Musik på ABC80

Hör du till dom som undrar hur man gör musik på ABC80? Om så är fallet, är du sannerligen inte ensam om detta. Jag (med flera andra) har läst igenom (nästan) alla böcker som finns för ABC80 utan att hitta en förklaring till hur man får fram toner. Visst stod det om OUT 6,(udda tal mellan 1 och 256),men förutom diverse brus och sirener fick jag bara fram ungefär tre olika (rena) toner.

På ABC-kassett 5 fanns det ett program som producerade musik på kassett. Detta var ju visserligen bra men jag ville ju ha mina låtar i själva programmet och då lämpade sig inte detta program.

Vidare kom det då och då på kassetterna program som spelade låtar typ Sloop John B med flera. Då trodde jag att jag var nära lösningen, men icke. När jag listade programmen fann jag till min stora sorg att en stor del av programmen bestod av datasatser som inte gav mig ett dugg.

Lösningen kom en dag när jag spelade ett spel som heter Missile Command eller nånting åt det hållet. Mellan omgångarna spelade datorn en liten låt, visserligen inte vacker, men ändock en låt. Jag ska nu förklara hur man kan göra sina egna låtar. Börja med att skriva in följande rader:

```
100 ONERRORGOTO 200
110 READ A%,B%
120 FOR C%=1% TO B%
130 OUT 6%,0%,6%,121%
140 FOR D%=1% TO A%
150 NEXT D%
160 NEXT C%
170 GOTO 100
200 END
```

Detta är grundprogrammet. Om du kör det nu kommer det inte att hända nåt, men om du lägger till raden:

```
10 DATA 50,100
```

och sen skriver RUN så börjar datorn tjuta. Vad beror nu detta på? Ja inte vet jag! Första siffran i datasatsen är i alla fall tonens höjd och andra siffran längden på tonen. Tonhöjden bör vara mellan 22 och 78, annars låter det lite falskt (testa gärna ändå!). Tonlängden kan vara hur lång som helst (tror jag).

Här kommer några datasatser som spelar en trevlig låt:

```
10 DATA 50,25,60,20,76,30
20 DATA 60,38,50,46,36,110
30 DATA 27,37,31,35,36,60
45 DATA 60,37,52,45,49,90
50 DATA 49,47,27,120,31,35
55 DATA 36,60,38,110
60 DATA 43,26,38,30
70 DATA 35,60,35,60
80 DATA 49,45,60,40,76,47
85 DATA 50,25,60,20,76,30
90 DATA 60,38,50,46,36,110
95 DATA 27,37,31,35,36,60
97 DATA 38,55,31,70,36,120
```

Skriv nu RUN och njut(?) av musiken!!! Kanske inte helt fulländat, men ändå! Ta väck datasatserna och skriv in följande:

```
10 DATA 70,15,70,18,70,14,52,35,52,35,
   45,37,45,37, 33,80,40,25,52,37,40,30,
   52,15,63,30,38,90, 45,30,52,15,52,80
```

Ack så vackert! Eller hur? Nu tar vi en sista rolig grej. Skriv NEW och:

```
10 FOR A%=10% TO -10% STEP -1%
   : FOR S%=1% TO 10% : OUT 6%,
   0%,6%,121% : FOR D%=1% TO A%
   :NEXT D% : NEXT S% : NEXT A%
```

Slutligen: Nu vet ni hur ni ska skriva låtar. Skicka gärna in era datasatser till ABC-klubben när ni har komponerat färdigt så att andra kan få njuta av dom!

Hej då!

<5486>
Fredrik Tengroth

Varför blir det på detta viset?

Jag skulle vara tacksam över om jag kunde få hjälpmed att lösa följande lilla triviala men ändock så förargliga problem.

Jag är tyvärr alltför okunnig i programmering för att kunna lista ut varför följande lilla utskriftsprogram inte fungerar som jag tänkt. Som den intelligente säkert omedelbart ser och hånler åt, får jag dubbelt radavstånd vid printerutskrifter på text som t ex är inlagd med hjälp av VDO-editorn eller TV-editorn.

Varför blir det så?

```
5 DIM A$=100
10 OPEN "PR:" ASFILE 3
15 PRINT "ANGE FILNAMN";
16 INPUT B$
20 OPEN B$ ASFILE 2
30 INPUTLINE §2,A$
40 PRINT §3,A$
50 GOTO 30
60 CLOSE 3
70 CLOSE 2
80 END
```

Hälsningar

<5621>
Olle Wiren

Förklaringsförsök

Först måste konstateras att varje gång vi skriver på filnumret §3, dvs printern, sker en radmatning. Detta kan visas genom att vi skriver ut en tom sträng, med PRINT §3,"". Sedan kommer Olles variabel A$ som har läst in från fil §2 på rad 30 med en INPUTLINE. Den avslutas alltid med CR och LF. Det är dessa som inte skall finnas.

Man kan då t ex ta bort två tecken i slutet av variabeln med raden

```
35 A$=LEFT$(A$,LEN(A$)-2))
```

LEN(A$) ger aktuell längd av strängen A$. Vi minskar detta tal med 2, (CR o LF). I LEFT$(argument 1,argument 2) blir då den ursprungliga strängen A$ argument 1 och dess längd minskad med 2 argument 2. Resultatet A$ blir de ursprungliga tecknen i strängen förutom de två sista.

Och skrivs strängen nu ut på printer finns ju nu inte någon "extra" radmatning. varför Olle inte får dubbelt radavstånd i sina utskrifter.

<1208>
Ulf Sjöstrand

Angående flexskiveenheter

Man blir minst sagt bedrövad då man ser vilka priser flexskiveenheterna i handeln betingar. Till vissa andra på marknaden förekommande datorer existerar discminnen som är prissatta till 3-4000 kronor, men till ABC-maskinerna finns ingenting under ca 10 000 kr. (Dock har det börjat dyka upp utrustning kring 6 000 kr.)

Jag har tittat i den senaste medlemsmatirkeln och funnit många medlemmar inte äger flexminne. Detta beror säkerligen helt och hållet på de höga priserna. Jag undrar: skulle ingen av alla klubbens tekniskt sinnade medlemmar kunna konstruera ett billigt alternativ till de dyra flexminnena? Många skulle säkert nöja sig med enbart en drive, men kompatibilitet med exempelvis Luxors 830 skulle naturligtvis vara en fördel. (Detta skulle kansek uppfattas som plagiat?). Det finns säkert också många som skulle kunna utföra själva komponentmonteringen och sammansättningen på egen hand, under förutsättning att ritningar och byggbeskrivning finns tillgängligt. (Kanske en artikelserie i bladet?) Ett pris på ca tretusen skulle nog vara överkomligt.

Är detta genomförbart? När jag tänker efter minns jag att jag hört om ett liknade projekt, möjligen hos ABC-Väst. Hur gick det?

Jag inser att det skulle bli svårt med PROM-bränning och kretskortstillverkning, men om tillräckligt många var intresserade kunde kanske en mindre serie PROM och kretskort till verkas av någon firma.

<4166>
Bengt Ask

Ps Hur vore det med ett liknade projekt för modem och minnesexpansion för ABC80?

Utdrag ur MSG angående buggar i programmet VÄXLAPGM.UTL på ABC-kassett nummer16.

(Text 1191) Anders Franzen <5258>
Ärende: VXLPGM
Programmet VÄXLAPGM.UTL innehåller några allvarliga buggar. Det går inte att läsa filer från kassett och man får inte skriva in för långa rader när VÄXLAPGM är aktivt.

ABC80 dyker när man läser från kassett därför att avbrottsvektorn för kassettinterupt saknas. Om en BASIC-rad skrivs in och som kompilerad upptar mer än 72 bytes dyker datorn därför att avbrottsvektorn för tangenttryck placerats i kompileringsbufferten!

Man kan åtgärda det första felet på ett enkelt sätt. Det andra är mer svårlösligt. Det bästa vore om maskinkoden till VÄXLAPGM placerades på något annat ställe än i POKE-arean högst upp i minnet. Man kan emellertid flytta avbrottsvektorn till annan plats.

Dessa rader gör VÄXLAPGM lite bättre:

```
151 IF PEEK(65064)=251 POKE 65064,250 :
REM sänk stack
540 POKE 64052,148,255,148,5 : REM interuptvektorer
541 POKE -112,250 : REM rätt värde till I-reg
```

Rad 540 ersätter gammal rad, de övriga är tillägg. Nu kan man läsa från kassett utan problem. Denna ändring ökar maxlängden på BASIC-rader till 199 bytes vilket dock är för litet ibland.

```
 10 REM ++++++++++++++++++++++++++++++++
 11 REM ! Program ....    PRIMELVA
 12 REM ! Utgåva 1.0      1986-01-05
 13 REM ! av L Lundström <2694>
 14 REM ! Minne 32 Kbytes
 15 REM ++++++++++++++++++++++++++++++++
100 A%=25000% : DIM A%(A%/2%),P%(483%)
120 B%=PEEK(65065%)+SWAP%(PEEK(65066%))
140 B1%=PEEK(B%+20%)+SWAP%(PEEK(B%+21%))
    -1%
150 B%=PEEK(B%+10%)+SWAP%(PEEK(B%+11%))
    : B2%=B%-1%
170 POKE 65408%,1%,B1%-B%,SWAP%(B1%-B%),
    17%,B1%-1%,SWAP%(B1%-1%),33%,B1%,SWA
    P%(B1%),237%,184%,201%
180 POKE 65427%,62%,0%,115%,25%,16%,252%
    ,185%,200%,13%,24%,247%
240 ; CHR$(12%);TAB(4%)'*** PRIMTAL  225
    < p < 1E+12 ***'
250 ; : ; 'Strax ...';
260 FOR I%=1% TO 1155% : A%(I%)=2%*I%+1%
    : NEXT I%
265 FOR J%=1% TO 5%
270 IF A%(J%)=0% THEN 295
275 P%(L%)=A%(J%) : L%=L%+1%
280 FOR K%=J% TO 1155% STEP A%(J%)
285 A%(K%)=0%
290 NEXT K%
295 NEXT J%
300 X%=1%
310 FOR I%=1% TO 1155%
315 IF A%(I%)=0% THEN 330
320 P%(L%)=A%(I%)-X% : L%=L%+1% : X%=A%(
    I%)
330 NEXT I%
340 ; CHR$(13%);TAB(6%)'Använd mellansla
    g för paus' : ;
350 ONERRORGOTO 350 : N$='' : ; 'Startta
    l:'SPACE$(12)'<';CUR(PEEK(65011%),8%
    ); : INPUT N1$
360 FOR I%=1% TO LEN(N1$) : Z%=ASC(MID$(
    N1$,I%,1%)) : IF Z%>47% AND Z%<58% T
    HEN N$=N$+CHR$(Z%) : NEXT I%
370 IF VAL(N$)<225 OR COMP%(N$,SUB$('999
    999999999',NUM$(2*A%),0%))=1% THEN ;
    'Utanför tillåtet talområde' : GOTO
    350
380 N1$=DIV$(N$,'2',1%) : IF RIGHT$(N1$,
    LEN(N1$))='0' THEN N$=ADD$(N$,'1',0%
    )
410 Z%=CALL(65408%)
420 M$=ADD$(N$,NUM$(2*A%),0%)
440 Z=SQR(VAL(M$)) : ; : ; 'SQR=';INT(Z)
445 N1$=SUB$(N$,'1',0) : ;
450 FOR I%=0% TO 3% : P$=NUM$(P%(I%)) :
    ; CHR$(13%);RIGHT$(P$,2%); : GOSUB 5
    00 : NEXT I%
455 P$='1'
460 FOR I%=4% TO 483% : P$=ADD$(P$,NUM$(
    P%(I%)),0)
462 ; CHR$(13);P$;
465 IF VAL(P$)>Z THEN 580 ELSE GOSUB 500
470 NEXT I% : GOTO 460
500 N9$=DIV$(N1$,P$,6%)
510 N9$=ADD$(LEFT$(N9$,LEN(N9$)-7%),'1',
    0%)
511 N9$=MUL$(N9$,P$,0%)
512 N9$=SUB$(N9$,N1$,0%)
520 S$=DIV$(N9$,'2',1%)
530 IF RIGHT$(S$,LEN(S$))='5' THEN S=VAL
    (S$)+.5 : GOTO 540
531 P1$=DIV$(P$,'2',1)
532 P1$=LEFT$(P1$,INSTR(1%,P1$,'.')-1%)
533 P1$=SUB$(P$,P1$,0)
534 S=VAL(S$)+VAL(P1$)
540 IF S>A% THEN RETURN ELSE T%=((A%-S)/
    VAL(P$))+1% : S%=B2%+S
550 IF T%>1% THEN POKE 65420%,6%,T%,14%,
    SWAP%(T%),33%,S%,SWAP%(S%) : Z%=CALL
    (65420%,VAL(P$)) : RETURN
570 POKE S%,1% : RETURN
580 ; CHR$(13); : FOR I%=B% TO B%+A%-1%
590 IF PEEK(I%) THEN 600 ELSE P9$=ADD$(N
    $,NUM$(2*(I%-B%)),0)
595 ; P9$;' '; : IF (INP(56) AND 127)=32
    % GET Ä$
600 NEXT I%
610 N$=ADD$(N$,NUM$(2*A%),0) : GOTO 410
620 END
```

Ljus i mörkret- EUREKA

(Text 1058) Tomas Wikström <1398>
Ärende: Ljus i mörkret - Eureka!
Efter att ha glatt sig åt paketet med Ge-Jo's ramfloppy som jag hämtade ut på posten härom veckan,blev allt snart svart i dubbel bemärkelse. Vad hände ? - Jo efter installationen verkade allt fungera,till en början,men då jag började använda ramfloppyn,dvs körde med den initierad,och *samtidigt* hade Supersmartaiden aktiverad började konstigheterna. Reset !?- Va? - Loop - Allsköns smörja skrollande på skärmen - Den lilla högtalaren tjutande som en stungen ... Nåväl,efter 7 sorger och 8 bedrövelser och en hel del funderande, + *mycket och goda* råd från vänner i den här eminenta klubben började vägen öppna sig ur eländet.
Om Ni bygger ut ABC80 (speciellt då den med nya 'lilla separata' nätdelen,se till att göra som följer nedan. FÖR ABC:n DRAR MER STRÖM ÄN SUPPLYET KAN LEVERERA. Den vetskapen har kostat mej 328 gråa hår,minst.

Så här gjorde jag:
Besökte ELFA för att skaffa 2 st. nya spänningsregulatorer: ELFA's best-nr: 73-0956-0 Pris 12:45 + moms. Dessa regulatorer är dubbelt så kraftiga som originalet i ABC80 (de sitter förresten på kylflänsen i tangentbordet,och är bara att byta ut mot de nya,lätt om man har lite lödvana). Man bör också,speciellt om man har den lilla nya nätdelen, byta likriktarbryggan till en kraftigare dito. (min likriktarbrygga gick sönder i ett mycket tidigt stadium, långt innan 80'an blev utbyggd till vad den är nu)

Det finns också en viss risk att inte transformatorn kan leverera den ström som kan tänkas behövas (på min maskin ligger jag på bara litet över 8 volt innan regulatorerna, och det bör upp litet,helst till minst 9 volt (men ej heller för högt då överspänningen då måste kylas bort). Detta med tanke på att man måste räkna med viss spridning på komponenter,gör att man bör kanske undersöka sin egen spänning innan likriktning för att se om det räckte med det ovanstående ändringarna eller om man måste se sig om efter ett kraftigare nätaggregat.

Det blev mycket skrivet,men det var inte ett ord för litet efter allt besvär och alla konstigheter jag haft på sista tiden.

Många och mycket välmenande hälsningar från Tomas i Solna.

(Text 1058)

Mottagare: ABC80

Ärende: Ljus i mörkret - Eureka!
Bra skrivet! Jag har haft samma problem och kommit på att det är spänning/ström som inte räcker. Däremot har jag inte haft konkreta uppgifter om VAD som måste göras. Nu ska det bli gjort här också!

Men jag tycker Gejo m fl borde tala om för att lla som köper tillsatserna att den nya kraftburken är för klen. Jag har hittills klarat Superbasic+32K+80tkn utan besvär, men Gejos ramfloppy blev för mycket.

(Brev 4296) Tomas Wikström <1398>
Mottagare: Sven Wickberg <1384>
Ärende: Strömförsörjningsproblem ABC80
Nu har jag äntligen fått stil på mitt system. Det tog sig en tur till,som jag vill varna Dej för: Normalt ska 5-volten ligga på just 5 V (eg 4.75 - 5 .25), men av någon anledning ville inte mina problem helt upphöra förrän jag gick upp *över* 5.2 V!
Så nu ligger jag på 5.4 (vid regulatorn - sjunker något på vägen till kapslarna) som är något för högt,men det måste ju fungera. Dessutom är det inget problem då garanterad breakdown-voltage ligger på 7 V (för TTL).

Nå till saken - jag gjorde på ett något annat sätt än som jag beskrev i MSG,då jag var tvungen att kunna finjustera spänningen. Komponenter till det har jag,inte i överflöd men i alla fall så att jag kan dela med mig.

Hör av Dej till mig,jag hjälper gärna till om det behövs,eller skriver ned enkla instruktioner som visar hur jag har (varit tvingad till att)gå tillväga.

Hälsningar Tomas.
(Brev 4296)

OUT-kommandon för tangentbordet ABC99

Koderna sänds från BASIC till tangentbordet som "OUT 34,Kod".

Kod	Funktion
0000 0111	alarm-signal
0000 0001	slå av larmfunktion
1000 0001	på -"-
0000 0010	slå av kort tangent-klick
1000 0010	på -"- -"-
000 0100	slå av långt -"-
1000 0100	på -"- -"-
0000 0101	slå av repetering
1000 0101	på -"-
0000 0110	ASCII-mode (normalt)
1000 0110	UP/DOWN-mode
0000 1000	CAPS LOCK normalt
000 1000	CAPS LOCK mode
0000 1001	INS normalt
1000 1001	INS mode
0000 1010	ALT normalt
1000 1010	ALT mode
0001 1000	sänd identifikationskod *1
0011 1001	skriv data till musens positions-register
0011 1010	skriv data till musens prescale register
0011 1011	skriv data till musens gränsregister
0011 1100	läs data från musen
0011 1110	läs data från musen auto mode
0011 1111	slå av auto mode för musen
0xxx 0000	tänd lysdiod x (1=000, 2=001, 3=010 osv)
1xxx 0000	släck -"- x

*1 Identifikationskod
Tangentbordet svarar med 7 bytes:

Byte	Kod	Betydelse
1	0001 1011	ESC
2	001x xxxx	nationalitet version
3	001x xxxx	LED-typ
4	0010 xxxx	LED's 5-8 1=tänd
5	0010 xxxx	1-4
6	0010 0xxx	INS, ALT, CAPS LOCK 1=påslagen
7	0010 0000	spare byte

Version	(byte 2)
0010 0001	Svenskt
0010 0010	Norskt
0010 0011	Danskt
0010 0100	USA
0010 0101	Engelskt
0010 0110	Spanskt
0010 0111	Franskt
0010 1000	Tyskt
0010 1001	Isländskt

Tgb-typ	(byte 3)
001x x001	ABC 55
001x x010	ABC 77
001x x011	ABC 99
001x 1xxx	UP/DOWN-mode
001x 0xxx	ASCII-mode (normalt)
0011 xxxx	Mus ansluten
0010 xxxx	ej ansluten

ABC99 MUS

Tangentbordet avkodar pulserna från musen och håller reda på dess position. Följande gäller både för x- och y-förflyttning. Pulserna från musen är ansluten till ett prescale register. När skalfaktorn är nådd så adderas/subtraheras increment registret till/från positions regitret. Om musen går utanför gränserna uppdateras inte positionsregistert förran man ändrar riktning.

Kodning:

Alla register är 12 bit långa. De delas upp i två bytes med off- set 20H vid sändning till/från tangentbordet. Varje byte har ett värde mellan 20H till 5FH. Den högsta byten sänds alltid först. Puls-jitter måste filtreras. Även tryckknapparna har en offset av 20H.

Register	Initialvärde (hex)
x-position	000
y- "	000
x-min	000
x-max	FFF
y-min	000
y-max	FFF
x-scale	001
y- "	001
x-increment	001
y- "	001

Sätta register:
Position:
0011 1001, x-pos hi, x-pos low, y-pos hi, y-pos low

Scale & increments:
0011 1010, x-scale hi, x-scale low, y-scale hi, y-scale low, x-increment hi, x-inc low, y-inc hi, y-inc low

Limits:
0011 1011, x-min hi, x-min low, x-max hi, x-max low, y-min hi, y-min low, y-max hi, y-max low

Avläsning av musen:

Det finns två sätt att läsa data från musen.
- När koden 0011 1100 sänds till tangenbordet, så svarar den EN gång med:

1001 0000, x-pos hi, x-pos low, y-pos hi, y-pos low, knappstatus

Ingen tangent får avbryta sekvensen.
- När koden 0011 1110 sänds till tangenbordet, så svarar den med:

1001 0001, x-pos hi, x-pos low, y-pos hi, y-pos low,
1001 0010, knappstatus

Efter denna första sekvens så sänds positionsregister eller knappstatus endast när dessa förändras. De kan sändas separat. Efter varje sekvens (6 alt. 3 bytes) kan tangentbordet sända tangentkoder.

För att avsluta denna auto read kommando, sänds koden 0011 1111 till tangentbordet.

Mus-kontaktens funktioner (15 pol. D-sub):

Pin	Funktion
1	5V/50 mA
2	XA
3	XB
4	YA
5	YB
6	Vänstra knappen
7	Mitt "
8	Högra "
9-13	NC
14	Mus-indikator, aktiv låg när mus är ansluten.
15	Jord

Överföringshastigheten ökar från 600 Baud till >= 4800 Baud när musen är ansluten.

