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 050504 Преподаватель

Сидорович Я.А 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.
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>  
using namespace std;  
int main()  
{  
 HANDLE COM\_Port\_1 = CreateFile("COM1", GENERIC\_READ | GENERIC\_WRITE, 0, NULL,OPEN\_EXISTING, 0, NULL);  
 HANDLE COM\_Port\_2 = CreateFile("COM2", GENERIC\_READ | GENERIC\_WRITE, 0, NULL,OPEN\_EXISTING, 0, NULL);  
 if (COM\_Port\_1 == INVALID\_HANDLE\_VALUE || COM\_Port\_2 == 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);  
 Serial\_Params.BaudRate = CBR\_9600;  
 Serial\_Params.ByteSize = 8;  
 Serial\_Params.StopBits = ONESTOPBIT;  
 Serial\_Params.Parity = NOPARITY;  
 if (!SetCommState(COM\_Port\_2, &Serial\_Params) || !SetCommState(COM\_Port\_1,&Serial\_Params))  
 {  
 cout << "Error setting serial port state.\n";  
 return 1;  
 }  
 char data = 'Y';  
 DWORD Size = sizeof(data);  
 DWORD Bytes\_Written;  
 cout<< Bytes\_Written<<" -- \n";  
 WriteFile(COM\_Port\_1, &data, Size, &Bytes\_Written,NULL);  
 cout << Size << " Bytes in string. " << Bytes\_Written << " Bytes sended. " << endl;  
 char Received\_Char;  
 ReadFile(COM\_Port\_2, &Received\_Char, 1, &Size, NULL);  
 if (Size > 0)  
 {  
 cout <<"Char sended: "<< Received\_Char;  
 }  
 return 0;  
}

3.2. Листинг программы, использующей BIOS прерывание 14h.

.model small

.stack 100h

.data

.code

jmp start

Init\_COM1 proc

   xor ax,ax

   mov al,10100011b

   mov dx,0

   int 14h

   ret

Init\_COM1 endp

Write\_COM1 proc

   mov al,'A'

   mov ah,1

   mov dx,0

   int 14h

   ret

Write\_COM1 endp

Read\_COM2 proc

    mov ah,2

    mov dx,1

    int 14h

    ret

Read\_COM2 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 Write\_COM1

   call Read\_COM2

   call Output

   call Exit

end start

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

.model small

.stack 100h

.data

Information db "Byte sent: $"

Data\_Byte db 'A'

Data\_Byte2 db ?

.code

IsWrite\_COM1 proc

    xor al,al

    mov dx,3FDh

    in al,dx

    ret

IsWrite\_COM1 endp

Send\_Byte proc

    mov dx,3F8h

    mov al,Data\_Byte

    out dx,al

    ret

Send\_Byte endp

IsRead\_COM2 proc

    xor al,al

    mov dx,3FDh

    in al,dx

    ret

IsRead\_COM2 endp

Read\_Byte proc

    mov dx,2F8h

    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 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

