

INTERNATIONAL COMPUTERS LIMITED A/S
KLAMPENBORGVEJ 232
2800 LYNGBY

Tlf.: 02- 88 94 88

REALTIDSUR
COMET MPS-25
BRUGERVEJLEDNING

COMET REALTIDSUR MPS-25

MPS-25 er et modul til COMET, som giver adgang til et realtidsur med følgende faciliteter:

- klokkeslet: timer - minutter - sekunder.
- dato: dag - måned - år.
- ugedag nummer.
- programmeret for skudår.
- valg mellem timer som 0 - 23 eller som 1 - 12 AM samt 1 - 12 PM.
- batteridrift, dvs. uret er i drift, også når strømmen til COMET'en er afbrudt.
- 2 kilo byte RAM lager, der ligeledes er batteridrevet.

1. PORTNUMRE.

Realtidsuret er at betragte som en ydre enhed med hvilken man kommunikerer via Z-80 processorens porte, der adresseres direkte fra brugerprogrammet.

Af de i alt 256 porte med numrene 0 - 255 er en del allerede reserveret, og ved valg af portnumre til MPS-25 eller andre COMET-moduler gør man klogt i at holde sig til portnumre mindre end 150. Har man i sin COMET flere MPS-moduler, må man endvidere være opmærksom på, at samme portnummer ikke anvendes i flere forbindelser.

I forbindelse med realtidsuret anvendes 5 porte med fortløbende nummerering, og før uret monteres i COMET'en skal disse portnumre fastlægges og kodes ind på selve modul kortet.

Ud over at være fortløbende skal de valgte portnumre også opfylde, at det laveste nummer er et helt multiplum af 8, altså 0, 8, 16 etc.

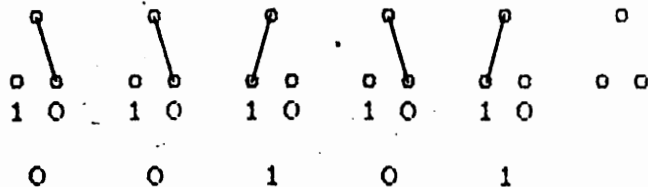
BEMÆRK: i det følgende antages, at man betragter MPS-25 modulet fra komponentsiden med kortets stik vendende mod højre.

Kodning af portnummeret sker nu på den blå sokkel med i alt 6 fjedre, som findes i modulets øverste højre hjørne. Af de 6 fjedre skal vi kun anvende 5, idet vi ser bort fra fjederen yderst til højre. Hver fjeder kan anbringes i 2 positioner:

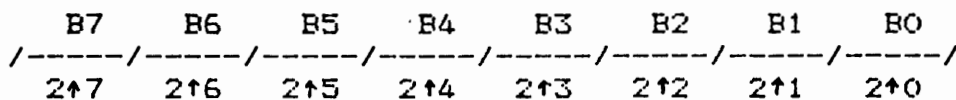
HØJRE position svarer til binært '0'
VENSTRE position svarer til binært '1'

(se figur 1 næste side).

Betragt nu det laveste af de 5 portnumre, der ønskes knyttet til realtidsuret, som et binært tal - se fig. 2 på næste side.



Figur 1.



Figur 2.

Som figur 1 og 2 antyder, skal de 5 aktive fjedre i den blå sokkel kodes således, at de repræsenterer de 5 mest betydende bit i portnummeret. De 3 mindst betydende bit i portnummeret er forud kodet på MPS-25 modulet til:

0 0 0 - 0 0 1 - 0 1 0 - 0 1 1 - 1 0 0

EKSEMPEL: kodes de 5 fjedre som vist på figur 1, bliver følgende portnumre knyttet til realtidsuret:

0 0 1 0 1 0 0 0 = 40 dec.
 0 0 1 0 1 0 0 1 = 41 dec.
 0 0 1 0 1 0 1 0 = 42 dec.
 0 0 1 0 1 0 1 1 = 43 dec.
 0 0 1 0 1 1 0 0 = 44 dec.

Stilles alle 5 fjedre i 0-positionen, bliver portene 0 til 4 knyttet til MPS-25.

2. MONTERING.

1. FJERN NETSTIKKET FRA STIKKONTAKTEN
2. Svøbet demonteres. 4 skruer, 2 stk. på hver side.
3. Monter MPS-25 i en ledig position i kortmagasinet.
4. Monter atter svøbet.

3. KOMMUNIKATION VIA PORTE.

Som allerede nævnt kommer man fra et program i COMET'en i forbindelse med realtidsuret ved at læse/skrive data via de 5 porte, der knyttes til modulet. De tre porte med lavest værdi (P0, P1 og P2) anvendes i forbindelse med modulets urfunktion, medens de to andre porte (P3 og P4) anvendes i forbindelse med modulets RAM-lager.

