

Porte i mikrodatamater II

Ved Helge Jensen

Praktisk eksempel på anvendelse af porte

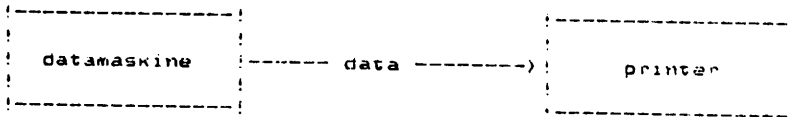
Portbegrebet dækker over mere end 'det sted hvor data bringes til / fra den ydre verden'. Portens anvendes også som hjælpefunktioner for at sikre datatransport. Det indebærer på mange måder fordele, at kombinere flere porte om e'n fælles opgave.

I det følgende gennemgås et eksempel på hvordan en port anvendes som parallel printerudgang og hvilken software der kræves for at kommunikere med en printer.

For at lette beskrivelsen vil eksemplet være baseret på parallel-stikket, som er placeret på bagsiden af en MPS-3000. Stikket er et 25 polet Canon-stik, hvis ben er forbundet som et 'Centronic-snit'. Denne type forbindelse anvendes af et stort antal printere og andre ydre anheder, som skal kommunikere med en datamaskine.

Hvis du ser i hardware brugermanualen for den pågældende maskine, kan du finde adresserne for de benyttede porte, normalt adresse 240 og 241.

Problemet kan deles i to afsnit datamaskinen og printeren, hvor datamaskinen skal afgive og printeren modtage informationerne.



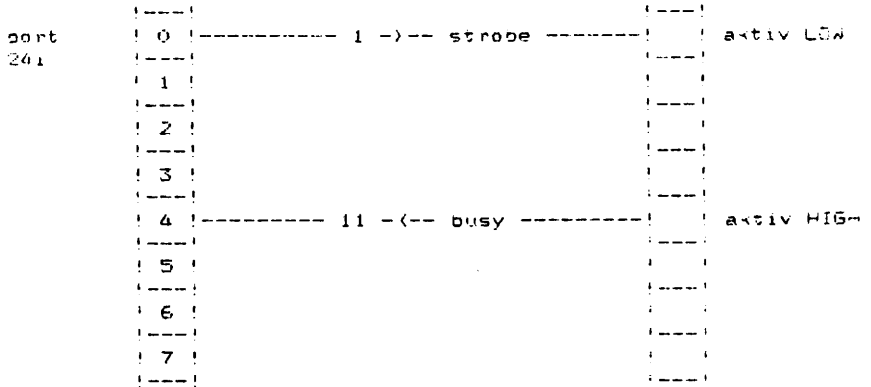
Datamaskinen skal anvende en port som udgang og printer en port som indgang, hvor data kan transporteres igennem. Idet data fylder 8 bit kræver det 'en hel port som "landevej for data", der i eksemplet har adresse 240 og er initieret som udgang.

Problemet er nu, hvad der skal sikre og indikere at data er kommet korrekt frem til printer, idet arbejdshastigheden for datamaskinen og printer er forskellige. Der findes her et signal fra datamaskinen som fortæller hvornår data er parat til overføring og kan låses ind i printer. Dette signal kaldes for 'strobe', det er en tidsmæssig indikation for valide data.

Et andet vigtigt problem er om printer er optaget af andet arbejde og overhovedet 'lytter til data', til at varetage dette findes der et signal fra printer som benævnes 'busy'. Det er nødvendigt fordi der er mange andre ting end 'lytte efter data' printer kan foretage sig, bla. betjene papirfremføring, vognretur, printe osv.

Model af opstilling:

datamaskine		ben nr.		printer
	-----*	på stik		*-----
port	0	2 ->		
240	1	3 ->		
	2	4 ->		
	3	5 ->		
	4	6 ->		
	5	7 ->		
	6	8 ->		
	7	9 ->		



Et helt forløb for overføring af data ser ud som følger:

- datamaskine ser efter busy
- data placeres i port 240
- strobe pulseres (går fra 1 til 0 tilbage til 1)

Data er nu placeret i printerens buffer og kan her overtages af printerens interne behandling, hvis det er en karakter af ASCII koden, vil den først fremkomme på papiret, når der sendes et lineskift eller en vognretur.

