**Serial Communications Lab**

**Objective:**

- To be familiar with PC serial ports.

**Requirement:**

Connect two PCs directly through their serial port and write a program that splits the screen into two halves. The upper half displays what is being typed on the PC and the lower half displays what is being typed on the other PC.

**Guide Lines:**

1. The program may contain two main procedures. One procedure for initializing the UART and the other procedure should detect any pressed keys and transmit it serially as well as detecting any received characters on the serial port.

The first procedure:

* 1. This procedure is to be written for initializing the UART (baud rate, parity, data bits, stop bits,…). Use the same parameters on both terminals.
* **Set Divisor Latch Access Bit**

mov dx,3fbh ; Line Control Register

mov al,10000000b ;Set Divisor Latch Access Bit

out dx,al ;Out it

* **Set LSB byte of the Baud Rate Divisor Latch register.**

mov dx,3f8h

mov al,0ch

out dx,al

* **Set MSB byte of the Baud Rate Divisor Latch register.**

mov dx,3f9h

mov al,00h

out dx,al

* **Set port configuration**

mov dx,3fbh

mov al,00011011b

* + 0:Access to Receiver buffer, Transmitter buffer
  + 0:Set Break disabled
  + 011:Even Parity
  + 0:One Stop Bit
  + 11:8bits

out dx,al

* 1. You can use the BIOS function Int10h/AH=00h to clear the screen. Setup your variables for the upper and lower halves of the screen. You could use different colors (attributer) for the upper and lower halves. You could also use windowing BIOS functions to support upper and lower half scrolling. For example; the following code uses Int 10h/AH=06 to initialize a window [(0,0) to (79,12)] or scroll window contents up

mov ah,6 ; function 6

mov al,1 ; scroll by 1 line

mov bh,7 ; normal video attribute

mov ch,0 ; upper left Y

mov cl,0 ; upper left X

mov dh,12 ; lower right Y

mov dl,79 ; lower right X

int 10h

The second procedure:

* 1. This procedure should detect any pressed keys and transmit it serially as well as detecting any received characters on the serial port.
     1. If a key is pressed, then a character is in the keyboard buffer (use the BIOS function Int 16h/AH=01h to check if a character is available in the keyboard buffer) then read it and display it in the upper half of the screen and **send it to the UART**. If a character is present you could read it using Int16h/AH=00h, then display it at any place on the screen. Use the BIOS functions Int10h/AH=2 to set the cursor position to the upper half of the screen then use Int10h/AH=9 to display the character on the screen at the preset cursor position.
     2. If a character is in the UART RDR (Receiver Data Register) then read it and display it in the lower half of the screen.
     3. You scroll the upper and lower half of the screen independently. Int10h can be used for that also.
     4. The above steps should be repeated until one of the users presses the ESC key. (The ASCII for ESC=27).

**Sending a value**

;Check that Transmitter Holding Register is Empty

mov dx , 3FDH ; Line Status Register

AGAIN: In al , dx ;Read Line Status

test al , 00100000b

JZ AGAIN ;Not empty

;If empty put the VALUE in Transmit data register

mov dx , 3F8H ; Transmit data register

mov al,VALUE

out dx , al

**Receiving a value**

;Check that Data is Ready

mov dx , 3FDH ; Line Status Register

CHK: in al , dx

test al , 1

JZ CHK ;Not Ready

;If Ready read the VALUE in Receive data register

mov dx , 03F8H

in al , dx

mov VALUE , al

1. Don’t write your whole program in one shot, divide your work into 3 steps:
   1. Initialize and send any dummy message from PC1 to PC2

2 codes, 1 for send, 1 for receive

* 1. Add to your code, so that it takes a character from KB on PC1 and sends it to PC2