Koder för plattan på ABC99:

Koderna som plattan avger kan avkodas på följande sätt:

```
bit

 7   6   5  4   3    2   1   0

! 1 ! 1 ! 0 ! 1 ! s ! x ! x ! y ! y !

 !   !   !   !   !   !   !   !
 ----!----   !   !   !   !   --- Upp/Ned
     !       !   !   !   0   1   Upp
     !       !   !   !   1   1   Ned
     !       !   !   !
     !       !   !   --- Höger/Vänster
     !       !   0   1   Höger
     !       !   1   1   Vänster
     !       !
     !       ----- SHIFT
     !
     ----- Kod för plattan
```

Exempel:
1010 1101 = uppåt, vänster
1011 0100 = höger, SHIFT nedtryckt

<4882>

Gerry Eriksson

Noteringar till kassett nr 16

Jag har nyligen mottagit kassett nr 16 och har efter att jag gjort några provkörningar ett par frågor och kommentarer som jag tror kan vara av intresse även för andra klubbmedlemmar.

1. Det intressanta programmet SORTERA bildar först med ABC80:s slumptalsgenerator ett önskat antal ord av 3 till 6 eller 7 teckens längd. Programmet jämför sedan sorteringstiden för denna ordmängd med 6 olika sorteringsmetoder. De 6 metoderna är:

1. BUBBLE-sortering
2. URVALS-sortering
3. DELAYED REPLACEMENT-sortering
4. BATCHER-sortering
5. SHELL-METZNER-sortering
6. QUICKSORT-sortering

Resultatet visar klart skillnaden i hastighet mellan olika sorteringsmetoder - N.B. om man startar upp programmet med RUN före varje nytt prov! Men programmet inbjuder på rad 580 till att direkt fortsätta med nästa sorteringsmetod, och då sorteras i fortsättningen DET REDAN SORTERADE ordbeståndet, inte det ursprungliga!

För att undvika detta bör programmet utökas med en ny rad 355:

```
355 FOR I%=1% TO N% : B$(I%)=A$(I%) : NEXT I%
```

varigenom det efter varje sortering återgår till samma osorterade ordföljd före varje nytt sorteringsalternativ.

Vill man sedan också ha slumpvis start vid bildningen av ordmängden tillfogas en rad:

```
265 RANDOMIZE
```

Det ursprungliga programmet gav emellertid en del oväntade resultat som motiverade en närmare undersökning. Den följer här.

Jag bildar t.ex. 200 ord och jämför med dessa olika sorteringsmetoder, dels som första metod och dels som andra, dvs sedan den första redan sorterat materialet. Då får jag de tider (i sek) som framgår av den övre delen av tabellen nedan, där MOL = max ordlängd, ST = sorteringstyp, AOB = antal ordbyten

			Sorteringsmetod					
MOL	ST	AOB	1	2	3	4	5	6
6	OS	0	67.2	49.8	55.0	13.5	11.3	9.4
7	OS	0	66.9	49.7	54.9	13.0	11.2	8.8
7	S	0	35.3	48.8	54.0	10.1	4.8	55.8
7	S	4	37.5	48.8	54.0	10.4	5.9	35.3
7	S	10	38.4	48.8	54.0	11.3	7.7	18.8
7	S	25	44.9	48.9	54.1	11.9	8.6	15.7
7	S	50	50.0	48.9	54.2	12.1	9.9	10.7
7	OS	0	593	437	487	53.1	43.1	32.6
7	S	0	318	433	484	41.0	18.9	490
7	S	3x10	326	433	484	43.3	22.3	243

Resultatet är intressant. De tre sista metoderna är påfallande snabbare än de tre första, och långsamheten hos dessa ökas kraftigt (något mindre än kvadatiskt) med antalet ord som sorteras.

Skillnaden i snabbhet mellan de tre senare metoderna är inte särskilt stor och lustigt nog går det för alla metoderna något fortare när största ordlängden ökas från 6 till 7 tecken.

Men när uppgiften blir att sortera en redan sorterad ordmängd går QUICKSORT bet. Medan SHELL-METZNER-metoden minskar tiden till mindre än hälften tar QUICKSORT på sig ungefär lika lång tid som de tre långsamma metoderna! Om nu tio slumpvis valda ordpar i den sorterade ordmängden byter plats sjunker sorteringstiden för QUICKSORT till ca 20 sek, dvs fortfarande mer än dubbla tiden för sortering med SHELL-METZNER. Först vid ca 50 ordbyten blir sorteringstiden lika, ca 10 sek.

Slutsatsen blir att metod 5, SHELL-METZNER, är något långsammare än QUICKSORT vid mycket osorterade ordmängder, men att QUICKSORT bör unvikas vid omsortering av en förut sorterad mängd som får ett tillskott av mindre än ca 20% nya ord. Och QUICKSORT skall tydligen skys som pesten vid små tillskott, sådana som ofta görs vid uppdatering i registerprogram. Det kan möjligen förklara varför ett välkänt registerprogram vid omsortering av några tusen poster efter tillskott av några få poster tar ett par timmar på sig för ny sortering.

Då SHELL-METZNER-metoden dessutom har en mycket enkel uppbyggnad, speciellt jämfört med QUICKSORT, och därtill tar mindre plats i minnet, bör den avgjort föredras när medlemmar (eller företag) behöver en subrutin för sortering.

Den undre delen av tabellen visar tiderna för en ordmängd på 600 ord. Slutsatserna ovan framhävs där ännu tydligare.

2. När jag med min ABC80 med 32kB försöker överföra FASTCAS.32K från kassetten med LOADCAS: stannar överföringen med ERR 11 på rad 29. Denna rad lyder "29 ??? 29 GOLV+". Kopiering till diskett med CASDISK3 fungerar, men när programmet sedan skall köras stannar det vid samma rad med ERR 11 LINE 29 ERR 59. Vid försök att läsa filen med VISA får jag STOP vid rad 29, nu utan frågetecken, och ERR 58 LINE 80. Alltså låst läge!

Jag har för mig att jag stött på de tre ??? någon gång tidigare i ett kassettprogram och då efter en del försök lämnade programmet åt sitt öde, och vidare att jag sett en sådan rad nämnas i någon uppsats i ABC-bladet - men i vilket nr?

Vad betyder den stoppande raden 29 ??? 29 etc och hur skall jag forcera den?

<1308>
Göran Tengner

Programredaktionen svarar:

Programredaktionen har uppmärksammat felet på kassett 16. Använd FASTCAS från ABC-kassett nr 17 i stället.

ABC VDP

Vad?

I ABC-bladet nr 4, 1985 finns en artikel 'Höjoplöselig Grafik med ABC80'. Den inspirerade mig till att skriva några rader om en del experiment jag gjort med grafik. Jag beskriver en tillsats till ABC80 som ger bättre grafik på bildskärmen. I stället för standardupplösningen 80*72 ger tillsatsen upplösningen 256*192. Den händige amatören kan bygga den för under 1000:- Den nya grafikbilden visas på den vanliga ABC80-bildskärmen. Man kan växla mellan den vanliga ABC80-bilden och grafikbilden med en enkel instruktion (OUT 2,x går även som kommando).

Varför?

När jag hade byggt om min ABC80 till 64 kbyte RAM så blev det gamla RAM-minnet över. Man skulle nämligen vid ombyggnaden sätta i nya minneskapslar, det gick inte att komplettera de gamla. Vad skulle jag då göra av dessa gamla minneskapslar? Högupplösningsgrafik är spännande! Men det behövs mer än RAM-minne. Jag studerade kopplingsschemat för ABC800 med grafik. Den grafiktillsatsen innehåller en väldig massa elektronik för att göra bilderna. Det verkade inte lockande att försöka med någon imitation av den.

Skoldatorn Compis var mycket omskriven vid denna tidpunkt. Den ger fin grafik med en speciell mikroprocessor (tex NEC 7220). Detta var en mycket intressant möjlighet ända tills jag fick veta priset på processorn. Den kostade 800 kronor på våren -84. Sedan hittade jag med hjälp av en artikel i den amerikanska tidskriften BYTE en enklare mikroprocessor (Texas TMS 9929) som bara kostade 135:-. Den fick det bli. Denna processor sitter i flera persondatorer t ex Spectravideo, TI 99/4A. Texas kallar den Video Display Processor förkortat VDP.

Teknik

I vertikal led är upplösningen hos grafiktillsatser oftast 256 linjer per bild. Det beror på att en standard TV-mottagare som man av prisskäl vill använda ger just 256 linjer per bild och 50 bilder per sekund. Vid Sveriges Radios TV sändningar får man dubbelt så stor upplösning genom att sända en bild till, lite förskjuten från den första. Det betyder att det ska ritas 2 skärmbilder innan man har fått se allt, det blir 25 hela bilder per sekund. Detta system duger inte för högupplösningsgrafik. Titta på TV2 pausbild och se hur 2:an hoppar! Det vore inte bra om ABC80:s tvåor hoppade på det viset. I horisontell led behöver upplösningen inte vara större än i vertikal led alltså 256.

Detta betyder att man på bildskärmen kan rita 256*256 = 65536 olika punkter. Om varje punkt ska kunna tändas i någon av 16 olika färger så behövs 4 bit/punkt i bildminnet (2**4 = 16). Bildminnets storlek blir då 65536*4 bit = 32768 byte. Den grafikprocessor jag valde är ursprungligen gjord till amerikansk TV-standard och bara delvis anpassad till europeisk standard. Den ritar därför bara på 192 linjer per bild. För att minska behovet av (på den tiden) dyrbart bildminne så kan processorn dessutom inte alltid ge olika färg till punkter som ligger bredvid varandra. Mer om denna lustighet senare.

Blockschemat

ABC-VDP byggs på ett labkort i enkelt Europaformat som ansluts till ABC-bussen, kanske i flexskivestationen. Även kabeln mellan tangentbordet och bildskärmen ska passera ABC-VDP Med omkopplaren i det läge som den är ritad i blockschemat så passerar signalerna från tangentbordet till bildskärmen och ABC80 beter sig helt normalt. Ett program i ABC80 får nu fylla bildminnet, via VDP:n, med bildinformationen. Programmet slår sedan om omkopplaren och högupplösnigsbilden kommer fram på bildskärmen.

Vill man absolut se färgerna så skaffar man en extra färgTV och bygger den anpassningselektronik som kallas MOD i blockschemat. MOD (=modulator) ska vara lite olika beroende på om man använder antenn- eller videoingång på färgTV:en.

Framtiden

Man kan ju undra vad det finns för anledning att ägna sig åt denna grafiktillsats när den som beskrevs i ABC-bladet 4, 1985 kan så mycket mer: tex. rita linjer, cirklar och bokstäver fruktansvärt snabbt och nästan på egen hand. Låt mig nämna några skäl:

1.ABC-VDP blir säkert billigare.

2.Man behöver ingen extra TV-apparat Det behöver man inte till den andra tillsatsen heller om man kompletterar den med omkopplaren. För den intresserade bifogar jag schema över omkopplaren. Den består av en CMOS omkopplaren MC 14016 som styrs från D-vippan 74LS74. Transistorerna på utgångarna minskar inverkan av omkopplarens resistans i tillslaget läge.

3.De olika färger som ABC-VDP kan göra blir på ABC80-bildskärm olika gråtoner. 'Sprites' från VDP är en form av grafik som kan vara rolig att prova.

Låt mig också nämna några skäl att man INTE ska bry sig om ABC-VDP:

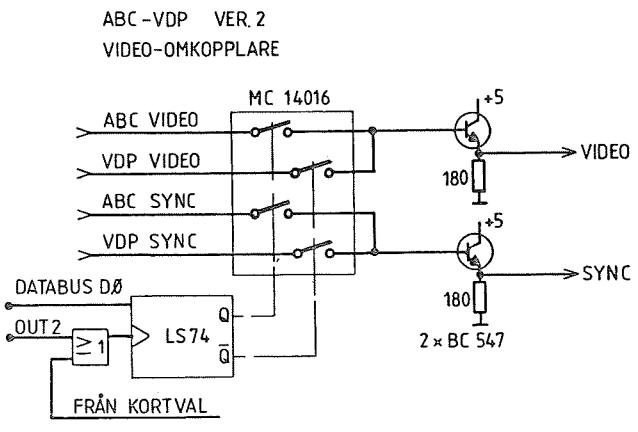
1.Det är ett ganska avancerat bygge, ingenting för förstagångslödare.

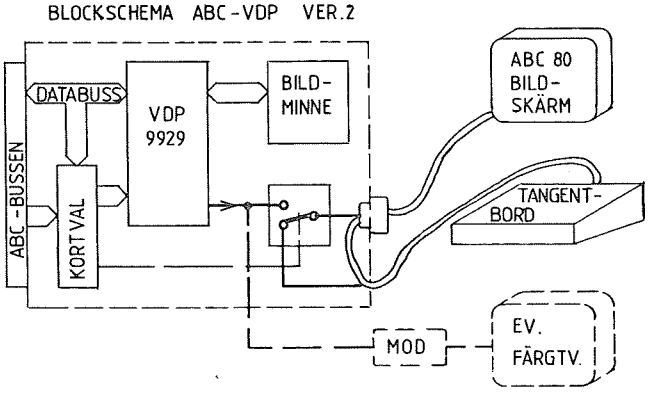
2.Texas vill nu istället sälja en processor som heter TMS 9129 Det enda som egentligen skiljer är anslutningen av bildminnet.

3.Det vore trevligt om man kunde visa både ABC80- och VDP-bilden samtidigt. Då måste båda två göra sina linjer och bilder samtidigt vilket ställer nya krav på elektroniken. Ibland går jag och funderar på hur detta ska lösas. Det kanske blir en ny version av ABC-VDP.

4.Det finns inget fullständigt kopplingsschema över ABC-VDP publicerat i ABC-bladet. Den som är så händig med lödkolven att han/hon törs ge sig på det här kan kanske göra ett schema själv. Eller komma på något sätt att skaffa sig ett.

<6555>
Lars Karis





Mera räknekul med CBS

Matematiker är ett lustigt släkte (har jag sagt det förut?). Man förströr sig med alla möjliga högst onödiga och möjligen onyttiga tankar om hur man kan mixtra med de hela talen. Ett sådant (obevisat, tror jag) teorem är att alla tal som behandlas med CBS-metoden förr eller senare konvergerar till 1.

Sanningen att säga har jag totalt glömt vad CBS står för. Jag utlyser därför en pristävling: Den som kan tala om vad det betyder skall få den absolut oöverträffade äran att vara ABC-klubbens matematiske Besserwisser. (Antalet pris ej begränsat.)

Till saken. Tänk på ett tal!

Om talet är jämnt, halvera det. Om det är udda, multiplicera med 3 och lägg till 1.

Gör om föregående stycke till dess du får 1. Teorin säger att det alltid skall inträffa till slut, men vägen dit kan vara mycket lång, och en matematikers livslängd är för kort för att man skall prova detta empiriskt för hand. Vi låter datorn göra det.

Jag har valt att arbeta med ASCII-räkning. Det ger på ABC80 upp till omkring 25 siffrors noggrannhet (på ABC800 bra mycket mer), medan vanliga flyttal är oanvändbara för heltalsjobb redan vid 1 000 000.

Vid ASCII-räkning görs talen om till teckensträngar, som sedan av datorns inre bearbetas med samma metoder som när man räknar för hand, men mycket fortare förstås.

På 70-80 läggs talen i A1$ och A9$, och sedan görs ett tabellhuvud som skall visa hur många steg man måste gå innan resp tal konvergerat till 1. Det påstående är förresten detsamma som att säga att förr eller senare blir talet av formen 2**N, där N är ett heltal, alltså ett tal i den inte helt obekanta serien 2, 4, 8, 16...

På 120 läser vi över värdet A1$ till A2$, som vi sedan bearbetar i subrutinen 170-. S% är en räknare som håller reda på antalet steg i serien.

Nu flyttar vi över A2$ till A$ och dividerar med 2, dels med noll decimaler, dels med 1 decimal. Blir resultaten B$ och C$ lika är det ett jämnt tal, som skulle divideras med 2. Det har vi redan gjort i B$, så vi läser över det till A$ och gör om manövern.

COMP%(B$,C$) är en funktion som jämför de numeriska strängarna B$ och C$. Om de är lika är COMP%()=0, annars -1 resp +1. (Experimentera litet!)

Observera att det går åt pipan i vissa fall om B$ och C$ inte har lika många decimaler! Det sammanhänger med sorteringen av strängar. Strängen "0.1" anses som mindre än "0.10".

Men det har ingen betydelse här, så jag har högaktningsfullt struntat i kontrollen. (Varsågod, vän av ordning!)

Rad 220: om värdena inte är lika är talet udda. Då multipliceras först med 3 (MUL$()), sedan adderas 1. Antalet decimaler sätts till noll.

Räknaren uppdateras och så da capo till dess A$="1" (rad 200). Då skrivs antalet steg ut, man hoppar till 140 och tar nästa tal, ända till dess man nått A9$.

Jag har hittills inte hittat något motbevis mot CBS-teoremet. Men skam den som ger sig. Det finns många oprövade tal lägre än 99999999999999999999999999.

Det finns ingen paufågel i programmet. Ett allvarligt förbiseende (vän av ordning - är du kvar?).

<1384>
Sven Wickberg

```
 10 REM CBS
 20 REM kollar konvergens till 1
 30 REM jämna termer halveras
 40 REM ojämna blir 3* +1
 50 REM ------------------
 60 ; CHR$(12%)'Undersökning av CSB-kon
    vergens'
 70 ; : ; 'Fr o m tal: '; : INPUT A1$
 80 ; ' T o m tal: '; : INPUT A9$
 90 REM ------------------------------
100 REM gör tabell
110 ; : ; 'Tal','Antal termer'
120 A2$=A1$
130 GOSUB 170
140 A2$=ADD$(A2$,'1',0%)
150 IF COMP%(A2$,A9$)>0% THEN END ELSE
    130
160 REM ------------------------
170 REM beräkning
180 S%=0%
190 A$=A2$
200 IF A$='1' THEN ; A2$,S% : RETURN
210 B$=DIV$(A$,'2',0%) : C$=DIV$(A$,'2'
    ,1%)
220 IF COMP%(B$,C$)=0% THEN A$=B$ ELSE
    A$=ADD$('1',MUL$(A$,'3',0%),0%)
230 S%=S%+1%
240 GOTO 200
```

Från 80 till 800

Ideer om programkonvertering

Den gamla trotjänaren ABC80 tillverkas inte längre. Även om mycket tyder på att den kommer att vara i omlopp länge än, måste man räkna med att många f d ABC80-ägare vill eller måste gå över till ABC800, dvs ABC800/802/806, och att de vill ta sina gamla program med sig. (I fortsättningen menar jag alla tre när jag säger ABC800.)

Det finns också en enorm programbank för ABC80-program som nytillkomna ABC800-ägare skulle vilja skörda ur.

Slutligen kan vi finna problemställningen att vilja köra i princip samma programkod på båda maskintyperna. Antingen vill eller måste man använda samma disketter eller också är båda maskintyperna anslutna till förslagsvis CAT-NET.

Eftersom jag har erfarenhet av samtliga alternativ har jag hunnit filosofera en del över Luxors s k kompatibilitet. Kanske kan mina erfarenheter vara till nytta också för andra.

Kompatibla uppåt, sa Luxor

I utvecklingsarbetet för nya datortyper har Luxor redovisat den lovvärda ambitionen att maskinerna skall vara "kompatibla uppåt", dvs att program från en tidigare maskin också skall kunna köras på en senare.

Detta innebär att det nästan alltid är möjligt att läsa in den gamla ABC80-programmen till den nya datorn. Däremot är det inte säkert att det går att köra dem där. Det måste ofta först göras en rad modifikationer. Det finns nämligen vissa hinder som har att göra med dels lagringsmediet (kassett, skiva) och dels programskrivningen.

Det betyder att det ingalunda är säkert att samma program kan köras på båda maskintyperna. Dock är det en avsevärd vinst att man inte behöver knappa in programkoden i dess helhet.

1: Kassetterna

Med den första ABC80 följde en bandspelare. Man hade visserligen en del trassel med den och det kom så småningom en ny modell. Samma bandspelare kunde kopplas till ABC800, men det skedde en produktutveckling så att ABC800 kunde spela in och av med mycket större hastighet. Det betyder att ABC80-kasseter fortfarande kan läsas av ABC800-bandspelare, men inte tvärtom.

Det gäller alltså att föra över sina program från kassett till diskett.

2: Skivorna

Skivspelarna har utvecklats: från enkel packningstäthet till dubbel, från enkelsidig till dubbelsidig och vidare till kvadrupel täthet. Inför varje steg har det funnits en möjlighet att läsa över informationen från föregående steg till det nya, men efter ett tag kan man vara i den situationen att man inte längre med sin nyaste maskinpark kan läsa sina äldsta disketter. Det gäller att som engångsföreteelse föra över allt man är intresserad av till det nya formatet.

Detta leder till problem för amatörer och skolor med de äldre typerna. Ofta är man hänvisad att behålla det man har - och varför inte? Grejorna kan fås billigt och de fungerar fint länge ännu. Men det kan som sagt bli problem med anpassning till eller snarare från senare varianter.

Ännu värre kan det bli om flera olika skivstorlekar blir populära. 8"-skivorna tycks vara mycket hårt standardiserade och under UFD-doset i ABC800 finns program som gör det möjligt att koppla ihop 5"- och 8"-apparater till samma dator och spela över från den ena till den andra.

Men det är knappast något för hobbyhackaren som skall bekosta allt ur sin privatekonomiska ficka.