Et eksempel på software der printer 'et 'd' og derefter skifter linie. Programmet er skrevet i Comal-80.

```
0010 DIM B$ OF 8
0015 //
0020 DATAREGISTER:=240
0040 STATUSREGISTER:=241
0055 //
0085 BITNUMMER:=4
0090 STROBE_ON:=255
0100 STROBE_OFF:=254
0110 CR:=13
0111 LF:=10
0112 data:=ord("d")
0120 ////////////////////////////////////////////////////////////////////
0130 // procedureer til at se om printeren er optaget //
0140 // samt give besked om at data er valide //
0145 ////////////////////////////////////////////////////////////////////
0146 //

* Her skal portene initieres hvis der benyttes en anden
* end den benyttede PARALLEL-port.

0150 PROC BUSY
0160 REPEAT
0170 B$:=BSTR$(INP(STATUSREGISTER))
0180 UNTIL IVAL(B$(8-bitnummer))
0190 ENDPROC BUSY
0200 //
0210 PROC STROBE
0220 OUT STATUSREGISTER, STROBE_OFF
0230 OUT STATUSREGISTER, STROBE_ON
0240 ENDPROC STROBE
0245 ////////////////////////////////////////////////////////////////////
0250 //***** BRINGER EN KARAETER TIL PRINTER *****
0280 EXEC BUSY
0310 OUT DATAREGISTER, data
0320 EXEC STROBE
0325 //***** GIVER ORDER OM VOGNRETUR *****
0330 EXEC BUSY
0351 OUT DATAREGISTER, CR
0352 EXEC STROBE
0353 //***** GIVER ORDR OM LINESKIFT *****
0354 EXEC BUSY
0355 OUT DATAREGISTER, LF
0356 EXEC STROBE
```

Samme eksempel skrevet i COMPAS-PASCAL.

```
PROGRAM printer_eks;
```

```
CONST
```

```
  dataregister   = 240;  
  statusregister = 241;
```

```
  strobe_on      = 255;  
  strobe_off     = 254;
```

```
  cr             = 13;  
  lf             = 10;  
  data          = 'd';
```

```
PROCEDURE busy;
```

```
BEGIN
```

```
  WHILE NOT(PORT(.statusregister.) AND 16 = 16 ) DO;
```

```
END;
```

```
PROCEDURE strobe;
```

```
BEGIN
```

```
  PORT(.statusregister.) := strobe_off;  
  PORT(.statusregister.) := strobe_on;
```

```
END;
```

```
BEGIN
```

- * Her skal portene initieres hvis der benyttes en anden
- * end den benyttede PARALLEL-port.

```
  busy;  
  PORT(.dataregister.) := ord(data);  
  strobe;
```

```
  busy;  
  PORT(.dataregister.) := cr;  
  strobe;
```

```
  busy;  
  PORT(.dataregister.) := lf;  
  strobe;
```

```
END.
```

Samme eksempel skrevet i ASSEMBLER-KODE:

```
                ORG      0100h
                ;.....
buffer EQU      240
contr EQU      241
                ;.....
busy EQU      0001$0000b
strooe EQU     1111$1110b
nvil EQU      1111$1111b
                ;.....
cr EQU        13
lf EQU        10
data EQU      'a'
                ;.....
```

***** bringer en karakter til printer *****

- * her skal portene initieres hvis der benyttes en anden
- * end den benyttede PARALLEL-port.

```
start:
        CALL     busyx
        MVI     a,data
        OUT     buffer
        CALL     stroox
;***** giver ordre om vognretur
        CALL     busyx
        MVI     a,cr
        OUT     buffer
        CALL     stroox
;***** giver ordre om lineskift
        CALL     busyx
        MVI     a,lf
        OUT     buffer
        CALL     stroox
;***** gå tilbage til CP/M operativsystem
        CALL     0000
        ;
;*****
; subrutiner til at se om printeren er optaget;
; samt give besked om data er valde
stroox:
        MVI     a,strooe
        OUT     contr
        MVI     a,nvil
        OUT     contr
        RET
        ;
busyx:
        IN      contr
        ANI     busy
        CPI     busy
        JNZ     busyx
        ;
        END
```

Det viste eksempel giver kun mulighed for at printe 'en karakter og går ud fra at portene er initieret, hvilket kun gælder så længe PARALLEL stikket benyttes. Hvis du ønsker en anden port til kommunikationsudgang har du initieringen af portfunktionerne. .

