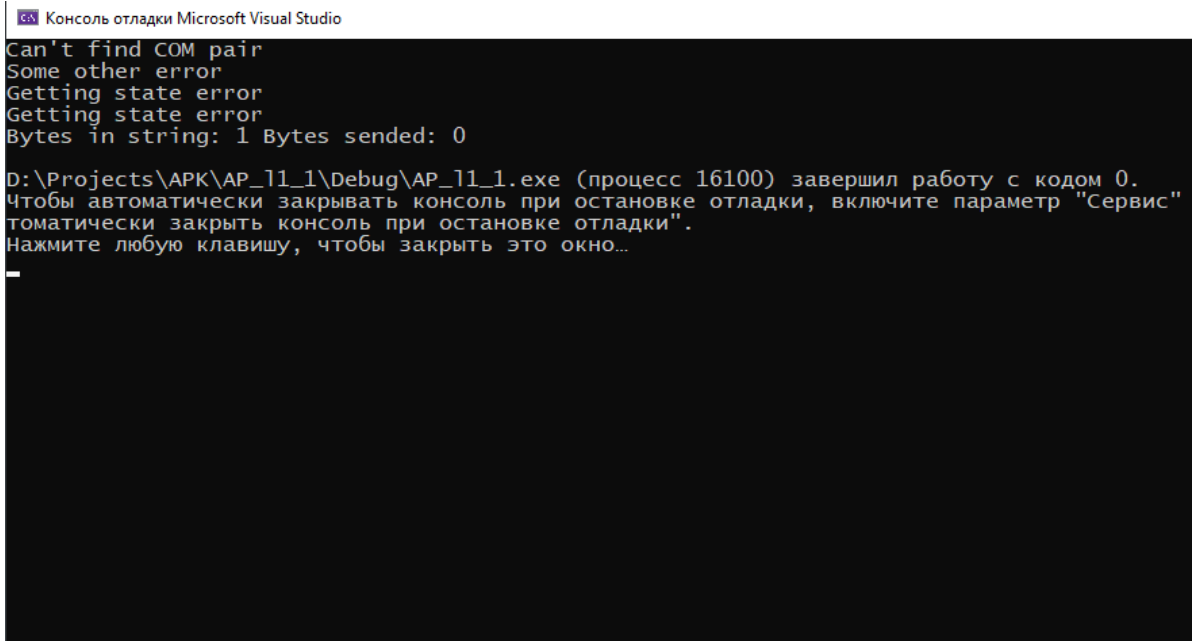
Exit proc
    mov ax,4C00h
    int 21h
    ret
Exit endp

start:
    mov ax,@data
    mov ds,ax
    call Init_COM1
    call IsWrite_COM1
    call Send_Byte
    mov al,02h
    call IsRead_COM2
    call Read_Byte
    mov dx,offset Information
    mov ah,09h
    int 21h
    mov ah,02h
    mov dl,Data_Byte2
    int 21h
    call Exit

end start

```

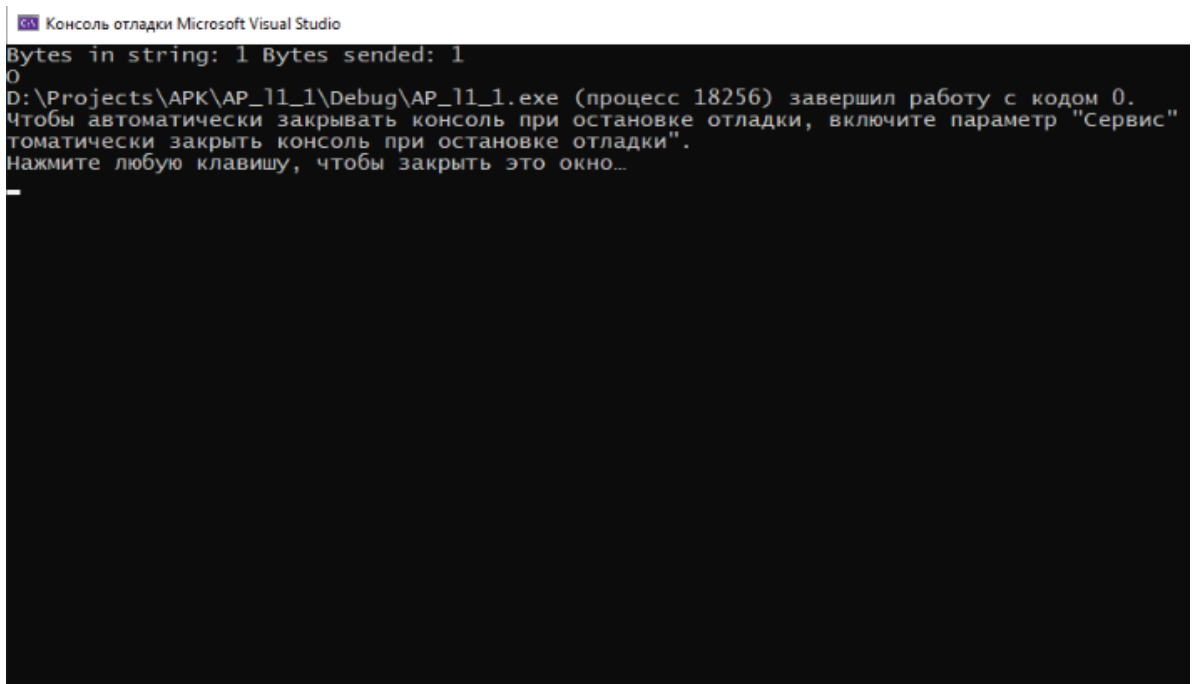
4. Тестирование программ



```
Консоль отладки Microsoft Visual Studio
Can't find COM pair
Some other error
Getting state error
Getting state error
Bytes in string: 1 Bytes sent: 0

D:\Projects\APK\AP_11_1\Debug\AP_11_1.exe (процесс 16100) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис"
томатически закрыть консоль при остановке отладки".
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
Bytes in string: 1 Bytes sent: 1
0
D:\Projects\APK\AP_11_1\Debug\AP_11_1.exe (процесс 18256) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис"
томатически закрыть консоль при остановке отладки".
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 и результат работы программы, работающей с COM-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 9, для эмуляции DOS используется DOSBox-0.74-3 на хосте 64-ех разрядной Windows 10.