Hvorledes man i et program skriver data til eller læser data fra en port afhænger af det valgte programmeringssprog:

COMAL-80

Skrivning til en port sker med sætningen:

OUT (portnr.) , (data)

{data} skrives til porten med nummeret {portnr.}.
Såvel {data} som {portnr.} skal være decimale tal, der
ligger indenfor intervallet 0 til 255. {data} og
{portnr.} kan være konstanter, variabelnavne eller arit-
metiske udtryk.

EKSEMPEL: Ønsker man at skrive ASCII-koden for 'A' til
port n. 9:

```
OUT 9, 65      eller      OUT 9, ORD("A")
```

Læsning af data fra en port sker ved funktionen:

```
INP( {portnr.} )
```

hvor {portnr.} er underkastet de samme regler, som nævnt
ovenfor. Efter udførelsen har funktionen INP en decimal
værdi mellem 0 og 255 svarende til de data, der findes
på porten med nummeret {portnr.}.

EKSEMPEL: Ønsker man at læse data fra port 8:

```
PORT8:=INP(8)
```

eller ønsker man at udprinte disse data:

```
PRINT INP(8)
```

8080-ASSEMBLER

Skrivning af data til en port sker ved instruktionen:

```
OUT {portnr.}
```

hvor {portnr.} er som tidligere defineret.
Denne instruktion bevirker, at indholdet af A-registeret
skives til porten {portnr.}.

EKSEMPEL: Følgende svarer til skriveeksemplet vist under
COMAL-80:

```
MVI  A,'A'      ;placer 'A' i reg. A  
OUT  9          ;skriv til port 9
```

Læsning af data fra en port sker ved instruktionen:

```
IN {portnr.}
```

hvorved data på porten {portnr.} (som tidligere define-
ret) placeres i A-registeret.

EKSEMPEL: Vi læser to byte fra port nr. 8 og beregner
summen af disse i A-registeret:

```
IN    8          ;læs byte 1 fra port 8  
MOV   C,A       ;mellemlagres i C-reg.  
IN    8          ;læs byte 2 fra port 8  
ADD   C         ;adder i A-reg.
```

De fleste andre programmeringssprog (f.eks. PASCAL) til-

byder faciliteter svarende til COMAL-80 eller muligheden for at indflette assemblerkode.

4. URFUNKTIONEN.

Urfunktionen på MPS-25 omfatter 13 registre:

REG. NR.:	INDHOLD:	VERDI-INTERVAL:
0	ettere i sekunder	0..9
1	tiere i sekunder	0..5
2	ettere i minutter	0..9
3	tiere i minutter	0..5
4	ettere i timer	0..9
5	tiere i timer	0..1 el. 0..2 (se note 1)
6	ugedag nr.	0..6
7	ettere i dag	0..9
8	tiere i dg	0..3 (se note 2)
9	ettere i måned	0..9
10	tiere i måned	0..1
11	ettere i år	0..9
12	tiere i år	0..9

Tabel 1.

Note 1: Bit 3 i register 5 er: "1" for 24 timers format
"0" for 12 timers format
I forbinelse med 12 timers format angiver bit 2
i register 5: PM hvis "1", AM hvis "0".

bit 3 bit 2 bit 2 bit 0
/-----/-----/-----/-----/

Note 2: Bit 2 i register 8 angiver skuår, hvis "1".

Disse tretten registre stilles enkeltvis fra et program ved at skrive data til registrene, hvorefter registrene opdateres så at sige med tiden.

Klokkeslet, dato og ugedag indlæses til et program ved at aflæse registrene enkeltvis.

4.1. AFLÆSNING AF REALTIDSURET

sker i to tempi:

1. Først skriver man nummeret på det ønskede register til porten P0 (laveste portnummer for MPS-25)
2. Dernæst læses data fra pågældende register ved at læse fra port P1 (næstlaveste portnummer).

BEMÆRK: den betydende del af de data, der indlæses under 2 fylder kun en del af den indlæste byte. Ikke benyttede bit er "1", hvilket der må tages højde for ved den videre anvendelse af dataene.

Det drejer sig ganske enkelt om at subtrahere 192 fra de indlæste data. Herudover må man naturligvis også tage højde for de informationer, der findes i bit 2 og 3 af register 5, samt i bit 2 af register 8.