Nu står 3.5"-disketterna i farstun, och då kan det bli mera problem. Men låt oss inte göra oss bekymmer i förväg.

3: Programmen

Processorn i en dator förstår sig bara på ettor och nollor. Om programmet är skrivet i BASIC måste det först tolkas och överföras till binär form som datorn förstår. Detta tar tid och gör at BASIC-program på de flesta datorer går ganska långsamt.

I Luxormaskinerna har man ett mellanläge. När programmet laddas in i arbetsminnet - vare sig man skriver från tangentbordet eller laddar in det från skiva/kassett - sker en första omvandling (kompilering) så att BASIC-texten förvandlas till sifferkoder som är lättare för maskinen att läsa. Därför går ABC-BASIC ganska fort om man jämför med de flesta andra BASIC-dialekter.

Sparas ett program med SAVE kommer det att lagras i denna kompilerade form. I programlistan står extension .BAC, om man inte "för hand" har valt en annan beteckning. I de flesta fall är det en fördel med .BAC-lagring. Programmen laddas och körs snabbare, även om de vanligen tar mer plats på skivan/kassetten.

ABC80 och ABC800 har olika sätt att kompilera BASICen och kan därför inte läsa varandras .BAC-program. Man får feltexten "fel programformat" om man försöker.

Sparas programmet i stället med LIST lagras det exakt som det skrivits på skärmen och i ASCII-format, dvs alla tecken lagras med siffervärde motsvarande ASCII-koden. I programlistan står extension .BAS.

Denna lagring har några fördelar. Dels tar den vanligen mindre plats än .BAC-formatet, dels kan den läsas som en textfil. En textfil är nämligen ASCII-lagrad på samma sätt. Och eftersom ASCII är en (nästan) gemensam standard för mikrodatorerna, kan program i ASCII-format överföras mellan dem med de vanliga överföringsrutiner som finns.

Skall man köra ett ABC80-program på en ABC800 spar man programmet med LIST <programnamn>, för över skivan till ABC800-systemet och skriver RUN eller LOAD på vanligt sätt. Då laddas programmet in och förvandlas samtidigt till det halvkompilerade format som ABC800 vill jobba med.

SÅ LÅNGT är det enkelt.

Programanpassning???

Har man TUR går nu programmet att köra. Men det har man nog inte. Redan vid inladdningen anges vanligen några radnummer med olika slag av felmarkering som först måste rättas till. Det finns nämligen skillnader i ABC80-BASIC och BASIC II, som maskinen inte själv kan reda ut. De aktuella programraderna markeras i listan med ett antal frågetecken, så de är relativt lätta att hitta.

Nödvändiga ändringar

Här är några av de vanligaste rättelser som måste göras.

* END måste stå ensamt på en rad. Om man har en rad som t ex

IF <villkor> THEN END

får man antingen byta END mot STOP eller också göra ett hopp till lämplig plats där man kan skriva sin END-rad.

* Instruktionen **ASC(tecken)** måste bytas till ASCII(tecken). På de ABC806:or där jag har provat tycks ändringen ske automatiskt vid inmatningen. Men OBSERVERA att om man då spar programmet från ABC800-maskinen, kommer programmet i fortsättningen att innehålla varianten ASCII som ABC80 inte kan läsa.

* I vissa fall tolereras inte en **REM-sats** i slutet på en programrad. Detta kan ha med kolonet att göra (se nedan) men jag har märkt det särskilt i rader som börjar med IF:

IF <villkor) THEN <instruktion> : REM <förklaring>

Här måste man antingen ta bort REM-satsen eller skriva med utropstecken:

IF <villkor> THEN <instruktion> ! <förklaring>

Det senare är ingen nackdel om man i fortsättningen bara kör programmet på ABC800.

I en rad andra fall, som jag inte kunnat totalt analysera, krånglar ABC800 om det kommer en kolonsats efter en IF-sats. Någon har påpekat att om man ändrar THEN till GOTO går det bra, men jag rekommenderar att man försöker dela upp raden på flera. Ev får man då upprepa villkoret på flera rader.

Dolda fel

Somliga fel visar sig inte i förväg utan först när man börjar köra. Några exempel:

* **ONERRORGOTO 0** och **BLANKTECKEN**

Det skall heta ON ERROR GOTO (ingen nolla och två mellanslag). Om datorn står i EXTEND-läge, vilket nog är det vanliga, blir blanktecknen i vissa lägen viktiga. Man kan inte skriva LIST130 utan det måste vara LIST 130.

På grund av en bug i ABC80-tolken skrivs ett extra blanktecken på vardera sidan om varje kolon vid LIST. Dessa blanktecken försöker en ABC800 läsa in en mening i, och ibland måste de tas bort för att progrmmet skall fungera.

Över huvud taget får man se upp med programrader som innehåller flera satser åtskilda av kolon. Jag har inte helt lyckats reda ut varför det ibland hakar upp sig. Ett ON ERROR GOTO <nr> måste ibland flyttas ut och stå i en rad för sig.

Speciellt lurigt kan det vara i satser av typ

CHAIN "DR0:MENY" och DATA DR0:MENY

eftersom ABC80 sätter in mellanslag "DR0 : MENY" som BASIC II inte kan tolka rätt.

* Mera **ON ERROR**

ABC800 fångar upp många fel som inte ABC80 klarade. På det lilla kortet under tangentbordet på ABC80 finns en prick vid de fel som kan fångas upp med ONERRORGOTO. På ABC800 kan alla fel uppfångas, även vissa sådana som på

ABC80 ger programavbrott av mera allvarligt slag. Detta kan ibland göra det svårt att hitta felet.

Jag har stött på följande exempel. I en meny skall man välja att göra CHAIN till ett visst program. Programmet hittas och ABC800-datorn försöker ladda in - men det visar sig vara ett ABC80-program i BAC-format. Då träder felhanteraren i funktion och man återvänder till menyn (i det här fallet) precis som om man inte gjort något val. ON ERROR skulle fånga upp att man t ex skrev en bokstav i st f en siffra vid valet, och man tänker inte genast på den här typen av fel.

Lösning: Ta bort ON ERROR och kör programmet igen. Då blir det programavbrott, man kan se på vilken rad och med ;ERRCODE får man veta vad det var för fel. Sätt sedan dit ON ERROR igen.

*** INPUT A$,B$**

Om ett ABC80-program kräver inmatning av flera variabler i samma INPUT-sats, skall inmatning ske med kommatecken emellan: "Sven","Wickberg". Om man matar in bara det ena värde skriver datorn ett nytt frågetecken och väntar på nästa.

Så fungerar inte BASIC II. Den SKALL ha BÅDA värdena med komma emellan, annars blir det felmarkering och ev programavbrott. Det gäller alltså att se till att man har ett ON ERROR etc på plats före en sådan inmatning. Och återhoppet måste ske till just den ON ERROR-raden, annars blir det programavbrott om felet upprepas.

*** RND och avrundning**

Instruktionen RND ger i båda maskintyperna decimaltal mellan noll och ett. Vill man slumpdra ett heltal mellan 1 och 6 (tärningskast) skriver man INT(RND*6)+1.

RND*6 ger tal större än noll men mindre än 6. INT() trunkerar (klipper bort) decimalerna och ger heltal 0-5. För att få dem 1-6 lägger man till 1.

Det hela tar mindre plats och går fortare med heltalsvärden och på ABC80 kan man enklast skriva:

```
X%=RND*6%+1%
```

Något INT() behövs inte eftersom decimaltalet på höger sida automatiskt förlorar sina decimaler genom trunkering när det tilldelas heltalet X%.,

Men i ABC800 spelar oss BASIC II ett spratt. När flyttal förvandlas till heltal på detta sätt, genom tilldelning till heltalsvariabel, sker inte TRUNKERING som i ABC80, utan KORREKT AVRUNDNING.

Man kan alltså råka ut för att få talet 7 med ovanstående rad!

Lösningen är förstås enkel: ingen smartprogrammering - ta med INT() som det var från början.

*** INP(56)**

Det säger sig kanske självt att det som inte är BASIC inte kan läsas av en annan maskin. Instruktioner som börjar POKE, PEEK, OUT, INP och CALL fungerar vanligen inte om man inte ändrar argumenten, dvs de siffror som anger adresser och portar som man vill göra något med.

INP(56) i ABC80 avläser tangentbordet och håller reda på dels om en tangent tryckts ned sedan sist och vilken tangent det var. Dessa funktioner ser annorlunda ut i BASIC II.

SYS(5) är tangentbordsflaggan - 0 om ingen tangent tryckts, annars 128. Nollställs med GET.

Z=SYS(6) för över ASCII-tecknet för senast nedtryckta tangent till inmatningsbufferten, där den kan avläsas med GET.

```
Z=SYS(6):GET W$
```

Uträttar alltså samma sak som:

```
W=INP(56):W$=CHR$(W)
```

*** CHAIN**

I ABC80 betyder CHAIN "" detsamma som END+NEW. Detta fungerar inte i BASIC II, där man i stället får skriva CHAIN "NUL:". Problemet kan dyka upp i en meny där olika menyval länkar till olika program och man vill ha ett val för "AVSLUTA". Man kan naturligtvis också nöja sig med att i det fallet få programavbrott och felmarkering, även om det inte ser snyggt ut och kan irritera den ovane.

*** EXTEND : INTEGER**

I EXTEND-läge, som vi redan varit inne på, kan BASIC II ta emot långa variabelnamn, vilket för det mesta är en fördel. Nackdelen är att mellanslag i vissa fall blir betydelsefulla på ett sätt som ställer till problem.

Ofta är det också en fördel med INTEGER-läget. Man slipper skriva alla % efter variabelnamnen. Men om man laddar in (resp sparar) ett ABC80-program som är skrivet utan % kan man råka ut för att variablerna tolkas som heltal. Detta kan vara mycket pinsamt om de i verkligheten är flyttal och skall avläsa decimaler. Felet kan vara svårt att se för den ovane, eftersom heltalen i INTEGER-läge ser ut "som vanligt" (medan flyttalen skall ha en punkt efter sig).

Det kan vara klokt att ligga i NO EXTEND:FLOAT-läge när ABC80-program laddas in.

*** Spara med SAVE**

När alla anpassningarna är gjorda och programmet fungerar är det vist att spara det i sin ABC800 med SAVE. Detta går bra om man i fortsättningen alltid skall köra det i ABC800. Men det skall man kanske inte alltid och då har vi nästa problem:

3: Samtidigt för ABC80/800

Hur gör man om man vill ha program så skrivna att de kan avläsas av och köras i både ABC80 och ABC800?

Ingenjörsfirman CAT har i sitt nätverk alla programrutiner tillgängliga för båda maskintyperna. Flertalet program är lika, men i några fall skiljer man dem åt och kallar dem då .80 resp .800.

Kan de så kan väl vi, i varje fall det som går att göra i BASIC.

För det första måste programmen vara lagrade i .BAS-format. Detta leder till att mycket långa program kommer att kännas en aning trögstartade, i varje fall i ABC800. Programmen skall först läsas in och sedan kompileras. Men det får man stå ut med. Är programmen någorlunda korta märks det inte så mycket.

För det andra gäller det att undvika instruktioner som ser olika ut i de olika maskinerna. Man måste givetvis avstå från BASIC II-finesser av alla slag, och man får sorgfälligt undvika de skrivsätt som resp maskin inte själv kan rätta till.

Man får lov att köra i NO EXTEND:FLOAT. Man kan skriva ONERRORGOTO <nr>, vilket kan tolkas av ABC800, men man får avstå från ONERRORGOTO 0. Ett sätt kan vara att i stället för nollan skriva ett radnummer som ligger sist i programmet med bara REM på. Man räknar ju inte med några fel i det läget. (Ännu smartare: i stället för rem-sats skriv ;'Oväntat fel!')

Man slipper felet med mellanslag kring kolon genom att skriva in och spara motsvarande program med ABC800-maskinen som inte gör några sådana mellanslag. Annars kan man alltid sätta ut anföringstecken:

```
DATA "DR1:PROGRAM"      CHAIN "SYS:MENY"
```

CHAIN"" kan ersättas med CHAIN "NUL:" som visserligen ger felmeddelande i ABC80, men det gör inte så mycket om meningen är att avsluta körningen.

*** INPUT och DATA**

I ABC80 tas alla mellanslag bort, medan i ABC800 bara mellanslag i början och slutet av strängarna avlägsnas.

Kan lösas med att sätta citationstecken kring alla data resp att använda INPUTLINE. (Det heter INPUT LINE i 800 och sparas på detta sätt om man skriver LIST från ABC800 - och då kan inte ABC80 läsa det.

När det inte går

Om man ändå står där med instruktioner som MÅSTE vara med, men som är inkompatibla: INP(56), PEEK och POKE?

Ett alternativ kan vara att låta programmet pröva vilken maskin som är inkopplad. Om detta har det varit en del diskussion i klubbens MSG-system. Men följande adresser kan vara användbara: 39 och 14.

Address(39) är noll för ABC80 och icke noll för de andra. Det betyder att PEEK(39) är sant om en ABC800 är inkopplad, annars ej. (Detsamma gäller om Address(14) - men jag har ännu inte kommit på vad det annars är för skillnad eller varför det här så här.)

Så man kan skriva:

```
IF PEEK(39) THEN <800-instr> ELSE <80-instr>
```

I värsta fall måste man ha olika program. Då kan man börja körningen med ett program som frågar vilken maskin man har (om det inte kan undersökas automatiskt som ovan) och sedan gör CHAIN till rätt variant.

Allting går om man bara har tid och pengar. Pengar har hackern inte för gott om, men påhittigheten har visat sig enorm inom klubben.

Jag väntar med stort intresse på synpunkter och ytterligare knep.

Nästa steg

Nu har (tyvärr) IBM intagit Luxors ledande plats i smådatorförsäljningen. Allt fler inrättar sig efter IBM-marknaden, så även CAT.

I ABC-bladet nummer 4, 1985 fanns inte mindre än TVÅ annonser om konvertering av ABC-program till IBM-standard. En vacker dag (den kanske är här när detta kommer i ABC-bladet) behöver vi en artikel om anpassning till IBM i CAT-NET med både Luxor och IBM anslutna.

Vem skriver den artikeln?

<1384>
Sven Wickberg

Reaktionstest

Det finns här och där i landet apparater med vilka man kan mäta sin reaktionshastighet. Man lägger i en slant (förr var det 10 öre) och väntar sedan på en signal. Vid signalen börjar myntet falla och man stoppar det genom att trycka på en knapp. Myntet fastnar då längs en skala som talar om hur lång falltiden har varit. Sälunda kan man få reda på sin egen reaktionsförmåga (och bedöma om man bör köra bil mer den dagen).

I Avancerad programmering på ABC80 fanns ett program som uträttade något liknande. Häromdagen hade jag anledning att snabbt smälla ihop ett reaktionsprogram och fann då att det kunde göras bra mycket enklare.

Programförklaring

Programmet är kanske inte så märkligt i sig, men det innehåller några intressanta delrutiner, nyttiga att lära för novisen - om vi nu längre har några noviser på ABC80 som inte längre tillverkas...

Först gäller det att få en signal att uppträda slumpvis. Med ABC80:s slumptalsgenerator kan man komma in på oförutsedda delar av lyckohjulet med RANDOMIZE, som alltså måste stå först (100). Sedan har jag valt att dra slumptal efter slumptal och göra signalen om slumptalet är mindre än 0.005. Vid större tal kom signalerna så fort att man knappt hann med, vid mindre är det för långt mellan dem, men det kan vara en smaksak som var och en kan pröva sig fram till.

Om RND-villkoret är uppfyllt sätter vi i gång klockan på rad 200. Eller rättare sagt, vi ställer klockan.

Klockan

ABC80:s klocka har den egenheten att den räknar ner klockbytarna baklänges. Adressen 65008 rör sig fortast och går från 255 ned till 1, ett steg för var femtiondels sekund.

När den räknat ut ett varv, vilket tar 5.12 sekunder, förs "ett i minne" över till 5009 som i sin tur räknar ned från 255 till 0 och behöver ungefär 20 minuter för att gå varvet runt och göra motsvarande manöver på 65010. (I själva verket är det krångligare än så, men detta räcker i vårt sammanhang.)

För de flesta ändamål behöver man bara de två första adresserna. Jag har valt att 255-ställa endast dem i rad 200.

I väntan på trycket

Nu väntar programmet på e? tangenttryckning som registreras genom att INP(56) får 128-biten ettställd. I de övriga bitarna lagras ascii-värdet för den nedtryckta tangenten, men det använder vi inte oss av här. Så länge INP(56)<128 stannar programmet på rad 210.

När tangenten nedtrycks går programmet till nästa rad och nu gäller det att ta tid. Den metod som visas är ett smart sätt att få fram tiden i sekunder. Den bygger på Booles algebra (dvs logiska variabler) och har utförligt beskrivits i ett tidigare nummer av ABC-bladet (1983:4 sid 12).

Raderna 500-520 är nog självförklarande, men observera att P och M inte får vara heltal. (Detta är en speciell fälla för ABC800-ägaren som kan ha apparaten i INTEGER-läge, varav mera nedan.)

På rad 600 väljer jag att låta skriva på skärmen tiden för rektionssnabbheten I, antal försök J och medelvärdet M.

M? Så står det inte! Nej, det som står är M korrekt avrundat till två decimaler.

Avrunda till 2 decimaler

Den kortaste tid man direkt kan mäta genom att avläsa adress 65008 (den som rör sig snabbast) är 1/50 av en sekund, eller 0,02 sek. Men datorn minns I som 0,02000 sek, även om slutnollorna inte skrivs ut. Vid medelvärdesberäkningen ,t ex division med 3, går det inte jämnt upp och det kan bli massor av onödiga decimaler, som inte ger någon som helst mening. Beräkningsnoggrannheten kan inte bli större än plus/minus 0,02 sek, alltså två decimaler i detta fall.

Egentligen är även detta för stor noggrannhet, eftersom datorn bara avläser jämna hundradels sekunder, medan medelvärdet även kan bli udda, men låt gå för det.

Om man vill klippa bort decimaler på ABC80 kan man inte göra det direkt. Man måste först se till att alla de siffror man vill spara finns i en heltalsdel. Sedan kan man med INT() klippa bort decimalerna. Därefter får man flytta tillbaka decimalkommat till rätt plats.

Med den metoden sker s k trunkering, dvs man bryr sig inte om hur de bortklippta decimalerna ser ut. Vill man ta hänsyn till att sista siffran skall höjas om den första borttagna siffran är 5 eller mer, måste man göra ett par ytterligare manövrer, och det är vad som skett här.

Först multipliceras M med 1000 vilket innebär att vi flyttar upp TRE decimaler till heltalsdelen. Sedan tilläggs 5, som ökar den sista siffran. Om denna är 5 eller mer kommer även den näst sista siffran (= decimal TVÅ) att höjas. Det var ju detta vi ville.

Nu divideras med 10, vilket gör att den tredje decimalen, som vi alltså inte vill ha, hamnar till höger om decimalkommat och försvinner när vi gör INT(resultatet).

Genom att dividera återstoden med 100 får vi tillbaka decimalkommat på rätt ställe och två, korrekt avkortade, decimaler.

Puh! Det är nog lättare att göra denna operation än att förklara den. Hängde du med? Om inte finns bara ett råd: sälj din ABC80 och köp en ABC800. Där kan du få hela operationen utförd "gratis" med PRINT USING "§§.§§" M - fast då måste du vara säker på att du inte behöver fler heltalssiffror än du sätter ut "brädgårdstecken" för till vänster om decimalpunkten.

Tangentbordsbuffertern

Nu återstår bara att förklara rad 120. Jag råkade ut för att trycka för tidigt, alltså innan det kommit något pip. Tangentbordet har en buffert som lagrar ett tecken, och därmed kom denna för tidiga tryckning att "ligga på lager" till nästa pip. Följaktligen fick jag tiden noll, och det blir en aning fuskigt.

Tangentbordsflaggan finns i adress 65013 på ABC80 och genom att nollställa den avlägsnas spåren av den för tidiga tryckningen. (Det går också bra med POKE 65060,0 som har samma verkan, men av en annan anledning.)

Så där, programmet är klart! Man måste avbryta med CRTL-C, såvida man inte utnyttjar de lägre bitarna i INP(56) för att definiera någon stopptangent. (Hemläxa)

Men vänta ett tag! I dessa tider blir en artikel om ABC80 inte komplett utan ett

Tillägg för ABC800-ägare

På ABC800-serien finns inte INP(56)-möjligheten, utan man får använda något annat. Dessutom fungerar klockan på ett annat sätt, varför programmet måste kompletteras med ett par speciella rutiner.

SYS(5,6,8)

Enligt handböckerna är SYS(5) en tangentbordsflagga. Det betyder här att SYS(5)<>0 ända till dess en tangent tryckts ned. Vid tangenttryckningen får SYS(5) värdet 128 som står kvar till dess man utför GET eller INPUT.

Om man inte behöver veta vilken tangent som tryckts (och det behöver man inte i programmet REAKTION) kan man alltså ersätta den aktuella raden med:

```
210 IF SYS(5) GET Key$ ELSE 210
```

Om man behöver veta vilken tangent som tryckts - man kanske skulle vilja avsluta med S i stället för med CTRL-C - använder man SYS(6). Den har egenskapen att återföra värdet för senast nedtryckta tangent till tangentbordsbufferten, där den avläses med t ex GET.

```
Z=SYS(6)          ! återför värdet av
                    senaste tangent
GET Tangent$      ! avläs tangentbordsbuf-
                    ferten
```

och sedan jobbar man vidare med värdet Tangent$ t ex

```
IF Tangent$='S' THEN STOP
```

Observera att ordet END måste stå ensamt på en rad i BASIC II. Det är därför jag använder STOP här.

