

INTRODUKTION TIL MIKRODATAMATSYSTEMET SPC/1

Note ved

P. Pedersen.

INDHOLDSFORTEGNELSE:

1.	SYSTEMBESKRIVELSE	s. 2
2.	OUTPUT CONTROL	s. 3
3.	OPSTART	s. 4
4.	STOP	s. 4
5.	PASCAL SYSTEM	s. 5
5.1.	MIKADOS EDITOR	s. 5
5.1.1.	OVERSIGT OVER EDITORORDRER	s. 6
5.2.	PASCAL COMPILER OG INTERPRETER	s. 7
5.2.1.	START AF COMPILER	s. 8
5.2.2.	START AF INTERPRETER	s. 9
5.2.3.	APPENDIX A OVERSÆTTER FEJLMEDDELELSER	s. 10
5.2.4.	APPENDIX B FORTOLKER FEJLMEDDELELSER	s. 14
6.	YDRE ENHEDER	s. 16
6.1.	GRUNDLÆGGENDE I/O OPERATIONER I PASCAL	s. 16
6.2.	A/D KONVERTERINGSMODUL	s. 16
6.2.1.	ESSENS VEDRØRENDE A/D-KONVERTERING	s. 18
6.3.	D/A KONVERTERING SMODUL	s. 19
6.3.1.	ESSENS VEDRØRENDE D/A KONVERTERING	s. 20
6.4.	INTERVAL TIMER MODUL	s. 20
6.4.1.	ESSENS VEDRØRENDE INTERVAL TIMER	s. 22

Rettelser til "Introduktion til SPC-1":

S. 1, 3 : (a) : printer er OKI-printer

S 4 : (b) : Før et andet systemprogram skal startes op, fjernes MIKODOS-disketten og disketten med det ønskede systemprogram monteres i P1.

Et systemprogram kan nu startes ved at taste :

'ESC-tast' :programnavn'

hvorpå programmet indlæses fra disketten i P1 (hvis det findes)

S 5 : (c) En diskette med editor-programmet skal være placeret i P1.

Editor findes på to systemdiskette marker :

1. EDIT and EDITFILES

2. PASCAL SYSTEM DISC

S 5 : (d) Editor-programmets navn er "EDIT"

S 6 : (e) Pladens navn er "PASCAL SYSTEM DISC"

1. BESKRIVELSE AF SYSTEM SPC/1 - Sohngaardsholmsvej 57.

Systemet består af (se figur 1).

- 1. Minidiskettedrev og spc/1-mikrodatamat 320K, 64K - RAM.
- x 2. Modemomskifter.
- 3. Konsol.
- 4. Printer.
- 5. I/O kasse.

x 1.1. Minidiskettedrev og spc/1-mikrodatamat.

Mikrodatamatsystemet er baseret på 8085A-2 mikrodatamat med 64 k RAM-lager i én bank med mulighed for lagerudvidelse til 16 banks (1024 k). Systemet er udbygget med 2 minidiskettedrev med 320 k-programlager.

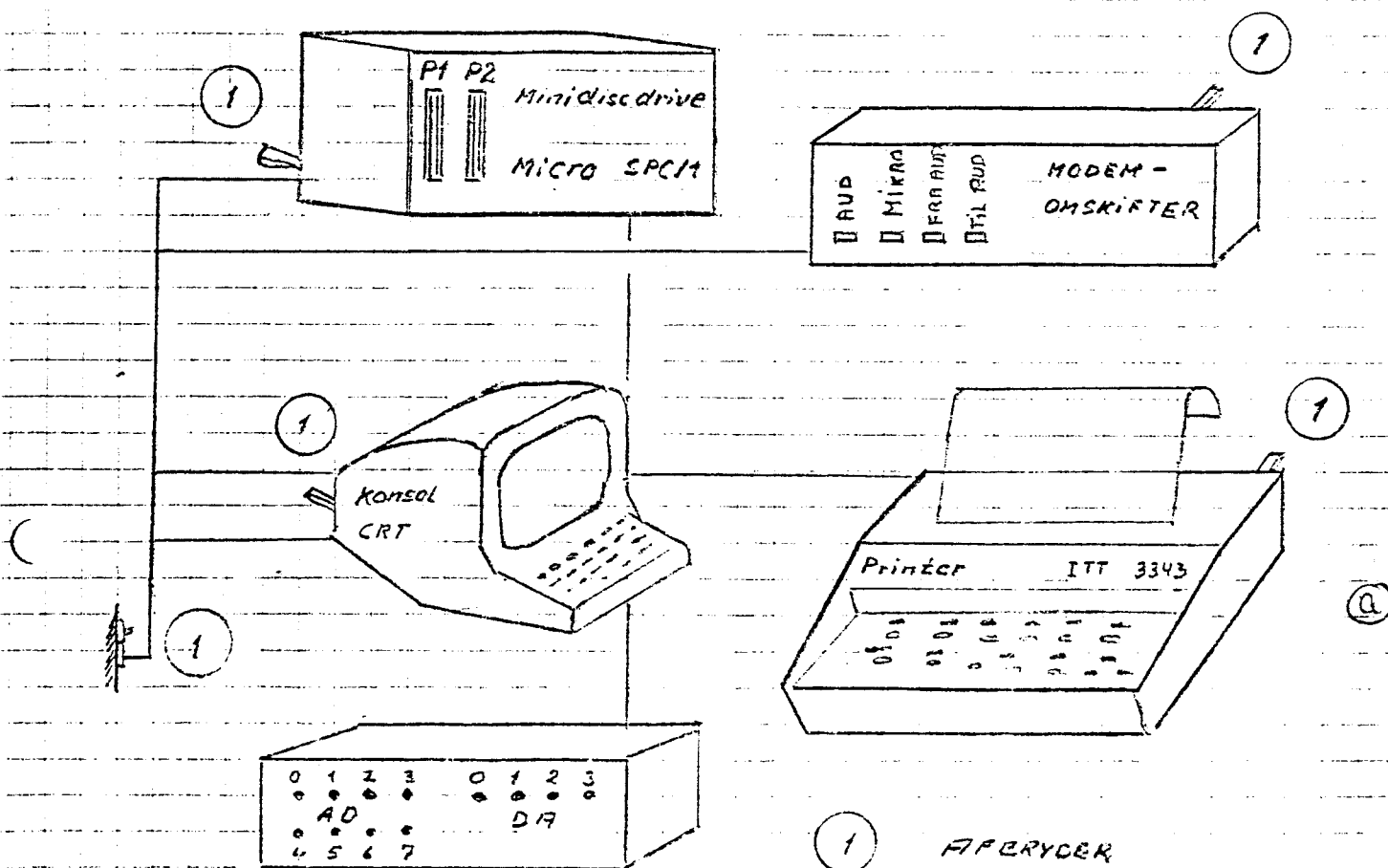


fig. 1

Programmeringen af mikrodatamaten kan foregå i forskellige sprog-niveauer:

ASSEMBLER (med makrofasciliteter).

