МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

кафедра САПР



**ЗВІТ**

до лабораторної роботи №5

**ПРОГРАМУВАННЯ ПОСЛІДОВНОГО ІНТЕРФЕЙСУ**

Виконав:

студент групи КНз-21

Чалий Михайло

Перевірив

Мазур В.В.

Львів-2014

# **Мета роботи**

Мета роботи – засвоїти основи організації обміну інформацією між ЕОМ і отримати практичнi навики програмування вводу-виводу через послідовний інтерфейс.

# **Короткі теоретичні відомості**

При асинхронному зв'язку машина посилає або приймає байти інформації порціями по одному біту. Тимчасові інтервали між байтами при цьому несуттєві, але дуже важливі інтервали між окремими бітами байта. Сигнал на лінії може бути високого або низького рівня, що відповідає логічним нулю і одиниці, і кажуть, що лінія відмічена (marking), коли рівень високий, і пуста (spacing), коли рівень низький.

Лінія підтримується у відміченому стані, коли по ній не передають даних. При початку передачі байта даних сигнал падає у 0, відмічаючи стартовий біт. Потім слідують вісім бітів даних (іноді менше) у вигляді набору високих і низьких рівнів. Останній біт даних може супроводжуватися бітом парності, що використовується для виявлення помилок, а потім у послідовність включаються 1 або більше стоп-біти, яким відповідає високий рівень. Ці стоп-біти починають відмічений стан, який буде зберігатися доти, поки не почнеться передача наступного байта даних; число стоп-бітів суттєве, оскільки вони встановлюють мінімальний час, який повиннен пройти перед наступним стартовим бітом. На рис. 1. показана ця послідовність.

Звичайно, передаюча і приймальна станції повинні застосовувати один і той же протокол для ланцюжків бітів і працювати з однією і тією ж швидкістю обміну (що вимірюється у бітах за секунду, які також називаються бодами). При обміні можуть легко виникати помилки, тому комунікаційне обладнання надає різносторонню інформацію про статус як самого порту, так і про приєднаного до нього модему. Задачею модему є перетворення сигналу, що генерується портом комунікації, в акустичний сигнал, який може потім бути переданий по телефонному каналу. Більшість модемів надає також додаткові комунікаційні можливості, такі, як автоматичний виклик і відповідь, які не підтримуються самим портом комунікації.

Стопові біти

8 біт даних

Біт парності

Наступний стартовий біт

Стартовий біт

Проміжок часу між бітами

Рис.1. Передача одного байта послідовних даних

# Програма передавача (Producer)

sseg segment stack

db 256 dup(?)

sseg ends

dseg segment

buf db "mike$"

error\_msg db "Error occured$"

dseg ends

cseg segment

assume cs:cseg,ds:dseg,ss:sseg

start:

jmp main

main:

push ds

mov ax,0

push ax

mov ax,dseg

mov ds,ax

; Initialization

mov ah,0

mov dx,0 ; Port number

mov al,11111111B

int 14h

mov si,0

mov cx,4 ; Size of the buffer

print\_symbol:

; Check status

mov ah,3

int 14h

test ah,100000B

jz print\_symbol

mov ah,1

mov al,buf[si]

int 14h

test ah,11110b

jnz err

inc si

loop print\_symbol

exit:

mov ax,4c00h

int 21h

err:

mov dx,offset error\_msg

mov ah,9

int 21h

jmp exit

cseg ends

end start

# Лістінг файл програми передавача

JWasm v2.11, Oct 20 2013

.\L5COMP.ASM

00000000 sseg segment stack

00000000 000000000000000000 db 256 dup(?)

00000100 sseg ends

00000000 dseg segment

00000000 6D696B6524 buf db "mike$"

00000005 4572726F72206F6363 error\_msg db "Error occured$"

00000013 dseg ends

00000000 cseg segment

assume cs:cseg,ds:dseg,ss:sseg

00000000 start:

00000000 EB00 jmp main

00000002 main:

00000002 1E push ds

00000003 B80000 mov ax,0

00000006 50 push ax

00000007 B80000 mov ax,dseg

0000000A 8ED8 mov ds,ax

; Initialization

0000000C B400 mov ah,0

0000000E BA0000 mov dx,0 ; Port number

00000011 B0FF mov al,11111111B

00000013 CD14 int 14h

00000015 BE0000 mov si,0

00000018 B90400 mov cx,4 ; Size of the buffer

0000001B print\_symbol:

; Check status

0000001B B403 mov ah,3

0000001D CD14 int 14h

0000001F F6C420 test ah,100000B

00000022 74F7 jz print\_symbol

00000024 B401 mov ah,1

00000026 8A840000 mov al,buf[si]

0000002A CD14 int 14h

0000002C F6C41E test ah,11110b

0000002F 7508 jnz err

00000031 46 inc si

00000032 E2E7 loop print\_symbol

00000034 exit:

00000034 B8004C mov ax,4c00h

00000037 CD21 int 21h

00000039 err:

00000039 BA0000 mov dx,offset error\_msg

0000003C B409 mov ah,9

0000003E CD21 int 21h

00000040 EBF2 jmp exit

00000042 cseg ends

end start

Binary Map:

Segment Pos(file) RVA Size(fil) Size(mem)

---------------------------------------------------------------

<header> 0 0 30 0

sseg 30 0 100 100

dseg 130 100 13 13

cseg 150 120 42 42

---------------------------------------------------------------

192 162

Macros:

N a m e Type

@CatStr . . . . . . . . . . . . Func

@Environ . . . . . . . . . . . . Func

@InStr . . . . . . . . . . . . . Func

@SizeStr . . . . . . . . . . . . Func

@SubStr . . . . . . . . . . . . Func

Segments and Groups:

N a m e Size Length Align Combine Class

cseg . . . . . . . . . . . . . . 16 Bit 0042 Para Private ''

dseg . . . . . . . . . . . . . . 16 Bit 0013 Para Private ''

sseg . . . . . . . . . . . . . . 16 Bit 0100 Para Stack ''

Symbols:

N a m e Type Value Attr

buf . . . . . . . . . . . . . . Byte[5] 0h dseg

err . . . . . . . . . . . . . . L Near 39h cseg

error\_msg . . . . . . . . . . . Byte[14] 5h dseg

exit . . . . . . . . . . . . . . L Near 34h cseg

main . . . . . . . . . . . . . . L Near 2h cseg

print\_symbol . . . . . . . . . . L Near 1Bh cseg

start . . . . . . . . . . . . . L Near 0h cseg Public

.\L5COMP.ASM: 58 lines, 2 passes, 0 ms, 0 warnings, 0 errors

# Програма для приймача (Consumer)

sseg segment stack

db 256 dup(?)

sseg ends

dseg segment

buf db 4 dup (?)

error\_msg db "Error occures$"

dseg ends

cseg segment

assume cs:cseg,ds:dseg,ss:sseg

start:

jmp main

main:

push ds

mov ax,0

push ax

mov ax,dseg

mov ds,ax

; Init port

mov ah,0

mov dx,0

mov al,11111111B

int 14h

mov si,0

mov cx,4

read\_symbol:

; Check status

mov ah,3

int 14h

test ah,1B

jz read\_symbol

; Read symbol

mov ah,2

int 14h

test ah,11110B

jnz err

mov buf[si],al

inc si

loop read\_symbol

exit:

mov ax,4c00h

int 21h

err:

mov dx,offset error\_msg

mov ah,9

int 21h

jmp exit

cseg ends

end start

# Лістінг програми приймача

JWasm v2.11, Oct 20 2013

.\L5COMC.ASM

00000000 sseg segment stack

00000000 000000000000000000 db 256 dup(?)

00000100 sseg ends

00000000 dseg segment

00000000 00000000 buf db 4 dup (?)

00000004 4572726F72206F6363 error\_msg db "Error occures$"

00000012 dseg ends

00000000 cseg segment

assume cs:cseg,ds:dseg,ss:sseg

00000000 start:

00000000 EB00 jmp main

00000002 main:

00000002 1E push ds

00000003 B80000 mov ax,0

00000006 50 push ax

00000007 B80000 mov ax,dseg

0000000A 8ED8 mov ds,ax

; Init port

0000000C B400 mov ah,0

0000000E BA0000 mov dx,0

00000011 B0FF mov al,11111111B

00000013 CD14 int 14h

00000015 BE0000 mov si,0

00000018 B90400 mov cx,4

0000001B read\_symbol:

; Check status

0000001B B403 mov ah,3

0000001D CD14 int 14h

0000001F F6C401 test ah,1B

00000022 74F7 jz read\_symbol

; Read symbol

00000024 B402 mov ah,2

00000026 CD14 int 14h

00000028 F6C41E test ah,11110B

0000002B 750C jnz err

0000002D 88840000 mov buf[si],al

00000031 46 inc si

00000032 E2E7 loop read\_symbol

00000034 exit:

00000034 B8004C mov ax,4c00h

00000037 CD21 int 21h

00000039 err:

00000039 BA0000 mov dx,offset error\_msg

0000003C B409 mov ah,9

0000003E CD21 int 21h

00000040 EBF2 jmp exit

00000042 cseg ends

end start

Binary Map:

Segment Pos(file) RVA Size(fil) Size(mem)

---------------------------------------------------------------

<header> 0 0 30 0

sseg 30 0 100 100

dseg 130 100 12 12

cseg 150 120 42 42

---------------------------------------------------------------

192 162

Macros:

N a m e Type

@CatStr . . . . . . . . . . . . Func

@Environ . . . . . . . . . . . . Func

@InStr . . . . . . . . . . . . . Func

@SizeStr . . . . . . . . . . . . Func

@SubStr . . . . . . . . . . . . Func

Segments and Groups:

N a m e Size Length Align Combine Class

cseg . . . . . . . . . . . . . . 16 Bit 0042 Para Private ''

dseg . . . . . . . . . . . . . . 16 Bit 0012 Para Private ''

sseg . . . . . . . . . . . . . . 16 Bit 0100 Para Stack ''

Symbols:

N a m e Type Value Attr

buf . . . . . . . . . . . . . . Byte[4] 0h dseg

err . . . . . . . . . . . . . . L Near 39h cseg

error\_msg . . . . . . . . . . . Byte[14] 4h dseg

exit . . . . . . . . . . . . . . L Near 34h cseg

main . . . . . . . . . . . . . . L Near 2h cseg

read\_symbol . . . . . . . . . . L Near 1Bh cseg

start . . . . . . . . . . . . . L Near 0h cseg Public

.\L5COMC.ASM: 58 lines, 2 passes, 0 ms, 0 warnings, 0 errors

# Висновок

В процесі виконання роботи засвоїв основи організації обміну інформацією між ЕОМ і отримати практичнi навики програмування вводу-виводу через послідовний інтерфейс.