Klockan

Klockan i ABC800-serien är också litet annorlunda än hos ABC80. För det första ligger är de aktuella adresserna

```
-14  timme
-13  minut
-12  sekund
-11  femtiondels sekunder
```

Det är alltså -11 som rör sig fortast (och bör avläsas först). Den går från 1 till 49 och lämnar "ett i minne" hos -12 när den kommer till 50.

Adressen -12 i sin tur går från 0 till 59 och lämnar ett i minne hos -13 vid 60.

(På samma sätt räknar -13 upp till 59 minuter och -14 upp till 23 timmar - och för den delen -15 till 31 dagar, -16 till 12 månader och -17 till 99 år. Allt det här

får man ut automatiskt med ;TIME$. I ABC806 finns en batteridriven klocka som man inte behöver ställa mer än en gång, medan man i ABC800 måste ställa klocka och datum efter varje kallstart. Hur det är med klockan i ABC802 vet jag inte - den tycks kunna gå litet hur som helst om man har otur.)

Om man för reaktionsprogrammet nöjer sig med att notera hela och delar av sekunder kan man skriva raderna 200 och 220 som nedan:

```
120 POKE 65506%,0%
200 POKE -12%,0%,0%
---
220 I=PEEK(-11%)/50+PEEK(-12%)
---
600 ; i, J, : ;USING "§§.§§" M
```

Jag har använt notationen PEEK(-11) i stället för PEEK(65525), eftersom det första verkar vara lättare att komma ihåg. På ABC80 bör man dels använda den positiva versionen och dels skriva den som heltal, annars både går det långsammare och tar mer plats. Jag vet inte om detta gäller även ABC800, men det kan vara ide att undersöka saken.

Varning för heltal

ABC800 har fördelen att kunna ställas i INTEGER-läge så att man slipper alla %-tecken när man skriver programmet. Har man otur skriver man alltså heltalsvariabler även där det inte skall vara några. Det hände mig när jag skulle konvertera programmet till ABC800. Jag blev förvånad över att få tiden noll, vad jag än gjorde!

Både I, P och M måste vara flyttal, endast J är heltal.

Bufferten, och PRINT USING

Rad 120 nollställer tangentbordsbufferten precis som i ABC80- programmet men det är en annan adress, och PRINT USING omnämndes tidigare i denna artikel. På min apparat kunde instruktionen USING inte användas på samma rad som ;I, J, - jag vet inte varför (har för mig att det har lyckats i andra sammanhang). I varje fall gick det bra med ett kolon emellan. Man kan också skriva PRINT USING på en särskild rad, men jag ville behålla likheten mellan programversionerna.

Om inte ABC80 fanns...

I ett program från början skrivet för ABC800 skulle jag troligen använda flerradiga funktioner för att definiera procedurer, och dessutom långa variabelnamn, men här ville jag bara göra en enkel konvertering av ABC80-programmet. Ofta kan man skriva programmen så att de går att läsa och köra på både ABC80 och ABC800, men det går jag inte inte på nu.

Hur fort går det?

En sista något undrande anmärkning. När jag provar ABC800-rutinen får jag kortare reaktionstider än i ABC80-rutinen. Eftersom det är hägst tveksamt om en ABC800 kan ha ett sådant förädlande inflytande på min fysik (jag har förresten jobbat med en ABC80 försedd med Gejos RAM-minne och inladdad ABC800-tolk), måste det bero på mindre "spilltid" vid ställning och avläsning av klockan. Hur det hänger ihop med den saken överlämnas i vanlig ordning åt läsekretsen att försöka fundera ut.

<1384>
Sven Wickberg

```
 10 REM reaktion
100 RANDOMIZE
110 IF RND<.005 ; CHR$(7%); ELSE 110
120 POKE 65013%,0% : REM neutralisera f
    eltryck
200 POKE 65008%,255%,255%
210 IF INP(56%)<127% THEN 210
220 I=(PEEK(65008%) XOR 255%)/50+5.12*(
    PEEK(65009%) XOR 255%)
500 J=J+1 : REM antal försök
510 P=P+I : REM totaltid för försöken
520 M=P/J : REM medelvärde
600 ; I,J,INT((M*1000+5)/10)/100
700 GOTO 110
```

```
 10 REM reaktion.800
100 RANDOMIZE
110 IF RND<.005 ; CHR$(7%); ELSE 110
120 POKE 65506%,0% : REM neutralisera f
    eltryck
200 POKE -12%,0%,0%
210 IF SYS(5%) GET W$ ELSE 210
220 I=PEEK(-11%)/50+PEEK(-12%)
500 J%=J%+1% : REM antal försök
510 P=P+I : REM totaltid för försöken
520 M=P/J% : REM medelvärde
600 ; I,J%, : ; USING "§§.§§" M
700 GOTO 110
```

Möte ABC800

Utdrag ur MSG-systemet

(Text 1803) Kent Berggren <6019>
Ärende: ABC806 BILDMINNET
Hur hanterar den det vanliga bildminnet. Ligger det på samma plats som i en ABC802?

(Text 1808) Benny Löfgren * <2615>
Ärende: Step på FD-55F
Jodå, man kan läsa 40-tracksskivor på 80-tracksdrivar, allt man behöver göra är att ställa om en switch på controllerkortet, det låter sig göras även på 4112. Vidare behöver man ett program som är speciellt förberett för ändamålet (man måste t ex konvertera clusterstorleken), ett sådant program är COPYLIB, vilket ju alla har.

:t 1810) Benny Löfgren * <2615>
.de: ABC806 BILDMINNET
Det ligger på precis samma plats. Däremot så måste man göra lite special för att komma åt attributminnet, om du vill det.

(Text 1811) Göran Engelbo <369>
Ärende: Angående adress 39
Kommentar till inlägg i ABC-Bladet 3/85, sid 36. Det där om ABC80 är helt fel!
är är det:
800M
2 - 800C
3 - 802
4 - 806
På gamla tolkar (800M, 800C och faktiskt också de allra första 802:orna) ligger i denna byte koden 10 som innebär att man inte vet vilken maskin det är. (DTC är alltså fel slutsats!)
Ett tips är att titta på 65364. Den är 40 på 800C och 80 på 800M om den inte ställts om från programmet. Den metoden täcker dock inte in de allra första 802:orna, men de är *mycket* få.
Mvh/Eng

(Text 1812) Sven Wickberg <1384>
Ärende: CAPS LOCK
Jag testade OUT 34,8 etc från ABC-bladet och fann:

1) att det är tvärtom på den 806 jag provade på (PÅ=AV, AV=PÅ)
2) när man slagit av den på detta sätt kunde man inte längre få på den (eller också tvärtom - ursäkta att jag inte skrev upp det och jag kommer inte åt att prova nu - prova själva får ni se)

Det vore kul med kommentarer från allvetarna.

(Text 1814) Benny Löfgren * <2615>
*ende: CAPS LOCK
:tum är att denna 'finess' faktist står i anualen till datorn (minns inte om det var dator- eller BASIC-manualen).

(Text 1817) Patric Ljung <5455>
Ärende: Mer om ABCUTE
Vad syftar du på?
Att man kan sätta flera prommar på varann går ju lika bra på vilken burk som helst.
Man kan iofs använda 256 tecken på C802 men då måste man ange grafikbd, vilket kan bli jobbigt. Eftersom bit 7 (värde 128) inte är kopplad till teckenprommet utan till en inverterare som inverterar tecknet på skärmen. Förövrigt genereras inte grafik för att aktuell kod ligger i bildminnet utan för kodens teckenfont har bit 6 och 7 satta på ett sätt som medför grafik. De möjligheter som finns är flash, Hide,Graphic,Normal.
Detta innebär att man kan sätta en rad i grafisk-mod med valfri kod. Förresten, är det någon som kan få Flash på sin 802a??? Jag kan med min, men inte min kompis. Jag tror det beror på teckenprommet.
Vill ni som har 802or testa? Skriv ; FLSH 'Blinkar det???'
Moss / Patric

(Text 1821) Patric Ljung <5455>
Ärende: CAPS LOCK
Det ligger faktiskt till så här:
OUT 34,8 sätter på Capslock och gör det därmed omöjligt att stänga av den. När man sedan skriver OUT 34,136 stängs inte Capslock av utan det blir bara möjligt att stänga av den genom att trycka på den.
Moss / Patric

(Text 1863) Kent Berggren <6019>
Ärende: Bild-minnet ABC 802
Jag har labbat lite med port 56 och 57. Men jag får det inte att fungera riktigt.
out 56,12,57,190
Det borde flyttat upp minnet till 49152, men jag får med massa skräp från gamla skärmen. Vidare skrivs ny inknappad text oxo på nya skärmen. Det borde den väll inte göra. Jag har inte ändrat reg 14 o 15.
Är det någon som kan reda ut vad jag skall göra, för att få det att fungera riktigt.

(Text 1867) Kent Berggren <6019>
Ärende: Bild-minnet ABC 802
Är det så att den skall ha ascii värdet ut på porten?
out 56,12,57,(+3 ascii värden)
Jag blir inte riktigt klok när jag läser i bit för bit om CRTC 6845
HJÄLP MEJ!

(Text 1888) Harald Nautsch <4479>
Ärende: HR-grafik i CPM (?)
En annan fråga: Skulle det vara möjligt att använda grafiken när man kör CPM på 806?

(Text 1889) Patric Ljung <5455>
Ärende: HR-Grafik (adressering)
Hur du adresserar HR-minnet 128K på ABC806 står i ABC-bladet Nr 2 1984. Om du inte har den kan jag förklara, annars tycker jag det är onödigt, jag kan kommentera den litet.
mvh / Patric

(Text 1909) Kent Berggren <6019>
Ärende: Bild-minnet ABC 802
Skriv out 56,13,57,190
i ett prg, där efter gör du ett GET skall du se att minnet hoppar tillbaka igen.

(Text 1910) Patric Ljung <5455>
Ärende: Bild-minnet ABC 802
Tja GET är väl inte samma sak som interuptet, prova att skriva:
10 OUT 56,13,57,190
20 WHILE INP(5)=0
30 WEND
40 FOR I=0 TO 5000 : NEXT I
35 OUT 34,7 ! Piiip
Om du kör denna rutin kommer nog inte skärmen att hoppa tillbaka förrän loopen har körts efter piipet, och du trycker någon tangent.
Det är nog GET-funktionen som ställer tillbaka minnet, inte interuptet.
Moss / Patric

(Text 1913) Magnus Bodin <4304>
Ärende: CTRC: Skillnader ABC800/2/6
Vilka är de egentliga skillnaderna?
Jag har bl.a. försökt ställa om markörens utseende, men inte fått samma resultat på de olika maskinerna.
Fråga2: Vilka tecken visas egentligen på skärmen om man lägger ned koder direkt i bildminnet? Hur ser PF-koder ut? Och koder < 32? Även här blir det skillnader på 800/2/6. Finns det några "osynliga" tecken på 806:an?
Magiska hälsningar Magnus

(Text 1914) Bo Kullmar * <1789>
Ärende: Comporian
Enligt ett brev från en advokat så gick Comporian i konkurs den 17:e dennes! Synd, för de är skyldiga klubben 3 775 för annonsen i senaste ABC-Bladet. Vi får väl inget, misstänker jag. Detta skriver jag som en upplysning till er som använder deras program. Kanske tar något annat företag över, det skulle ine förvåna mig. Är det någon Linköpingsbo som vet mera? I så fall skulle vi kanske flytta diskusionen till fritt.

(Text 1919) Patric Ljung <5455>
Ärende: CTRC: Skillnader ABC800/2/6
1. På ABC800C är det annorlunda, där kan man inte ställa om markören. Det är väl nåt liknande som på ABC80. 802/806/800M ska väl vara samma. Man kan på vissa burkar ange startlinje nedanför slutlinje, då får man markören "delad".
2. På ABC806 visas alla tecken på skärmen, det vill säga att alla adress ledningar är kopplade => 256 tkn. från 0 till 255. Här ligger de första grafikteckner på 1-31 (0 är blank), 32-127 ligger de vanliga teckner. 129-159 ligger grafiktecknens andra del, (128 är blank), 160-255 ligger vanliga tecken igen (används inte) kan användas som alternativ tecken uppsättning. Jag gör det i min 806a.
På ABC802 är bara 0-127 kopplade 128-255 ger samma tecken fast inverterat. 0 är tom, 1-31 innehåller div styrkoder för grafik, hide, flash, normal. 32-127 är vanliga tecken. Grafik teknen återfås genom ; GWHT eller nåt dylikt.
På 800M är jag inte så säker, det är väl bara 0-127 som används där med, men 1-31 innehåller inte några styrkoder, tja, jag vet inte.
Hmm, var det nåt mer?
Moss / Patric

(Text 1920) Patric Ljung <5455>
Ärende: Comporian
Hmm, de har väl inte gått i konkurs men måste se över läget. Jag pratade med Örjan häromdagen och han sa att det eventuellt skulle reda upp sig. Nästa vecka (första hela veckan i November) skulle saken vara avklarad, dvs så skulle man veta då hur det blir.
Vi får avakta och se. ABC-klubben är inte de enda, men det är inte så kul, jag har jobbat litet åt Comporian.
Moss / Patric

(Text 1925) Benny Löfgren * <2615>
Ärende: CTRC: Skillnader ABC800/2/6
Ett tillägg: i 800M kan man få fullständig 256 teckens uppsättning om man byglar om videokortet och sätter i en 2732-teckenprom i stället. Då fungerar det som på 806, dvs alla 8 bitarna används för att slå upp rätt tecken i teckengeneratorn. Kan med fördel utnyttjas för att få inverterade tecken eller underline.

(Text 1936) John Stagg <5741>
Ärende: CTRC: Skillnader ABC800/2/6
1. Nej, ABC802 har inte precis likadant styrning av markören som andra 800 varianter. Jag har tittat på två olika 802 promversioner och både betedde sig något annorlunda jämfört med de inlägg man läser här i monitorn beträffande kursorutseendestyrning. I senaste promverionen är det väldigt svårt att ha vissa cursorutseende därför att tolken "gör om" cursorn rätt ofta.
Mvh TAP

(Text 1940) Patric Ljung <5455>
Va' de' nåt mer? -Ja.
Man kan då känna av om flera tangenter är nertryckta samtidigt, detta resulterar i att man både kan gasa och svänga höger och skjuta i ett spel till exempel.
Men man måste alltså hålla reda vilka tangenter som är nertryckta. När man trycker ner en tangent genereras ett interupt, och ytterligare ett när man släpper. Man måste således vara ganska snabb, om man vill hållas i basic bör man lägga upp en buffert i nån variabel som interuptet lägger upp alla koder i, annars kan man låta interuptet sätta bitar som indikerar att en viss tangent är nertryckt. Om man nu vill kunna shifta en tangent måste man således ha en flagga som indikerar om shift ärе nere eller inte.
OBS. Tidiga versioner av ABC99 klara inte UP/DOWN-mod
Vidare finns en del finnesser hos ABC77 (ABC55)

1	Bell från
129	Bell till
2	5 ms klick från (gäller tangent tryck)
130	5 ms klick till
4	20 ms klick från
132	20 ms klick till
5	Repetering från
133	Repetering till
6	Normal ASCII-mod
134	UP/DOWN-mod
7	Bell (pip i högtalaren)
8	CAPS LOCK från (Normal mod)
136	CAPS LOCK till (Normal mod)
8	CAPS LOCK indikator från (UP/DOWN-mod)
136	CAPS LOCK indikator till (UP/DOWN-mod)
9	ger 5 ms klick
10	ger 20 ms klick

Det var nog allt.
Moss / Patric

(Text 1961) Magnus Bodin <4304>
Fråga: Nya kommandon å ABC800.
Vad jag menar alltså, är att om jag skapar ett nytt kommando med namnet "lax" som exekveras direkt, dvs direkt i syntaxchecken, hur kan jag då i "lax-rutinen" känna av om användaren skrivit
lax
eller
1049 ;'Hejsan' : Lax
Så att jag kan säga ifrån med felkoden 144 eller liknande.
Hur kollar man om man är i RADCOMPILE-mode, alltså??
Magiska hälsningar Magnus

(Text 1962) Kent Berggren <6019>
Ärende: V24:
Följande rutin kan läsa av DCD och CTS på V24: och PR:

```
; 8        32
; DCD      CTS
;  LÄSER AV PÅ PORT-A
BAUD:   LD   A,5
        OUT  (33),A
        LD   A,226
        OUT  (33),A
        LD   A,16
        OUT  (33),A
        IN   A,(33)
        RET
;  LÄSER AV PÅ PORT-B
        LD   A,5
        OUT  (65),A
        LD   A,226
        OUT  (65),A
        LD   A,16
        OUT  (65),A
        IN   A,(65)
        RET
```

De är de enda två bitarna jag vet att man kan läsa av på porten. Hoppas det hjälper någon lite.

(Text 1966) Valter Pettersson <2008>
Ärende: Överföring V24:
Ett liknande problem som jag ej lyckats lösa :
Jag läser indata på V24: men vet ej mängden indata, ej heller filsluttecken (som inte alltid sänds). Programmet vet alltså ej att data från sändaren är slut utan hänger i nån GET eller INPUT.
Hur kan detta tillstånd brytas av "operatören" från tangentbordet gärna med Ctrl... tryckning)?
Ctrl-C går ju men det bryter ju programmet (Basic) och är oacceptabelt.
Assembler går väl på nåt sätt men finns inget enklare?

(Text 1970) Gunnar Faith-Ell * <2733>
Ärende: Överföring V24:
Du bör ha optionsprom av lägst version 7-2, annars kan ett byte rekommenderas.
Från BASIC kan du sedan känna av om det finns tecken i bufferten med "IF PEEK2(PEEK2(65500)+6)<>0 GOTO <hämta-ettteckenmedget>". Se "manualen" till optionsprommet, den beskriver det ordentligt med programextempel.

(Text 1996) Kristoffer Eriksson <5357>
Fråga: Nya kommandon å ABC800.
Vad är det för mystiskt kommando du hittat på som måste exekveras redan vid syntaxkollen/kompileringen, och dessutom gör skillnad på om det ingår i ett program eller ges direkt??? Räcker det inte att kompilera den som vanligt och kolla vid exekveringen; kommandos utan radnr exekveras ju direkt, och då kan du helt enkelt kolla bit 2 i adress 65315 (INT.DIR i Y.INT, du har väl Bit för Bit...). Annars skulle jag föreslå att du kollar om radnr i den delvis kompilerade raden är .0, vilket betyder avsaknad av radnr. Radnumret ligger i två bytes på adressen PEEK2(65306)-250 (Y.GENEND) tror jag, men jag har inte testat.
HL = Pekar till textrad som kompileras (inbuffert)
DE = Pekar på kompileringsbufferten där internkoden byggs upp (DEstination)

(Text 2005) Gerry Eriksson <4842>
Ärende: 31250 bit/sek
SIO (V24:) kan du öppna på vanligt sätt (OPEN 'V24:....). Ändra sedan baudraten för kanalen med CTC:n
CTC ch0 TxC vid splitt speed, CTC ch1 RxC och TxC när splitt speed ej byglad.
För 31250 Baud så blir det:
OUT 96,71,96,3 ! TxC CTC ch0
OUT 97,71,97,3 ! RxC 1
Detta förutsätter att SIO:n jobbar i /16-mod vilket den gör om man öppnar kanalen via OPEN. CTC är i countermode och klockas via Exteral Clock input som är ansluten till en 1.5 MHz signal (CPUclockan/2).
Delningsfaktorn är 3.
För andra baudrates mellan 366 - 38k Baud räknas delningsfaktorn som skall laddas ut med:
N=1.5E6/(Baudrate*16)
För baudrates under 366 Baud skall CTC:n sättas i TIMER-mode.

(Text 2015) Kristoffer Eriksson <5357>
Ärende: FGCTL
LD C,7
LD B,Färgregister
OUT (C),Färgvärde
Det finns 16 färgregister (0-15) en för varje färgnr som kan lagras i grafikminnet. Färgvärdet bestämmer vad det blir för färg på skärmen och består egentligen (om jag inte minns fel) av två färgkoder om vardera fyra bits, varav det mest värda biten anger förgrunds/bakgrunsfärg och de övriga Röd, Grön och Blå i någon viss ordning.
LD A,Text+4
OUT (54),A
där Text är 0 eller 128 väljer om textskärmen ska visas eller inte.

(Text 2017) Bo Kullmar * <1789>
Ärende: Hur ska det gå för Luxor (och oss)?
Luxor struntar i hobbyisterna. Hade det inte varit så att vi numera har många "företagsmedlemmar" så hade de inte varit intresserade av oss.
Hållera annars med om att ABC-Klubben nog är borta om 10 år, men några år till bli väl vi och Luxor kvar.

(Text 1943) Bo Kullmar * <1789>
Ärende: Comporian
Ja, Patric har nu uppmärksamma mig på den exakta lydelsen i brevet. Jag har det inte här och skrev bara vad jag kom ihåg. De har alltså inställt betalningarna och uppdragit åt en advokat att utreda förutsättningarna för fortsatt verksamhet. Alltså inte konkurs.