COMAL (Struktureret Basic)

PASCAL (kompatibelt med standard Pascal).

Kun fasciliteter til programmering i højniveau-sproget PASCAL, som kendes fra langt større datamater, skal omtales i denne fremstilling, idet erfaringerne viser, at programudviklingstiden for at skrive et program i et højniveau-sprog er væsentligt mindre end den tid det tager for at skrive et tilsvarende program i f.eks. Assembler.

1.2. Modemomskifter.

Trykkes på MIKRO, kører SPC/1 - systemet som selvstændig datamat.

Trykkes på AUD kører konsol som almindelig terminal til AUD.

Til AUD/fra AUD anvendes ved dataoverførsel fra MIKRO til AUD et vice versa.

1.3. Konsol.

Konsollen virker som en almindelig skærmterminal med 80 karakterer pr. linie og 24 linier pr. side.

1.4. Printer.

Printer ITT 3343 anvendes kun som output medium (d.v.s. tastaturet benyttes ikke).

1.5. I/O-kasse.

Beskyttelses- og koblingsanordning for 8 analoge indgange og 4 analoge udgange for henholdsvis 8 kanaler A/D konverteringsmodul og 4 kanaler D/A konverteringsmodul.

2. OUTPUT CONTROL.

Listedevise kan via konsol defineres efter ønske. Skrives på konsol

>.LI,D defineres konsol som outputmedium

>.LI,P defineres printer som outputmedium.

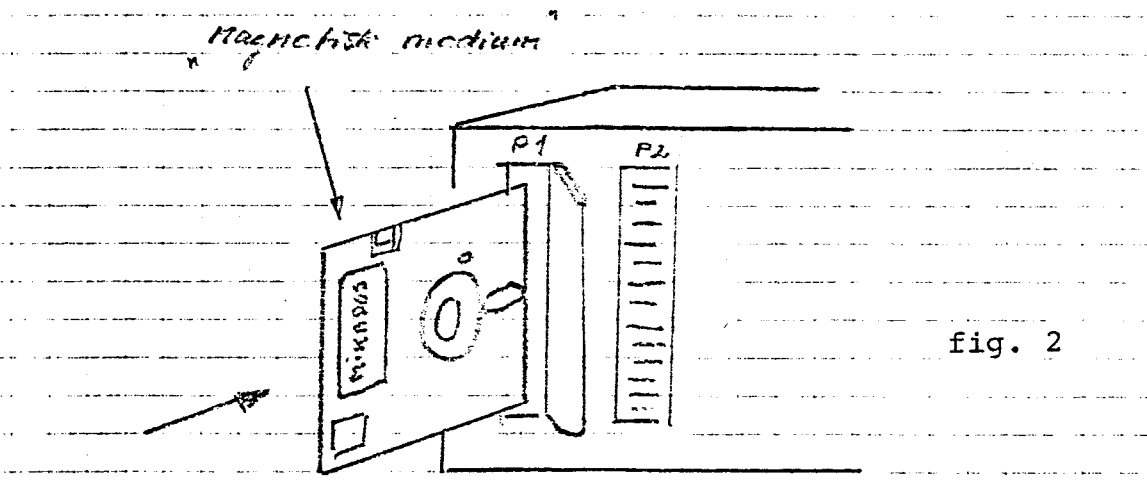
For at skrive tegnet > trykkes esc -tasten på konsol, hvilket generelt indleder enhver ordre til styresystemet.

I et Pascal-program skrives f.eks. Write(LIST,X,Y)

Denne sætning foranlediger talsættet x,y udskrevet på konsol eller printer afhængig af definition af listedevise.

3. OPSTART.

1. Sørg for at diskettedrev er tomme. Dette er vigtigt, idet evt. disketter i drev ved start kan ødelægges.
2. Tænd for netforsyning.
Tænd for modem og selektor. (Knap trykkes ind på f.eks. MIKRO).
Tænd for konsol.
Tænd for printer.
Tænd for mikro.
3. Anbring systemdiskette Mikados i P1 orienteret som vist på fig. 2.



Luk lågen i P1 og afvent svar på konsol.

Systemet er nu "Loaded" med Mikados styresystem, som er et multiprogrammerbart operativ-system.

(b)

4. STOP.

1. Fjern alle disketter fra drev. Vigtigt ellers kan disketter ødelægges.
2. Sluk for mikro, modem, konsol, printer og netforsyning.

5. PASCAL SYSTEM.

Dette system kører under MIKADOS-operativsystem (d.v.s. opstart-proceduren beskrevet i det foregående skal gennemføres).

Systemet består af følgende hovedprogrammer:

1. MIKADOS EDITOR.
2. PASCAL COMPILER og INTERPRETER.

5.1. Mikados Editor.

Editorfilen tillader oprettelse og redigering af filer. Editoren findes på diskette som endvidere indeholder 1Editfil, 2Editfil.

a) START AF EDITOR. Skriv på konsol:

'> EDITOR' Afvent systemets svar på konsol

esc - taster på konsol for tegnet >

Hvis filen eksisterer, skal 'filnavn' indtastes.

Hvis filen skal oprettes taster return.

Skrivning eller editering af program kan nu finde sted under anvendelse af de specielle editorordrer, som kort er beskrevet på den følgende side.

b) STOP AF EDITOR.

Efter skrivning og/eller editering afsluttes typisk med en af følgende ordrer, der placeres umiddelbart efter editorsvaret /

/'A'

A-ordren afslutter editeringen uden ændringer i det oprindelige dokument

/'EE'

EE-ordren (kort variant af EE-filnavn) gemmer dokumentet i den eksisterende fil 'filnavn', hvis indhold hermed erstattes af det nye dokument.

/'EOMYPROG:P2:4'

EO-ordren gemmer dokumentet i den nyoprettede fil MYPROG ('filnavn') på drev P2 med pladsreservation 4 (småtekster under 2 sider).

Ved krav om mere plads erstattes 4 med 10 (2-20 sider) eller 25 (over 20 sider).