Det kræver ikke meget at initiere en port som udgang, der skal kun placeres decimalværdien 15 i kontrolregistret, hvorimod en port hvor der er blandede ind - og udgange kræves en ny metode for initiering.

For at kunne blande ind og udgange skal der først placeres decimalværdien 207 i kontrolregistret og derefter det ønskede bitmønster på samme adresse, hvor bitnærværdien 1 er udgang og 0 er indgang.

Lad os antage at bit nr. 7 og 0 i port adresse nr. 0 skal være udgang og resten indgange.

Eksempel skrevet i Comal-80:

```
0005 kontrolregister:=2
0010 bitmønster:=128+0+0+0+0+0+1
0020 OUT kontrolregister,207
0030 OUT kontrolregister,bitmønster
//
```

Opgave:

Fremstil en printerdriver (procedure), som kan printe en vilkårlig streg med max. 80 karakterer og skifte linie, men kun hvis der gives besked med et flag ved navn linie. Afprøv rutinen på den eksisterende printerudgang og omskriv programmet så en plotter med 'Centronic-snit' kan tilsluttes en ny port som ligger på adresse 0 og 1. Adresse 0 er 8 bit parallel udgang og adresse 1 er mix-ind/ud. Rutinerne som er beskrevet kan fås ved at sende formateret diskette til forfatteren eller DaTS.

Løsningseksempl på opgaverne i blad nr.1 :

(6)-7-6-5-4-3-2-1-0

32	=	0 0 1 0 0 0 0 0	=	32+0	=	32
22	=	0 0 0 1 0 1 1 0	=	16+4+2	=	22
35	=	0 0 1 0 0 0 1 1	=	32+2+1	=	35
156	=	1 0 0 1 1 1 0 0	=	128+16+8+4	=	156
255	=	1 1 1 1 1 1 1 1	=	128+64+32+16+8+4+2+1	=	255
256	=	(1) 0 0 0 0 0 0 0 0	=	256+0	=	256

Løsningen skrevet i COMAL-80 :

```
0010 DATAREGISTER:=0          // GRUNDADRESSE
0020 KONTROLREGISTER:=0+2    // GRUNDADRESSE + 2
0021 FUNKTION:=15           // UDGANGSPORT
0022 motor1 :=1
0023 motor2 :=2
0024 lampe1 :=32
0025 lampe2 :=64
0042 //*****
0045 // Programmet initierer PORT-A som udgang og
0046 // giver løsningen på opgaven
0047 //-----
0049 //
0050 OUT KONTROLREGISTER,FUNKTION
0060 OUT DATAREGISTER, INP(DATAREGISTER)+motor1+lampe1
0070 OUT DATAREGISTER, INP(DATAREGISTER)+motor2+lampe2
```


Samme løsning i assembler :

```
start: ORG 0100H
;
datareg EQU 0000H
kontreg EQU 0000H + 2
funk EQU 15
motor1 EQU 1
motor2 EQU 2
lampe1 EQU 32
lampe2 EQU 64
;
START: MVI A, funk      ;Initier port-A
      OUT kontreg      ;som udgang
;
      IN  datareg
      ORI motor1+lampe1
      OUT datareg
;
      IN  datareg
      ORI motor2+lampe2
      OUT datareg
;
      END
```

Samme løsning i COMPAS FASCAL :

```
PROGRAM port_eksempel;
const
  kontrolregister = 2;
  dataregister   = 0;
  funktion       = 15;
  motor1        = 1;
  motor2        = 2;
  lampe1        = 32;
  lampe2        = 64;

BEGIN
  PORT(.kontrolregister.):=funktion;
  PORT(.dataregister.):=PORT(.dataregister.)+mo-
tor1+lampe1;
  PORT(.dataregister.):=PORT(.dataregister.)+mo-
tor2+lampe2;
END.
```