

NASCOM ^{Z80} NYT

NASCOM BRUGERGRUPPE
2730 Herlev

Sidevolden 23
Giro 6742602

NR: 2
3. årgang

Februar 1982.

Det er med glæde, at jeg ønsker Erik Hansen tilbage til foreningsarbejdet efter overstået sygdomsperiode. Det skulle betyde, at alle ekspeditioner gennem foreningen vil gå hurtigt og planmæssigt. For en ordens skyld skal jeg nævne, at du skal modtage en udskrift over programbiblioteket, når du har fået et medlemnummer. Dette nummer står over dit navn på kuverten. Hvis du ikke har fået det, bedes du henvende dig til Erik Hansen. Til alle: vær venlig at læse bagsiden! Vi vil meget gerne indtage vor middag uforstyret, det er ret almindeligt, at I ringer kl. 18.00, for der er vi jo nok hjemme, men maden bliver kold.

Er der overhovedet ingen, der har lyst til at mødes med mig og lave blad

si'r

ASBJØRN

INDHOLD

side 2	Struktureret programmerind del 2
side 4	Bytefinder
side 5	Anmeldelse af POLYDOS
side 11	Orgel
side 12	String Save
side 13	Referat fra monitorgruppen
side 17	Styring af DC-motor
side 18	Hvad så ?
side 19	Hexadecimale tal
side 20	Programbibliotek, tips nyheder
side 21	Salg
side 22	Teac FD 50A & uP765 U1
side 23	Nye medlemmer

```

:#####
:## IF (X OR Y OR Z) THEN ..... ; ELSE .....
:##
:IF: set (true) / reset (false) flag X
:OR: JP X, THEN....
:OR: set (true) / reset (false) flag Y
:OR: JP Y, THEN....
:OR: set (true) / reset (false) flag Z
:OR: JP Z, THEN.... ; hvis man tør, så
:OR: JP ELSE....
:OR: THEN....: rutine
:OR: JP SLUT
:OR: ELSE....: rutine
:OR: SLUT: ...
:##
:## IF (X AND Y AND Z) THEN ..... ; ELSE .....
:##
:IF: set (true) / reset (false) flag X
:AND: JP NX, ELSE....
:AND: set (true) / reset (false) flag Y
:AND: JP NY, ELSE....
:AND: set (true) / reset (false) flag Z
:AND: JP NZ, ELSE....
:OR: THEN....: rutine
:OR: JP SLUT
:OR: ELSE....: rutine
:OR: SLUT: ...
:##

```

```

:#####
: 4. WARNIER ORR diagrammet
:#####
:
: Gennem et eksempel vil jeg vise, at det
: nærmest er så enkelt at anvende opstillingen,
: at det er selvforklarende. Teksten i venstre
: side skal man altid presse smal og høj, så
: udnyttes pladsen bedst. Venstre søjle er hele
: programmet i arøv fremstilling, næste søjle
: er noget mere detaljeret, men søjlen er det
: samme program, blot beskrevet på et lavere
: hierarkisk niveau.
:
: Mod højre møder man fremstillingen mere

```

3b. KOMPLEKS LOGIK med betingede hop

Se, det er just et alvorligt problem for enhver helhjertet amatør: man laver sine logiske tests ved at demonstrere et enormt kendskab til, hvordan flåsen i ens vndlings-computer virker og hvordan de kan snydes. Spændende assembler-artisteri uden sikkerhedsnet.

Oftest er en "test" en sammensat beregning, hvis resultat er "sand" eller "falsk". Denne beregning kunne man bruge de indbyggede logiske funktioner til ved at så frem som følger:

1. Giv hvert led (del-test) sin egen byte.
2. Beregn hvert led for sig og brus denne byte til at samme resultatet, falsk = 00 og sand = -1 = FFh. Beregningen af eet led kræver kun eet betingset hop.
3. Beregn det sammensatte resultat med computerens logiske funktioner OR, AND, XOR og CPL (= inverter). Resultat => A.
4. Hop betingset: OR A // JP Z,falsk

Det er omtrent sådan, BASIC og PASCAL løser testen

```
IF ((A>B) OR ((N=2) AND NOT Y)) THEN ..
```

Sådanne algorithmer giver sikre, men voluminøse resultater på assemblerniveau, de er lidt uoverskuelige. Altså tilbage til hop. På, enkle typer, som man har en chance for at lære at kende, er bedre end skræddersyede superalgorithmer, der kan spare både to og tre bytes.

Jeg vil give to typer, det er nok, for ved at vende del-testen, f. eks. (X) til (not X) kan man altid komme igennem. BOOLE's algebra kan hjælpe til at forenkle logiske udtryk, men nu til eksemplerne:

PolyZap V2.0

Bytefinder af Ole Brandt

PAGE 01

```

;Følgende program er modtaget af Ole brandt og
;skrevet ind på Anders Hejlsberg nye assembler
;PZAP, der bl.a. tillader en labelliste fra en
;anden assemblering, her NASSYS.

```

```

                                REFS    SYSEQU ;labelliste fra NASSYS
                                REF      ;dem alle !!

0000
0C80                                ORG     OC80H
0C80                                IDNT   $,$    ;load og startadresse for
0C80                                ;assemblerede fil

```

```

;Dette program finder alle adresser en bestemt
;byte findes på
;Ole Brandt, 811226

```

```

;Initier registre

```

```

0C80                                START:
0C80 310010                          LD     SP,STACK
0C83 EF                              RST   PRS
0C84 494E444C                        DB     'INDLÆS: STARTADR. BYTE-X LÆNGDE'
0CA3 0D00                            DB     CR,0
0CA5 DF63                            SCAL  ZINLIN
0CA7 DF79                            SCAL  ZRLIN
0CA9 DF60                            SCAL  ZARGS
0CAB
0CAB 7B                              LOOP: LD     A,E    ;byte der søges

```

```

;SAMMENLIGN *****

```

```

0CAC EDB1                            CPIR

```

```

;CHECK OM FÆRDIG *****

```

```

0CAE 79                              LD     A,C    ;BC=0=slut?
0CAF 80                              OR     B
0CB0 2808                            JR     Z,SLUT

```

```

;UDSKRIV ADRESSE *****

```

```

0CB2 C5                              PUSH   BC
0CB3 2B                              DEC    HL
0CB4 DF66                            SCAL  ZTBCD3
0CB6 23                              INC    HL
0CB7 C1                              POP    BC
0CB8 18F1                            JR     LOOP

```

```

0CBA                                SLUT:
0CBA EF                              RST   PRS
0CBB 0D00                            DB     CR,0
0CBD DF5B                            SCAL  ZMRET
0CBF                                END

```


PolyDos

- et disksystem.

Polydos er POLYDATA's (Anders Hejlsberg) seneste softwarepakke. Med dette system kan man betjene sig af et disksystem. Det findes i to udgaver mellem hvilke, der kan udveksles disketter. Dette ene betjener sig af NASCOM's PIO (port 4-7), mens det andet kræver FDC-kort fra Gemini (G 809).

Forskellen på de to systemer er tætheden, hvormed der lagres data på disken. På første system kan opbevares 175 K bytes (single densitet), mens der på det andet kan vælges mellem 175 K og 300 K bytes (dobbelt densitet) pr. diskette.

Når man får systemet består det af 3 dele. 1) 2K EPROM med startadresse i D000H. 2) En diskette med 13 forskellige programmer, der benyttes på forskellig måde - herom senere. 3) En engelsk manual, der giver en beskrivelse af systemet - med listning af EPROM'en.

Installerings.

Med single densitet placeres EPROM'erne i soklerne på NASCOM'en og stikket fra disken sluttes til PIO-udgangen. Disketten stikkes i disken og systemet kører! Og det kører med den sædvanlige monitor (NASSYS 1 - 3); med den sædvanlige VDU-ram; og med den sædvanlig basic-ROM. Altså et system specielt designet til Nascom-fan med alt hvad, der knytter sig til det! Du bliver ikke afskåret fra nogen mulighed, dog du kan jo ikke køre CPM - men gør det noget ????