(Text 1949) Thomas Andersson <3917>
Ärende: Listskydd
Vad är egentligen vitsen med att listskydda systemprogram ?
Om man kör t ex Lib och har en 806:a så skulle det ju vara trevligt om man fick se filnamnen i färg och inte som nu i svart-vitt. Detta är ju lätt att ordna om man blev insläppt i programmet.
Mvh T Andersson

(Text 2020) Jan Holmberg <3141>
Ärende: Lite info från Luxor:
Som bekant går f.n. inte tangnetbord 99 att använda ihop med programmen Teledata-inmatning 800, Teledatahämtning 800 och VT-102 (från Autocode).
Det hela beror på en bugg i 99 bordet.
Luxor kommer nu att gratis byta ut ett prom i bordet så att programmen sedan går. Felet har att göra emd UP/DOWN någonting.
Alltså ni som har 99 bord och vill byta prom kontakta er återförsäljare om några veckor så byter de ut det (garantiärende).
En ny version av Teledatainmatning 800 och även av Teledatahämtning 800 kommer runt jultiden.
Den nya versionen kommer att innehålla en del nyheter såsom styrning av modem 800 m.m.
De nya programversionerna kommer även att gå att lagra på winchester (vilket de inte går idag).

(Text 2024) Peter Goldmann <5080>
Ärende: Autostart/Reset
ABC806,ABC832,gamla controllerkortet ger med skiva som innehåller ADDOPT.ABS och BASICINI autostart vid tillslag av strömmen. Men om jag ändrar i den ********.BAC fil som jag sedan skall köra så blir det problem med Autostart/Reset.
Exempelvis så har jag lagt QZ.806 på en skiva i BAC-format. När jag sedan läste om inverterad bild här i monitorn, så la jag in en ny programrad som ger blå bakrund och vit text. När jag sedan sparar den nya versionen med SAVE så tappar jag Autostart funktionen. Men jag kan tyvärr inte kolla detta direkt utan det verkar som om några bitar finns satta i minnet som inte påverkas av RESET, när väl programmet väl är kört en gång.
Följande händer då vid kallstart:
Jag trycker på RESET-knappen upprepade gånger och kanske jag får napp - och det startar - men si - ofta kan då ERROR 43 eller meddelandet SKIVAN SKRIVSKYDDAD komma upp på skärmen.
Lägger jag nu in en liten snutt som ;'TEXT' eller ett litet program som jag kör, då kan jag få igång min skiva med RESET utan fel.
Vad är det för liten flagga eller BIT som sätts?
Har det att göra med UFD-DOSET? Jag låter nämligen BASICINI starta DRINI som startar QZ.806.
Bo Kullmar har ju beskrivit hur man gör i senaste numret av ABC-bladet Och det är noga med att man har en nyformatterad skiva. Men jag tycker ju att om man nu har QZ.BAC sist på skivan i ordning räknat borde ju vid SAVE av en ny version av QZ.806 med ev ändrat kodord eller att jag vill ha * före medlemsnumret - så vill jag ju helst slippa göra om hela proceduren med formattering och kopiering på nytt.
Hälsningar Peter

(Text 2029) Kristoffer Eriksson <5357>
Ärende: Fler frågor om FGCTL
Grafiken visas alltid. När den inte syns beror det på att alla färger är svarta, dvs nollor i alla färgregister.
Som du (väl) vet kan man använda 806:ans grafik i två lägen, en med 256 punkter (eller 240) per linje och 16 färger vilket kräver 4 bits för en färgkod i grafikminnet, och en med 512 punkter och 4 färger där bara 2 bits behövs för färgkoden. Nu är det så att hårdvaran som sköter grafiken alltid arbetar likadant, oavsett vilken upplösning som används. Skillnaden åstadkomms helt mjukvarumässigt.
När en del av en linje ska visas på skärmen hämtas således en halv byte = 4 bits från grafikminnet som får bestämma vilket färgregister som används. Detta register bestämmer färgerna på två pixels på skärmen. Har man valt upplösningen 256 punkter/linje har dessa två pixels samma färg och uppträder som EN punkt, då finns 16 färgregister att välja på med varsin färg. (Får man pixlarna att visa olika färg i stället får man blandfärger som man kan experimentera lite med. Går att göra helt i BASIC). Med 512 punkter/linje och 4 färger används alltså de två pixlarna i varje färgregister var för sig, men eftersom fortfarande 4 bits i stöten hämtas från grafikminnet och dessa ska ge två punkter på linjen och dessa ska kunna vara vilken som helst kombination av de fyra utvalda färgerna blir det lite mer komplicerat. Varje färgregister får representera en kombination av två färger.
Två pixels, 4 färger ==> 4U2=16 kombinationer ==> 16 färgregister ==> Två pixels, 16 färger.(4U2:4 upphöjt till 2)(red)
Översättningen går alltså från 2-bits-färge r till 4-bits-färger (Röd, Grön, Blå, För-/Bakgrund)
Jag förmodar att detta arangemang ger vissa tekniska fördelar. Hoppas du får ut något av beskrivningen, den blev inte så genomtänkt.

(Text 2034) Gunnar Faith-Ell * <2733>
Ärende: Autostart/Reset
Autostarten av program görs med en kommandorad som skrivs in med hjälp av PREABS.BAC som ligger på systemdisken, du kör igång programmet och lägger in ADDOTP.ABS som maskinkodsfil och CHAIN 'QZ.806' som basickommando och sedan så ska det väl inte vara några problem under förutsättning att ADDOPT.ABS är rätt initierad. Några ändringar i programmet kan inte påverka autostarten av programet eftersom den hela tiden ligger lagrad på disken.

(Text 2085) Helene Bodin <5133>
Ärende: ABC expansionslådor
Luxor säljer inte längre expansionslådan ABC890, utan har nu sedan juni 1985 sagt att det 'strax' kommer en ny expansionslåda som då skall heta ABC894, och tydligen bygger på den lilla lådan från ABC834 driven. Den kommer då förmodligen att bara ha 3-4 kortplatser. Ingen på Luxors försäljningsavdelning kan dock lämna några tekniska specifikationer eller leveranstider. Alltså HJÄLP! Vet någon något om exp lådor från Luxor? Vem på Luxor arbetar med att utveckla nya prylar. Alla som jag pratat med skall undersök saken och sedan ringa upp....

(Text 2093) Bo Kullmar * <1789>
Ärende: ABC expansionslådor
Vad jag har hört så är den nya expansionslådan redan ute. Björnhem påstår i alla fall att han har sålt sådana lådor. Den är alltså gjord för att hänga på "andra sidan" på ABC802:an.
Tranfor har dock alltid själv sålt en expansionslåda med 7 kortplatser och den kostar numera 3 600 SEK exkl moms om inte 894:an passar.

(Text 2096) Martin Björkman <3854>
Ärende: ABC expansionslådor
Ett företag i Gbg (Microsupport tel 031-/416665) gör en utmärkt låda med 10 platser och ett rejält tilltaget nätagg (mkt bättre än det i ABC890)
MVH Martin i Gbg

(Text 2104) John Stagg <5741>
Ärende: ABC802 MEM: <=> CAS:
Jag borde kanske förtydliga detta. 65502/3 är alltså systempekaren till MEMBUF. Bufferten flyttas till DOSBUF5 med :
POKE 65502,64000,swap%(64000) .
Sedan borde man spärra DOSBUF5 genom POKE 64913,0 (detta enligt ett tidigare inlägg i detta möte av någon annan medlem).
Hoppas det är klart nu !
John

(Text 2109) Göran Sundqvist <1255>
Ärende: Släcka och tända skärm
kan man göra med:
OUT 56,0,57,126 ! Släck skärm resp.
OUT 56,0,57,127 ! Tänd skärm
informationen liggr hela tiden kvar i bildminnet.

(Text 2111) Kristoffer Eriksson <5357>
Ärende: Släcka och tända skärm
CONSI-rutinen använder (åtminstone på den maskin jag kollat) OUT 56,1,57,0 för att släcka och OUT 56,1,57,80 för att tända. Vad som händer är att CRTC:n får visa 0 resp 80 tecken per rad. Det tycks f.ö duga med vilket udda tal som helst för att bilden ska slockna. Man kan också få den att visa 0 rader, med samma resultat. Det är alltså inte någon särskild IC inblandad som Ulf Hedlund påstod. Bygeln han nämde avläses av CONSI-rutinen med
OUT 67,16:X=INP(67)AND 8
(dvs, motsvarande i maskinkod), då hamnar resultateti X. Den här bygeln är kopplad till DCD-signalen på SIO:ns kanal B.
Vill du ändra tiden innan skärmen slocknar, får du ersätta CONSI-rutinen med en egen.

(Text 2122) Petter Stahre <4501>
Ärende: Hur fungerar FGPICTURE?
Det hade jag också mycket problem med först, men nu förstår jag det! Det fungerar så här:
I ABC806:an så finns det ju minne reserverat för 4 grafikbilder. När du startar datorn, så skriver du i bild nr 0. Datorn är alltså startad så här:
FGPICTURE 0,0,4
Där den 1:a nollan betyder att man ritar i bild nr 0.
Den 2:a nollan betyder att bild nr 0 visas (du kan alltså rita i bild nr 2 tex, och visa bild nr 0).
4:an betyder att det är 4 bilder du använder (bild nr 0,1,2,3).
Alltså:
Om du skriver FGPICTURE 0,3,4 så visas bild 3, medans du kan rita i bild 0.
Om du skriver FGPICTURE 2,0,4 så visas bild 0, medans du ritar i bild 2 osv osv...
Den fungerar alltså som 4 papper.
Du visar någon papper nummer 2, där har du ritat en dator. När han/hon tittar på pappret så ritar du något annat på papper nummer 0.
Detta skulle se ut så här på ABC806:an...
FGPICTURE 0,2,4
Hoppas att du förstår!
/ mvh Petter Stahre

(Text 2132) Ulf Sjöstrand <1208>
Ärende: 3270-emulering
Vad behövs för att emulera 3270 på en "naken" 806? Vore trevligt om kostnadsuppskattning ingick i svar(en)et. 806:an är bara försedd med 815 monokrom men jag vet att 3270 kan ge färg på skärmen. I systemet finns också en 832(DD12), Selicmodem 16. Kan ej användas 3270 är väl synkron kommunikation?
Kan man få simultan utskrift på skrivare som med BoK:s QZ.BAC?
Mvh Ulf S

(Text 2135) Ulf Hedlund <4868>
Ärende: Portarna i ABC 800 M
Dvs, ut går det lika bra att sända tecken på bägge portarna, men in är B-porten lämpligare, då det sitter en SIO där och en DART på kanal A.

(Text 2136) Ulf Hedlund <4868>
Ärende: ABC99
Du tänder diod 1-8 med
OUT 34,(diod-1)*16
och släcker med
OUT 34,(diod-1)*16+128

(Text 2141) Benny Löfgren * <2615>
Ärende: 3270-emulering
Det finns flera olika typer av SDLC 3270-emulering. T ex finns RJE, SNA och BSC. Samtliga dessa är (som SDLC antyder) synkrona protokoll, och de finns att få tag i från både Luxor och DIAB, som har gjort dem. Jag skulle tro att de kostar runt en 5.000 :- styck.

(Text 2145) Harald Sjöztedt <75>
Ärende: Bugg i '800-'806
Nu har jag hittat senaste buggen i BASICII. Ja bugg och bugg, men det är fel i alla fall. Om man händelsevis skulle knappa in följande program:
10 I=17
20 poke 65329,129 : poke 65293,129
30 chain "x2"
Och sedan, av en annan händelse (p<0.001), skulle knappa in något i den här stilen:
10 ;i
Observera att variablerna heter SAMMA ASCII-bokstav ! Spara porg1 med SAVE och porg 2 som "x2" med list. Buggen då ? Jo kör porg1 som chainar "x2" och vad ser ni ? Ja vad ser ni Jo ERROR 2 (Que?) ! Står inte i manualen ! Noggrann som jag är konsulterar jag min lilla röda och min debugger varvid jag finner att:
När COMTOP/BOTTOM pokas upp läses .BAS-filen in på en högre (större) adress än .BAC-filen. När .BAS-filen kompileras tillverkas ett programblock på 20 byte först i filen (Här var den förra buggen, 37 bytes "tappas bort") där bland annat pekaren till vanliga variabler finns. En annan pekare till vanliga variabler finns också, nämligen adress 0ff2eh. Den förra pekaren sätts upp till variablernas nya adress 08100h, men den senare ändras INTE ! Således kommer BASICII vid förkollen att leta efter .BAS-I i .BAC-filens variabellista ! När sedan VARPEK skall räknas ut i rutinen som i min bok kallas VAR FINNS (adress 02f9ch) kommer denna bli NEGATIV = CARRY = ERROR ! Detta förutsätter att det är samma variabel.
Testa med heltal (%) => Error 1
Testa med flyttal (.) => Error 2
Testa med matris => STOP
Testa med sträng ($) => Error 2
Testa med en variabel med annat namn och den hamnar UNDER BOTTOM !
Detta måste alltså vara:
1 :BAC-program med variabel flyttar upp bottom
2 .BAS-program med "samma" variabel qkar ur
Som errcode erhålles en funktion av variabeltypen. Vad sägs om det, mina herrar !

(Text 2147) Gerry Eriksson <4842>
Ärende: Portarna i ABC 800 M
Så länge det inte är frågan om synkron kommunikation så spelar det inte någon roll då SIO:n och DART:en uppför sig lika. Hårdvaran runt omkring sätter i och för sig stopp för splitt-speed (kanske anledningen till att Terminal är förbehållet V24:)

(Text 2149) Benny Löfgren * <2615>
Ärende: Portarna i ABC 800 M
Nja, det spelar en betydande roll, nämligen den att SIO:n i ABC:n kan köras med interrupt, och det kan så vitt jag vet inte DARTen.

(Text 2153) Bernt Johansson <3384>
Ärende: Portarna i ABC 800 M
Det går bra att köra alla 4 SIO/DART kanalerna med interrupt.

(Text 2170) Benny Löfgren * <2615>
Ärende: DART, CTC...
DART = Dual Asynchronous Receiver Transmitter
CTC = Counter Timer Circuit
CPU = Central Processing Unit
PIO = Parallel Input/Output
SIO = Serial Input/Output
CIO = Counter Input/Output
SCC = Serial Communications Controller

DMA = Direct Memory Access
Några fler kommer jag inte på för tillfället...

(Text 2173) Lars Michael Jogbäck <5862>
Ärende: ABC99

Lysdiod	OUT 34,
1	128
2	144
3	160
4	176
5	192
6	208
7	224
8	240

Hoppas du blev nöjd med detta

(Text 2175) Benny Löfgren * <2615>
Ärende: Portarna i ABC 800 M
Blev det rätt det där??? Kanal A är ju PR:, dvs DARTen och kanal B är V24:, dvs SIOn. På kanal B kan man köra split speed, dock ej på kanal A, och det innebär att kanal B har separat Receive och Transmit Clock, medan kanal A har gemensam.

(Text 2222) Sven Wickberg <1384>
Ärende: 80 -> 800
Jag stöter på mystiska fel när jag försöker konvertera en rad ABC80-program till ABC800-maskiner.
I princip lagrar jag programmet i .BAS-format med LIST <programnamn>. Sedan hämtar jag .BAS programmet till en ABC806 med LOAD. Då hämtas den in och kompileras till 806:ans specialkod. Men när jag gör RUN visar sig en rad fel. Självklart vet jag vissa förändringar i BASICen (jag har läst Bit för bit), men här uppträder saker som jag inte förstår.
Exempel:
I många fall godkänns inte rader som avslutas med : REM xxx Tar jag bort REM-satsen eller sätter ! i stället för : REM går det bra. Men det FINNS också rader där REM-satsen accepteras! ONERRORGOTO 0 tycks IBLAND läsas in allright och göras om till ON ERROR GOTO, ibland inte.
I några fall har jag måst dela upp en rad med ett stort antal instruktioner i flera rader. Har ABC800 svårt att svälja många semikolon på en rad.
Lustigt nog tycks maskinen acceptera ASC(x) och göra om den till ASCII(x), i varje fall ibland.
En annan varning. Har man INTEGER-läge kommer alla variabler att bli heltal även utan %. Skulle man av någon anledning vilja göra en ändring så att programmet kan köras för både 80 och 800 (t ex i ett CAT-NET) så kan man inte utan vidare spara programmet med LIST igen och sedan köra på 80. Då försvinner nämligen alla heltals%. Man måste ha FLOAT-läge innan man spar. Det är inte så lätt som det verkar att köra .BAS program i både 800 och 80. Närmast tycks kompatibiliteten innebära att man inte behöver knappa om hela programmet, utan kan slippa undan med en del mindre modifikationer.

(Text 2233) Benny Löfgren * <2615>
Ärende: 80 -> 800
Det första man måste tänka på är givetvis att ha FLOAT och NO EXTEND inkopplat. Att du får fel i slutet på raden med ': REM bla bla' kan bero på att 800 kräver att vissa instruktioner står ensamma på en rad, t ex END.
ONERRORGOTO 0 görs INTE om till ON ERROR GOTO, utan det får man göra själv.

ASC konverteras alltid till ASCII, oavsett vilken mode man står i. Att du måst dela upp en rad i flera kan ha samband med det jag sa tidigare, att vissa instruktioner måste vara ensamma på en rad. Jag antar att du menar KOLON, och inte SEMI-dito. ABC800 har en större radbuffert och borde inte ha något problem att kompilera ABC80-rader.
För övrigt bör nämnas att det inte är att rekommendera att köra program medvetet i .BAS-format på ABC800, det tar nämligen en j-vla tid att kompilera program vid inladdning, speciellt om det är stora program.

(Text 2235) Benny Löfgren * <2615>
Fråga: Hur skriver man ett FIFO?
Ja, för det första så ska väl nämnas att en bättre (tycker jag) benämning på ringbuffert är FIFO, First In, First Out, till skillnad från LIFO, Last In First Out - även kallat stack.
Hur man gör? Ja, man måste hålla rätt på fyra pekare:
- buffertens startadress i minnet
- buffertens slutadress i minnet
- aktuell 'skriv'position i bufferten
- aktuell 'läs'position

När man ska SKRIVA något i bufferten gör man så här:

1. Hämta skrivpekaren
2. Kolla mot läspekaren
2.1 Om (skrivpekaren=läspekaren-1) så är bufferten full. Tänk på 'wrap-around'!
3. Skriv ner tecknet i bufferten
4. Öka skrivpekaren
Kolla om skrivpekaren passerat slutadressen
4.1.1 I så fall, sätt skrivpekaren = startadressen

För att LÄSA från bufferten gör man så här:

1. Hämta läspekaren
2. Kolla mot skrivpekaren
2.1 Om (läspekare=skrivpekare) så är bufferten tom. Tänk även här på wrap-around!
3. Hämta upp tecknet från bufferten
4. Öka läspekaren
4.1 Kolla om läspekaren passerat slutadressen
4.1.1 I så fall, sätt läspekaren = startadressen

Det är rätt enkelt, det svåra är att hålla rätt på när man ska göra wrap-around, dvs börja om från början i bufferten.

(Text 2249) Stefan Lennerbrant <4364>
Ärende: UTE - ABC9000
Är det någon som vet hur man fixar över textfiler med ABCUTE ? Jag hörde att man kunde använda 'Skicka kommandofil' så det testade jag ju såklart - gick inte alls (Kör mot ABC9000)
Flera tecken försvann & något CHR$(4) måste kommit med för ABC9000 trodde plötsligt att filen var slut och tolkade allt som kommandon (Tur att inget "rm" fanns med...)
Alltså: Hur göra för att skicka textfiler ed UTE ?
/STL

(Text 2250) Stefan Lennerbrant <4364>
Ärende: Optioner vid CHAIN
Vid CHAIN 'KALLE,MF0:TESTFIL'
Var läggs 'MF0:TESTFIL' eller läggs det inte någonstans alls ? Jag hörde att man kunde få något av XS0's program att starta filen 'KALLE' om man skrev 'CHAIN ROGNAMN,KALLE' Alltså måste 'KALLE' läggas nån'stans
/STL

(Text 2255) Patric Ljung <5455>
Ärende: UTE - ABC9000
SAFT men det har de nog inte på 9000 där. Körde du en 800 som terminal till 9000? (hur stor fil?) XON/XOFF?
Hurra, nu kommer min färgskärm!
Moss / PAtrioc

(Text 2258) Benny Löfgren * <2615>
Ärende: Optioner vid CHAIN
Jag vet inte, men jag tror att det hamnar på samma ställe som det man skriver i kommandomod. Adressen dit fås mha CALL(81). Det är mycket praktiskt, då kan man göra program som kan hämta parametrar från kommandoraden, t ex "RUN BLAHA,INFIL".

(Text 2262) Kristoffer Eriksson <5357>
Ärende: Optioner vid CHAIN
Så fungerar det, ja. Man kan göra sig en funktion som plockar fram kommandoraden:

```
DEF FNCmdline$ LOCAL S$=0,P
P=CALL(81)
POKE VAROOT(S$),160,0,P,SWAP%(P),160,0
P=INSTR(1,S$,CHR$(13))
IF P THEN RETURN LEFT$(S$,P-1) ELSE
RETURN ""
FNEND
```

Då får man hela kommandoraden som resultat. Om programmet startas me CHAIN fås det som står inom citationstecknen, t ex ger CHAIN "KALLE,SOCKER" resultatet KALLE,SOCKER från FNCmdline$.