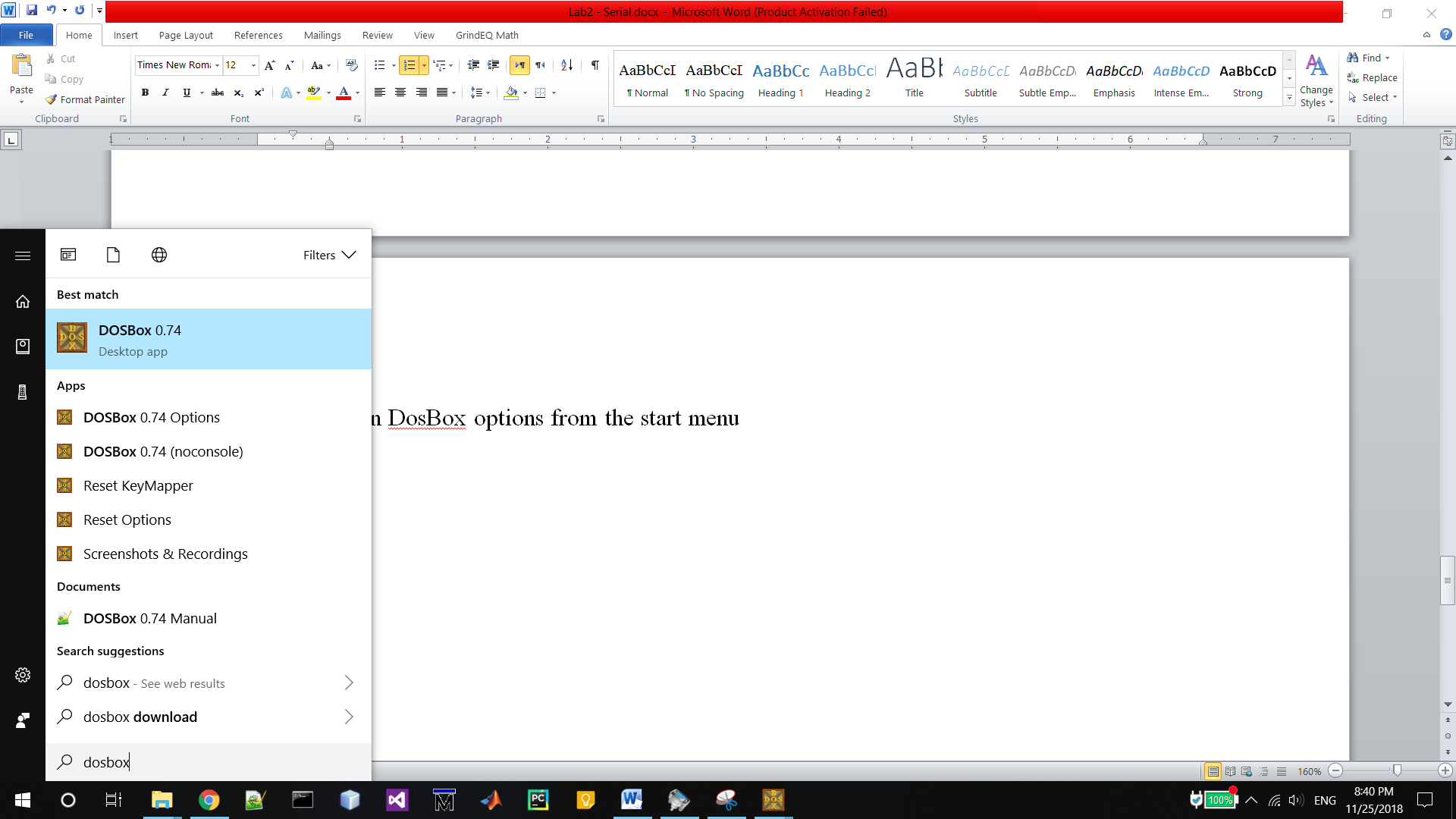
2 codes, 1 for send, 1 for receive

* 1. Integrate the send/receive into a single code that checks for serial input and prints it, then checks for KB input and sends it.