Med dobbelt densitet skal der tilføjes et et nyt kort i bussen, hvorefter ovennævnte procedure gentages - og det kører også. (jeg har prøvet begge dele!).

Beskrivelse af systemet.

Først vil jeg lave en udskrift af diskens indhold; det gøres ved: DIR ;ELDP <NL>. DIR betyder selvfølgelig directory, mens ELDP er options man kan tilføje for at udvide kommandoen. E = udvidet, L = også de lukkede, D = også de slettede og P betyder send til printer.

```

Drive 0: PolyDos 2.0 SYSTEM
13 files in use, 1 deleted, 36 free.
91 sectors in use, 1 deleted, 1168 free.
Sect Nsec Load Exec F Name
0004 0008 C800 0000 L Exec.OV
000C 0005 C800 0000 L Emsg.OV
0011 0006 C800 0000 L Dfun.OV
0017 0001 C200 C200 L Info.IN
0018 0003 C800 0000 L Ecmd.OV
001B 0008 C800 0000 L Edit.OV
0023 0001 C800 0000 L BSfh.OV
0024 0010 B000 B000 L BSdr.BR
0034 0016 1000 1000 PZAP.GO
004A 0005 0000 0000 SYSEQU.SY
004F 0005 1000 1000 FORMAT.GO
0054 0003 1000 1000 BACKUP.GO
0057 0004 1000 1000 SZAP.GO
005B 0001 0000 0000 D TEST.TX

```

Som man ser består navne af navn op til 8 bogstaver/cifre, hvortil der er knyttet udvidelse på 2 bogstaver. Udvidelsen har betydning for arten af programmer. Systemet forstår fra fødslen følgende udvidelser TX (tekst filer), GO (maskinkode program filer), OV ('overlay' filer), BS (basic program filer), DT (basic data filer) og IN (informationsfil).

Det er muligt, selv at danne nye navne for udvidelser ved at lave en 'overlay'-fil med den ønskede effekt. Forstået således: Når du vil have udført et program, skriver du blot dets navn, systemet undersøger derefter udvidelsen og finder det overlay, der passer hertil (BSfh.OV til BS-filer osv.). Herefter indlæses overlay'et og kontrollen overgives hertil til videre ekspedition, som f.eks. kan være indlæsning af runtimepakker m.m.; til slut indlæses programmet og det udføres derefter automatisk!

Systemet tilgodeser familierelationer og kan behandle dem ens. F.eks. vil TEST betyde alle filer med dette navn på master drive. :0 vil betyde alle filer på disk 0, .GO:1 betyder så maskinkodepgr. på disk 1.

Kommandoer.

Når systemet er startet fremkommer der en \$-karakter som virker som prompt. Hvis denne slettes virker det, som du var i NASSYS. Man kan altid komme tilbage til kommando-mode, lige meget fra hvilket program, ved at trykke 'nul'-karakter (ctrl/shift/@). Her følger en liste over samtlige kommandoer, der findes på Polydos 2.0

DIR ;ELDP	Directory
COPY ;YS	Kopierer
REN ;Y	Omdøber filnavne
DEL ;Y	Sletter fil
UNDEL	Genskaber fil
LOCK ;Y	Sikre mod sletning
UNLOCK ;Y	Kan derefter slettes
PACK	Fjerner fysisk, slettede filer
SAVE	Læser til disk
LOAD	Læser fra disk
ATTRIB	Ændre Load- og Execadresser
LIST ;P	Lister fil
SKIP ;P	Blanke linier
BUFFER	Definerer RAM til COPY m.m.
NAME	Omdøber diskette
READ	Læser til memory fra sektorer
WRITE	Læser fra memory til sektorer
NEW	Ny disk kan indsættes
BOOT	Software opstart

Herudover findes to specielle

EDIT ;W	
BASIC ;W	W betyder varmstart

Her skal igen bemærkes at ; med efterstillede bogstaver er options, der ikke skal benyttes. Herefter gennemgås nogle af de vigtigste kommandoer.

COPY kopierer fra disk til disk eller fra diskette (options S) til diskette. Det kan gøres familievis, hvorefter der spørges inden hver kopiering, om det er ønskeligt, hvis man ikke har sat Y som option (d.v.s. alle på en gang).

DEL og UNDEL sletter og genindsætter filnavne. Slettede filer er ikke slettet fysisk på disken, dette gøres med PACK - og så er der ingen vej tilbage !

Det er også muligt at danne en 'command file', der vil fungere som om du selv indtastede kommandoer fra keyboard. Kan specielt benyttes, hvis man flere gange skal udføre den samme rækkefølge af opgaver, f.eks. kopiering af bestemte filer fra disk til disk eller listning af en række programmer.

Med disse kommandoer er man fuldt ud i stand til at klare alle funktioner, der er ønskelige i forbindelse med disken. Her skal tilføjes, at det er muligt ved hjælp af disknumre, at styre single/dobbel density. Disknumre 0,1,2 og 3 er dobbelt, mens disknumre 4,5,6 og 7 er single density; skønt disk 0 og 4 fysisk er identiske (1/5, 2/6 og 3/7 det samme). Man kan f.eks. COPY fra :0 til :5 og få sin fil på begge formater.

Det er muligt at 'generate' også ved Polydos. Det gøres ved at kalde en fil for "Init", der så ved bootprocessen vil blive udført. Det kan således arrangeres at starte i Basic eller Pascal eller i hvilket andet maskinkode program.

Alle fejlmeddelelser bliver givet i klart sprog også i basic. For meget enkelte findes ingen speciel fejlmeddelelse, hvorefter man får at vide: "?No message for error".

Systembeskrivelse

Systemmet bruger maksimalt fra adresse A900H til D800H ved Basic, men for alle andre fra C000H til D800H. Af disse benyttes C000H - C400H til variable og tabeller; C400H - C800H som directory buffer; C800H - D000H til overlay og fra D000H - D800H Polydos rutiner og DOS (disk operation system).

Disketten er formateret med 35 spor med hver 10 sektorer i single og 18 sektorer i dobbelt density. På disketten er der plads til 50 filer (måske skulle man hellere sige, at det er directory-buffer, der kan indeholde 50 filnavne med diverse oplysninger).

Polydos indeholder 16 rutiner til opbygning af diskbehandling. Fuldstændig listning og vejledning er indeholdt i manualen. Det er forbavsende let at kreere sine egne supplerende Overlays (f.eks. til Pascal, Naspen m.m.). Derudover inderholder det en opdatering fra NASSYS 1 til 3; en printrutine, der kan variere linier pr. side, margin i bund og venstre side. Du skal dog selv indtaste din egen printerafhængige rutine, der vil læse A-reg. til printer og vente på klar signal til næste karakter. Foruden er nyindsat mulighed for TAB ved ctrl/I, der deler skærmen over i 6 dele (tab 8).

De enkelte programmer.

Vi har nu set på selve kommandooperationer, der behandles af Exec, Dfun, Emsg og Info-filerne. Derudover er der følgende:

FORMAT. Et program der kan formatere og verificere disketter. Et godt program, hvor der er mulighed for at vælge i hvilken rækkefølge de enkelte sektorer skal formateres - så man undgår de store ventetider, når læsehovedet skal finde den næste sektor efter behandling af den nys læste. Kan vælges til 'skew faktor' 0-4, vælges 1 kommer sektorerne i rækkefølgen 0, 5, 1, 6, 2, 7, ... , 4, 9 i single density.

BACKUP. Et program der tillader hurtig kopiering af disketter. Læsning og skrivning foregår med blokke af sektorer og ikke med enkelte filer ad gangen, som f.eks. COPY :0 :1 ;Y

SZAP. Programmet bliver brugt til at undersøge og ændre indhold af diskettens indhold. Når programmet køres fås indhold af en halv sektor udskrevet i T-format med ASCII-oversættelse. Det er så muligt at rette i koden eller i ASCII-karakterne - lige som ved Modify. Uhyggeligt nemt at forandre diskindhold, men pas på at du ved, hvad du gør !

EDIT. Dette program gør brug af Overlay Edit og Ecmd. Det benyttes til at skrive tekst til assembler, basic eller som diskpen og snart til pascal. Polydata har dog lavet et specielt tekstprogram, som vil blive omtalt ved en senere lejlighed.