(Text 2269) Peter Goldmann <5080>
Ärende: varning för blå skärm med vit text
JAg berättade i ett tidigare inlägg hur bra det syntes och hur vilsamt det är med nomnda kombination - men den har en nackdel man skall akta sig för blått ljus - det kan på lång sikt skada ögat. Det fick jag höra så sent som idag. Av en ögonläkare.
Den våglängd som ögat kan urskilja flest nyanser hos är de gröna. Gult är vilsamt också, men som sagt kan starkt blått ljus ge synförändringa på lång sikt - därför är blått inte att rekommendera.
Hälsningar Peter Goldmann 5080

(Text 2271) John Stagg <5741>
Ärende: HARDCOPY.REL
Har nyss försökt prova denna rutin med det gick inte så bra. Jag har en del andra rutiner på olika platser i dosbuffertar och systemram (tangentbuffert, lib-kommandot mm) och jag misstänker att det blir krock nånstans.
Kan man möjligtvis få veta var HARDCOPY ligger i minnet, och även bättre få en listning av assemblerrutinen ? Då kan jag assemblera om så att rutinen ligger rätt bland mina andra rutiner. Det gäller att ha "städat" i minnet om man ska ha så himla många extrarutiner så att ABC'n blir någalunda användbart nufördtiden! (Annars skulle jag slänga den och köpa en PC !)
Mvh John

(Text 2272) Benny Löfgren * <2615>
Ärende: HARDCOPY.REL
Hardcopy har som du ser extension '.REL', vilket innebär att programmet är RELokerbart, dvs man bestämmer först vid laddningen var programmet ska hamna. Denna inladdning sker som du säkert vet med hjälp av ADDOPT.ABS.

(Text 2290) Kjell Stolpe <6025>
Ärende: Optioner vid CHAIN
Förlåt en yngling som halkat in i monitorn..., men jag tyckte att det här med CHAIN och optioner verkar intressant. Menar du att man kan lägga in olika styrparametrar, ex-vis till olika Ufd: eller liknande i samband med att man ger ett CHAIN-kommando eller har jag missat något väsentligt i sammanhanget? Tacksam för en något "fylligare" kommentar om hur man kan utnyttja dessa optioner.
Mvh K.S gm Bo Thydell

(Text 2292) Benny Löfgren * <2615>
Ärende: Optioner vid CHAIN
Ja, precis som vanligt när det gäller datorer så är det bara fantasin som sätter gränsen - du kan t ex skicka med parametrar till ett program utan att behöva använda COMMON-variabler. En annan 'feature' som man får på köpet är att man kan ange parametrar när man startar ett program med RUN, t ex "RUN PROG,FIL.TXT". Man använder exakt samma metod för att få fram startparametrarna.

(Text 2294) Kristoffer Eriksson <5357>
Ärende: Optioner vid CHAIN
Man får förstås inte glömma att det program som ska ta emot optionerna måste vara avsett för detta redan från början. Det går inte att skicka optioner på det här sättet till de flesta program, eftersom de inte bryr sig om dessa. Det är när man konstruerar egna program som det kan bli tal om att utnyttja denna finess, bl a med hjälp av den funktion jag beskrev tidigare.

(Text 2307) David Andersson <5201>
Fråga: Hur skriver man ett FIFO?
Man kan förenkla lite om bufferten är på pricken 256 byte. När man ökar läspekaren eller skrivpekaren incrementerar man bara låga delen av pekaren.
pekarnas värden behöver bara initieras med startadressen en gång. Köa data:

1. Kolla först att ((skrivpek-läspek) MOD 256)<=maxkölängd
2. POKE skrivpek,data
3. INCREMENT skrivpek.lågabyten

Hämta data:
1. Kolla först att skrivpek<>läspek
2. data=PEEK(läspek)
3. INCREMENT läspek.lågabyten

Många moderna terminaler har en intern mottagningsbuffert om 256 tecken. (Utoin äkta VT100 som har 32 tecken). Normalt låter man terminalen sända XOFF när sådär en 64 eller 128 tecken är köade, (det är lite olika).
Det gällde väl en terminal rutin, vad jag förstod.
Vid testen av maxkölängd (som f.ö. ska < 256) i punkt 1 ovan ska resulatet av MOD operationen alltid tolkas som positiva värden. (vilket inte är så svårt i assembler, det blir ganska automatiskt så) Hoppas det blev rätt.
(PS. Kan 'over-run' eller 'over-flow' verkligen kallas 'wrap-around'?)

(Text 2327) Gunnar Faith-Ell * <2733>
Ärende: KABLAR
Jag öppnade nyss kontakterna till bildskärmskabeln på ABC800M och fann att kopparflagor spretade ut från lödningarna på ett par ställen så pass mycket att det lätt kunde blivit kortslutning !!!
Tidagare har jag sett samma sak i kontakterna på kabeln till diskdriven, det är otroligt att dessa kablar blivit godtjända och släppts ut på marknaden.
Kanske är det dags att se över sina gamla kablar innan något händer...

(Text 2338) Johan Hedberg <2755>
Ärende: Interrupt
Om du vill utnyttja interrupten för tangen bordet och klockan så kan du ju lägga in dina egna adresser dit du vill hoppa i systemvariabellistan. Adr. tangenbordsint. 65460-1 där du lägger adressen dit du vill hoppa. Adr. klockint. 65494-5 där du lägger adressen dit du vill hoppa. Detta var kanske inte vad du frågade efter, men i så fall kan du ju höra av dig.

(Text 2340) Torsten Frank <5452>
Ärende: Öppna fil i assembler (5DH)
Jag har följande rutin i ett program:

```
ÖPPNAFIL  LD    HL,FILNAMN
                ! Filnamn
          LD    IX,0FF63H
                ! Temporärt LU-block.

          LD    A,00H
                ! Kod för "OPEN".
          CALL  5DH
                ! OPNCMD
          RET
          ! Återvänd till huvudprogram
FILNAMN   DEFM  "FIL.BAS"
          DEFB  0DH
```

Om nu inte filen finns, eller om något diskfel inträffar, tar BASIC över och skriver ut ett felmeddelande (= assemblerprogrammet avbryts). Finns det inget sätt att komma runt det? Kan jag inte behandla felen själv, med någon egen felrutin el dyl?

(Text 2341) Magnus Bodin <4304>
Ärende: Öppna fil i assembler (5DH)
Har för mig att det ligger en felhoppsvektor på adress FFAC, (Gillar inte HEX själv, men kom i detta fall endast ihåg den versionen) Om nu (FFAC), heltalet inte =0 är det adressen till felrutinen. Tror jag, har inte tillgång till Bit för Bit just här, ligger i skolans bibliotek..
Magnus :-)

(Text 2343) Magnus Hedner <530>
Ärende: Öppna fil i assembler (5DH)
Jag har däremot Bit föt Bit här hos mig: Adress FFAC är "DOS-anrop för att hämta error-text. A=felnr" Däremot adresserna FD21 - FD32 innehåller hoppadresser till felrutiner för fel 35 - 43 respektive (dvs PEEK2(FD21) = Err 35, PEEK2(FD23) = err 36, osv), samt i adress FD33-FD34 finns en "default-adress" om adressen i de bytear man skulle till råkar vara noll.
Något klarare???
/mvh mh

(Text 2350) Torsten Frank <5452>
Ärende: Mystisk låsning
Jag har ett ass-program som använder adress 45000-63231 för data-lagring. När nästan hela av detta utrymme är fyllt (återstår cirka 1680 bytes) låser sig maskinen och måste resetas. Dosbuffertarna som används har markerats som upptagna och används alltså inte av doset. Ligger det något precis innan buffertarna som inte får förstöras? Vad? Kan man flutta det?

(Text 2351) Harald Nautsch <4479>
Ärende: Mystisk låsning
Förmodligen kvaddar du stacken. Stackpekaren flyttar du med: LD SP,adress.
/H

(Text 2366) Magnus Hedner <530>
Ärende: .REL-länkare
Det går inte att addopt'a objektfiler gjorda av asmz, därför att dom är inte färdiga program! Det fattas en del info (som estab fixar) och det finns massa redundant info (gör ju inget förstås). Men skilj alltså på OBJEKTkod (gjord av asmz) och BINÄRkod (gjord av estab).
/mvh mh

(Text 2372) Anders Fällström <2700>
Ärende: 'Nytt DOS'
Jag håller på att försöka fixa lite med något slags eget dos, och vore nu glad om någon kunde hjälpa mig med några detaljer. Har kommit så långt att jag kan läsa alla sektorer på skivan med hjälp av några OUT-kommandon till kontrollkortet:

```
OUT 2,0,1,45
OUT 0,3  resp 0,12 för läsning resp skrivning
OUT 0,0  för drive 0
OUT 0,A  för spår A
OUT 0,B  för sektor B (/32)
```

Det funkar utmärkt, men nu undrar jag hur man ska kunna skriva data mellan de 'ordinarie' sektorerna, så att man kan 'gömma undan' data.
Någon som kan hjälpa till?

(Text 2374) Gunnar Faith-Ell * <2733>
Ärende: 'Nytt DOS'
(Är det inte så att du tänkt att kopiera skyddade skivor ?) Tänk på att fler kortadresser än 45 kan förekomma.

(Text 2375) Kent Berggren <6019>
Ärende: 'Nytt DOS'
Kodierings skydd. !!
Troligen finns det spår som inte kan nås av det vanliga doset. Spår som -1. De kan man läsa på men inte skriva på. De är nog hela hemligheten med kopierings skydden. Vist kan man oxo formatera in lite skit så att du inte får över det utan att formatera på samma sätt. Personligen tycker jag kopierings skydd är urk! Finns bättre sätt att ficka det hela. T ex med kontrakt med kunden.

(Text 2391) Patric Ljung <5455>
Ärende: Ljuspenna
Var kan man få tag i ljuspenna, och för hur mycket. Att användas till ABC806. Någon som har några erfarenheter?
Muh / Patric

(Text 2396) Gunnar Faith-Ell * <2733>
Ärende: Ljuspenna
Jag såg att Luxor för ett par år sedan hade en ljuspenna till en ABC806:a på mässan i Sollentuna, hör med dem.

(Text 2398) Sten Bergmar <4713>
Ärende: ASM-rutin för printerstyrning
Kan någon vänlig själ tala om hur man med ASM-rutin skriver ut en textsträng från given plats i internminnet på printern. Jag har försökt med OTIR till port 32 men printern reagerar inte. Vad fattas?
Mvh/Sten

(Text 2403) Gunnar Faith-Ell * <2733>
Ärende: ASM-rutin för printerstyrning
Har du satt baudreuaten till skrivaren ? Du måste avsluta texten med <CR> <LF> på flertalet skrivare.

(Text 2406) Jan Lundmark <2765>
Ärende: ASM-rutin för printerstyrning
Öppna printern som en fil istället.

(Text 2408) Gerry Eriksson <4842>
Ärende: ASM-rutin för printerstyrning
Du kan inte använda OTIR för att sända data till PR-porten då den inte hinner sända ut tecknen i den takt du skickar dem till kretsen. Vad du skall göra förutom att sätta rätt baud-rate m.m. som man enklast gör genom att öppna den genom en filanrop enligt en annan kommentar är att du skall testa TX Buffer Empty-flaggan och CTS-flaggan i DARTen för att se dels om DART'en (=PR-porten) kan ta emot ett tecken till och att printern även kan det.
Tex:

```
      LD    HL,textadress
      LD    B,antal
      LD    C,32D
            ; dataporten
L1    INC   C                 ; kommando-
                              porten
L2    LD    A,16D
            ; reset ext/status
      OUT   (C),A
      IN    A,(C)
            ; läs statusreg 0
      AND   32D OR 4D
            ; maska CTS och Tx buffer
              empty
      CP    36D
            ; båda skall vara satta
      JR    NZ,L2
      DEC   C
            ; dataporten
      OUTI
            ; sänd tecken
      JR    NZ,L1
            ; om ej sista tecken att sända
            ; klart
```

Mvh Gerry

(Text 2410) Kristoffer Eriksson <5357>
Ärende: ASM-rutin för printerstyrning
Jag föreslår att du utnyttjar filsystemet, det ger större flexibilitet. Så här kan man göra:

```
            CALL   USEUTLU
                   ; = 105
            LD     A,0
                   ; 0 = Normal OPEN
            LD     HL,FILNAMN
            CALL   OPNCMD
                   ; = 93
            LD     A,4
                   ; 4 = Operationen PRINT
            LD     HL,TEXT
            LD     BC,TEXTEND-TEXT
            CALL   IO
                   ; = 90
            RET
FILNAMN:    DEFM "PR:"
            DEFB 13
TEXT:       DEFM "Denna text skrivs på
            skrivaren"
TEXTEND: *
```

Denna variant passar bäst för korta utskrifter. USEUTLU åstadkommer vid RET i ovanstående rutin ett tvångsanrop till en filstängningsrutin. Vill man undvika det kan man POP:a den adress som ligger överst på stacken efter USEUTLU-anropet. En mer generell variant är:

```
            LD     IX,LUBLOCK
            LD     HL,FILNAMN
            LD     A,0
            CALL   OPNCMD
            ... osv
LUBLOCK:    DEFS 23
CLOSE:      LD     IX,LUBLOCK
            LD     A,2
                   ; 2 = Operationen CLOSE
            CALL   IO ...
```

På det här sättet öppnar man först filen, använder den då och då, och stänger den till sist. Varje gång filen används måste man se till att IX innehåller adressen till LUBLOCK.

(Text 2430) Patric Ljung <5455>
Ärende: Lagring av maskinspråksrutin
Om maskinkoden är bestämd till att ligga på ett bestämt ställe så är följande rutin MYCKET snabb och lätt.

```
OPEN 'MFIL.ASM' AS FILE 1
DIM Maskinkod$=0
GET §1,Q$ COUNT 2 : Storlek=CVT$%(Q$)
POKE VAROOT(Maskinkod$),Storlek,SWAP%-
(Storlek),Adress,SWAP%(Adress),0,0
GET §1,Maskinkod$ COUNT Storlek
CLOSE 1
```

Då kanske man skulle förklara alla stegen.
Först öppnas filen. Filen är uppbyggd som så att först ligger två bytes som anger längden (storleken) på maskinkoden. Men först DIMar vi variabel till noll så den inte tar upp onödig plats i minnet. Därefter läses storleken in i Q$ och konverteras till Storlek. Följande steg är att Ändra på variabel roten så att DATA arean börjar där maskinkoden skall börja och längden på variabel är den längs (storlek) som maskinkoden har. Sedan läses maskinkoden in och filen stängs.
Notera en sak! Allra först måste pekarna ändras så att datorn inte lägger in BASIC-program på minnes arean som maskinkoden skall ligga på. Man går nästan tillväga på samma sätt när du skall spara maskinkodebn om den ligger i minnet.

```
Storlek=(storlek på maskinkoden)
DIM Maskinkod$=0
POKE VAROOT(Maskinkod$),Storlek,SWAP%
(Storlek),Adress,SWAP%(Adress), Storlek,
SWAP%(Storlek)
PREPARE 'MFIL.ASM' AS FILE 1
PUT §1,CVT%$(Storlek)
PUT §1,Maskinkod$
CLOSE 1
```

Vidare kanske en del då förstår andra smarta lösningar på andra problem.

```
DIM Screen$=0
POKE VAROOT(Screen$),2000,SWAP%(2000)
,0,120,2000,SWAP%(2000)
```

Variabeln Screen$ 'innehåller' således skärmen.
Ex: Gör ovanstående tryck CTRL+L (rensa skärmen) hoppa ner ett par rader och skriv Screen$='Hej'.
Hej skrivs ut överst på skärmen.

```
MID(Screen$,Rad*80+Kolumn,Längd)=Text$
```

Alternativt PRINT.
Moss / Patric

(Text 2438) Torsten Frank <5452>
Ärende: PLC med ASMZ
Om man gör en utlistning (CON:/PR:) av en assembler-fil med assemblern ASMZ skrivs det på varje sida PLC (och en hel del annat). Det står tydligen för "Current location counter". Problemet är att jag inte kan få PLC:n att ändra sig. Det står inte någon ting alls i detta fält på utskriften, hur jag än bär mig åt. Finns det någon instruktion som ändrar PLC:n (?), eller ändras den automatiskt vid nytt ORG, utan att det syns på listan?

(Text 2439) Jan Lundmark <2765>
Ärende: PLC med ASMZ
PLC ändrar man med 'PLC <nr>' där <nr> är ett heltal i intervallet 0-31. När du sedan länkar och ska orga de olika PLC'erna till olika adresser så skriver du:

```
exvis:
PLCNR 3
ORG 8000H
PLCNR 4
ORG 0C000H
.
.
```

i din kommandofil till länkaren.
Jo just det ja, det är mycket trevligare att orga vid länkningen än att skriva in orgar i källkoden.

(Text 2463) Curt Rehnborg <1121>
Ärende: Rad 25 och inverterad text på ABC80X
Är det någon som har kläm på hur man på ABC802 och ABC806 kan fixa till inverterad text på någon rad(er).
Hur man får till rad 25 vet jag, men hur fixa inverterad text där ? Helst ska man kunna köra inverterat ibland och inte ibland.
Någon som vet, hör av er .....
Gott Nytt År ....
Hälsning Curt Rehnborg

(Text 2464) Kent Berggren <6019>
Ärende: Rad 25 och inverterad text på ABC80X
Bit 7 hög. Funkar inte det?

(Text 2466) Gerry Eriksson <4842>
Ärende: Rad 25 och inverterad text på ABC80X
Att sätta bit 7 hög går bara på ABC802. På 806 så kan du sätta en bakgrundsfärg och skriva text med svart (BLK).

(Text 2467) Gerry Eriksson <4842>
Ärende: V24:ans interrupt-vektorer.
Vektorerna innehåller adressen till interrupt-rutinerna.
Vektorerna för V24: (SIO'n chA) är som följer:

Adress	Interruptfunktion
65480	Transmitter buffer empty
65482	External/Status Change (DCD, CTS, SYNC och Break seqence)
65484	Receive Character Available
65486	Special Receive Condition (Parity error,RX overrun error, Framing error)

Du får då ändra vektorn i 65482 till adressen för en rutin som undersöker om det var DCD som orsakade interrupet.
Givetvis måste även SIO'n vara programmerad för interrupt (WriteReg 1).

(Text 2472) Einar Eriksson <1720>
Ärende: Överfört från KOM - Printerkabel V24 till RS 232C
Har skrivaren RS 232 C ingång så ska kabeln se ut så här.

V24 (9-pol)		RS 232 C (25-pol)
1 DTR	till	6
2 TxD	till	3
3 RxD	till	2
4 RTS	till	8
5 CTS	till	20
6 + 12 V		NC
7 GND	till	7
8 DCD		NC
9 - 12 V		NC

NC = Not connected
Så här är orginalkabeln kopplad som Luxor levererar med skrivaren Epson.
Glöm inte att ställa in rätt hastighet på skrivaren samt att ställa in rätt paritetsbitar.
Dessutom måste man ställa in om skrivaren ska använda XON/XOFF eller DTR som handskakning.

(Text 2485) Stefan Westman <4969>
Ärende: Program lagrade som strängar.
Tack för alla tips om lagring av program på strängar.
Jag fick mitt program att fungera.
Då jag nu har upptäckt att det går snabbare att läsa in ett program på detta sätt tänkte jag lagra fler program som strängar.
Problemet är dock följande.
Ett program som har lyft golvet till 237*256 och använder grafikminnet skall sparas som sträng.
Jag gjode då följande.

1. Sparade minnescell 0-237*256 på en sträng.
2. Sparade minnescell 134+241*256 - 65535 på en sträng.
3. Sparade grafikminnets alla celler på disk med hrsave.

Nu händer följande: Om jag läser in alla sparade delar i ordning 3-2-1 utan att resetta kan jag fortsätta och köra programmet. Sker inläsningen efter resett så kommer jag in och kan köra delar av programmet men vissa rutiner gör att maskinen autostartar om.
Finns det någon som kan ge några hintar om var problemet kan sökas?
Jag kör på ABC 806
mvh Stefan

(Text 2489) Kristoffer Eriksson <5357>
Ärende: Rad 25 och inverterad text på ABC80X
På 806 är det ganska enkelt, eftersom man kan skriva t.ex PRINT CUR(24,0) YEL NWBG BLK "Detta är rad 25";
På 802 däremot är man tvungen att POKE:a in texten eller använda andra knep, varvid bit 7 ska vara ett-ställd. Dock finns det en del 802:or med större teckenprom som har andra tecken än inverterade normala tecken på de koderna. Med "andra knep" menar jag exempelvis:

```
DIM Line25$=0
POKE VAROOT(Line25$),80,0,32640,SWAP%
(32640),80,0
Text$="Detta blir inverterat på rad 25"
FOR I=1 TO LEN(Text$)
MID$(Line25$,I,1)=CHR$(ASC(MID$(Text$,I,1))
OR 128)
NEXT I
```

fast när man ändå måste behandla varje tecken för sig, går det ju lika bra med ren POKE.

(Text 2493) Torsten Frank <5452>
Ärende: Clustersize mm
Finns det något enkelt sätt att ta reda på clusterstorleken för en viss enhet och ta reda på om DOS = ABC 6-1X (då ska ju tydligen fysiska sektornummret adderas med 32) ?

(Text 2495) Gunnar Faith-Ell * <2733>
Ärende: Clustersize mm
Det finns ett program på systemskivorna till UFD-dos som kollar vilket dos som sitter i burken.

(Text 2505) Kristoffer Eriksson <5357>
Ärende: Rad 25 och inverterad text på ABC80X
För att invertera endast en bit av en rad på 806 skriver man i stil med detta exempel:

```
PRINT CUR(10,30) YEL NWBG BLK "Inver-
terat" BLBG " "
```

BLBG på slutet återställer resten av raden till svart bakgrund och blanktecknet behövs för att den nya färgen verkligen ska skrivas ut i bildminnet, i alla fall när man har ATTRIBUTE 1. Ska man genast fortsätta med ej inverterad text får man förstås ställa in en förgrundfärg också, annars blir det ju svart på svart. Om det redan står något mera i slutet av raden behövs inte 'BLBG " "' eftersom den gamla texten ändå behåller sin färg, det är bara på helt o-skrivna rader som bakgrundsfärgen sträcks ut över hela raden.
Tänk på att om print-satsen avslutas med ";" fortsätter de inställda färgerna och övriga attribut att gälla även vid nästa print oavsett om man har ATTRIBUTE 0 eller 4.
Grafik- och färghanteringen skiljer ju ganska mycket mellan 806 och 802. På 802 skriver man i förväg ut styrtecken på lämpliga ställen i bilden, medan det i 806 hela tiden är senast använt styrtecken som gäller.