**Using Real Serial Port:**

1. Open DosBox Options from the start menu

Or open the configuration file in “C:\Users\\_\_\_\_\_\_\_\AppData\Local\DOSBox\dosbox-0.74.conf”



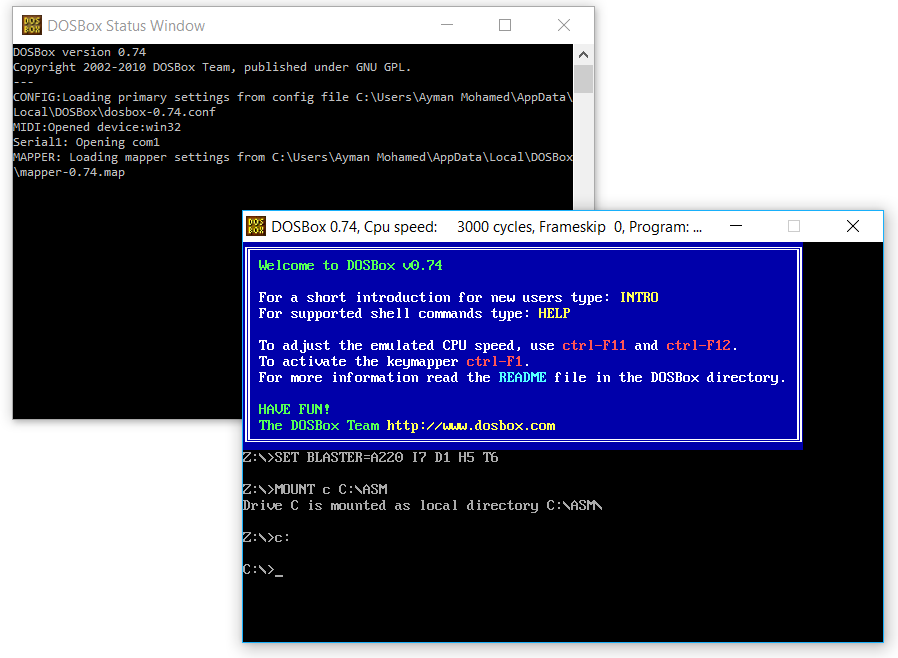
1. Then scroll up to the “[serial]” in the file



1. Modify the “serial1” line

serial1 = directserial realport:COM1

1. Now Open DosBox



**Running Virtual Serial Ports:**

1. Download and install DosBox & Eltima Virtual Serial Port
   1. Download DosBox from:

<https://www.dosbox.com/download.php?main=1>

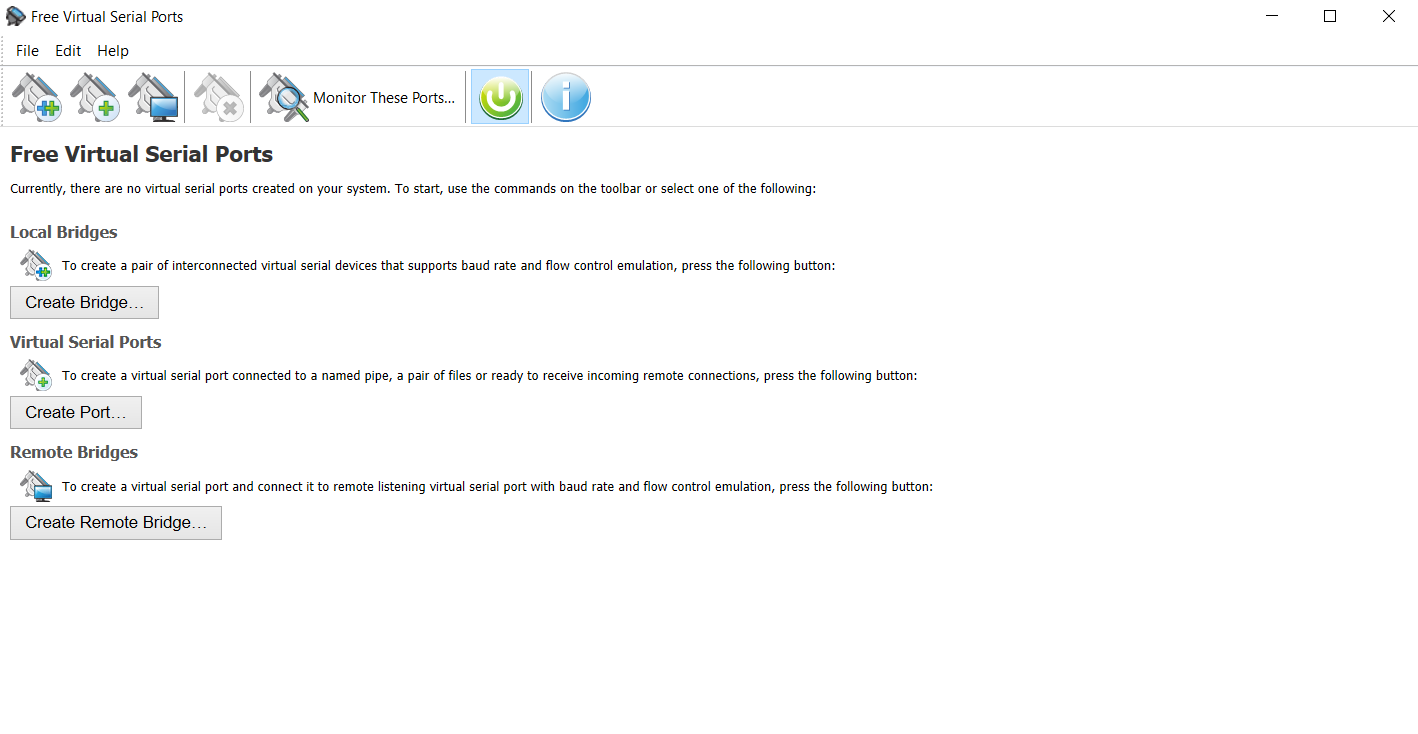
* 1. Download Virtual Serial Port from:

<https://www.eltima.com/virtual-com-port/>

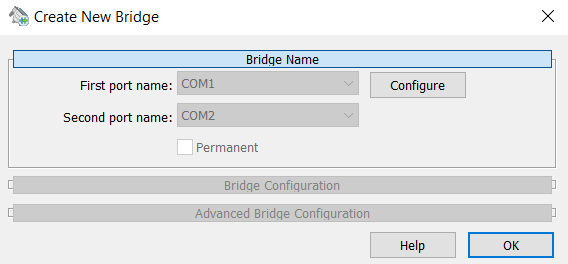
* 1. You could use ProGammaX, that is a program that configure and run DosBox

<https://progammax.webs.com/data/pages/en/download.html>

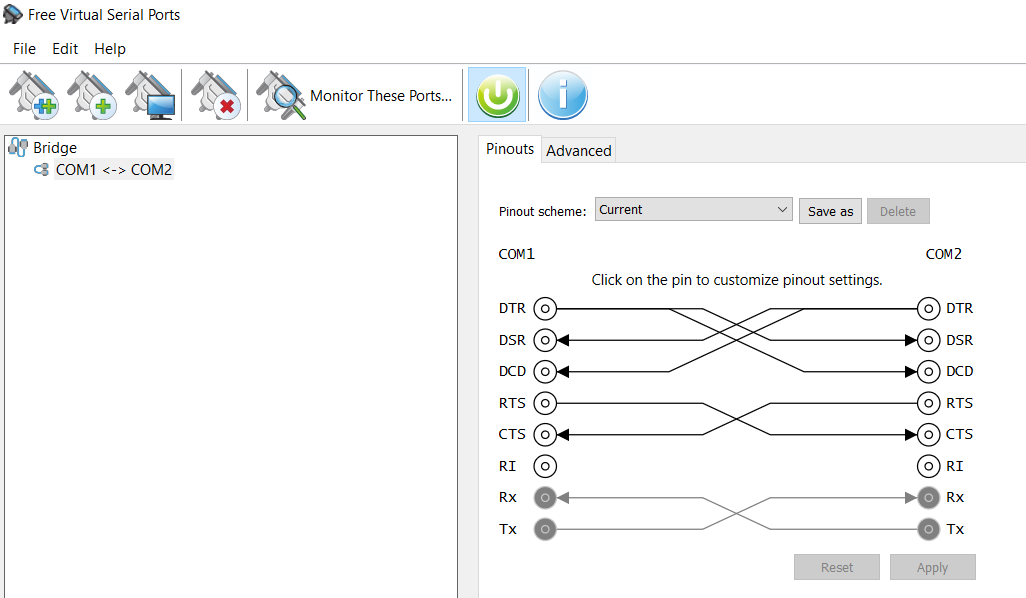
1. Open Virtual Serial Port program



1. Create a new virtual port bridge (click on  icon)



1. The newly created ports will be connected to each other



1. Don’t close the virtual serial port program
2. Open DosBox Options from the start menu

Or open the configuration file in “C:\Users\\_\_\_\_\_\_\_\AppData\Local\DOSBox\dosbox-0.74.conf”