Det er mulig at få to forskellige cursorfunktioner til at virke i PolyEdit. Enten en blinkende som i NASSYS (du kan forresten selv bestemme din cursors udseende) eller en fast, der klemmer sig ind mellem bogstaverne, hvor tilføjelser til teksten altid sker til venstre for cursor.

Det brugbare skærmareal er udvidet med godt 6%, idet den øverste linie udnyttes på lige fod med de øvrige. Funktioner i editoren har fællestræk med NAP og BLS pascal, men der er alvorlige forbedringer. Anders har forladt ideen med de 80 karakterer på linie (man kunne jo ikke læse dem alle på en gang - irriterende) og er gået i retning af NASPEN, hvor man krøller linierne rundt. Cursorfunktioner er ens bortset fra SH/PU og SH/PN, der flytter cursor til henholdsvis første og sidste karakter på linien.

Editerings kommandoerne er ændrede til følgende:

BACK	Slet karakter før cursor
CS	Slet ord før cursor
ESC	Slet linie før cursor
LF	Fremkald slettet karakter eller ord, slettet ved fejltagelse
CH	Tabuler med 8
ENTER	Indsætter lineslut og skifter linie
CTRL/A	Skifter fra små til store bogstaver eller omvendt fra cursor og rest af linien.

Det er muligt at søge efter en bestemt tekst og få den erstattet af en anden tekst med en længde op til 255 karakterer.

Blokmærker kan sættes nemt ved CTRL/P. Første gang bliver det startmærke, hvis man gør det en gang til ændres det til slutmærke. Meget brugervenligt. De sædvanlige blokkommandoer: slet og flyt findes, samt mulighed for at slette alle blokmærker.

Tilføjlet foruden CTRL/G (der bruges som alfa-lock) er CTRL/I, der får alle tab- og CR-karakterer til at træde frem på skærmen. De slettes med samme nøgle. CTRL/SHIFT/O afslutter editeringen og skriver resultatet til disk.

Det er en særdeles god editor med mange fine egenskaber. Bl.a. står skærbilledet uhyggeligt stille, der må være lavet noget softwarestyring af billedet, således at skærbilledet bliver opdateret på de rigtige tidspunkter (men det kan jo ikke lade sig gøre - men hvordan så Anders ???)

Det er også muligt at benytte editoren i dit eget program, så kom bare i gang.

PZAP. En nykonstrueret assembler til disksystemmet. Source-koden skrives i editoren og gemmes på en fil. Når assembleren kaldes skal den følges et navn, der angiver sourcekilden, og hvis man skal gemme objekt-koden, skal der følge et navn, der skal være objekt-fil-navn.

Ved sourceskrivningen er det lige gyldigt, om man benytter store eller små bogstaver, ligeledes er det option, at sætte ":" efter labelnavne. En label er et ord der starter helt til venstre - med vilkårlig længde (anbefales til max. 7). Hvis man vil have label'en til at stå alle uden efterfølgende opcode, skulle man i NAP skrive 'EQU \$', det er i PZAP ikke nødvendig at skrive noget (se program andet sted i bladet) for at opnå det samme.

Konstanter kan udtrykkes i forskellige talsystemmer: Hexadecimal (postfix 'H'), decimal ('D' eller ' '), octal ('O' eller 'Q') eller i binær kode ('B'). Dog skal alle tal starte med et ciffer mellem 0 og 9 - for at skelne mellem labels, der skal starte med et bogstav eller en specialkarakter.

Der kan regnes med udtryk, der udregnes i 16-bits værdier uden mente. Der kan benyttes en række operatører ved udregningerne. De er nævnt med øverste prioritet først: HIGH og LOW, der isolerer mest og mindst betyden byte, + og -, fortegn, * og /, multiplikation og division, MOD eller \, der finder modulo, SHL og SHR, der skifter bit i tallet mod venstre og højre, + og -, addition og subtraktion, alle relationer enten udtrykt som i basic eller med engelske ordforkortelser (EQ, NE osv.), NOT, AND, OR og sidst XOR.

Pseudo operatører findes i et utal, da assembleren gerne skulle sælge også på det engelske marked. Derfor findes den danske NAP's PO samt den engelske ZEAP's PO, så kan man jo vælge, hvad man er vant til at bruge. (F.eks. DB/DEFB, DW/DEFW, DS/DEFS, DB/DEFM)

Af ganske normale findes selvfølgelig ORG, EQU og END. Derudover er følgende:

IDNT	Definerer load- og startadresse for objektfil
SET	Som EQU, dog opstår ingen fejl, hvis label er brugt før
DEFL	Som SET
REFS	Giver besked om at indlæse en bestemt symbolfil f.eks. SYSEQU, der indeholder alle labels fra NASSYS og Polydos.
DC	Sætter bit 7 høj i sidste byte af en karakterstreng
IF eller COND	Hvis udtrykket efter IF er falsk, springes til
ELSE	eller hvis det mangler til
ENDIF eller ENDC	i modsat fald gøres det modsatte.

Der findes en del options til assembleren, der afgør, om den skal listes (VDU/printer), udskrive symboltabel, danne en ny symbolfil, udskrive hele objekt-koden ved DB-sætninger (mod sædvanlig 4), overspringelse af ikke udførte sætninger i IF-sætninger samt mulighed for at tvinge assemblering igennem trods fejl.

Ved assemblering rapporteres løbende fejl i 1. gennemløb ved at linien vises, og hvilken fejl der var årsagen. Hele 1. gennemløb fortsættes, mens anden gennemløb standses, hvorefter der kommer meddelelse om, at der ikke er dannet nogen objektfil.

BASIC Basicprogrammer gør brug af følgende filer: Emsg.OV (der altid er tilstede i RAM under udførelse af program - de udvidede fejlmeddelelser), BSfh.OV og BSdr.BR, der er en runtimepakke (udvidelse) til basic.

Det er muligt at skrive sine basicprogrammer i editoren, hvor det er nemt at skrive til de maximale 72 karakterer; det eneste man skal huske er, at udvidelsen skal være '.BS'.

I runtimepakken ligger rutiner til betjening af disken fra basic. I den direkte mode fra keyboard er der følgende:

LOAD Indlæser et basicprogram fra disk
SAVE Gemmer program på disk i maskinkode
SAVET Gemmer program som .TX (tekstfil, der kan editeres i EDIT)
EXEC Indlæser og udfører et program
AUTO Automatisk linienummerering
REN Omnummererer alle linienr.
FIND Finder strenge i programmet.

Global kommandoer kan benyttes i programmer og direkte fra keyboard. Disse kan deles i fire områder: 1) data ind/ud, 2) Program filbehandling, 3) Printerkontrol og 4) diverse.

DATA IND/UD Det er det fornemste og vigtigste område i hele Polydos, da man kan få adgang til datafiler - både som 'lager med direkte adgang - random fil' og som 'lager med adgang i rækkefølge - sekventiel fil'.

I sekventiel fil er hver post en streng af ASCII-værdier, der afsluttes med et CR. Mens random fil kan defineres til at indeholde enhver kombination af heltal, reelle tal og strenge med forudvalgt længde.

I stedet for i programmet at benævne filen med et navn, tilknytter man en enhed (fra 0 - 3), som betegnelse for filen. Selve processen med at åbne, finde og lukke datafiler virker overordenlig god og nem (i modsætning til f.eks. Regnecentralens - der for øvrigt kun tilbyder random filer). Der er tilknyttet nogle systemvariable, der fortæller om tilstanden af datafil, dimension, end-of-file m.m. Det er muligt i randomfiler at flytte pegepinden, så du kan finde lige det du ønsker. Det er ligeledes muligt at indlæse det antal variable ad gangen, der er ønskeligt, blot skal du huske at variabelen og det indlæste skal passe sammen.

PROGRAM FIL Der er to instruktioner, der kan benyttes til indlæsning af andre programmer fra et arbejdende program. SETLOAD bruges fortrinsvis til indlæsning af maskinkodeprogrammer, der skal bruges under programafvikling. SETCHAIN vil indlæse og udføre et basicprogram uden at forrige programs variableværdier SLETTES !!