5.1.1. Kortfattet oversigt over ordrer til editoren

I oversigten anvendes følgende specialudtryk:

filnavn op til 8 bogstaver eller tal, filnavnet kan efterfølges af ':Pl', dvs. en pladebetegnelse, samt af ':filstørrelse i sektorer'

tttt et decimalt tal, højst 9999

abc, defg tegnstreng (ord, dele af ord)

aktive linie den linie der umiddelbart kan modificeres

Systemet accepterer følgende ordrer:

NB¹ ingenting(RETURN ved liniestart) editer den aktive linie

NB¹ blank indsæt denne linie efter den aktive linie

A afslut editeringen uden at modificere det oprindelige dokument (ændringer i dokumentet kasseres)

Bttt linien som ligger tttt linier før den aktive linie gøres aktiv

^ECEfilnavn gem dokumentet i den eksisterende fil 'filnavn' og genstart editoren

COfilnavn gem dokumentet i den nyoprettede fil 'filnavn' og genstart editoren

D - del den aktive linie (med skilletegn)

D / del den aktive linie (uden skilletegn)

EEfilnavn gem dokumentet i den eksisterende fil 'filnavn' og stop editoren

EOfilnavn gem dokumentet i den nyoprettede fil 'filnavn' og stop editoren

Fabc find tegnstrengen abc

I indsæt denne linie før den aktive linie

Ltttt vis de næste tttt linier på dataskærmen

Mfilnavn kopier dokumentet i 'filnavn' ind i dokumentet
 efter den aktive linie

N vis nummeret på den aktive linie

Rabc/defg/ erstat tegnstrengen abc i den aktive linie med
 tegnstrengen defg

Sx definer nyt skilletegn ('x')

T komprimer det afsnit som den aktive linie står i
 startende fra den aktive linie

Znn,mm definer marginzone (nn >= 10, nn < mm <= 77)

Z vis marginzone

-tttt slet tttt linier fra og med den aktive linie

/ vis den næste linie og gør den aktiv

tttt vis linien der har nummer tttt, og gør den aktiv

NB: Editeringen skal afsluttes med f.eks. A eller E-ordre ellers kan editoren ødelægges.

En mere detaljeret beskrivelse af de enkelte editorordrer findes i MIKADOS EDITOR -Brugervejledning, som er placeret ved SPC/1.

5.2. PASCAL COMPILER OG INTERPRETER.

Pascal oversættereren accepterer kildetexten skrevet (ved hjælp af Editor) i standard Pascal (med få afvigelser) og oversætter dem til en kode for en hypotetisk computer benævnt "P-maskinen". Denne kode ("P-kode") fortolkes af et fortolkerprogram som simulerer den hypotetiske computer på SPC/1 computeren under MIKADOS operativsystem.

Pascal reference manual:

Katleen Jensen and Niklaus Wirth

Pascal: User manual and report, second edition.

SPRINGER VERLAG NEW YORK Inc.

En detaljeret beskrivelse af afvigelser fra referencemanualen findes i

MIKADOS: PASCAL USER'S GUIDE DDE, som er fremlagt ved SPC/1-systemet.

5.2.1. START AF COMPILER (oversætter).

For at starte oversætteren indsætter brugeren "Pascal system diskette" i P1 drev samt Pascal kildetext på diskette i P2 drev.

Herefter skrives:

```
>PASCALE, programnavn, listoption, externaloption, Pn
```

hvor

programnavn er filnavnet for det kildetekst modul, der skal oversættes. Filnavnet skal være et lovligt Mikodos filnavn (se definition under editor afsnit 5.1.1) f.eks.

PROGR(:P2). Diskettedrevidentifikation (her P2) kan udelades hvis der på såvel P1 som P2 kun findes en fil med filnavnet PROGR.

Listoption Denne option styrer listeoutput som produceres af oversætteren. Der kan vælges mellem L og Q. Vælges L som Listoption styres output fra oversætteren af den output-controldefinition operatøren har foretaget med sin >.LI kommando (jævnfør afsnit 2). Et eksempel på Listeoutput på printer kan ses på følgende udsnit af et Pascal-program:

```

1 0 PROGRAM GEN;
2 0 TYPE
3 0     SEMAFOR=ARRAY (1..3) OF INTEGER;
4 0     MESSAGE=ARRAY (1..7) OF INTEGER;
5 0     BUFFER=ARRAY (1..16) OF INTEGER;
6 0 VAR TSAMPLE,D,E,E6,E7,E8,E9: REAL;
7 0     TSAMPL,X,J,I,K,M,SUM,SEMA: INTEGER;
8 0     FIL: STRING;
9 0     DATA: TEXT;
10 0    A: ARRAY(.1..9.) OF INTEGER;
11 0    B: ARRAY(.1..9.) OF REAL;
12 0    SEM: SEMAFOR;
13 0    MES,MES1: MESSAGE;
14 0    BUF,BUF1: BUFFER;
15 0    OK: BOOLEAN;
16 0 PROCEDURE CSEM(VAR I: INTEGER);EXTERNAL;
17 0 PROCEDURE LOC(VAR I: INTEGER;SEM: SEMAFOR); EXTERNAL;
18 0 PROCEDURE SIGNA(VAR B: BOOLEAN;SEM: INTEGER;VAR MESS: MESSAGE
19 0     VAR BUF: BUFFER); EXTERNAL;
20 0 PROCEDURE WAITP(VAR SEM:SEMAFOR; VAR MESS:MESSAGE;
21 0     VAR BUF: BUFFER); EXTERNAL;
22 0 PROCEDURE ANADIG(VAR AI,PORTNR :INTEGER);
23 0 BEGIN
24 1     OUT80(8,PORTNR);
25 1     OUT80(9,0);
26 1     WHILE (IN80(9,16))=0 DO;
27 1     AI:=IN80(9,3);
28 1     AI:=IN80(8,255)*4+AI-512;
29 1     IF AI=511 THEN WRITELN('OVERFLOW IN PA PORT ',PORTNR);
30 1     IF AI=-512 THEN WRITELN('-OVERFLOW IN PA PORTNR ',PORTNR);

```

Bemærk linienumre og blokniveauskift efter begin, som letter editeringen. Eventuelle fejlmeddelelser kommer umiddelbart efter den fejlbehæftede linie. Vælges Q option undertrykkes programlistningen. Kun fejlmeddelelser listes på listedevice.

Externaloption Denne option anvendes kun, hvis der refereres til eksterne procedurer i kildeteksten, f.eks. til assembler programmer. Der findes følgende options E, C, F og D. For nærmere beskrivelse henvises til PASCAL USERS GUIDE.

Pn Er diskettedrevsidentifikation for disketten på hvilken modulet (P-kode), som oversætteren producerer skal gemmes. Hvis denne ikke specificeres er standard P2.

TYPISK EKSEMPEL:

```
>PASCALE, MYPROG, L
```

P-coden og evt. et relokerbart modul, som produceres af oversætteren placeres i en type P og evt. R-fil under samme navn som kilde-tekstfilen, der er placeret som en type k-fil. Hvis en eller flere af disse filer eksisterer erstattes det eksisterende indhold med de nye moduler produceret af oversætteren.

OBS. GØR INGEN FORSØG PÅ AT EKSEKVERE ET FEJLBEHÆFTET PROGRAM.

Vedrørende fejlmeddelelser se uddrag af Pascal-USERS GUIDE, Appendiks A og B.

5.2.2. Interpreter.

Er det oversatte program uden fejl kan programmet eksekveres ved at skrive

```
> INTRE, programnavn
```

EKSEMPEL:

```
> INTRE, MYPROG
```

5.2.3. Appendix A. Compile time error messages.

Most of the error numbers are similar to those defined in appendix E of J&W. Errors with numbers > 400 cause the compiler to terminate. Only the first error detected on a source line is reported.