(Text 2507) Kristoffer Eriksson <5357>
Ärende: Program lagrade som strängar.
Det verkar klart överdrivet att även spara PROM-delen av minnet på filen! (Adress 0-30719 är prom, om man inte mappat in annat minne. 30720-32767 är bildminne för text.)
Att sedan spara alla systemvariabler inklusive stacken måste vara rena självmordet! (Varför inte spara CPU-registren också, eller åtminstone stackpekaren?) När stacken vid inläsningen byts ut kommer ju BASIC-tolken att spåra ur. Eventuella omlänkningar av vissa systemvariabler bör göras en i taget, inte med strängmetoden.
Är det verkligen nödvändigt att spara grafikminnet också?

(Text 2563) Nils Hansson <519>
Ärende: RAM:-floppy på ABC806
Jag har fått anledning att titta på hur jag kan använd ramflopyn i olika egna tillämpningar i kombination med grafikminnet. Vad jag inte har förstått är vad det innebär att RAM: är spärrat.
Finns det något sätt att göra reset utan att ramfloppyrutinen rammlar ur minnet? Det vore praktiskt för alla filer ligger ju i alla fall kvar i RAM:.
Mvh Nils H

(Text 2565) Gunnar Faith-Ell * <2733>
Ärende: RAM:-floppy på ABC806
Om du har den äldre drivrutinen för RAM-floppyn så stannar minnet kvar men då finns det risk för att det blir fel på innehållet. Eventuellt kan man nog byta RAM--kapslar till några som håller sig längre utan refresh och lösa problemet med minnesförluster på det sättet.

(Text 2597) Mikael Liden <5651>
Ärende: GRAFIK & RAMFLOPPY samtidig t
Här är några nyttiga adresser:

65267	- max antal tillåtna HR-bilder
65268	- max antal HR-bilder

gäller ABC806

(Text 2608) Mikael Liden <5651>
Ärende: Formattering
Kan någon beskriva vilka kommandon och data som behöver skickas till kontrollern för att formattera disketten. Antar att det är olika för varje diskett-storlek. Är i första hand intresserad av 640Kbyte-disketter. (Orkar inte dissassemblera DOSGEN)
Tänkte göra ett snabbt helkopieringsprogram av disketter, som dels formatterar och dels kopierar snabbt de sektorer som är använda på disketten (dvs de som är markerade i bitmappen). Har tidigare gjort ett snabbt helkopieringsprogram för 160K som sköter kopieringen genom att först läsa upp sektorerna i kontrollern och sedan skriva ut dem på den andra disken (max 3 sektor åt gången), bufferten i kontrollern är inte större), det man vinner i hastighet på att förfara på detta sätt är att slipper läsa in sektorerna till datorn. Detta program formatterar dock inte disketterna, det vore naturligtvis bra att få till ett snabbt kopieringsprogram som både formatterar och tar kopia Om intresse finns för ett dyligt program så skickar jag naturligtvis in det.

(Text 2609) Jan Lundmark <2765>
Ärende: Formattering
Är det den gamla controllern ? I så fall blir det knepigt För att formattera ett spår fixt och färdigt med data i sektorern a så måste controllern ha hela spåret i sin buffert inklusive adressmärken. till detta kommer sedan kodsnutten som ska göra't. Innan man ger formatkommandot till controllern så måste bufferten fyllas med info om hur adressmärkena ska se ut. Sedan så sköter controllern om att fylla sektorerna med 'E5'. Detta gäller alltså den gamla luxor/diabcontrollern med 1k minne på kortet. Hur luxor/diabs nya / myabs gör har jag ingen aaaaning om.

(Text 2610) Gerry Eriksson <4842>
Ärende: Formattering
I ABC bladet nr 2/85 sid 32 hittar du en text av Göran Altius <123> (= f.d. text 259 i detta möte) som beskriver kommandon för MYABs UNIDISK-kort bl.a. formattering. Gjorde en testsnutt och provade med en DD12 (spec.) och det fungerade fint. (Formattering av bara ett spår) 12an utrustad med DIABs ny kontroller.
Har du inte nr. så hör av dig så får vi ordna det på annat sätt.

(Text 2612) Bo Michaelsson <913>
Ärende: Rad 25 på 806 och 802
Följande lilla program ger en ny rad, rad 25, på ABC 800 M.
Raden påverkas inte av CHR$(12).

```
DEF FNRad25
OUT 56,6,57,25
POKE 65362,0,24
; SPACE$(80);
; "På denna rad skrivs en text";
RETURN 0
FNEND
```

Rutinen fungerar utmärkt på 800 M men inte på 806. Har någon en ide som fungerar på båda?

(Text 2614) Gunnar Faith-Ell * <2733>
Ärende: Rad 25 på 806 och 802
På ABC800M kan man om man vill lägga in TVÅ nya rader men då får man flytta upp BASICen lite.

(Text 2617) Göran Sundqvist <1255>
Ärende: Rad 25 på 806 och 802
genom att programmera om CRTC-kretsen, kan man få 30 rader om 64 tecken.

(Text 2630) Kristoffer Eriksson <5357>
Ärende: Rad 25 på 806 och 802
Om man betänker att man på 806:an kan nå den 25:e raden med vanlig PRINT och att den enda skillnaden mot övriga rader på skärmen är att den inte scrollar, ser man lätt vad som är felet med din lilla snutt. När SPACE$(80) har skrivits ut så görs ett radbyte till den 24:e raden som är skärmen sista rad enligt scroll-rutinen, och skärmen scrolla. Därför hamnar den efterföjande texten inte på rätt rad. Använd T$="På denna rad krivs en text"

```
; T$ SPACE$(80-LEN(T$));
```

istället. Då fungerar det på alla maskiner.
F.ö. är det en ganska smart metod att POKE:a radräknaren, jag kom själv inte på den för mer än några månader sedan.
Tilläggas bör att på 806:an tömmer CHR$(12) ALLA rader, även den 25:e.

(Text 2661) Henrik Ahlen <5133>
Ärende: Skärmflimmer
Min ABC800 lider svårt av att skärmen ofta plötsligt börjar flimra kraftigt. Avhjälps genom att slå på datorn eller vänta en stund. Är värst den första timmen efter påslag,v arför jag numera låter den vara på dygnet runt. Datorn har varit på TDX för service, men kom tillbaka likadan som förut. Vad kan göras åt detta? Tacksam för tips.

(Text 2664) Gunnar Faith-Ell * <2733>
Ärende: Skärmflimmer
Lyft på kortet på bildröret i bildskärmen och spraya lite. Dra ut videokortet i datorn och spraya kontakterna lyft på kretsarna och sprya, kolla kontakterna på sladden, så ska den nog fungera. Kontaktspray hjälper alltid...

(Text 2711) Patric Ljung <5455>
Ärende: Rad 25 på 806 och 802
Nej, nej!
; FNR25$(10) 'Detta är rad 25';
Här används rutinen precis som ett vanligt
R. Observera att ett semikolon måste
sist för att skärmen inte skall scrolla.
DEF FNR25$(Kolumn) LOCAL Kolumn
POKE 65362,Kolumn,24
RETURN "
FNEND
Om man först vill rensa rad 25 får man skriva:
; FNR25$(0) SPACE$(80) FNR25$(0) Text$
(Jämför med CUR)
UR(24,0) SPACE$(80) CUR(24,0) Text$
man så kan man....
DEF FNC$(Rad,Kolumn) LOCAL Rad,Kolumn
POKE 65362,Kolumn,Rad
RETURN "
FNEND
Så är funktionen helt utbytbar mot CUR.
OBS att den 25 raden måste initieras först.
OUT 56,6,57,25 ! Sätt 25 rader
Moss / Patric
PS. Hoppas alla problem är utredda

(Text 2739) Robert Svedjehammar <2350>
Ärende: Kablar & skatbon
Visst gör det någonting om man börjar plocka bort de skydd som finns för att förhindra utstrålning, kanske inte för dig, men för omgivningen som försöker använda någon form av radiomottagare. Det finns en orsak till varför de sitter där. Tillverkaren skulle knappast fördyrat konstruktionen om det inte behövdes.
Hälsning från en radioamatör med mycket störningar.

(Text 2761) Curt Rehnborg <1121>
Ärende: Kablar & skatbon
De kåpor och skärmar m.m. som sitter monterat på Luxormaskinerna sitter inte bara för att förhindra utstrålning av oönskade störningar. Det är även för att hindra att störningar av olika slag kommer IN, kanske mer det sista än det första.
som är viktigt i sammanhanget är att
an är s.a.s obruten. Vissa kontakter
an Luxor är dock tyvärr "slarvigt" monterade varför just kortslutningar inte är ovanligt. Speciellt en variant av kabel mellan 806 och 815 (Den korta)
H-ning CR

(Text 2767) Lars Hedberg <2497>
Ärende: Programstatus
ddress SYS(10)+38 ligger informationen
318d):
- Det har varit förkoll
b1 - LISTSKYDDAT program
b2 - Double på flyttal
b3 - Flyttal finnes
b4 - COMMONflyttal finnes
b5 - Programmet ligger högt i minnet
b6 - HR-Grafik användes
b7 - RESUME användes
På address SYS(10)+29 ligger lite annat av intresse (=65309d):
b0 - Integer mod
b1 - Extend mod
b2 - Mellanslag signifikant
b3 - Variabler finns ej (?)
b4 - COMMON bort (?)
b5 - Lägger in COMMON (?)
b6
b7
En lustig variant är att ändra SYS(10)+29 så att man befinner sig i extend mod men UTAN signifikanta mellanslg.
Hoppas det svara på eventuella ?.
Hälsningar TFP Lars Hedberg

Möte ABC80

(Text 929) Guido Petz <1098>
Ärende: basicii till abc80
Enligt Börje Larsson på Sala Teknik finns nu Basic2-kortet på marknaden.
Priset är 2500+moms!
På kortet finns följande faciliteter:
1 anpassad ABC 802-Basic
2 totalt 80k ram, varav 48k ramfloppy
3 UFD-dos
4 Seriell printerrutin
5 förberedd för CP/M
6 förberedd för nätverk

Enl. Börje är man intresserad att förhandla om rabatter för ABC-medlemmar. Jag undrar nu om någon vet om klubben har börjat förhandla med Sala Teknik, som tydligen har övertagit basic2-projektet från Liber? Vidare vore det bra att veta hur många ABC80-ägare vill hotta upp sin gamla kärra! För något år sen antydde Gunnar Tidner för mig att klubben var i kontakt med Liber för att förhandla fram en rabatt för medlemmarna!
Hör av er ni som vet!!

(Text 930) Bo Kullmar * <1789>
Ärende: basicii till abc80
Nej, det är nog ingen som har varit i kontakt med Sala Teknik för att förhandla med rabatter. Det som vi har gjort är bara så att vi för några år sedan gav lite synpunkter till en konsult som Luxor hade anlitat för projektet.
Kan du skriva dett brev till syrelsemötet här och ange telefonnummer till Sala Teknik, så hoppas jag att vi kan få någon ABC80 fantast i styrelsen att ta kontakt med Sala Teknik för att får fram någon form av rabattavtal.
Vi har ganska ont om tid, eftersom matrial till ABC-Bladet nr 4 skall vara inne snart.

(Text 940) Jörgen Gustavsson <3528>
Ärende: CMDINT.SYS
I våras fördes en diskussion i MSG om man behövde detta till annat än formattering. Jag kommenterade då att det fanns ett program i basic som klarade formattering och att jag skulle försöka få författaren till programmet att sända in det. Börje Gustavsson har nu sänt in sitt FORM.BAS. Nu skulle jag vilja återuppliva diskussionen: Har man någon nytta av CMDINT.SYS?

(Text 941) Bo Kullmar * <1789>
Ärende: CMDINT.SYS
Svaret är Nej, om man frågar Luxor. Se bara på Meny0 där allt ligger i BASIC-program. Svaret är Ja, om man frågar DAIB. DAIB kör mest under DOS och brukar till och med se till så att man kan anropa ett BASIC program från DOS.
Jag tror det är en vana. Det är nog svårare att köra från DOS, eftersom man använder switchar. För övrit av samma typ som används i Monitorn. Något som t ex Meny0 blir väldigt enkelt att använda.
Själv använder jag ofta ett formaterinsprogram i BASIC som fanns med på någon skiva till BASIC II boken för ABC806. Jag har fått programmet av någon. Jag tror dock inte att det är fritt att lägga in här. Fördelen med detta program är att man slipper anropa DOSGEN varje gång på nytt. För min egen del så skulle jag nog vilja svara nej på frågan. På CAT-NET kan man t ex överhuvudtaget inte köra program under DOS! Kan man anropa DOS program från BASIC så räcker det.

(Text 944) Anders Franzen <5258>
Ärende: CMDINT.SYS
Det enda skälet till att använda CMDINT.SYS är väl möjligheten att lagra maskinkod i ABS-filer som kan läsas av både ABC80 och ABC800. Man kan ju förstås fråga sig vilka program som går att köra båda maskinerna. Nej, det finns nog ingen anledning att använda CMDINT. Program skrivna i BASIC kan göras mycket mer användarvänliga, t ex finare menyer.

(Text 945) Benny Löfgren * <2615>
Ärende: CMDINT.SYS
Varför säger du så? Man kan göra precis samma användarvänliga menyer om man kör program under DOS som om man skriver programmen i BASIC. DOS är inte bara assembler, om nu någon tycks tro det.
Jag tycker att hela diskussionen är löjlig, ställ samma fråga till en IBM PC-ägare: Behöver du verkligen ditt PC-DOS, klarar du dig inte med BASICen? och vad tror ni man skulle få för svar?
Användarvänlighet är nog bra - så länge man är nybörjare. När man börjar lära känna sitt system så börjar man tycka att det är bekvämt att kunna ange alla parametrar till det program man ska köra på kommandoraden, så man slipper vänta på att programmet ska laddas in, och sen svara på ett antal frågor och sen vänta på att programmet är klart.

(Text 946) Jörgen Gustavsson <3528>
Ärende: CMDINT.SYS
Det kan hända att diskussionen förefaller löjlig för en del.
Bakgrunden till att jag ställde frågan är följande:
- Jag kan inte assembler eller annan maskinnära programmering. Jag använder min dator flera timmar om dagen och då till 95% färdiga program. Min egen programering inkränker sig till en del basic-program samt anpassning av andra program till mina behov.
- På systemskivan som följde med datorn har jag kopierat in en hel del andra program. En del av dessa förutsätter CMDINT. Det ena efter det andra av dessa program har jag hittat basicvarianter till. Detta ser jag som en fördel: Jag kan läsa och ev ändra programmen + endast detta program behövs på den skiva jag använder. Det enda programmet jag till slut använde under CMDINT var DOSGEN. När sedan FORM.BAS kom ställda jag denna fråga.

(Text 953) Henrik Schyffert <2314>
Ärende: CP/M och KERMIT
Som glad amatör så har jag fått veta att Kermit inte går att använda på ABC-80 och CP/M för V24-porten stämmer inte. Är det rätt och i så fall vad skall man använda under CP/M för att kunna köra kom ??
För det andra så förstår jag att det finns massor av program i CP/M på 8'skivor i IBM-format 3740 (tror jag). Dessa program har jag möjlighet att använda men jag har 51/4' skivor i format ABC-831 (som är samma som 832 men bara på ena sidan). Kan någon kopiara ???? Skulle det bli flera kopior så vet ju inte jag om det.
Ang ABC-ord så finns det 'debuggade versioner' —brevlådan—.
Är någon intresserad av en 'deal' ???
Henrik och Stig

(Text 957) Ola Arwidsson <5275>
Ärende: EJ CP/M och KERMIT
KERMIT finns till ABC800 (basic), skulle inte den kunna gå att använda ? (eller är jag ute och cyklar).

(Text 958) Bo Kullmar * <1789>
Ärende: EJ CP/M och KERMIT
Ja, du är ute och cyklar. KERMIT:en till ABC800 är skriven i BASIC II och kan ej köras på ABC80 varken under CPM eller under BASIC I. Sedan använder sig nämnda KERMIT av den sk inbyggda terminalrutinen i ABC800-serien och en sådan finns inte i ABC80. ABC80:s hårdvara saknar ju sådant som normalt behövs för kommunikation.
På ABC80 måste man köra bitbang och sinumera en SIO. Jmf David Andersson terminalrutin SOFTSIO till ABC80. Jag har för mig att Myab säljer någon terminalrutin som man kan köra på ABC80 under CPM. Kolla med MYAB.

(Text 959) Anders Franzen <5258>
Ärende: EJ CP/M och KERMIT
Vad är bitbang?

(Text 963) Karl Lindström <837>
Ärende: EJ CP/M och KERMIT
Bit-bagning är att du har en byte (= 8 bitar), slänger ut den (t ex) den högraste biten på en tråd, roterar en bit åt höger, slänger ut den högraste igen roter åt höger en gång till osv tills du har slängt ut 8 bitar. Då har man slängt ut en byte seriellt på en tråd.

(Text 970) Anders Olsson <1019>
Ärende: 64K a la Microdatorn
Jag har sett en del rättelser till artikeln i senare nummer. Kärnan är dock att just 8-bitars refreshen inte finns med, och det har vad jag vet inte publicerats någon rättelse om det. (Har jag fel?)
Jag håller på att skriva på en beskrivning på den minnesombyggnad som jag själv har gjort. Om jag får den färdig är det inte otänkbart att jag skickar den till bladet.
Kruxet är att minnesombyggnaden förutsätter ett specialskjutet bipolärt prom för minnesavkodningen. Så jag kanske inleder med en byggbeskrivning på en promskjutare för d:o, vem vet? Den som har tålamod får se!

(Text 978) Kjell Svensson <5318>
Ärende:RAM-kort
Vet någon vad som skiljer mellan de olika expansionslådorna när det gäller minnesbussen? Jag har ett CMOS ramminne som fungerar utmärkt i min expansionslåda (ABC), men när jag provade i klubblokalen (DATA-DISK) ville det inte längre. Jag misstänker (i min låda):

stift A4= $\overline{\text{XMEMFL}}$

stift A5= $\overline{\text{XMEMW}}$
Hoppas på hjälp.
Kjell.

(Text 979) Kent Berggren <6019>
Ärende:RAM-kort
Det kan inte vara så enkelt att du satt in den på fel sida i Datadisken. Den har en I/O och en mem sida.

(Text 980) Bo Kullmar * <1789>
Ärende:RAM-kort
Så enkelt kan det inte vara för även ABC-exp har IO och MEM på olika sidor. Jag vet att det skiljer sig också, för det funkar inte att köra CAT-NET kort i en Tranfor IO låda. Däremot får det utmärkt i en ABC-låda. Jag vet inte vad som skiljer, men LG vet i alla fall.

(Text 1004) Arne Hartelius * <2731>
Ärende: Gejo's 80-tecken och printer
Nej, jag har Gejo's 80-tecken och är nöjd med det. Men OK, det är betydligt krångligare än Myab's innan man har fått fason på det hela. Vad gäller krocken mellan bildminne och printerrutin så har det bara betydelse vid hardcopy. Har man däremot Ramdator som jag har så måste man ju välja bort printerrutinen. Men den kan man ju då lägga på den plats där Myabs 80-tecken ligger. Det går väl på ett ut, men är som sagt krångligare i början med Gejo's. Gejo's var vid det tillfälle jag köpte den ca 500 kr billigare än Myab's. Den stora fördelen med Gejo's är att när jag laddar in 800-tolken så kan jag köra både 40 och 80-tecken. Har man vant sig vid 80-tecken från ABC80 är det inget lajbans att köra 40 på 800:an. Enligt uppgift har dock Gejo slutat sälja sin 80-teckenstillsats.

(Text 1007) Hans Johansson <1678>
ÄRENDE: EJ CP/M och KERMIT
Trots att man måste fuska lite med metoden för överföringen så tycker jag att det borde gå att fixa en KERMIT till ABC80. Jobbar för tillfället med KERMIT i ett program från Autocode och som host en UNIX dator, funkar fint, menm visst vore det kul att även kunna labba lite med ABC80'an. Så skicka gärna ett brev till mig om det är någon som vet mer eller om det är någon som har lust att sätta sig och skriva ett KERMIT prog till 80'an. Jag har en del dokumentation och protokollbeskrivningar till KERMIT.

(Text 1017) Johan Struwe <3338>
Ärende: Konstigt Z80-register?
Jag har en skiss över Z80-CPU:ns uppbyggnad framför mig, och har till min stora förvåning upptäckt ett register med namnet WZ! Vad används detta register till? Det går inte att komma åt ifrån maskinkod. Från vissa håll har jag hört att registernamnen HL,BC och DE skulle stå för en förkortning:
Hi-Low,Byte-Counter & DEstination Är detta en efterkonstruktion månne?
Varma kalsingar !!!!

(Text 1022) Peter Öhlen <4659>
Ärende: Konstigt Z80-register?
WZ-registret har den funktionen att vid instruktioner när man behöver mellanlagra ett värde.
T ex: LD A,(nn)
Först hämtas opcoden, två följande bytes hämtas och läggs i WZ-reg.
I nästa steg läggs WZ ut på adr.bussen och läsning beordras, varefter det som finns på databussen läggs i A.
HL heter så för att det används som adr.reg., och då var det ju logiskt med High-Low.
Mr Qwerty

(Text 1038) Sven Wickberg <1384>
Ärende: Formatering m m
När man formaterar en skiva med DOSGEN får man en "blank" skiva, vilket innebör att man fr o m lägsta minnesaddressen (32768) får FF och 00 i varannan byte. Litet högre upp (möjligtvis vid starten på det gamla 16K minnet) blir det plötsligt 00 i varenda byte.
Varför?
När letade efter en adress i basictolken upptäckte jag att ;CALL(292) på min ABC80 med checksumma 9913 (8881 vid 80 tkn) blanker hela minnet med 00 i varje byte. Någon som vet varför?