DIVERSE Printeren kan slukkes eller tændes med følgende ordre: SETPRON og SETPROFF. SETERR nnnn vil dirigere programpilen til et bestemt linienummer, hvor man da kan afgøre, om programmet skal standses eller om programmet kan fortsættes - trods fejl. SETREAD er en helt vidunderlig editeringsfacilitet under programafvikling. Du kan rette i en strengvariabel, dvs. indsætte og slette med NASSYS pilene, men kun i den oprindelige længde. Editeringen afsluttes med <ENTER> og programmet fortsætter. SETCLEAR vil på lignende måde som CLEAR slettes alle programvariable, men det er muligt med denne ordre foruden at sætte strengareal også at ændre memorytop for basic'en.

FINAL.

Det er en meget god programpakke, der serveres fra Poly Data. Den er helt støbt og gennemtænkt, og det man eventuelt savner, kan man selv udbygge. Det skyldes den gode og instruktive manual (der dog kun foreligger på engelsk), som indeholder alle relevante oplysninger.

Prisen på nuværende tidspunkt er 2000 kr., men det er absolut en rimelig pris, når man tager omfanget i betragtning: 4K udvidelse til basic, en assembler, en editor, en superzap, to mindre hjælpeprogrammer (små, men effektive) og så selvfølgelig selve PolyDos 2.0

Asbjørn Lind

24.1.82

PolyZap V2.0

Orgel af Erling Thorup Madsen

PAGE 01

OC01	KMAP	REFS REF	SYSEQU
1000		ORG	1000H
1000		IDNT	,\$,\$
1000	START		
1000	21010C	LD	HL,KMAP
1003	3E02	LD	A,2
1005	0E00	LD	C,0
1007	ED79	OUT	(C),A
1009	3E00	LD	A,0
100B	ED79	OUT	(C),A
100D	0608	LD	B,8
100F	A1		
100F	DB00	IN	A,(0)
1011	2F	CPL	
1012	77	LD	(HL),A
1013	23	INC	HL
1014	3E01	LD	A,1
1016	ED79	OUT	(C),A
1018	3E00	LD	A,0
101A	ED79	OUT	(C),A
101C	10F1	DJNZ	A1
101E	DD21010C	LD	IX,KMAP
1022	110502	LD	DE,205H ;/
1025	DDCB064E	BIT	1,(IX+6)
1029	C29310	JP	NZ,TONE
102C	111502	LD	DE,215H ;.
102F	DDCB054E	BIT	1,(IX+5)
1033	C29310	JP	NZ,TONE
1036	113C02	LD	DE,23CH ;,
1039	DDCB044E	BIT	1,(IX+4)
103D	C29310	JP	NZ,TONE
1040	116402	LD	DE,264H ;M
1043	DDCB034E	BIT	1,(IX+3)
1047	C29310	JP	NZ,TONE
104A	117D02	LD	DE,27DH ;N
104D	DDCB024E	BIT	1,(IX+2)
1051	C29310	JP	NZ,TONE
1054	11AA02	LD	DE,2AAH ;B
1057	DDCB014E	BIT	1,(IX+1)
105B	C29310	JP	NZ,TONE
105E	11E102	LD	DE,2E1H ;V
1061	DDCB074E	BIT	1,(IX+7)
1065	C29310	JP	NZ,TONE
1068	113E03	LD	DE,33EH ;C
106B	DDCB075E	BIT	3,(IX+7)
106F	C29310	JP	NZ,TONE
1072	118103	LD	DE,381H ;X
1075	DDCB0166	BIT	4,(IX+1)
1079	C29310	JP	NZ,TONE
107C	11FF03	LD	DE,3FFH ;Z
107F	DDCB0266	BIT	4,(IX+2)
1083	C29310	JP	NZ,TONE
1086	112106	LD	DE,621H ;SPACE
1089	DDCB0766	BIT	4,(IX+7)
108D	C29310	JP	NZ,TONE

PolyZap V2.0

Orgel af Erling Thorup Madsen

PAGE 02

```

1090 C30010          JP      START
1093                TONE
1093 D5             PUSH    DE
1094 0E00           LD      C,0
1096 0610           LD      B,10H
1098 ED41           OUT     (C),B
109A                B1
109A 1D             DEC     E
109B 20FD           JR      NZ,B1
109D 15             DEC     D
109E 20FA           JR      NZ,B1
10A0 0600           LD      B,0
10A2 ED41           OUT     (C),B
10A4 D1             POP     DE
10A5                B2
10A5 1D             DEC     E
10A6 20FD           JR      NZ,B2
10A8 15             DEC     D
10A9 20FA           JR      NZ,B2
10AB C30010          JP      START

10AE                END

```

Dette program er taget fra IMNC 3
 Skrevet af David Reddington

Programmet giver en nem måde at gemme strengvariable på bånd.
 Problemmet klares igen ved at 'liste' til bånd, som man også
 gør ved subrutiner.

Et typisk demo-program vil se således ud:

```

10 REM PROGRAM FOR CSAVE X,Y,Z,T$ OG A$(100)
20 DEVICE=3187:ZCRT=1919:ZSRLX=1915
30 PRINT "START BANDOPTAGER 'OPTAGELSE'"
40 INPUT "TRYK <RETURN>";A$:PRINT "VENT !!"
50 DOKE DEVICE,ZSRLX
60 PRINT X;Y;Z;
70 PRINT T$;
80 FOR N=1 TO 100
90 PRINT A$(N);
100 NEXT
110 DOKE DEVICE,ZCRT
120 PRINT "DATA GEMT PA BÅND - SLUK BANDOPTAGER"
For at indlæse variablene igen skal følgende udføres:

180 PRINT "TÆND BANDOPTAGER FOR INDLÆSNING AF DATA"
190 INPUT "KLAR TRYK <ENTER>";A$
200 INPUT X,Y,Z
210 INPUT T$
220 FOR A=1 TO 100
230 INPUT A$(A)
240 NEXT
250 PRINT "DATA INDLÆST"

```

Referat fra kursus om NASSYS monitor.

Forklaring af kommandoer.

A Arithmetic kommando. EKS. A 1200 1220

Denne kommando udregner sum, diff. og relativ tal til brug ved relativ jump.

B Breakpoint kommando. EKS. B 2000

Denne kommando stopper programmet, når den møder den adresse, man har sat ind efter B xxxx. Den virker på samme måde som RST 20. RST 20 kan bruges i et program under test som automatisk breakpoint. Virkemåde: Når man kalder kommandoen bliver den byte der står på adressen gemt i breakpointbuffer og RST 20 (E7) bliver indsat på den plads

C Copy kommando. EKS. C 2000 2200 100

Denne kommando bruges til at flytte byte i memory.

ARG 1: start adresse

ARG 2: ny startadresse

ARG 3: hvor mange byte der skal flyttes.

NB. denne kommando har en god ting, den kan bruges til af slette eller indsætte en bestemt værdi i et område af memory.

EKS. C 3000 3001 5FF

3000 = startadresse

3001 = startadresse +1

5FF = længden af området-1. område = 600.

Indsæt først den ønskede værdi i startadressen med M kommando.

E Execute kommando EKS. E 2000 ARG 2 ARG 3

Denne kommando starter et program der begynder i ADR. 2000

G Generate kommando. EKS. G 2000 2600 2100

Denne kommando overfører et program til tape, med automatisk start. 2000 = startadresse på programmet. 2600 = slutadresse på programmet. 2100 = den adresse hvor man vil starte programmet fra.

I Intelligent copy kommando

Denne kommando virker på samme måde som C kommandoen, med den fordel, at den ikke overskriver den del af programmet, den skal kopiere fra.

J Kold start af BASIC og PASCAL.M Modify (memory) kommando

EKS. M 2000

Den byte, der er i adresse 2000, vil komme op på skærmen.

EKS. M 2000 2000 AF 24 46 57 68 68 36 45 . Nu vil adresse 2000 have en værdi på AF og adresse 2007 vil have en værdi på 45.