- 1: error in simple type
- 2: identifier expected
- 3: 'PROGRAM' expected
- 4: ')' expected
- 5: ':' or '..' expected
- 6: symbol illegal in context (maybe missing ';' on the line above or ';' in front of ELSE)
- 7: error in parameter list
- 8: 'OF' expected
- 9: '(' expected
- 10: error in type
- 11: '(.' expected
- 12: '.)' expected
- 13: 'END' expected
- 14: ';' expected
- 15: integer expected
- 16: '=' expected
- 17: 'BEGIN' expected
- 18: error in declaration part
- 19: error in <field list>
- 20: ',' expected
- 21: '.' expected

- 50: error in constant
- 51: ':=' expected
- 52: 'THEN' expected
- 53: 'UNTIL' expected
- 54: 'DO' expected
- 55: 'TO' or 'DOWNTO' expected in FOR-statement
- 58: error in <factor> (bad expression)
- 59: error in variable

- 101: identifier declared twice
- 102: low bound exceeds high bound
- 103: identifier is not of the appropriate class
- 104: undeclared identifier
- 105: sign not allowed
- 106: number expected
- 107: incompatible subrange types
- 108: file not allowed here
- 109: type must not be real

- 110: <tagfield> type must be scalar or subrange
- 111: incompatible with <tagfield> part
- 113: index type must be a scalar or a subrange
- 114: base type must not be real
- 115: base type must be a scalar or a subrange
- 116: error in type of standard procedure parameter
- 117: unsatisfied forward reference
- 119: re-specified parameters not ok for a forward declared procedure
- 120: function result type must be scalar, subrange or pointer
- 121: file value parameter not allowed
- 122: a forward declared function's result type cannot be re-specified
- 123: missing result type in function declaration
- 124: f-format for reals only
- 125: error in type of standard function parameter
- 126: number of parameters does not agree with declaration
- 127: illegal parameter substitution
- 128: result type does not agree with declaration
- 129: type conflict of operands
- 130: expression is not of set type
- 131: tests on equality allowed only
- 132: strict inclusion not allowed
- 133: file comparison not allowed
- 134: illegal type of operand(s)
- 135: type of operand must be boolean
- 136: set element type must be scalar or subrange
- 137: set element types must be compatible
- 138: type of variable is not array
- 139: index type is not compatible with the declaration
- 140: type of variable is not record
- 141: type of variable must be file or pointer
- 142: illegal parameter solution
- 143: illegal type of loop control variable
- 144: illegal type of expression
- 145: type conflict
- 146: assignment of files not allowed
- 147: label type incompatible with selecting expression
- 148: subrange bounds must be scalar
- 149: index type must not be integer
- 150: assignment to standard function is not allowed
- 152: no such field in this record
- 153: type error in read
- 154: actual parameter must be a variable
- 155: control variable cannot be formal or non-local
- 156: multidefined case label
- 157: too many cases in case statement
- 158: no such variant in this record
- 159: real or string tagfields not allowed
- 160: previous declaration was not forward
- 161: again forward declared
- 162: parameter size must be constant
- 163: missing variant in declaration

164: substitution of standard procedure/function not allowed
165: multidefined label
166: multideclared label
167: undeclared label
168: undefined label
169: error in base set
173: externaloption not specified in RUN command (see section 4.1)

201: error in real number - digit expected
202: string constant must not exceed source line
203: integer constant exceeds range

250: too many scopes of nested identifiers
251: too many nested procedures or functions
252: too many forward references of procedure entries
253: procedure too long
257: too many external procedures

320: READREAL or WRITEREAL procedure not included (see section 7.12 or 7.13)

398: implementation restriction
399: implementation restriction

400: illegal character in text
401: unexpected end of input
408: include control comment not allowed in inclusion file
10xx: error during open of inclusion file
11xx: error during open of source file
12xx: error during create/open of P-code file
13xx: error during create/open of relocatable file
14xx: error during output to relocatable file
15xx: error during output to P-code file

In the error messages with numbers ≥ 1000 the last two digits represent the MIKADOS file system error code. The MIKADOS file system error codes are:

1: a file with the specified name does not exist
2: a file with the specified name already exists
3: no more room on disc
4: illegal record length
5: the file is being used by another user
6: the specified DCB is not open
8: attempt to extend a file more than 60 times
9: the file name is illegal (the first character is not printable)
10: illegal disc identification
11: attempt to position file to non-existent record
12: attempt to read or write a record that lies partly or completely outside the file boundaries
13: the file has not been opened for writing

- 14: the catalog on the disc is full
- 15: illegal DCB length
- 16: illegal number of sectors in file
- 17: illegal file type
- 18: the file name has not been reserved (returned by CLOSE if file not open or DCB bombed)
- 19: error in variable length record file format

- 40: disc drive not ready
- 42: hard error on disc
- 44: disc drive is write protected
- 48: illegal track/sector number or illegal buffer length
- 50: transfer extends past last sector of disc drive
- 52: illegal disc identification

- 1: end-of-file disregarded
- 2: input error (READREAL)
- 3: input error (READ integer)

MIKADOS file system errors 4, 15, 18, 48, 50, and 52 normally should not occur in Pascal systems. If they do the system has probably been destroyed by a pointer or index error.

5.2.4. Appendix B. Run time (interpreter) error messages.

During interpretation of a Pascal program (including the Pascal compiler itself), the interpreter checks for a number of error conditions. If an error condition is detected it is reported with a message similar to the following:

```
RUN TIME ERROR x NEAR LINE yyyy
```

where yyyy is the line number of the last executed Pascal statement that was compiled with the debug option enabled (see section 4.2). If the debug option was not enabled during any part of the compilation of the program section executed before the error occurred, yyyy will be 0000. x will be replaced by

- D - division by zero
- E - external procedure error (probably wrong interpreter used to execute program with external references)
- I - invalid index (if index and range checking disabled only string index < 1 or > 255 detected)
- K - stack or heap overflow
- L - string too long or parameter error in intrinsic string procedure
- M - standard procedure not implemented
- N - P-code instruction not implemented
- O - floating point overflow
- P - floating point error (error in PWROFTEN call)
- R - execution error (XERR instruction executed)
- S - non-existent segment called (system error)
- U - user i/o error (only if i/o checking not disabled, see section 5.2)
- X - exiting procedure never called (system error)

The error codes returned by IORESULT are described in appendix A.

If the Pascal compiler fails with an interpreter error message other than K, M, O, or P, start by correcting all syntax errors reported during the compilation. Then recompile the program. If the error persists, contact DDE.

If the Pascal compiler fails with interpreter error message K, increase the compiler data area (increase MAXADR, see section 4.4), or reduce the number of identifiers declared in the program.

If the Pascal compiler fails with interpreter error message M, then the interpreter used is not suitable for compiling the Pascal program (special compiler module or floating point module required in the interpreter).

If the Pascal compiler fails with interpreter error message O or P, then the source statement just after the last one listed on the list device probably contains a (real) constant which is not acceptable to the compiler.