1. Тестирование программ

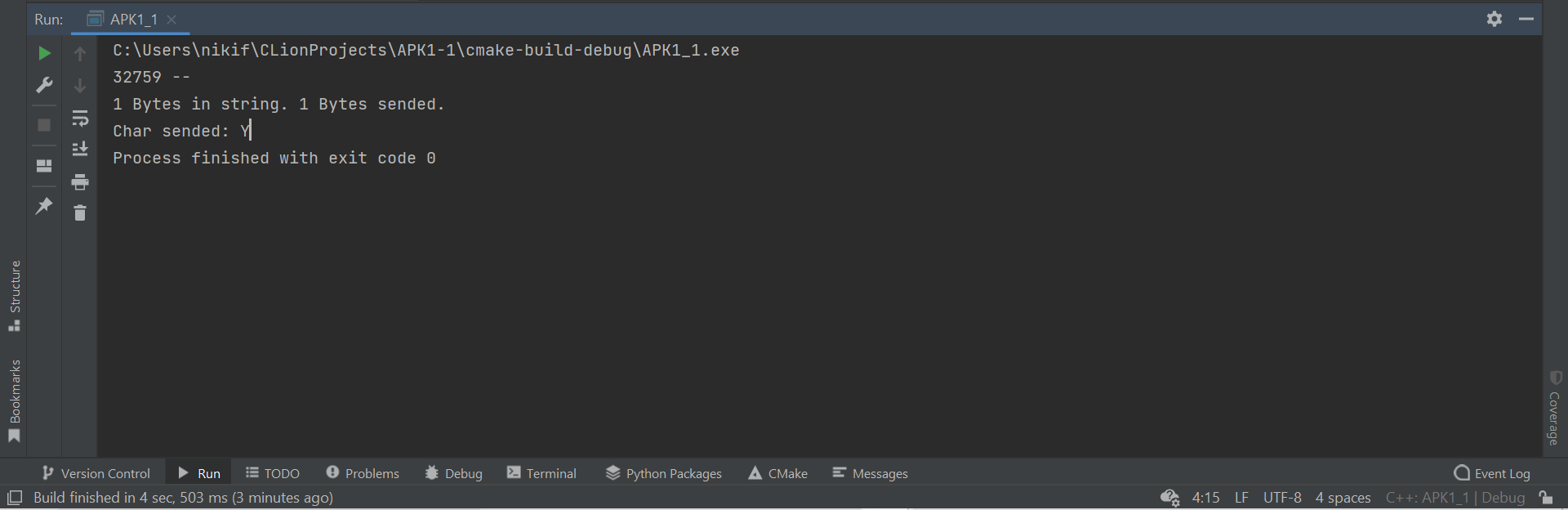


Рисунок 4.1 – Результат работы программы, взаимодействующей с портами ввода-вывода, при включенной эмуляции COM-портов.

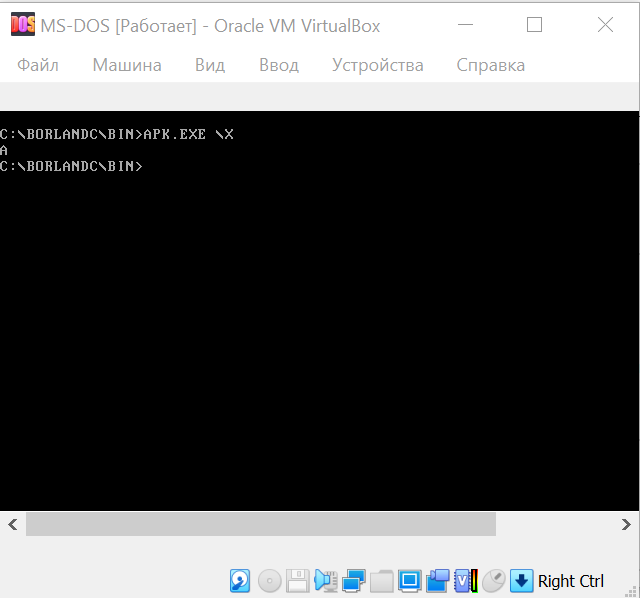


Рисунок 4.3 – Результат работы программы, использующей BIOS прерывание 14h.

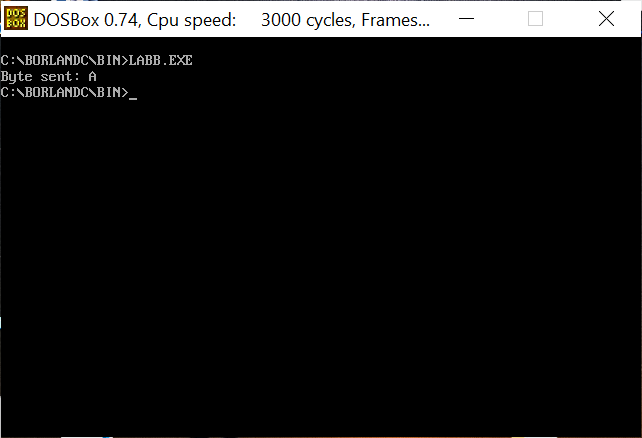


Рисунок 4.4 – Результат работы программы, работающей с COM-портами через регистры как с устройствами ввода-вывода.

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