. = Gå tilbage til kommandomode. : = Gå en byte tilbage. / = Gå til ny adresse. EKS. /3000 Gå til ADR. 3000. Hvis man vil have en tekst ind i memory kan det gøres på følgende måde.

M 2000 ,N,A,S,C,O,M

Nu vil der stå NASCOM fra adresse 2000.

O Output kommando. EKS. O Outputdata Portadresse. O 67
5 Kommandoen vil sende byten 67 til PORT 5.

Q Input kommando. EKS. Q PORTnr.

Q 6 . Kommandoen vil læse data fra port 6 til A reg., og vise det på skærmen.

S single step kommando. EKS. S 2000

Maskinen vil udføre en instruktion af gangen, og derefter vise alle registre på skærmen.

R Read kommando EKS. R

Efter R og new line lyser read lysdioden, og maskinen er klar til at modtage data fra tape eller modem.

Hvis man taster R 2000, flytter man de data man modtager med 2000 byte.

EKS. R 2000
Man modtager normalt data fra 1500, og maskinen vil skrive 1500 0010 . , men programmet placeres i adresse 3500 og opefter.

T Tabulate kommando. EKS. T 2000 2500 E WW HHAA

2000 = Startadresse

2500 = Slutadresse

E = Antal af linier der skal være på en side

WW = Antal af byte på hver linie +8.

WW = 5 vil der være 13 byte på hver linie.

HH = Hvis HH=0 skrives hex-delen paa skærmen.

AA = hvis AA=0 skrives ASCII paa skærmen.

EKS. T 1000 1100 E O 100

Ved danne kommando skrives der kun ASCII på skærmen.

W Write kommando. EKS W 3000 4000

3000 = startadresse

4000 = slutadresse.

Ved denne kommando overføres byte 3FFF som den sidste byte.

V Verify kommando. Denne kommando læser tape, men data bliver ikke overført til memory. Dette er en kontrol af data, der er overført til tape.

Restart 10H. Relative call.

Denne rutine kalder en subrutine med en relativ adresse, som er placeret i den efterfølgende byte. Det er ikke en Z80 kommando, men laves af monitoren på følgende måde.

EKS. Dette program er ikke det samme som er i monitoren, men virkemåden er den samme, og dette er lettere at forklare.

```
RCAL:  PUSH  HL
        PUSH  DE
        PUSH  AF
        LD   HL,0006H
        ADD  HL,SP
        LD   E,(HL)
        INC  HL
        LD   D,(HL)
        LD   A,(DE)
        INC  DE
        LD   (HL),D
        DEC  HL
        LD   (HL),E
        LD   L,A
        ADD  A,A
        SBC  A,A
        LD   H,A
        ADD  HL,DE
        POP  AF
        POP  DE
        EX   (SP),HL
        RET
```

Forklaring!

Først gemmes HL, DE, AF på stakken. Nu loades HL med 6. Derefter forøges stackpointer med 6.

Stackpointer adresser

```
adr.  FFF + FFE  Returadresse for subcall.
        FFD + FFC  HL REG. data
        FFB + FFA  DE  -----
        FF9 + FF8  AF  -----
        FF8                Næste stack adresse
```

Stackpointer peger på FF8 inden man ADD HL,SP og efter er HL REG. =FFE =Returadressen. Derefter overføres returadressen til DE REG. Denne returadresse er ikke brugbar, da det er det offset, man skal springe. Dette loades ind i REG. A. REG. DE INC med en og giver den nye returadresse, som derefter gemmes i adresse FFF og FFE. Derefter gemmes offsetværdien i REG. L Linie ADD A,A og SBC A,A undersøger om tallet er over eller under 127, eller om man skal springe frem eller tilbage. Det sker på følgende måde. Hvis bit 7 i REG. A er 0 bliver summen ikke over 256H, og SBC giver 00. Hvis bit 7 i REG. A er 1 bliver summen over 256, og der bliver sat et carry flag, når man SBC dette tal giver det FF. Dette tal gemmes i REG. H som det mest betydende tal. Nu ADD REG. DE til HL og HL har adressen på rutinen, som skal kaldes.

HVIS BIT 7 = 0 REG. A = 0
 HVIS BIT 7 = 1 REG. A = FF

Efter ADD

REG. A	REG. HL
er = 0 - 7	0 - 7F
er = 80 - FF	FF80 - FFFF

Efter SBC

Nu er der kun at give AF og DE REG. de gamle værdier igen, og derefter ombytte den nye værdi i HL med den gamle og det sker i linien EX (SP),HL, da SP REG. peger på indholdet af REG. HL i stacken. Nu springer den tilbage, og den adresse den springer til er den nye offsetadresse, og her udfører den subrutinen og næste gang den møder en RET kommando springer den tilbage til sin næste adresse i programmet som kaldte dette RCAL.I

SCAL SUBROUTINE DF 5BH.

Denne rutine kalder en subrutine. Adressen er i den næste byte. Denne rutine virker næsten som RCAL.

RCAL	SCAL
LD (HL),D	
DEC HL	
LD (HL),E	

LD L,A	LD E,A
ADD A,A	LD D,D
SBC A,A	LD HL,(STAB)
LD H,A	ADD HL,DE
ADD HL,DE	LD E,HL
	INC HL
	LD D,(HL)
	EX DE,HL

POP AF	POP AF
POP DE	POP DE

De rutiner, der er i NASSYS-3, har adressen fra 41H til 7FH, og adressen fra 41H til 5AH kan kaldes fra keyboard. Det er det samme, som de kommandoer der gives. F.eks. 43H er COPY kommandoen. Hvis man kalder SCAL 'M' i et program vil man kalde MODIFY kommandoen. VIRKEMÅDE! Nummeret på den rutine loades ind i REG DE, og HL loades med adressen på STAB. STAB er startadressen på tabellen, og da der skal 2 byte til en adresse vil den første brugbare adresse være STAB + 41*2.

STAB = 700H
 +2*41 = 82H
 TABST = 782H

Derefter ADD DE til HL REG, og HL har nu den værdi, hvor den rutine begynder. Denne adresse gemmes i DE REG., og derefter ombyttes DE,HL REG. Nu har REG. HL værdien af startadressen, og AF DE tages tilbage, samt REG. SP loades med den nye adresse, og subrutinen udføres.

PER THOMSEN.

NASCOM

did

Styring af en DC motor til en Casetteoptager.

Nascom Nyt behøver ikke kun at være fyldt med programmer. Vi ser gerne at vore medlemmer kommer med små nyttige diagrammer.

Jeg har denne gang valgt, at vise, hvordan man holder en konstant hastighed på capstanmotoren i en cassetteoptager.

Det kan laves på forskellige måder: Man kunne forestille sig, at en kontakt afbrydes når hastigheden er kommet op på den ønskede. Dette system ses i fjernskrivere. Det er et udemærket princip. Hastigheden justeres ved hjælp af en stilleskrue, og startmomentet er meget stort, således at motoren lynhurtigt når op på den rigtige hastighed. Man kunne også sætte en lille dynamo på selve motorakslen, og lave en tilbagestyring fra den spænding dette tacsystem giver.

På den cassetteoptager jeg har, sidder der en skive på selve capstanakslen. Denne skive er fyldt med fine sorte streger hele vejen rundt. På den ene side sidder en infrarød lyssender, mens der på den modsatte side sidder en fotodiodemodtager.