If a program (including the Pascal compiler) fails with an interpreter error, then the files opened by this program at the time when the error occurred are not closed. This may result in file system error 5 (file in use) in subsequent attempts to open the file. The error condition can be removed by restarting the operating system.

6. YDRE ENHEDER.

I det følgende skal omtales muligheder for at kommunikere med ydre enheder i højsproget Pascal. Fremstillingen omfatter kommunikation med A/D - D/A konverteringsmoduler samt intervalltimermodul.

6.1. Grundlæggende input/output operationer i Pascal.

Følgende 2 rutiner udfører de basale INTEL 8080 input/Output operationer på specificerede i/o adresser (porte).

FUNKTION IN80 (IOADRESSE, MASK:INTEGER): INTEGER;

PROCEDURE OUT80 (IOADRESSE, VALUE:INTEGER);

IN80 funktionen er, når der ses bort fra fortolkningen, ækvivalent med følgende assembler sekvens:

IN IOADRESSE (8 bits læses fra den specificerede i/o adresse)

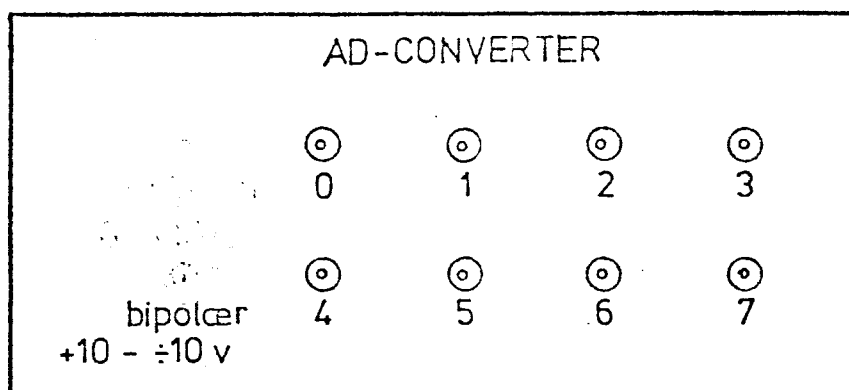
ANI MASK (De læste 8 bits AND-opereres med MASK)

OUT80 proceduren er ækvivalent med følgende assembler sekvens:

LDA VALUE (8 mindst betydende bits af VALUE læses)

OUT IOADRESSE (sendes ud til i/o adressen)

6.2 A/D Konverteringsmodul

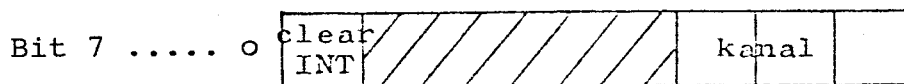


Modulet benyttes med henblik på at forbinde op til 8 analoge indgange til SPC/1-mikrodatamat, som det fremgår af ovenstående skitse af frontpanel på tilslutningskasse for A/D konvertering. De analoge spændinger må for den aktuelle tilslutningskasse variere bipolarart (-10 volt til +10 volt) svarende til en 10 bits (binary digits) repræsentation af det analoge måleområde, som belyst i følgende tabel:

Analog input	Digital output	Decimalrepræsentation (standard binær kode)
	9 8 7 6 5 4 3 2 1 0 2 2 2 2 2 2 2 2 2 2	
+ 9,98 volt	1 1 1 1 1 1 1 1 1 1	1024
+ 5,00 volt	1 1 0 0 0 0 0 0 0 0	768
0,00 volt	1 0 0 0 0 0 0 0 0 0	512
- 5,00 volt	0 1 0 0 0 0 0 0 0 0	256
- 10,00 volt	0 0 0 0 0 0 0 0 0 0	0

Der benyttes 2 i/o adresser, - henholdsvis 8 og 9 -, når der kommunikeres med A/D modulet. Pascal sætningen:

OUT80(8,HELTAL); (* HELTAL 0 til 255 *) vil bevirke at kanalnummer og interrupt-flip-flop status på A/D-konverteringsmodulet vælges efter følgende regler:



Bit 2-0 : Kanalnummer

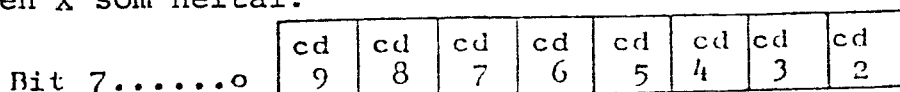
Bit 7 : Hvis 0 foretages ingenting.

Hvis 1 cleares INT.

OUT80(9,HELTAL); (* Hvor heltal vilkårligt *) vil bevirke, at INT flip-flop cleares og konverteringen startes. Når denne er endt og de konverterede data er klar i bufferen, sættes INT. Konverteringen tager ca. 200µs.

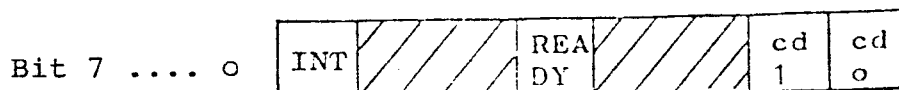
X: = IN80(8,255)

vil bevirke, at de 8 mest betydende af de 10 bits indlæses i cellen X som heltal:



Y: = IN80(9,MASK); (*MASK 0 til 255*)

vil bevirke at statusordet og de 2 mindst betydende bit indlæses i Y.



Bit 1-0: De 2 mindst betydende bit af det konverterede signal.

Bit 4 : Bitten er 1 når konverteringen er endt, ellers 0.

Bit 7 : Angiver status af INT-f.f.

EKSEMPLER:

MASK = 3 bevirker at Y tilskrives heltalsværdien af den binære repræsentation for de 2 mindst betydende bits af det konverterede signal. D.v.s. af $Y_e(0,1,2,3)$.

MASK = 16 bevirker test på ready-bit

$Y = 0 \Rightarrow \text{ready-bit} = 0$

$Y = 16 \Rightarrow \text{ready-bit} = 1$

MASK = 128 bevirker test på INT-bit

$Y = 0 \Rightarrow \text{INT} = 0$

$Y = 128 \Rightarrow \text{INT} = 1$

En decimalrepræsentation Z (fra 0 til 1024) for et analogt målesignal på kanal 2 kan opnås ved følgende Pascal programsekvens:

OUT80(8,2);

OUT80(9,2);

X: = IN80(8,255);

Y: = IN80(9,3);

Z: = 4*X+Y;

Ovenstående program tager ikke hensyn til indstillings- og konverteringstider. De 8 analoge indgange er multiplexet af tekniske årsager. Multiplexeren kræver en indstillingstid på 70 μ s til kanalskift, d.v.s. at der mellem en OUT80(8,2) og en OUT80(9,2) skal gå mindst 70 μ s. Efter oplysninger fra DDE kræver fortolkning af en OUT80 ordre ca. 30 instruktioner af ca. 8 maskincykler af 0,5 μ s svarende til 120 μ s. Således kan konkluderes, at multiplexerens indstillingstid ikke kræver programtekniske overvejelser. Derimod kræver OUT80(9,2), som varer 200 μ s, en test på f.eks. ready-bit før IN80 ordrene.

