

# UTILIZAREA COMUNICATIEI SERIALE ÎN IBM – PC

## 1. Memento teoretic

Standardul RS 232C este cel mai folosit standard de cuplare seriala a doua echipamente de calcul. Cuplele de legatura seriala prezente în PC sunt cu 9, respectiv 25 de pini. Semnalele prezente în cupla cu 9 pini sunt:

pin 1: DCD (Data Carrier Detect) - intrare, indica detectarea de catre modem a purtatoarei.

pin 2: RD - intrare, receptie date.

pin 3: TD - iesire, transmisie date.

pin 4: DTR (Data Terminal Ready) - iesire, calculator pregatit pentru receptie.

pin 5: GND - masa.

pin 6: DSR (Data Send Ready) -intrare, modem cuplat la linia telefonica.

pin 7: RTS (Request To Send) - iesire, cerere de emisie din partea calculatorului.

pin 8: CTS (Clear To Send) -intrare, gata de emisie, raspuns al modemului la activarea RTS.

pin 9: RI (Ring Indicator) - intrare, indica detectia de catre modem a unei frecvente de apel.

Din punct de vedere electric la transmisia seriala prin acest standard se opereaza cu urmatoarele nivele de tensiune:

	0 logic	1 logic
emisie	-15 ÷ -5	5 ÷ 15
receptie	-15 ÷ -3	3 ÷ 15

Lungimea maxima a liniei de comunicatie pentru care schimbul de informatie este corect este de 15 m.

În PC comunicatia seriala se poate face prin intermediul a patru porturi COM. Zona de date BIOS contine la adresele 0:0400h..0:0407h o lista a celor patru adrese de baza corespunzatoare celor patru porturi.

În cursul rutinei de initializare (POST), se testeaza si se initializeaza porturile COM1 si COM2. Adresele de port corespunzatoare acestora sunt:

COM1 - 3F8h..3FFh

COM2 - 2F8h..2FFh

Asupra celor patru porturi se poate opera direct, prin instructiuni de intrare/iesire (IN, OUT), sau prin intermediul întreruperii BIOS 14h. Aceasta întrerupere va lucra cu oricare din cele patru porturi, cu conditia ca adresa de baza a portului serial solicitat sa se gaseasca în tabela din zona de date BIOS. Totodata, este necesar ca doua porturi sa nu împarta acelasi spatiu de adresare, caz în care nici unul nu va functiona corect.

Utilizarea porturilor seriale se poate face prin interogare (*polling*), sau prin întreruperi. În acest din urma caz liniile de întrerupere folosite, corespunzatoare controlorului de întreruperi 8259A sunt:

COM1 - IRQ4 - INT 0Ch

COM2 - IRQ3 - INT 0Bh

Registri asociati unui cuplor de comunicatie seriala sunt (adresele sunt date pentru COM1):

Adresa

3F8h *Scriere*: registru de date, continând cei 8 biti ai caracterului ce trebuie transmis. Când bitul DLAB=1, contine octetul inferior al divizorului frecventei

ceasului, care împreună cu octetul superior, scris la adresa 3F9h determină rata transmisiei seriale după cum urmează:

Rata (baud)	Valoare de divizare (zecimal)
110	1040
150	768
300	384
600	192
1200	96
2400	48
4800	24
9600	12

*Citire:* buffer de recepție, conținând cei 8 biți ai caracterului recepționat.

3F9h *Scrisoare:* Când bitul DLAB=1 conține octetul superior al valorii de divizare a frecvenței ceasului. Când DLAB=0 reprezintă registrul de validare întreruperi:

Bit: 0 - 1=validează generarea unei întreruperi la recepție de date.

1 - 1=validează generarea unei întreruperi când bufferul de emisie este gol.

2 - 1=validează generarea unei întreruperi la detectarea unei erori sau oprire.

3 - 1=validează generarea unei întreruperi la schimbarea stării modem-ului (CTS, DSR, RI, RLSD).

4..7 - au valoarea 0.

3FAh *Citire:* registrul cauză a întreruperii, La apariția unei întreruperi citirea acestui registru determină natura întreruperii.

Bit: 0 - 1=întrerupere activă (poate fi folosit pentru interogare).

- 1, 2 -00=întrerupere cauzata de o eroare (suprascrisiere, paritate, încadrare), sau oprire. Este resetat prin citirea registrului de stare a liniei (3fdh).  
 -01=date receptionate disponibile. Este resetat prin citirea bufferului de receptie (3F8h).  
 -10=buffer de emisie gol. Este resetat prin scrierea în bufferul de emisie (3F8h).  
 -11=întrerupere cauzata de schimbarea starii modem-ului. Este resetat prin citirea registrului de stare a modem-ului (3FEh).  
 3..7=sunt 0.