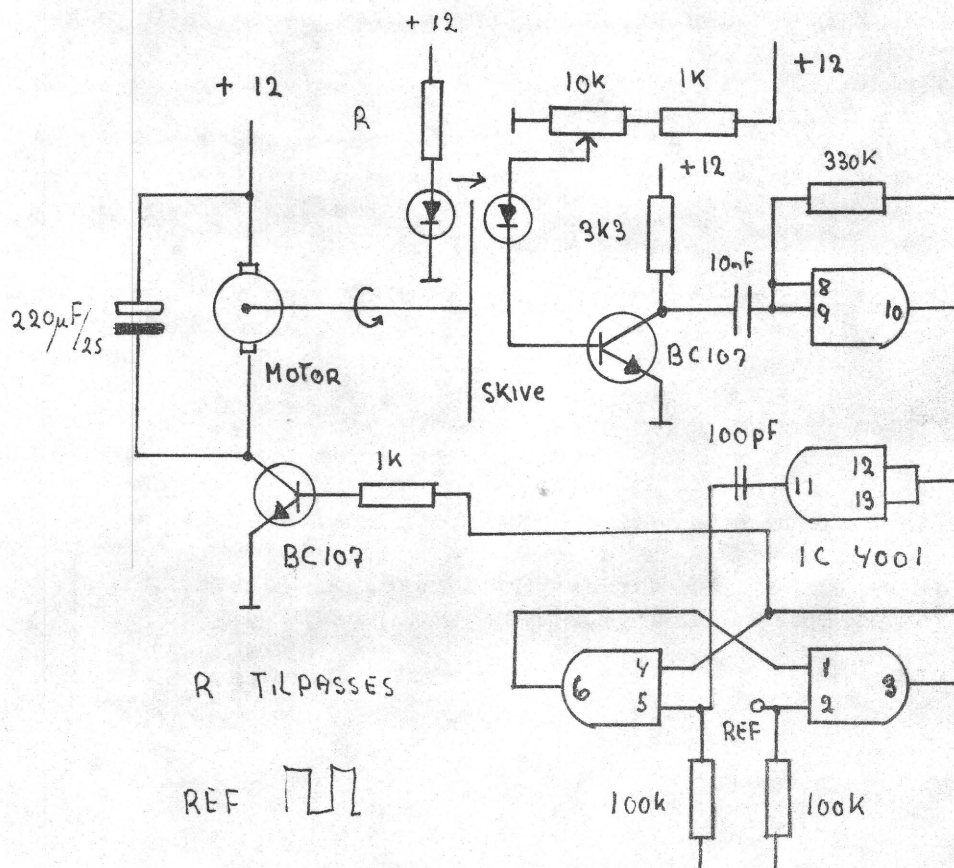
Ideen er nu følgende: Når motoren begynder at køre rundt vil der på fotodioden opstå nogle pulser. Disse impulser forstærkes op i en almindelig transistor og efter endnu lidt forstærkning tilføres disse impulser til en flip-flop. Denne flip-flop tilføres også impulser, den såkaldte reference. Udgangen fra denne flip-flop føres over i en transistor, hvis kollektor sidder i serie med selve motoren. Hvis nu alt er rigtigt forbundet, vil der ske følgende:

Drivetransistoren til motoren er helt åben. Motoren vil lynhurtigt gå op i hastighed, men nu vil flip-floppen jo modtage et sæt impulser på den anden indgang, og dette vil i princippet stoppe motoren. Nu da motoren går ned i hastighed bevirker at der opstår en ligevægt mellem reference og pulser fra fotosystemet. Og motoren vil indstille sig på en bestemt hastighed.

De impulser motoren arbejder med glattes ud ved hjælp af en passende stor kondensator, der ligger over motoren. Man kan stille og roligt styre motoren ved hjælp af en tonegenerator, som man så senere kan erstatte med en fast reference.

Diagrammet skulle nok tale for sig selv.

OH



Så blev Nascom færdigbygget, hvad nu?

Jeg får som teknisk redaktør en masse opringninger. Det er jeg naturligvis glad for, og jeg har da også tænkt meget over, hvad der ligger bag mange af spørgsmålene. Det er nu ikke kun gennem telefonsamtalerne, man får det indtryk. Nej som radioamatør træffer man også mange nye Nascombrugere, og jeg må så sandelig sige, at her bliver der hjulpet, så det batter. Sagen er jo den, at der har vi jo mulighed for, at få spørgeren til at afprøve sit anlæg næsten med det samme. Det er jo oftest sådan, at computeren er bygget sammen med radioanlæg og modem. Den måde som en nybegynder starter på er typisk og efter min egen opfattelse meget normal:

Nu virker maskinen og nu skal den til at tænke for mig?

Gør Dig helt klart, at maskinen ikke tænker, og at den i det hele taget er dum til alt, hvis de oplysninger den får er forkerte.

Hvis man derimod giver den rigtige oplysninger, er den overmåde hurtig til at løse opgaven, men det er stadig ikke intelligens.

Jeg vil tro, der er et stort behov for en lille beskrivelse af computeren. En lille bog som fra starten af begynder med de helt små ting, som er så vigtige for at få den til at funktionere?

Jeg kan nu selv se, at jeg fra starten har grebet det hele forkert an. Skulle jeg idag gøre status, ville jeg starte på en helt anden måde, men en gammel hest er nu glad for sin stald.

Jeg ville nok starte med en Nascom med kun et par kilo ram. Den kan naturligvis ikke køre BASIC, men man kan lære de mest elementære ting i maskinsprog, og det er så sandelig ikke værst, selv om det i starten virker meget tørt og trist.

Er man først helt inde i de funktioner, har man en klar opfattelse af hvilke muligheder man har.

Jeg vil benytte denne side til lidt reklame for Nascom. Jeg har nemlig det indtryk, at mange ryster lidt på hovedet, når den omtales. Der findes et uhyre antal forskellige computere på markedet, men hvor mange kan, for de samme penge, gøre det samme som Nascom?

Det er mest de drevne computerfolk, der er afvisende over for den, men jeg har nu adskillige eksempler på, at den alligevel er brugt som udviklingscomputer. Grunden er ganske simpelt den, at det er et byggesæt og alle vigtige funktioner er tilgængelige.

Nascom 2 kan også bringes til at samarbejde med nogle af de andre systemer, og jeg ved, at nogle er igang med den opgave.

Jeg kunne tænke mig, at en kreds af os fik lavet en beskrivelse af vores Nascom I kunne jo skrive en lille beskrivelse, hvori der står hvilke besværligheder, der var, da computeren blev startet op, og så ville jeg sammen med et par andre lave et sammenkog af alle disse erfaringer.

Selv om I har valgt mig som teknisk redaktør, må I ikke forvente, at jeg kan klare at besvare alle forespørgelser. Jeg er ikke programmør, og bliver det næppe, men ved hjælp af gode venners råd og en varm loddekolbe, kan man nu godt komme langt.

Jeg skriver ikke et ord om noget, uden jeg selv har prøvet de instruktioner, der skulle til for at få det til at virke, og det er så sandelig noget, man kan lære noget af.

Vist ved vi alle lidt om hexadecimale tal, men prøv selv at nedfælde det på et stykke papir. Det er sandelig ikke nemt.

Jeg håber at disse tanker, jeg her har nedskrevet, vil blive ført ud i noget praktisk, og venter spændt på nogle tilbagemeldinger.

Ole H.

En lille indføring i det hexadecimale system.

Denne artikel er blevet til, fordi jeg ikke er videre stiv i det hexadecimale talsystem. Skal man finde fejl i sin computer, er man tvunget til at sætte sig ind i det. I det decimale talsystem tælles til 10 og hertil lægges så menten. Det hexadecimale tæller til 15 på følgende måde: 123456789ABCDEF.

En almindelig tideler f.eks 7490 tæller til 9 i binær form, den kendte BCD code.

8 4 2 1	
0 0 0 0 = 0	
0 0 0 1 = 1	Bogstaverne A B C D E F har værdierne 10 11 12 13 14 15.
0 0 1 0 = 2	1+F= 10, hvilket er 16
0 0 1 1 = 3	F+F= 1E, hvilket er 30
0 1 0 0 = 4	
0 1 0 1 = 5	Du lægger mærke til at menten i de hexadecimale system først optræder ved springet 16.
0 1 1 0 = 6	
0 1 1 1 = 7	
1 0 0 0 = 8	I tabellen vil du også lægge mærke til, at jeg har skrevet 1 2 4 8.
1 0 0 1 = 9	
1 0 1 0 = A	Disse tal vil gælde hele vejen ned så lang tid du regner med binære tal.
1 0 1 1 = B	
1 1 0 0 = C	Binært 1111 = 15 og F i hexadecimal værdi.
1 1 0 1 = D	
1 1 1 0 = E	En 8 bits microprocessor kan altså adressere op til 64 K eller helt præcis: 1+2+4+8+16+32+64+128+256+512+1024+2048+4096+8192+16384+32768= 65536.
1 1 1 1 = F	

Husk dog lige på at et 64K ramkort ikke kan bruges i bunden, når du bruger en Nascom, idet monitoren ligger der, men køb det alligevel. Kloge folk siger, at det skaa nok blive klaret en dag.