6.2.1. Essens vedrørende A/D-konvertering.

Følgende reviderede program med test af ready-bit skulle således forestå A/D konverteringen af et analogt signal på kanal N:

OUT80(8,N); (*N HELTAL FRA 0 TIL 7 - KANALNR-*)

OUT80(9,N);

XX: = 0

REPEAT

XX: = IN80(9,16); (*Test på ready-bit*)

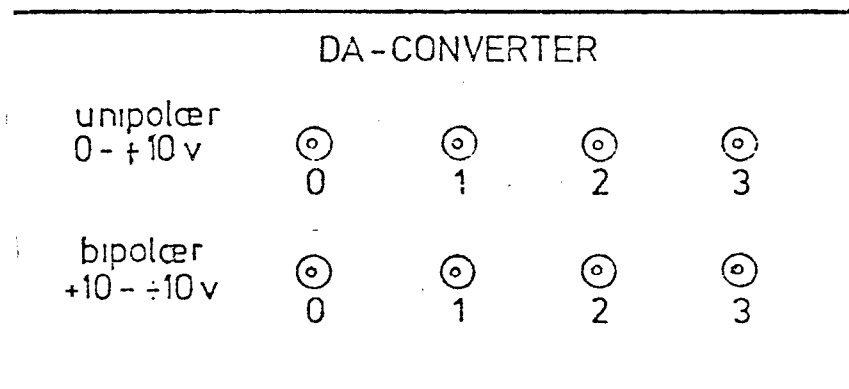
UNTIL XX = 16

X: = IN80(8,255);

Y: = IN80(9,3);

Z = 4*X+Y;

6.3. D/A Konverteringsmodul.




Modulet benyttes med henblik på at forbinde SPC/1-mikrodatamat med op til 4 analoge udgange, som det fremgår af ovenstående skitse af frontpanel på tilslutningskasse for A/D-konvertering. De analoge spændinger leveres unipolær eller bipolær efter ønske, som vist på skitsen. Spændingerne svarer til en analog repræsentation af henholdsvis heltalsområdet fra 0 til 1024 og binær repræsentation af samme område som illustreret i nedenstående tabel:

DECIMAL HELTAL	BINÆR repr. af INPUT	UNIPOLÆR OUTPUT	BIPOLÆR OUTPUT
1024	1111111111	+ 9,98V	+ 9,98V
512	1000000000	+ 5,00V	0,00V
0	0000000000	0,00V	- 10,00V

For kanalerne 0,1,2 og 3 gælder følgende:

`OUT80(2*KANALNR,X); (*KANALNR,X: INTEGER*)`

bevirker at de 2 mest betydende bits af den binære repræsentation af X sendes til D/A-modulets buffer.

Bit 7.....0 

`OUT80(2*KANALNR+1,X); (*KANALNR 0 til 3*)` bevirker at de mindst betydende bit sendes til D/A-modulets buffer. Konverteringen startes, og det endelige signal sendes ud på den specificerede output-

kanal.

cd	cd	cd	cd	cd	cd	cd	cd
7	6	5	4	3	2	1	0

Bit 7.....0

6.3.1. Essens vedrørende D/A konvertering.

Følgende program vil konvertere en variabel XXX til en analog spænding på kanal 1:

```
VAR XXX: REAL; (*XXX ventes i interval - 10.0 til 10.0*)
```

```
YYY,ZZZ: INTEGER;
```

```
XXX:= 51.2*XXX;
```

```
IF XXX > 511.0 THEN
```

```
XXX:= 511.0;
```

```
IF XXX < - 512.0 THEN
```

```
XXX:= -512.0;
```

```
YYY:= (TRUNC(XXX+512.0))DIV256;
```

```
OUT80(2,YYY); (*2MSB på port 1*)
```

```
ZZZ:= (TRUNC(XXX+512.0))MOD256;
```

```
OUT80(3,ZZZ); (*8LSB på port 1*)
```

6.4. Intervaltimer Modul.

Interval timer modulet kan programmeres til at foretage interrupt i intervallet 50 μ s til 12,8 sek.

Intervallet fastlægges dels af DIVIDE-tælleren og dels af MODULUS-tælleregistret. DIVIDE-tælleren leverer clockpulser til MODULUS-tælleregistret. Disse clockpulser vælges af programmøren til en af følgende tider: 50 μ s, 0,5ms, 5ms eller 50ms. Disse Clockpulser tælles så op i et 8 bit MODULUS-tælleregister, der efter at have nået til decimaltallet 255 vil sætte interruptflip-floppen, INT, når næste clockpuls fra DIVIDE-tælleren ankommer. Dette MODULUS-tælleregister er programmerbart, således at programmøren bestemmer fra hvilket starttal (mellem 0 og 255) optællingen skal begynde. Efter at den har rundet 255, og INT er sat, indsættes starttallet automatisk igen. Hvis der ønskes et interrupt for hver N-clockpulser, skal modulusregistret lades med N-1.

Eksempel:

Der ønskes en intervallængde på 10 sec. DIVIDE-tælleren vælges til perioden 0,05 sec, hvorved der ønskes 200 clockpulser pr. modulusperiode. N er således 200, og modulusregistret lades med 200-1 = 199.

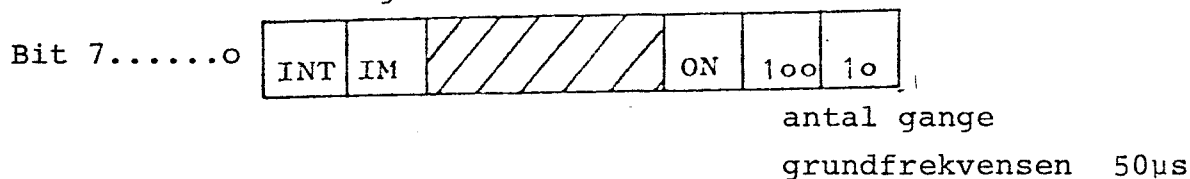
Foruden INT'f.f. indeholder modulet en interrupt mask flip flop, IM. Når $INT \cdot IM = 1$, sendes et interrupt request til CPU-enheden, på prioritetsniveau nr. 5 (start adresse 28). Da IM af programøren kan sættes til 0, respektivt 1, er det således muligt at bestemme hvorvidt der ønskes interrupt eller ej, selvom timeren kører. ITM benytter 2 adresser, 14 og 15, hvorved dets in- og outinstruktioner fastlægges:

OUT80(14,N-1); (*N antal clockpulser*)

bevirker indstilling af modulregister

OUT80(15,MASK); (*N fra 0 til 255*)

bevirker indstilling af divide tæller samt status:



Kun 5 bit benyttes

Bit 1-0 : Fastlægger frekvensen af DIVIDE-tælleren.