3FBh *Citire/scrisiere*: registrul de control al liniei.

- Bit: 0, 1: lungimea cuvântului de date: 00=5, 01=6, 10=7, 11=8.  
 2: numarul de biti de stop: 0=1, 1=2.  
 3,4: paritate: x0=fara, 01=impara, 11=para.  
 5: nu este folosit de catre BIOS.  
 6: valideaza controlul opririi. 1=transmisia începe prin emiterea de 0-uri (spatii).  
 7: DLAB (Divisor Latch Access Bit). 1=se programeaza rata de transmisie, 0=normal.

3FCh *Scriere*: registrul de control al modem-ului.

- Bit: 0: 1=activeaza DTR, 0=deactiveaza DTR.  
 1: 1=activeaza RTS, 0=deactiveaza RTS.  
 2: 1=activeaza OUT1  
 3: 1=activeaza OUT2 (necesar în cazul lucrului prin întreruperi).  
 4: 1=activeaza bucla pentru testare.  
 5..7: sunt 0.

3FDh *Citire*: registrul de stare a liniei. Bitii 1..4 cauzeaza generarea unei întreruperi daca aceasta este validata.

- Bit:
- 0: 1=date receptionate. Este resetat prin citirea bufferului de receptie.
  - 1: 1=eroare de suprascriere, caracterul precedent fiind pierdut.
  - 2: 1=eroare de paritate. Este resetat prin citirea registrului de stare a liniei.
  - 3: 1=eroare de încadrare; caracterul receptionat contine un bit de stop invalid.
  - 4: 1=este indicata oprirea; se receptioneaza spatii.
  - 5: 1=bufferul de emisie gol, este ceruta emisia urmatorului caracter.
  - 6: 1=transmitatorul este inactiv, nici o data nefiind procesata.

3FEh *Citire*: registrul de stare al modem-ului. Bitii 0..3 determina generarea unei întreruperi daca aceasta este validata.

- Bit:
- 0: 1=delta CTS si-a schimbat starea.
  - 1: 1=delta DSR si-a schimbat starea.
  - 2: 1=se detecteaza fronturi ale semnalului RI.
  - 3: 1=delta DCD si-a schimbat starea.
  - 4: 1=CTS este activ.
  - 5: 1=DSR este activ.
  - 6: 1=RI este activ.
  - 7: 1=DCD este activ.

Întreruperea 14h pune la dispozitie urmatoarele servicii pentru interfata seriala:

AH=00h - initializarea portului serial.

La apel:

DX=numarul portului (0,1).

AL=parametri de initializare, conform urmatoarelor câmpuri:

Bit: 0,1: lungimea cuvântului de date (10=7 biti, 11=8 biti).

2: numarul bitilor de stop (0=1, 1=2).

3,4: paritatea (x0=fara, 01=impara, 11=para).

5..7: rata de transmisie: 000=110, 001=150, 010=300, 011=600,  
100=1200, 101=2400, 110=4800, 111=9600.

La revenire:

AH=starea liniei seriale (vezi mai jos).

AH=01h - emisie caracter.

La apel:

DX=numarul portului (0,1).

AL=caracterul ce trebuie emis.

La revenire:

AL=caracterul emis.

Daca bitul 7 al lui AH este setat a intervenit o eroare si ceilalti 7 biti ai lui AH contin starea liniei de comunicatie (vezi mai jos).

AH=02h - receptie caracter.

La apel:

DX=numarul portului (0,1).

La revenire:

AL=caracterul emis.

AH este diferit de zero daca a aparut o eroare.

AH=03h - citirea starii portului serial.

La apel:

DX=numarul portului (0,1).

La revenire:

AX=starea portului serial

AH - starea liniei de comunicatie

bit 7: timeout.

bit 6: registrul de deplasare la transmisie gol.

bit 5: buffer de transmisie gol.

bit 4: oprire detectata.

bit 3: eroare de încadrare.

bit 2: eroare de paritate.

bit 1: eroare de suprascriere.

bit 0: date receptionate.

AL - starea modem-ului

bit 7: DCD

bit 6: RI

bit 5: DSR

bit 4: CTS

bit 3: delta DCD.

bit 2: fronturi RI.

bit 1: delta DSR.

bit 0: delta CTS.