Den højeste adresse, du kan komme op på, er altså FFFF. Det er normalt nok, men der er muligheder, hvis man skal have lagerplads, men det vil jeg ikke komme ind på her.

Derimod vil jeg vise, hvordan man kontrollerer om de data, man har i sin EProm nu også kommer over i ramlageret, når ens prombrænder skal afprøves. Det kan nemlig godt lade sig gøre uden at have nogen IC i. En 2708 eller 2716 har 8 udgange. Er de alle høje: Altså 1 1 1 1 1 1 1 1 vil man ved en copiering over i ram se, at der står FF Hvorfor nu det? Jo 1111 i binær =15 som er F i hexa.

Lad os se på en 2708: Vi har benene 9 10 11 13 14 15 16 17 Ben 12 er stel.

Nedenunder skriver jeg i binær form 1 1 1 1 1 1 1 1

Deler dem i to blokke a fire.

Altså kommer tabellen til at se sådan ud: 1 1 1 1 1 1 1 1 = F F

0 1 1 1 1 1 1 1 = 7 F

0 svarer til en stel til ben 12.

1 0 1 1 1 1 1 1 = B F

1 1 0 1 1 1 1 1 = D F

1 1 1 0 1 1 1 1 = E F

1 1 1 1 0 1 1 1 = F 7

Du vil lægge mærke til, at der i det

1 1 1 1 1 0 1 1 = F B

hexadecimale system ikke optræder

1 1 1 1 1 1 0 1 = F D

noget bogstav efter F. Prøv at lave

1 1 1 1 1 1 1 0 = F E

nogle eksempler selv, så lærer man det faktisk ret hurtigt.

Jeg vil i nogle kommende artikler omtale PIO, og hvordan man får noget ud og ind i den. I det hele taget kan jeg på de mange telefonsamtaler, der kommer, mærke, at der er et meget stort behov for at få nogen mere viden om betjening af Nascom. Lad mig få nogle reaktioner, så jeg kan komme med nogle svar.

MEDDELELSER FRA PROGRAMBIBLIOTEKET.

RETTELSE TIL LISTEN OVER PROGRAMBIBLIOTEKET FRA 1-SEP-1981.

Først og fremmest skal jeg gøre nye medlemmer opmærksom på, at der i Nascom Nyt fra december 1981 (nr. 10/81) er et rettelserblad til ovennævnte liste. Sørg for at få indført rettelserne i din liste så misforståelser undgås ved bestillinger.

I omtalte rettelserblad manglede der desværre to rettelser, så de kommer her:

**SIDE 11: ØVERSTE LINIE I PB1-B:
"SYS 1" skal være "SYS 1-3".
SAMME RETTELSE VED PB3.**

DIVERSE TIPS OG NYHEDER.

Af Jesper Skavin.

NY SERVICE FOR MEDLEMMER MED POLYDOS EL. MIDICOS.

Det vil fra næste måned være muligt at købe software fra programbiblioteket leveret på disk (Polydos 2 system) eller minidigitalkassette (Midicos system). Foruden de i biblioteket eksisterende programmer, vil NAP-teksten til de fleste maskinkodeprogrammer blive tilgængelig. Liste over software og tilhørende priser kommer i næste blad. Anmeldelse af Midicos systemet er på trapperne. Yderligere oplysninger hos næstformanden.

SPACE INVADERS. Hvis du har programmet "Space Invaders (M9)" og spiller det jævnligt, vil du også vide hvor irriterende det er, når man kun mangler tre invaders og telefonen pludselig ringer. Tast "P" og programmet standser øjeblikkeligt. Enhver tast herefter får spillet til at fortsætte.

BLS PASCAL har nu været fremme i et års tid og diverse forbedringer har medført, at der har været tre versioner på markedet. Disse er version 1.1, 1.2 og 1.3. (Version 1.0 var en "prototype" med en del fejl, og forhåbentlig er der ikke nogen, der kører med den). Den nyeste af disse er version 1.3 og den adskiller sig en del fra de to andre. Bl.a. er keyboardrutinen lavet om, hvilket desværre har medført at programmer skrevet under 1.1 & 1.2 i visse tilfælde ikke udføres korrekt under version 1.3. Det er nu muligt at indtaste hex-tal ved "read" og <ENTER> alene accepteres, hvilket bevirker at den pågældende variabel beholder sin værdi (som i BASIC). Runtimepakken er kopt ned så dens slutadresse er 2140H (E140H for ROM-versionen). Compileren starter nu ved 29C0H (E9C0H). Den er i øvrigt lavet om, så den ikke skelner mellem store og små bogstaver i identifiere. Udover de her nævnte ændringer er der en del mindre. Yderligere oplysninger hos Poly-Data. (BLS PASCAL V1.3 markedsføres i England under navnet NASCOM PASCAL).

NASCOM 3 er ikke en ny computer fra Lucas, som man kunne få indtryk af i diverse engelske blade, men derimod en ganske almindelig NASCOM 2 med 64K RAM anbragt i en plastickasse og med to stk. floppydisk incl. controller og strømforsyning. Den vil efter sigende ikke blive markedsført i Danmark.

POLYSYS 4 er en ny 2K udvidelse til NASCOM ROM BASIC. Den har alle kommandoerne fra Polysys 2 og 3 plus selvfølgelig en hel del nyt. Bl.a. er RENUMBER lavet om, så i stedet for at renummere hele basicprogrammet, kan man specificere hvilke linier, der skal renummereres til ny linienumre. UNNEW ophæver en utilsigtet NEW eller CLOAD, og TRACE kan udskrive variabelværdier. Derudover er der printerrutiner til såvel seriel som parallel printer, hvor man kan vælge mellem udskrift til printer alene eller printer og skærm. Men det mest interessante ved Polysys 4 er, at den leveres med en relokerings-rutine som gør, at man selv kan bestemme, hvor i lageret man ønsker Polysys 4 placeret, eller man kan lægge den i EPROM. En decideret anmeldelse er på vej; i mellemtiden fås yderligere oplysninger hos Poly-Data.

Privatundervisning i Basic, Pascal og assembler gives. Gennemgang og skrivning af forskellige programmer, gerne efter eget ønske, alt på Nascom med Epson printer.

Henvendelse:

E. Th. Madsen

(02) 42 49 97

Pris: 65,- kr. pr time.

SÆLGES:

brugt blæser 220 V. (110 mm Ø)	50.00 kr.
25 stk. 24 V relæ 2skift a'	5.00 kr.
IC sokler 8 ben, 14 ben, 16 ben, 24 ben	
til	1.00 2.00 2.50 4.50 kr.
DIP stik	2.50 kr.
Lysdioder rød/grøn	1.00/1.50 kr.
2N 3055	5.00 kr.
BC 547 / BC 557	1.00 kr.
IC 555	6.50 kr.
IC 741	5.00 kr.
MANGE andre typer transistorer	
Diode 1N 4148	0.25 kr.
Diode 4005	1.00 kr.
Kredse i 74xx	
Kredse i 4xxx	
24 leder fladkabel pr. meter	25.00 kr.
Printspyd med hunstik	0.15 kr.
Komponetpakke til M.K. EPROM-brænder uden 24 bens sokkel	60.00 kr.
Regulator 78xx og 79xx	11.00 kr.
Krystalfilter 10,7 MHz	
Nascom 1, Ram- og bufferkort	

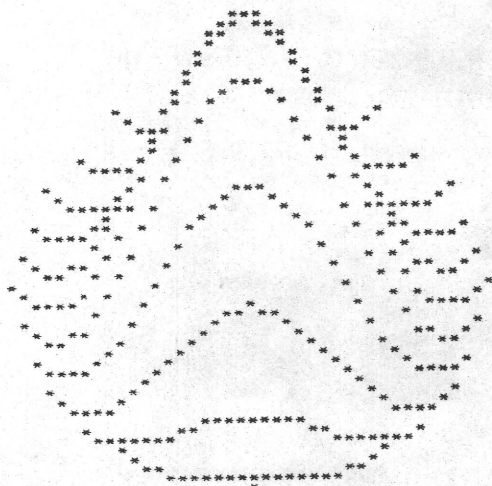
KØBES

Netdel til NASCOM

HUSK

Jeg laver print efter opgave og har print til sneplov og EPROMbrænder.