0 0 = 50µs

0 1 = 0,5ms

1 0 = 5ms

1 1 = 50ms

Bit 2 : starter/stopper tælleren

0 = stopper og resetter tælleren

1 = starter tælleren/lader tælleren køre videre

Bit 5-3 : Ikke i brug

Bit 6 : Interrupt mask flip-flop

0 = IM = 0. Disable interrupt

1 = IM = 1. Enable interrupt.

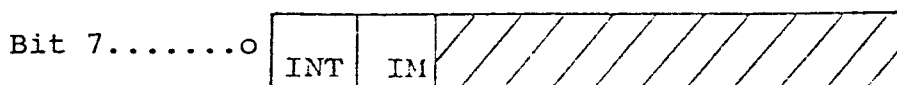
Bit 7 : Reset INT flip-flop.

0 : INT = 0

1 : INT. = INT. (som forrige tilstand)

Y: = IN80(15,255);

Bevirker test af INT og IM



Kun 2 bit benyttes:

Bit 7 : INT flip-floppens status (interrupt f.f.)

Bit 6 : IM flip-floppens status (interrupt mask f.f.)

6.4.1. Essens vedrørende interval timer.

Ønskes et program, som starter en hændelse for hver N sekunder kan det have følgende udseende:

```
REPEAT
```

```
OUT80(14,20*N-1); (*MODULUSREGISTER TIL 20 N*)
```

```
OUT80(15,3) ; (*NULSTIL TÆLLER*)
```

```
OUT80(15,7) ; (*START TÆLLER,DIVIDE-50MS*)
```

```
{ Program som laver en hændelse
  '
  '
  '
  '
}
```

```
Y: = 0
```

```
REPEAT
```

```
Y: = IN80(15,128) } (*vent på at tiden går*)
```

```
UNTIL Y = 128
```

```
UNTIL
```

7. MIKADOS FILSYSTEM ROUTINER.

Dette afsnit beskriver kort typiske anvendelser af en række filsystemroutiner: PLINI, KATLG, PKOPI, FCOPY, KOMPR, RENAME. En mere detaljeret beskrivelse kan findes i DDE: Mikados utility programs and subroutines.

7.1 PLINI - initialisering af en diskette.

Programmet initialiserer (sletter) en diskette og anvendes til at fjerne alle filer fra en diskette samt til at formatere disketten.

En typisk dialog er vist i det følgende (med operatør-input understreget)

>PLINI

Enter disc drive identifikation P2

Enter/Change disc label DDE

Enter/Change disc type MIK

Enter/Change date of last back-up 06.02.1980

Label sector output to disc, Initialization desired (Y/N)?Y

Read-back disired (Y/N)?Y

En initialisering med PLINI anvendes fortrinsvis på fejlbehæftede disketter f.eks. på en ødelagt editor (file system error 42).

7.2 KATLG - output diskette katalog.

Routinen sender information om indhold på en specificeret diskette til liste device.

>KATLG,P2. vil foranledige en listning af indhold på diskette i P2, som vist i det følgende:

FILENAME	TYPE	EXTENT	TRACK/SECT.	#SECT.	REC.LEN.	#EXT.	DISC P2 PAGE 0001
SKRIV	K	A	0001 0002	0004	0254	0000	
SKRIV	P	A	0002 0003	0010	0128	0000	
STEEN	K	A	0003 0004	0004	0254	0000	
STEEN	P	A	0003 0008	0010	0128	0000	
HENRIKN	K	A	0005 0000	0004	0254	0000	
HENRIKN	P	A	0005 0004	0010	0128	0000	
LØKKE	K	A	0006 0005	0004	0254	0000	
LØKKE	P	A	0007 0000	0010	0128	0000	
PETER	K	A	0008 0001	0004	0254	0000	
PETER	P	A	0008 0005	0010	0128	0000	

First unused track/sector: 0009 0006. Number of unused sectors: 00264.
 Disc type: MIK . Date of last backup: 03.09.1979. Disc label: DDE1
 Number of purged sectors: 00000.

Mere generelt kan KATLG-ordren lyde:

>KATLG,P2,OPTION

OPTION er her enten O,F,X,N,Q eller S for nærmere beskrivelse se DDE MANUAL.

7.3 PKOPI - kopier en diskette.

Ordren kopierer al information fra en diskette til en anden ved f.eks. følgende dialog:

>PKOPI

Source disc: P1

Destination disc: P2

Før kopieringen starter skal følgende spørgsmål bekræftes:

Source disc type: DIS1*. Destination disc type: DIS10

DO YOU STILL WANT COPYING? Y/N Y

Bemærk at PKOPI stiller krav om en passende formatering af destinationsdisketten i relation til den diskette, der kopieres fra, hvilket nærmere er uddybet i DDE-manualen. Er man i tvivl om formateringen er korrekt bør FCOPY anvendes.

7.4 FCOPY - kopier fil(er).

Routinen kopierer alle filer som korresponderer med et specificeret filnavn (master file name).

Følgende dialog viser hvorledes alle kildetext filer overføres fra diskette P1 til diskette P2, idet ***** (8 stjerner) kan bruges for alle filnavne.

>FCOPY

Enter source disc identifikation: P1

Enter destination disc identifikation: P2

Enter master file type: K

Enter master file name: *****

7.5 FPURG - slet fil(er).

Routinen sletter alle filer med et specificeret filnavn, som vist i følgende dialog:

>FPURG

Enter master file name: FPURG

Enter master file type: *

Enter disc identification: P2

Skal hele disketten slettes bør PLINI anvendes.

7.6 KOMPR - komprimér diskette.

Routinen genvinder plads, som bliver ledig efter FPURG. Følgende dialog viser anvendelse af ordren:

>KOMPR

Enter disc identification: P2

Routinen holder operatøren underrettet under eksekveringen.

7.7 RENAME - omdøb fil.

Routinen omdøber en fil ved en dialog som vist i det følgende:

>RENAME

Enter old file-name: disc identification OLDNAME: P1

Enter new file-name: NEWNAME

Enter file type K

7.8 OPROFIL - opret/modificer nulfil.

Speciel routine som tillader operationer på nulfiler (MIKADOS SYSTEM FILER). Da denne routine kun bruges ved specielle anvendelser, henvises til DDE manual.

8. Dataoverførsel til AUD.

8.1 Opkobling til jobterminalsystemet (export/import) via KN-switch. (modemomskifter)

Dialog-terminaler tilsluttet en ID-7000 koncentrator vil normalt kommunikere med terminalsystemet Telex, men kan ved anvendelse af en speciel ordre sættes i forbindelse med jobterminalsystemet.

Herefter vil terminalen blive behandlet som en almindelig jobterminal med tilkoblet strimmeludstyr og kortlæser. Filer, der skal hules på strimmelhuller udskrives i stedet på dataskærmen, mens indlæsning af strimmel- og kortdata foretages fra dataskærmens tastatur.