## **2. Programe demonstrative.**

*Exemplu 1:* Utilizarea interfetei seriale prin polling.

.model small

;tabela adreselor de port ale registrelor interfetei seriale

```
COM2_DATE          equ  02F8h
COM2_VAL_INTR      equ  02F9h
COM2_TIP_INTR      equ  02FAh
COM2_FORM_DATE     equ  02FBh
COM2_CTRL_MODEM    equ  02FCh
COM2_STARE         equ  02FDh
COM2_VITEZA        equ  02F8h
```

.stack 100h

.data

.code

start:

;initializarea interfetei seriale

```
mov ax,@data
mov ds,ax
mov dx,COM2_FORM_DATE
mov al,80h                ;urmeaza programarea ratei de transmisie
out dx,al
mov dx,COM2_VITEZA        ;programarea octetului mai putin semnificativ
mov al,60h                ;9600 baud
out dx,al
xor al,al                 ;programarea octetului mai semnificativ
inc dx
out dx,al
mov al,03h                ;setarea formatului datelor:
mov dx,COM2_FORM_DATE     ;8 biti, 1 bit stop, fara paritate
```



```

    out dx,al
    mov al,0Bh                ;activare DTR si RTS
    mov dx,COM2_CTRL_MODEM
    out dx,al
    xor al,al                 ;invalidare întreruperi
    mov dx,COM2_VAL_INTR
    out dx,al
    mov dx,COM2_STARE         ;verificare daca exista un caracter
    in al,dx                  ;în bufferul de receptie
    and al,01h
    jz bucla
    mov dx,COM2_DATE          ;daca da, este citit
    in al,dx

bucla: mov ah,0               ;asteptarea unei taste
    int 16h
    cmp al,1Bh                ;daca este ESC
    je iesire                  ;se face saltul la iesire
    push ax
    call print                 ;afisarea caracterului citit
    mov dx,COM2_STARE         ;asteptare buffer emisie gol
et1:  in al,dx
    and al,20h
    jz et1
    pop ax
    mov dx,COM2_DATE          ;emitere caracter citit
    out dx,al

```

```

        mov dx,COM2_STARE           ;asteptare buffer receptie plin
et2:    in al,dx
        and al,01h
        jz et2
        mov dx,COM2_DATE           ;citire caracter receptionat
        in al,dx
        call print                  ;afisare caracter receptionat
        jmp bucla
iesire: mov ax,4C00h
        int 21h

print:  push ax                     ;afisare caracter
        push dx
        mov dl,al
        mov ah,02h
        int 21h
        pop dx
        pop ax
        ret

end start

```

*Exemplu 2:* Utilizarea interfetei seriale prin întreruperi cu monitorizare pe display.

```

.model tiny

;tabela cu deplasamentele registrelor fata de baza
rxdat    equ    0    ;registru date receptie

```

```

txdat      equ    0      ;registru date emisie
lowsp      equ    0      ;registru divizor viteza low
higsp      equ    1      ;registru divizor viteza high
itenb      equ    1      ;registru validare întreruperi
intid      equ    2      ;registru identificare întreruperi
lcont      equ    3      ;registru control linie
mcont      equ    4      ;registru control modem
txint      equ    2      ;masca indicator întrerupere emisie

```

```

;lungime barete de monitorizare a functionarii pe display

```

```

lbe        equ    1ceh   ;bareta solicitari emisie
lbr        equ    30eh   ;bareta solicitari receptie

```

```

;masti

```

```

iniv4      equ    0efh   ;masca nivelului de întrerupere 4

```

```

cseg  segment

```

```

    org 100h

```

```

    assume cs:cseg, ds:cseg, es:cseg,ss:cseg

```

```

exmp  proc far

```

```

    mov ax,40h

```

```

    mov es,ax

```

```

    mov dx,es:[0]      ;adresa pentru COM1 (0:0400)

```

```

    mov comx,dx        ;se memoreaza în segmentul program

```

```

;initializare UART si întreruperi

```

```

    mov ax,lcont

```

```

    mov bl,10000000b ;DLAB=1

```

```

call putb          ;trimite octetul din bl în registrul dat de ax
mov ax,higsp
mov bl,3           ;3 în divizor high
call putb
mov ax, lowsp
xorbx,bx          ;0 în divizor low
call putb          ;viteza va fi 150 baud
mov ax,lcont
mov bl,1           ;8 biti, fara paritate, 1 bit stop
call putb
mov ax,mcont
mov bl,0bh         ;activeaza DTR,RTS
call putb