Henvendelse til
Hans Ole Groth
09 58 16 03



TEAC FD 50A - uP 765 U1

Kort hardwarebeskrivelse:

Systemet er bygget op på et NASBUS-compatible prototypekort (Winchester). Alle forbindelser er wirewrapped. Ved hver IC anbringes en afkoblingskondensator over +5/GND.

Interfacet til Floppy-Drivet er beregnet til TEAC FD 50 A, men kan forbeholdsvist enkelt ændres.

Interfacet til NASBUS'en er buffered. Der bruges 3 porte: 16, 17, og 18 (decimalt). Port 16 = FDC-kontrol, 17 = DFC-data og 18 = FDD-moter/FDC-reset/FDC-terminate - et bit til hver funktion.

FDC (Floppy disk controller) er en NEC 765, der bare 'kan det hele'. Forhandler: MER-EL A/S, Ved Klædebo 18, 2770 Hørsholm. Pris incl. moms 410,- kr. (1.7.1981).

FDD'et (Floppy disk drive) er et TEAC FD 50 a, single side, double density. Kapacitet ca. 92 Kbyte pr. disketteside (184 ialt pr. diskette) ved single density. Forhandler: DANBIT A/S, Plantagevej 23, 2860 Solrød Strand. Pris incl. moms 2318,- kr. (1.7.1981).

Systemet kan hardwaremæssigt klare 4 FDD's, double-side/single density (DS/SD) - double density kræver ekstra hardware. FDD'et kræver: +12 volt/800 mA og +5 volt/350 mA
FDC-kort kræver: +5 volt/300 mA.

Kort beskrivelse af software:

DOS'en fylder 4K og kan leveres på diskette med bootstrap-loader på tape, eller som 2 stk 2716 EPROM.

Følgende kommandoer findes:

A(llocation-map display)	- Oversigt over forbrugt plads
B(ackup)	- Overførel af filer mellem 2 disketter
C(opy)	- Dan kopi af fil
D(isplay)	- 'Tabulering' af data på diskette
E(xecute)	- Indlæs fil og udfør
F(ormater)	- Initierer ny diskette
I(ndex)	- Display filer på diskette
K(eep)	- Udskriv fil til diskette
L(oad)	- Indlæs fil fra diskette
N(ewname)	- Ændre filnavn
O(rganize)	- Komprimer diskette
P(urge)	- Slet fil

Alle kommandoer består af en masterspace '@' efterfulgt af en et-bytes kommando (A-P) samt diverse parametre. Filnavn kan være 1-8 karakterer. Følgende filtyper kendes: Basic, Nap, Pascal, ? og Forth.

Der findes rutiner til kald fra bruger-programmer til læsning/skrivning af random/seriel filer (rutinerne er dog ikke på nuværende 100 % færdige - 10.2.82).

DOS'et fungerer som en udvidelse af NASSYS, og kommandoerne kaldes som alle andre NASSYS kommandoer.

Eksempler på kommandoer:

'@P GEMFIL'	- Sletter filen GEMFIL fra directory
'@K NYFIL.NAP'	- Danner en fil med navnet NYFIL. Data hentes fra NAP's teksbuffer.

Programmet er foreløbig kun beregnet til at styre et drive, men det vil senere blive ændret.

For-slag til pris: Diskette-version: 150,-
EPROM-version: 250,-

Manualer til FDD/FDC (5-600 sider) 75,- (kan dog undværes).

Interessant hardware/software kan eventuelt indgå i bytte.

Man er velkommen til at kontakte mig for assistance eller

yderligere informationer (helst ikke i week-enden).

Lars Rugård Jensen
Nørregade 221, 3.th
2200 N

Tlf. privat 01 85 31 51
arb. 02 96 60 88

DIAGRAM BRINGES I NN 3 D.

VELKOMMEN til nye MEDLEMMER:

218. Ejnar Lund.

Jellingevej 103.
7100 Vejle.

219. Kurt Hald.

Vestergade 17.
Frederiks.
7470 Karup J.

220. Flemming Larsen.

Skiftevej 18.
2820 Gentofte.

221. Thorsten Jacobsen.

Elsdyrsgade 18.st.
1324 Kbh.K.

222. Freddy Nielsen.

Kong Georgsvej 92.1.
2000 Frb.

223. Jørgen Abildgren.

Fjernskriv.vrkst.
Oldenborggade 1.A.
7000 Fredericia.

235. Ole Madsen.

Bragesgade 26.E.1.tv.
2200 Kbh.N.

239. Jesper Refslund

Andersen.

Buskevej 2.
2830 Virum.

240. Leif B. Andersen.

Gl. Kongevej 86.C.1.
1850 Kbh.V.

241. Ole Borup.

Strandboulevarden 156.
2100 Kbh.Ø.

242. Eriksson Information

Systems A/S.
att. Henrik Jørgensen.
Generatorvej 80.
2730 Herlev.

243. Theo W. Tams.

Amalie Skrams Alle 1.
2500 Valby.

244. Kurt Winther.

Gelstedvej 8.
5591 Gelsted.

245. Allan Randa.

Kordilgade 63.
4400 Kalundborg.

246. Bo Estrup.

Lundbjerggårdsvej
62.st.tv.
2740 Skovlunde.

247. DANPOWER APS:

att. Peter Johansen.
Svendborgvej 46.
8900 Randers.



RUSTENBORGVEJ 1, 2800 LYNGBY

NÆSTE MØDE:

1982-03-14-13.30.

EMNE: DOS i flere vers.

Indledere: Anders.

Erik Palsbo.-Steen Lærke.
+ arb.anlæg.

NB. Husk: Abent Hus Aften
hos Asbjørn.

Sidevolden 23, Herlev.

1982.02.23.19.00.

PASCAL-programmeringskredsen starter op igen s.d.

kl. 10.30. ALLE interesserede er velkommen.

+ vi vil gerne se interesserede medlemmer til dannelsen af en københavnerklub.

ALMINDELIGE OPLYSNINGER OM FORENINGEN :

Bestyrelsens sammensætning:

Formand Asbjørn Lind
Sidevolden 23
2730 Herlev
02 91 71 82
(helst efter kl. 20.00)

Næstformand Jesper Skavin
Broholms Alle 3
2920 Charlottenlund
01 64 03 14

Kasserer Erik Hansen
Lyngby Kirkestræde 6.1
2800 Lyngby
Helst skriftlig henv. Ellers
torsdag mellem kl. 17 og 18.
02 88 60 55

Sekretær Carsten Senholt
Blommevangen 6
2760 Måløv
02 66 19 65

Ole Hasselbalch
Vibeskrænten 9
2750 Ballerup
02 97 70 13

Søren Sørensen
Højlundvej 13
3500 Værløse
02 48 31 01

Frank Damgård
Kastebjergvej 26A
2750 Ballerup
02 97 10 20

Henvendelse til foreningen:

Indmeldelse, adresseændringer o.l. til kasserer
Programbibliotek til næstformanden

Øvrige henvendelser til formanden
(herunder annoncer/stof til NASCOM NYT)

Indmeldelsesgebyr: 25,00 kr.
Kontingent 1.1.82 - 1.7.82: 40,00 kr.

Oplag: 300

Redaktionen sluttet den 11.02.82

Husk at gamle numre kan købes hos Ole for 10 kr./stk +porto
Printerservice hos formanden

Bånd og bokse kan købes hos Carsten til følgende priser:
10 bånd 45 kr., ekstra etiketter 0,25 kr./stk og bokse
1,50 kr./stk + porto.

Annoncepris 0.75 kr. pr. A4 side (siderne 4 - n-2)*oplag
Indlevering foreningens adresse.