Via en KN-switch kan terminalen tilkobles en anden datamat hvorved det er muligt, at overføre data mellem denne og CDC-anlægget i samme format som via strimmel-/kortudstyr.

Af hensyn til afslutning af indlæsning m.v. er der defineret en række specialtegn/ordrer, der beskrives i det følgende.

- EXPORT** medfører opkobling til jobterminalsystemet. Denne ordre kan kun anvendes, når terminalen ikke er logget ind til telex. Bemærk, at kun store bogstaver accepteres. Dette gælder iøvrigt alle ordrer afgivet til jobterminalsystemet.
- QUIT** anvendes til at afbryde forbindelsen til jobterminalsystemet. Terminalen bør være logget ud når denne ordre afgives. I modsætning til alle øvrige ordrer kvitteres ikke med line-feed, men ordren er accepteret når et indtastet tegn ekkoes på skærmen.

Følgende specialtegn er defineret:

Bemærk, at specialtegn ikke skal afsluttes med return med mindre andet er angivet.

ESC anvendes til at markere afslutning af strimmelindlæsning.

CTRL C anvendes til midlertidig afbrydelse af indlæsning. denne kan herefter genoptages med ordren GO.

De følgende fire specialtegn kan kun anvendes hvis terminalen er defineret som overførselsterminal, hvilket sker når en fil-overførsel til jobterminalsystemet er initieret.

Skal disse kontroltegn anvendes vil det altså i nogle tilfælde være nødvendigt at sende en tom fil til CDC-anlægget inden brugen. Skal terminalen efter overførsel anvendes som en normal terminal kan jobterminal-status fjernes ved CTRL A umiddelbart efterfulgt af CR.

CTRL ATTTT anvendes til definition af det antal tidsenheder, der skal ventes uden inddata før en indlæsning automatisk afsluttes. Er en tidsperiode defineret kan der ikke foretages afslutning ved hjælp af ESC.

CTRL A ekkoer som et T.

Hvorefter antal tidsenheder kan indtastes afsluttet af CR. En tidsenhed er ca. det samme som et sekund. Indtastes et RETURN umiddelbart efter CTRL A nulstilles tidsmåling og ESC kan atter anvendes som sluttegn. Som nævnt ovenfor vender terminalen tilbage til normalttilstanden.

LINEFEED har samme effekt som CR, men ekkoes ikke på dataskærmen. Dette anvendes i forbindelse med overførsel til en lokal datamat, hvor man ikke ønsker den RETURN der sætter overførsel igang indlæst.

CTRL N medfører at efterfølgende tegn indlæses fra/udskrives på terminalen som 8 bits uden paritet.

CTRL E medfører at efterfølgende tegn indlæses fra/udskrives på terminalen som 7 bits med lige paritet. Dette er DEFAULT ved opstart.

Overførsel til CDC-anlægget kan foretages ved brug af export/import ordrene R(EAD) og T(APE) jvf. beskrivelse af jobterminalsystemet. Ved brug af READ-ordren (overførsel af kortbilleder) skal følgende konventioner overholdes:

- 1) De første linier skal være et styrekortsæt formatteret som hulkort med 80 tegn pr. linie og afsluttet af EOR.
- 2) EOR signaleres som ASCII-EOR (tegnværdi 30D) efterfulgt af 79 mellemrum.

- 3) Kun tegnværdier i intervallet 32D til 126D accepteres. Øvrige tegn frasorteres i koncentrator.
- 4) Overførte tegn vil af systemet blive opfattet som tekstdata. Det vil sige, at 80 tegn bliver opfattet som en linie. Det er muligt at øge overføringshastigheden ved korte linier, ved selv at definere et lineskiftsymbol og derefter foretage en konvertering på CDC-anlægget, men i det tilfælde bør man være opmærksom på at systemet fjerner efterstillede blanke i en linie.

8.2 Bemærkninger.

- 1) Systemet er ikke i stand til at håndtere inddata og uddata samtidig. Det er derfor nødvendigt at sikre sig, at der ikke er uddata til terminalen, inden en indlæsning startes, dette kan f.eks. gøres ved en LISTQ inden opkobling til jobterminalsystemet.
- 2) Der foretages ingen kontrol af, at data overføres korrekt. Normalt vil der ikke opstå problemer, men ved ekstrem høj belastning eller linienedbrud kan data formentlig tabes. Det vil derfor være en god ide, i det omfang det er muligt, at formatere data således at fejl let konstateres (tilføjelse af linienumre, kontrolsummer o.lign.).
- 3) Visse koncentratorer er ikke forsynet med afslutning på tid, men som beskrevet ovenfor kan det let konstateres, om definition af tidsenheder kan finde sted.
- 4) Ikke alle koncentratorer rummer mulighed for filoverførsel. Filoverførsel kan for øjeblikket finde sted fra koncentratorerne på Strandvejen, Sohngaardsholmsvej, Danmarksgade.
- 5) Der er TIME-OUT på opkobling til jobterminalsystemet. Hvis der ikke inden for et kvarter er foretaget en dialog med export/import afbrydes opkobling til dette system og den sædvanlige forbindelse til telex etableres påny.

NOJ/800529

En opdateret version af denne beskrivelse kan fåes ved følgende ordre:

```
GET(KBESK/UN=DANOJ)
```

Filen KBESK er formatteret til udskrift og kan umiddelbart udskrives med

```
PRINTQ(KBESK/EI=<JOBTERM>)
```

3
8.3 Essens vedrørende dataoverførsel til AUD.

Eksempel på overførsel til AUD.

TAST		SVAR
	switch sættes i stilling 'AUD'	
EXPORT	Terminalen sættes i forbindelse med jobterminalsystemet.	
LOGIN, <BRUGER>, <LØSEN>	Der logges ind, hvorefter systemet hvis alt er i orden svarer med Derefter kobles switch i stilling 'MICRO', og der gøres klar til overførsel uden at starte denne. Switch sættes i stilling 'AUD'	IDLE
T, <MODTBRUGER>, TTY	Der startes en strimmelindlæsning. Bemærk, at <MODTBRUGER> ikke må være det samme brugernavn, som man er logget ind under, da de indlæste data så vil blive returneret med det samme. Når systemet er klar svares
	Switch sættes i stilling 'TIL AUD' og overførsel startes. De overførte data udskrives samtidig på skærmen. Når overførsel er afsluttet sættes switch i stillingen 'AUD' og der tages	
ESC	Hvis alt er i orden svares	IDLE
L	Er der ikke flere data logges ud med hvorefter systemet svarer med dato/klokkeslet, og der frakobles med	
QUIT		
<MODTBRUGER>	Der logges ind under <MODTBRUGER>	
LISTQ		KØ NR. XXX
FETCHQ, UD=XXX	Hent køfil	
PTAPEC, UD, UDE. TTYN	Omsæt kode til lokalfil UDE	
PRINTQ (UDE/EI=I7UT)	Print lokalfil UDE	