```

;pregatirea vectorului de întrerupere

```

mov dx,offset ioint ;deplasament rutina în segment
mov ax,250ch        ;functia DOS 25h, vectorul 0c (COM1)
int 21h             ;apel DOS

```

;activarea înrruperilor în interfata

```

mov ax,itenb
mov bl,3           ;validare înrruperi emisie si receptie
call putb
in al,21h          ;citeste registrul de mascare al 8259A
and al,iniv4
nop
nop

```

```

nop
nop
out 21h,al      ;se demascheaza nivelul 4 de intreruperi

```

;din acest moment UC si-a terminat sarcina de initializare a interfetei. În continuare ;controlul UC este predat sistemului de operare care de regula alocă UC altor ;programe. Prin sistemul de întreruperi însă, de câte ori interfata termina de emis sau ;de receptionat un caracter, UC este întrerupta si obligata sa execute rutina de ;servire a interfetei seriale, numita aici IOINT. Durata de executie a rutinei este mult ;inferioara intervalului la care este ea solicitata. În restul timpului UC este la ;dispozitia altor utilizatori. Controlul este dat aici sistemului de operare cu optiunea ;ca zona de memorie în care se gaseste acest program, deci si IONIT sa nu mai fie ;alocata altor programe, astfel ca rutina de servire sa ramâna nealterata.

```

lea dx,rezrm    ;limita rezervarii de memorie
int 27h         ;revenire în DOS cu ramânere rezidenta

```

;urmeaza rutina de servire a întreruperii propriu-zisa. Spre deosebire de partea ;anterioara, rutina se executa la fiecare solicitare de asistenta a UC din partea ;interfetei seriale.

;

```

ioint: sti      ;validarea întreruperilor
push ds        ;salvarea starii curente
push es
push di
push ax
push bx

```

```

    push dx
    mov ax,cs
    mov ds,ax           ;initializarea segmentului de date
reia:  mov ax,intid
       mov bl,txint      ;identificarea sursei întreruperii (emisie sau receptie)
       call gets
       jz recep

```

;întrerupere la emisie

```

    mov ax,txdat
    mov bl,txb
    call putb           ;emite octetul din txb
    inc txb             ;trece la codul urmator

    mov ax,lbe          ;secventa de monitorizare pe display
    les di,spote
    call flash
    mov word ptr spote,di ;actualizarea pozitiei pe display pentru emisie
    jmp alte            ;daca între timp s-a terminat o receptie

```

;întrerupere la receptie

```

recep: mov ax,rxdat
       call getb        ;citeste octetul receptionat
       mov ah,rxb       ;octetul referinta
       inc rxb          ;trece la codul urmator
       cmp ah,al        ;compara receptia cu referinta
       nop              ;eroarea nu este tratata în acest exemplu

```

```

    mov ax,lbr          ;secventa de monitorizare pe display
    les di,spotr
    call flash
    mov word ptr spotr,di    ;actualizarea pozitiei pe display pentru receptie

alte:  mov ax,intid
       mov bl,1           ;se verifica daca mai sunt solicitari active (când emisia si
                           ;receptia apar simultan, sau în timpul tratarii uneia survine
                           ;cealata).

       call gets
       jz reia            ;se reia secventa de identificare si tratare

       mov al,20h         ;EOI pentr 8259A
       out 20h,al
       pop dx             ;restaurari registre
       pop bx
       pop ax
       pop di
       pop es
       pop ds
       iret               ;revenirea în programul întrerupt

;subrutine si rezervari de memorie

gets:  mov dx,comx        ;testeaza starea bitilor mascati în bl din registrul indicat de
                           ;ax

       add dx,ax
       in al,dx
       test al,bl

```

retn

```
putb: mov dx, comx      ;trimite octetul din bl în registrul indicat de ax
      add dx,ax
      mov al,bl
      out dx,al
      retn
```

```
getb: mov dx,comx      ;aduce în al octetul din registrul indicat de ax
      add dx,ax
      in al,dx
      retn
```

```
flash: mov byte ptr es:[di],20h
       cmp di,ax
       jne et1
       sub di,40h
et1:   inc di
       inc di
       mov byte ptr es:[di],0dbh
       retn
```

exmp endp

;rezervari si definitii

```
comx      dw    0      ;adresa interfetei COM1
txb        db    0      ;coduri emise
rxb        db    0      ;coduri receptionate
spote     dd    0b0000190h
```



```

spotr      dd    0b00002d0h
rezrm      equ   $
cseg ends
end    exmp