EKEMPEL: Det antages, at portnummereringen er P0 = 0, P1 = 1 og P2 = 2. Endvidere regnes med 24 timers format. Vi ønsker nu at indlæse timer:

COMAL-80:

```
OUT 0, 4           // læs ettere
TIMER01 := INP(1) - 192
OUT 0, 5           // læs tiere
TIMER10 := INP(1) - 198
TIMER := TIMER10 * 10 + TIMER01
```

ASSEMBLER:

```
MVI  A,4           ;læs ettere
OUT  0
IN   1
ANI  00001111B    ;afmask betydende del
MOV  L,A
MVI  A,5           ;læs tiere
OUT  0
IN   1
ANI  00000011B    ;afmask betydende del
MOV  H,A
```

4.2. STILLE REALTIDSURET.

Dette foregår ligesom aflæsningen i to omgange:

1. Først skrives nummeret for det register, der ønskes stillet til porten med laveste nummer.
2. Dernæst skrives data til registeret ved at skrive til porten P3 (tredie laveste nummer).

BEMÆRK: Det registernummer, der skrives under 1, skal være 240 større end de registernumre, der er anført i tabel 1.

BEMÆRK: Uanset hvad der skrives til registrene 0 og 1 (her adresseret ved hhv 240 og 241) vil begge disse registre blive nulstillet, blot der skrives til det ene.

BEMÆRK: Realtidsuret stopper, når der skrives til et af registrene 0 til 5. Uret genstarter, når der læses fra et af disse registre.

EKEMPEL: Lad os stille årstallet under de samme betingelser, som gjalt i eksemplet i afsnit 4.1.:

COMAL-80:

```
INPUT "ARSTAL: ":AR
OUT 0, 252 //skriv til reg. 12
OUT 2, AR DIV 10
OUT 0, 251 //skriv til reg. 11
OUT 2, AR MOD 10
```

ASSEMBLER: (det antages at ettere findes i
L-reg. og tiere i H-reg.)

```
MVI A, 252 ;tiere til reg. 12
OUT 0
MOV A, H
OUT 2
MVI A, 251 ;ettere til reg. 11
OUT 0
MOV A, L
OUT 2
```

Til yderligere information medfølger et COMAL-80 program som kan vise klokkeslet, ugedag (navn) og dato. Endvidere kan man via programmet stille på disse størrelser. Det anbefales, at man anvender dette program til at blive fortrolig med realtidsurets funktion, samt evt. studerer fremgangsmåden ved programmering af uret.

5. RAM LAGER.

De 2 kB RAM-lager, der findes på MPS-25 modulet kan anvendes til lagring af data helt efter eget ønske. Da lageret er batteridrevet, vil lagrede data blive bevaret selv om strømmen til COMET'en afbrydes. RAM-lageret er derfor velegnet til lagring af væsentlige data, der for eksempel indsamles under processtyring o.lg., men naturligvis kan lageret anvendes til mange andre formål.

BEMÆRK: kapaciteten af det anvendte batteri er stort nok til at såvel ur som lager vil få tilstrækkelig forsyning i op til 5 måneder efter at der sidst har været strøm på COMET'en. Hver gang der tændes for COMET'en genoplades batteriet til fuld kapacitet.

RAM-lageret kan betragtes som 2048 bytes, hvor der findes en pegepind, der udpeger "næste ledige". Hver gang man skriver en byte data til lageret, placeres denne byte i den af pegepinden (PP) udpegede byte, hvorefter PP tælles en op. Ved læsning fra lageret får man indholdet af den byte, der udpeges af PP, hvorefter PP tælles en op.

PP sættes til at pege på første byte i RAM-lageret, når der skrives et vilkårligt tal (0..255) til porten med det højeste nummer (P4).

Man læser data fra eller skriver data til RAM-lageret via porten med det næsthøjeste nummer (P3).

EKSEMPEL: En måde at anvende RAM-lageret på er ved at skrive til og læse fra bestemte bytes ved hjælp af følgende procedurer:

```
PROC SKRIVRAM(BYTENR, DATA)
//
// skriver en byte 'DATA' til byten 'BYTENR'
//
OUT 4, 255 // reset pegepind
// læs frem til BYTENR-1
FOR PP := 0 TO BYTENR-1
    DUMMY := INP(3)
NEXT PP
OUT 3, DATA
ENDPROC SKRIVRAM
```

```
PROC LÆSRAM(BYTENR, REF DATA)
//
// læser en byte 'DATA' fra byten 'BYTENR'
//
OUT 4, 255 // reset pegepind
// læs frem til 'BYTENR'
FOR PP := 0 TO BYTENR
    DATA := INP(3)
NEXT PP
ENDPROC LÆSRAM
```

RAM-lageret er på grund af den hurtige tilgang velegnet til f.eks. opbevaring af tabeller, mindre filer o.lg. Dg så behøver man ikke engang at indlæse disse data til lageret, hver gang man starter op.