1. Then scroll up to the “[serial]” in the file
2. Modify the “serial1” line

serial1 = directserial realport:COM1

1. To connect 2 DosBox, modify the DosBox options again

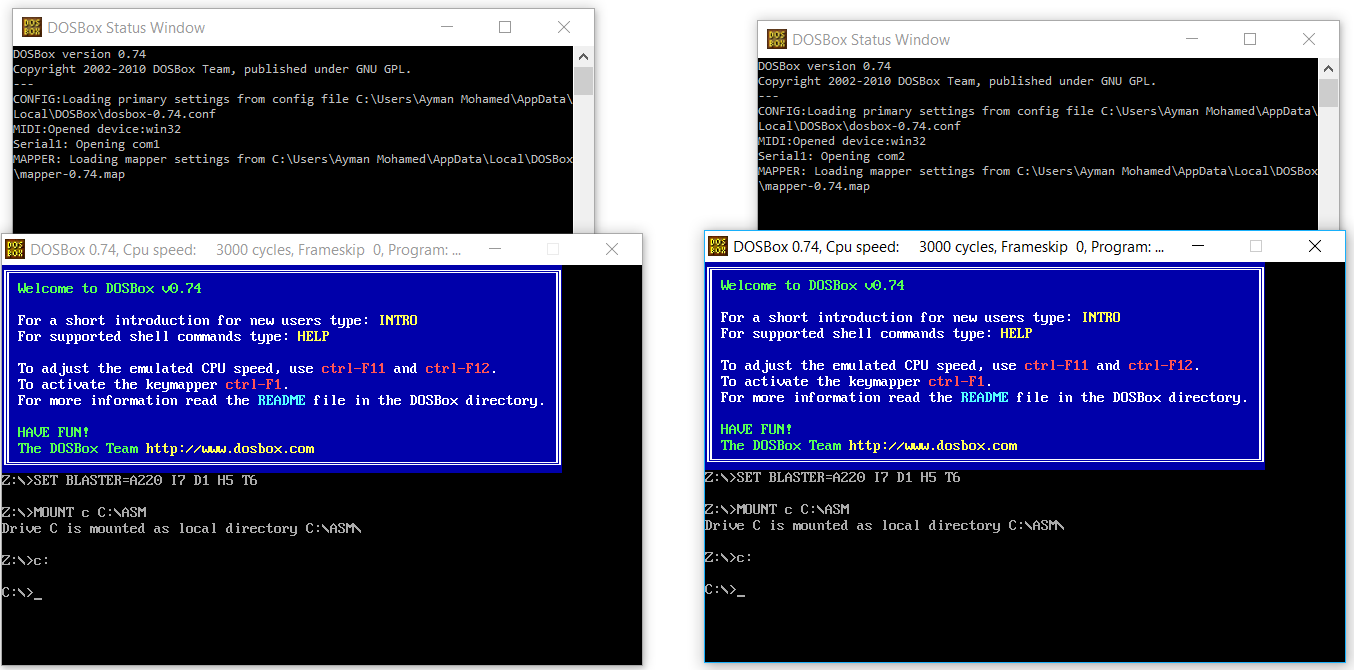
This time choose COM2 instead of COM1

serial1 = directserial realport:COM2

1. Then open a new DosBox

Now the 1st DosBox is connected to the 2nd DosBox through the virtual ports

(COM1<=>COM2)



1. Run your Chat program on both DosBox

**NB**:

1. Create a folder “ASM” on drive “C”
2. Open DosBox Options from the start menu

Or open the configuration file in “C:\Users\\_\_\_\_\_\_\_\AppData\Local\DOSBox\dosbox-0.74.conf”

1. Add the following lines at the end of the file

MOUNT C C:\ASM

C:

1. The DosBox is now open and can see all files in the “C:\ASM” directory