```

O problema obisnuita existenta în transmisiile seriale este aceea a protocolului de dialog între emitator si receptor. La emitator/receptor exista în mod obisnuit câte un buffer de dimensiune finita pentru datele ce urmeaza a fi emise/receptionate. Când acest buffer este plin calculatorul ignora datele noi pâna ce bufferul se goleste suficient. Pentru evitarea fenomenului mai sus amintit se implementeaza protocoale hard sau soft, prin intermediul carora emitatorul si receptorul își semnaleaza unul altuia umplerea bufferului propriu. Protocolul XON/XOFF este un protocol soft care functioneaza pe urmatorul principiu: când bufferul de receptie este plin, receptorul trimite un caracter XOFF (în zecimal 19) pentru a comunica emitatorului sa opreasca emisia datelor. Când bufferul de receptie devine suficient de gol, receptorul trimite un caracter XON (în zecimal 17) pentru a indica faptul ca transmisia datelor poate reîncepe. Când este folosit protocolul XON/XOFF codurile XON si XOFF vor fi folosite întotdeauna ca si coduri de control si nu ca date, deci acest protocol nu este recomandat în cazul transmisiilor binare.

*Exemplu 3:* Program de folosire a interfetei seriale prin întreruperi cu implementarea protocolului XON/XOFF.

```

.model small

COM1 equ 03F8h

.code

```

start:

jmp begin

caracter db 30h

handleCOM1 PROC FAR ;rutina de tratare int 0Ch

push ax ;salvarea registrilor folositi în rutina

push dx

push ds

push bx

mov ax,COM1+2

in al,dx

and al,00000111b

cmp al,03h ;este intrerupere la receptie?

jnz alfa ;nu - salt la alfa

mov dx,COM1

in al,dx

cmp al,13h ;caracterul XOFF?

jz beta

mov dx,COM1+1 ;tratarea caracterului XON

mov al,03h

out dx,al ;validare întreruperi emisie/receptie

jmp alfa

beta: mov dx,COM1+1 ;tratarea caracterului XOFF

mov al,01h ;invalidare întrerupere emisie

out dx,al

jmp exit1

```

alfa:  mov ax,0B000h          ;adresa de inceput a memoriei video
        mov ds,ax
        mov dl,cs:caracter    ;încarcarea caracterului curent
        mov bl,dl
        xor bl,3Ah            ;trece de '9'?
        jn et1
        mov dl,30h            ;caracterul '0'
        mov cs:caracter,dl
et1:    mov ds:[20],dl        ;scriere caracter curent
        inc cs:caracter
        mov dx,COM1           ;emitere caracter 'A'
        mov al,'A'
        out dx,al

exit1:  mov al,20h            ;EOI pentru 8259A
        out 20h,al
        pop bx                ;refacere stare registri
        pop ds
        pop dx
        pop ax
        iret

handleCOM1    ENDP

sfirsit      db 0

;rutina de initializare port serial
initCOM1     PROC NEAR

```

```

        mov ah,00h                ;setarea parametrilor de comunicatie folosind serviciul
BIOS
        mov al,67h                ;600 baud, 8 biti, 2 biti stop
        mov dx,0                  ;fara paritate
        int 14h
        mov ax,250Ch
        mov dx,offset handleCOM1
        int 21h

        mov dx,COM1+4             ;activare DTR,RTS
        mov al,0Bh
        out dx,al
        mov dx,COM1+1
        mov al,3
        out dx,al                 ;validare intreruperi emisie/receptie

        in al,21h                 ;activare IRQ4 în 8259A
        and al,11101111b
        out 21,al
        ret

initCOM1    ENDP

begin:
        mov ax,@CODE
        mov ds,ax
        call initCOM1
        mov dx,offset sfirsit
        mov cl,4                  ;calcularea lungimii codului rezident în paragrafe

```

```
shr dx,cl
inc dx                ;rotunjire superioara
mov ax,3100h
int 21h              ;ramânere rezidenta
END start
```

### **3. Probleme propuse.**

1. Sa se modifice exemplul 2 de folosire a interfetei seriale prin întreruperi pentru emiterea seriala a unor caractere introduse de la tastatura si afisarea lor în rutina de tratare a întreruperii de receptie. În acest scop se va folosi o cupla cu bucla locala.

2. Sa se modifice exemplul 1 pentru citirea si interpretarea erorilor ce pot aparea la transmisia seriala.

3. Sa se testeze exemplul 3 de folosire a protocolului XON/XOFF pentru comunicarea cu o consola programabila conectata serial la PC.