(Text 1043) Arne Hartelius * <2731>
Ärende: Formatering m m
Vad menar du ? Reultatet av DOSGEN,F är en skiva med idel E (E-FNUTT) i varje sektor utom vissa biblioteks-sektorer. Resultatet av DOSGEN utan påföljande F är att biblioteksspåren nyinitieras, men att de gamla programen ligger kvar på skivan och går att leta upp och rädda med t ex Börje Gustavssons förträffliga RESCUE.BAS (ABC80/DISKHANT-biblioteket).
Eftersom .ABSprogram inte kan förutsätta annat än grundutrusning dvs att man bara har minne från 49152 -> så arbetar dessa program bara i den övre delen av minnet. ABS betyder ju ABSolute-adress. Formatering eller dosgenerering av en skiva har inte med den undre delen av minnet att skaffa. Den enda del av minnet som hammnar på skiva är dosbuffert 0 slumpmässiga innehåll som inte töms på skräp innan den kopieras till sektor noll på skivan (den sektor där skivnamnet står).
Jag är säker på att du kan allt det här sedan länge. Men varför blandar du då ihop skivformatering med minneskartan ?

(Text 1048) Sven Wickberg <1384>
Ärende: RAM-minnet efter uppstart
Rätt, Arne, jag skrev förmodligen i nattmössan. Jag menade som det står i rubriken nu, att RAM-minnet i min ABC80 efter uppstart innehåller FF resp 00 omväxlande från 32768 och uppåt något på 40000 där det plötsligt blir 00 i varje.
Jag undrade varför det är så. Det har kanske ingen praktisk betydelse, men man tycker det vore naturligt att ett junfruligt blankt minne innehöll idel nollor (eller idel ettor) och inte ibland noll och ibland 1.

(Text 1049) Arne Hartelius * <2731>
Ärende: RAM-minnet efter kallstart
Jag undrar också över det där. Helt oviktigt kanske det inte är eftersom den enda byte som skiljer checksumman 10042 från 9913 är en som har till uppgift att ge 'de sista opröGramerade byten' ett värde.
Varför besvära sig med ett omständigt checksummebyte för en sån detalj om det saknar vikt.

(Text 1054) Sven Wickberg <1384>
Ärende: Varning för tillsatser
Min ABC80 har jag liksom mång andra försett med extra minne och 80 tkn. Det har inte givit några problem. Mend et finns mer att välja på: RAM-floppy, UNIDISK, CP/M. Jag har satt in Gejos RAM-floppy för att komma åt BASIC II på ABC80, och då har jag haft alla möjliga konstiga fel: programbortfall, underliga felmeddelanden, låsningar. Eftersom jag körde den "nya" ABC80 med separat kraftpaket och extra monitor trodde jag det var fel på dem, men Håkan hos Björnhems förklarade att de var ok.
Det har nu påpekats för mig att jag antagligen har för mycket grejor i min burk. Likriktaraggregaten till ABC80 tappar orken med tiden, och marginalerna är så små att man lätt råkar ut för att ström och spänning inte räcker till för alla extragrejor. Detta skulle förklara varför vi tidvis haft besvär med alla handikapptillsatser, som också skall dra nytta av spänning och ström från ordinarie strömförsörjning.
Är det någon som rett ut detta och vet exakt besked?
Vad jag vet i dag är att t ex CAT-NETs kretskort drar nästan dubbelt så mycket som tillsatser skall få göra totalt - det har mätts upp och påpekats för mig. Men det tycks gå bra om man inte har något annat än 32 K RAM exta.

(Text 1055) Gunnar Faith-Ell * <2733>
Ärende: Varning för tillsatser
Jo, det stämmer att nätagregatet till "nya" ABC80:n endast är dimensionerat för datordelen, pröva att byta likriktarbrygga och säkringar (transformatorn är överdimensionerad). Ev kan spänningsregulatorn behöva bytas men det beror på hur mycket ström man behöver. Jag har hört talas om att vissa nätagregat inte ens orkade driva datordelen, men efter ovastående åtgärder så gick de bra.

(Text 1062) Mikael Liden <5651>
Ärende: 16K RAM extra ABC80
Vilket sätt är enklast och billigast att bygga ut ABC80:s minne med 16K RAM ?

(Text 1063) Peter Öhlen <4659>
Ärende: 16K RAM extra ABC80
Enklast är att köpa ett kort som du sätter i disken. Ett sådant kort kostar runt en tusenlapp tror jag. Det billigaste är en sats som ABC-Data säljer och den kostar 595 kr+moms. Det krävs dock att man gör fyra lödningar och plocka bort några kretsar och sätta två sm åavlånga kort i stället för de DRAM som satt där innan. Jag har för mig att ABC-Data kan utföra det om man tycker att det verkar vara för svårt. Det billigaste är förståss om man kommer över ett kort begagnat.

(Text 1064) Kent Berggren <6019>
Ärende: 16K RAM extra ABC80
Det finns yterligare ett sätt och det är att löda på den nya kapslarna på de gamla. "pigi back". Beskrivning har funnits i någon gammal Microdatorn. Vilken vet jag ej. Doch finns beskrivningen nere på klubben.

(Text 1065) Bo Michaelsson <913>
Ärende: Standardbandspelare till ABC 80
Finns någon tillsats för att kunna koppla en vanlig bandspelare till ABC 80 ? Det vore mycket trevligt att kunna spara på kassetter eller rullband nu när man inte kan köpa det vanliga kassettminnet längre. Rullband skulle också kunna användas som en billig förvaringsplats för filer då banden rymmer mycket.

(Text 1066) Sven Wickberg <1384>
Ärende: Standardbandspelare till ABC 80
Såvitt jag förstått går en standardbandspelare utmärkt till ABC80 om man sätter på och stänger av den för hand. Den saknar således ett relä som startar och stoppar motorn resp inkopplingen av kapstan-rullen i takt med programmet, annars är det som vanligt.

(Text 1067) Bernt Johansson <3384>
Ärende: Standardbandspelare till ABC 80
Det går aldrig att läsa tillbaka filer från en audiobandspelare. Tror att det beeror på att den inte har DC-utgång. Kassettminnet skickar ut en TTL-kompatibel signal med fast DC-nivå.

(Text 1070) Stefan Berg <216>
Ärende: Standardbandspelare till ABC 80
Är du säker på det ? Jag lyckades efter diverse justeringar både spela in och av på en vanlig kassettbandspelare, utan några fel alls. Bergis nå't elektroniskt eller så...

(Text 1071) Kent Berggren <6019>
Ärende: Standardbandspelare till ABC 80
Va då skulle det inte kunna gå med en standard bandspelare. Man kan väll ha en nivå omvandlare och en pulsformare så måste det kunna fungera. I alla fall på dessa låga farter.

(Text 1072) Bernt Johansson <3384>
Ärende: Standardbandspelare till ABC 80
Ja, det är just det man måste ha. En flankdetektor och pulsformare och det finns i kasstettminnet. Dom allra allra första ABC80 gick att köra med en audiospelare. Det följde faktiskt med en kassettradio som standard. Men säkerheten blev för dålig.

(Text 1080) Bo Engborg <2369>
Ärende: Interrupt och Chr$(5)
Vad händer (före),under och efter en interrupt?
Ex..

```
Rutin  xyz
       xyz
       EXX
       CALL NÅGONTING  ; Tar lång tid

       EXX
       FORTS...
```

Hur ska man förhindra att interrupt sker under Call någonting (för att kunna rädda registren).
Kanske med rätt kombination av DI,EI ..??
Vad sker när en Videotexbas begär ENQuire, Chr$(5) ??
Vad ska datorn svara för att därefter kunna ge kodord ??
Hälsningar Bos!!

(Text 1083) Benny Löfgren * <2615>
Ärende: Interrupt och Chr$(5)
En interruptrutin FÅR INTE förstöra något register, varken de ordinarie eller primregistren. För att stänga av alla interrupt utom NMI gör man DI, för att slå på dem igen gör man EI. Sedan kan vissa av perifierienheterna, t ex PIO- och SIO-kretsarna fås att stänga av sina egna interrupt, men det är olika beroende på vad man pratar med.

(Text 1114) Arne Hartelius * <2731>
Ärende: Processorns odelade uppmärksamhet
Jag förstår inte heller vad jag pratar om. Det är därför jag vill att du ska skriva en artikel om det. En disasemblering av mina inlägg ! Nä allvarligt talat. Om man använder sig av en hjälpare typ SuperSmartAid, SuperBasic, Lökaren eller varför inte din egen SLAV så kommer dessa att sluta fungera om man använder sig av programet VÄXLAPGM. Jag antar att detta beror på att den avbrottsrutin som finns i VÄXLAPGM inte gör retur till den avbrottsrutin som finns i ens hjälpare utan istället går direkt tillbaka till Bascitolken, adr 799, eller var det nu var.
Vad jag efterlyser är alltså en generell rutin som undersöker om det finns någon annan rutin som använder sig av tangentbordsinterupt och i så fall lagrar startadressen till den i slutet av sig själv. Om inte så gäller adressen för återhopp till tolken.
Rutinen ska alltså bara användas vid initieringen av en ny rutin. Den ska bara lämna ifrån sig en adress som den egentliga rutinen ska använda sig av vid återhopp. Sen kan den utplåna sig själv om den vill.
Tänk dig att du har den här snutten i början av Basicprogramet VÄXLAPGM. Den undersöker om det finns en hjälpare med i bilden (minneskartan). Denna hjälpares avbrottsrutin börjar på en viss adress. Just denna adress grabbar vår lilla snutt tag på och spar. Fanns det ingen hjälpare är adresen 799. När VÄXLAPGM sen hunnit till inpokandet av själva asemblerkoden läggs så denna adress in på därför avsedd plats.
Förstår du vad jag är ute efter ?

(Text 1119) Anders Franzen <5258>
Ärende: Processorns odelade uppmärksamhet
Jo, jag anar vad du menar. Det bästa vore kanske att göra som fadern till TERM100, att låta varje program som ställer till det med olika interuptrutiner städa upp efter sig när man lämnar programmet. Det finns dock många gamla program som inte gör detta och det kan vara mycket jobbigt, kanske onödigt, att skriva om dem. Jag kan inte så här på rak arm säga om man kan spara adressen till en interuptrutin INNAN den lagts in i minnet. Man kan tänka sig ett program som man exekverar först då man lagt in t ex VÄXLA och som på någotvis....
Nej, jag vet inte. Måste tänka, återkommer.

(Text 1133) Arne Hartelius * <2731>
Ärende: Mjuk respektive hård exit från program
Nej, nej. Det är ju just det jag (vi?) vill undvika. Massor av ABC80-ägare har inte printerrutin i prom utan måste ladda in detta vid uppstart. Det gäller alla som har bandspelare eller enkel densitet på på sin drive. Massor av medlemmar använder sig av en ramladdad hjälpare. En del har rammaskin och eller ramfloppy.
Ingen som tillhör någon av de uppräknade grupperna är speciellt förtjust i att behöva initiera sytemet efter varje programkörning, så det vore en katastrof om det blev kutym att lämna program med CALL(0). Idealet som jag ser det är som följer:
--
Att det program man kör återställer BOFA och STACK till de värden de hade innan och dessutom försäkrar sig om att enhetslistan inte kommer att förstöras av ny programinladdning.
Att information om varmstartsadress lämnas om sådan kan vara aktuell i sammanhanget.
Att man när det gäller assemblerrutiner i möjligaste mån lägger dessa under taket istället för ovanför golvet. Vid ett ev missöde har man då större möjligheter att initiera rutinen på nytt utan ny inladdning.
Att man inte använder sig av DOS-buffertar utan att undersöka om de är lediga, eller själv använder sig av dem utan vederbörlig upptagenmarkering. --
Jag betraktar RESET-knappen som en nödutgång och tycker att det är ett osnyggt sätt att (behöva) avsluta en programkörning på.

(Text 1136) David Andersson <5201>
Ärende: Processorns odelade uppmärksamhet
I förstora.bas kollar initieringsrutinen vad värdet är på I-reg. innan det ställs om. Sedan förutsätter den att låga byten av vektorn (lagrad i PIOn) är 52 och tittar i minnet (adress I*256+52,I*256+53) var tidigare interuptrutin ligger sedan avslutar den inte sin egen interuptrutin med

```
EI
RETI
```

utan med

```
JP adress
```

där adress är uträknad sålunda:

```
LD A,I
LD (SPARAI),A    ;spara gamla I-reg
LD H,A
LD L,52
LD E,(HL)        ;DE=adress till int-rutin
INC HL
LD D,(HL)
LD (adress),DE   ;självmodifierande kod
```

Med denna hantering får man automatiskt JP 798 om I=0.
Man bör inte glömma att ha en adress till cassett-interupten när man ändrar I.
Om alla program som använder int gör så här som måste de termineras i motsatt den ordningen de initierades.
Om PIOn används för annat än tangent interupt så blir det genast mycket krångligare.
(assembler rutinen ovan är inte testad)

(Text 1152) Sven Wickberg <1384>
Ärende: CALL(292)
På min ABC80 med cs 9913 innebär ;CALL-(292) att HELA internminnet NOLLSTÄLLS. Rubb och stubb inte bara FÖRSVINNER - varenda adress blir NOLL. Detta sker inte ens vid RESET eller CALL(0). Jag tycker mig ha märkt att CALL(0) ibland inte ens tar bort program ur internminnet utan bara sätter första adressen=1 vilket datorn tolkar som att minnet är ledigt.
Jag har Superbasic och med rutinen OLD kan jag alltså få tillbaka program efter reset (liksom efter NEW som också ettställer första adressen) samt vid program som är preparerade med den gamla run-only-rutinen (som ju verkar så att första byten blir 1 efter körning så att man inte kan lista eller köra igen utan ny inladdning från skivan).
Är det någon som kan förklara verkan av CALL(292) eller varför denna finns över huvud taget - när den inte används vid uppstart av systemet. Då är nämligen varannan adress = 0 och varannan = 255 och en hel del andra oregelbundenheter förekommer. (Jag har tittat efter med Superbasics SCAN.)
För övrigt kan man ta sig ur BYE-mod med $ABC om man har Magnus Lundbergs Superbasic + diverse andra program, utan att nolla alla andra rutiner i minnet. Vet inte hur det fungerar.

(Text 1153) Benny Löfgren * <2615>
Ärende: CALL(292)
Det är med största sannolikhet så att det inte är någon MEDVETEN rutin som ligger på adress 292. Det är säkert bara en konsekvens av att man hoppar in mitt i en rutin, eller till och med mitt i en instruktion.

(Text 1156) Gunnar Forssell <1631>
Ärende: CALL(292)
Var det inte Halmstads-David som hade gått djupare in i det där?
Jag har för mig att han upptäckte att ABC80'n försökte att kompilera en rad, men då man inte hade några ingångsvärden satta så satte den igång med att friskt kompilera ickeexisterande saker. Och då uppstår visst effekten med blankning av minnet...
(En snabbtitt i dissen av basicen visar att det mycket riktigt är i det området kompileringen sker. Inhoppet sker straxt efter sättning av vitala parametrar...)

(Text 1160) Magnus Bodin <4304>
Ärende: CALL(292)
Jo, CALL(0) gör endast NEW genom att sätta BOFA=EOFA, (BOFA)=1.
Call(292) är ett inhopp mitt i en rutin, resultatet blir som det blir pga att man kommer från BASICen. Till skillnad från assemblernivå kan man ju inte här kontrollera vad som ligger i varje register när man anropar en maskinkodsrutin. (Vilken man förresten kan i FORTH!) Resultatet blir ett slags LDIR-FILL. Helt oavsiktligt, men sen Kalle Lindström skrev ett brev till Bladet har det varit allmänt känt, trevligt uthopp för "diskreta" program..

(Text 1168) Gerry Eriksson <4842>
Ärende: Utslitet tangentbord
En kondensator är i princip två plattor med ett isolerande skikt emellan t.ex luft, polykarbonat m.m.
Det är även så som tangenterna i ABC80 fungerar. De kan ses som en variabel kondensator där kapacitansen ändras när du trycker ned dem. Det är denna ändring man känner av mellan de två halvorna och INTE om de är kortslutna (=MYCKET dålig kondensator). Det blir ett mindre luftgap och därmed ökad kapacitans d.v.s. tangent nedtryckt. Jämnför även en radio där ratten för frekvensinställning är kopplad till en vridkondesator som består av ett antal fasta plattor och en axel med plattor på som vrids in olika mycket imellan de fasta = frekvensen ändras. På samma sätt är det med aluminiumplattan på tangenten.

(Text 1178) Kent Berggren <6019>
Ärende: Utslitet tangentbord
Om du motorar ner en knapp ser du den ytan som skall tvätt på krets kortet. Ser ut som 2 halvor av en cirkel. Förtennad. Det bästa medlet att tvätta bort smuts är isopropanol efter som det inte länar några rester
Inlägg 1179 är raderat.

(Text 1191) Anders Franzen <5258>
Ärende: VXLPGM
Programmet VÄXLAPGM.UTL innehåller några allvarliga buggar. Det går inte att läsa filer från kassett och man får inte skriva in för långa rader när VÄXLAPGM är aktivt.
ABC80 dyker när man läser från kassett därför att avbrottsvektorn för kassettinterupt saknas. Om en BASIC-rad skrivs in

och som kompilerad upptar mer än 72 bytes dyker datorn därför att avbrottsvektorn för tangenttryck placerats i kompileringsbufferten! Man kan åtgärda det första felet på ett enkelt sätt.
Det andra är mer svårlösligt. Det bästa vore om maskinkoden till VÄXLAPGM placerades på något annat ställe än i POKE-arean högst upp i minnet. Man kan emellertid flytta avbrottsvektorn till annan plats. Dessa rader gör VÄXLAPGM lite bättre:
151 IF PEEK(65064)=251 POKE 65064,250

(sänk stack)
540 POKE 64052,148,255,148,5
(interuptvektorer)
541 POKE -112,250
(rätt värde till I-reg)
Rad 540 ersätter gammal rad, de övriga är tillägg. Nu kan man läsa från kassett utan problem. Denna ändring ökar maxlängden på BASIC-rader till 199 bytes vilket dock är för litet ibland.

(Text 1193) Olov Lindström <6411>
Ärende: Hur adresseras filerna på floppyn
Track 1 och 2 verkar innehålla pekare till ..et block (T*256+S*32) där ytterligare inf ..ns.
..en vad betyder innehållet i den sektorn som pekas på. Det vore fint om någon kunde svara på det.
O.Lindström

(Text 1195) Bengt Österholm <2077>
Ärende: Hur adresseras filerna på floppyn
Det är en headersektor som i sin tur pekar ..å olika sektorer där programmet finns. ..r kan man räkna ut hur många sektorer ..t program upptar. Man kan även utläsa på vilken plats i biblioteket filnamnet står på. Det här har beskrivits i ABC-bladet tidigare men jag kommer inte ihåg vilket nummer det var.
2-82(RED)

(Text 1200) Mikael Hovmöller <4688>
Ärende: String$
Varför klarar inte string$(x,y) större tal på x än antal bytes som finns kvar i minnet? Varför har man inte gjort en loop, som gör Call 11 x ggr? (Det kanske fungerar så, men då borde det inte bli ERR 3 när x är för stort.)
M.v.h. Micke

(Text 1202) Anders Franzen <5258>
Ärende: String$
När du skriver "; STRING$(x,y)" placeras den skapade strängen på stacken. Då måste det ju naturligtvis finnas tillräckligt med utrymme på stacken. Notera att SPACE$ fungerar på samma sätt. Att göra en loop som gör CALL 11 för varje tecken i strängen vore dumt, vem säger att vi nödvändigtvis ska skriva ut strängen? Tänk om vi vill göra "A$='x'+SPACE$(100)"! Genom att lägga den skapade strängen på stacken i samma format som övriga strängar kan den behandlas som en vanlig sträng också.
Här kommer ett tips i samband med detta: Skriv ej så här: 10 ; A$+space$(x)+b$+ ..ring$(x,y)
..kriv så här istället: 10 ; a$ space$(x) b$ string$(x,y)
Använd alltså inte "+" i printsatser! Summan av allt läggs på stacken och man kan råka ut för ERR 3 om programmet är stort!

(Text 1227) Stefan Persson <1980>
Ärende: Generera BREAK från TERM100
Kan man på något listigt sätt generera ..den för BREAK, en tangent som finns ..i vissa terminaler och som används i bl a LUNET (Lund University Network). Jag hittar inget tips i handledningen för Term-100. Jag vet inte ens vad BREAK-tangenten skickar ut för kod på nätet.
Tacksam för hjälp.
<1980> Stefan P.

(Text 1228) Stefan Berg <216>
Ärende: Generera BREAK från TERM100
Jodå. Om du lägger MENY-funktionen som finns på CTRL-E (E-FNUTT) på någon annan tangent (görs genom att gå in i MENY med CTRL-E (E-FNUTT) men att gå därifrån genom att trycka på t ex CTRL-Q) så kan du generera BREAK med CTRL-E (E-FNUTT). (Har för mig om att jag gjorde så.) Om jag inte missminner mig så skickas då ASCII 0 (NUL) ut på linan, d v s signalen går låg. (Rätta mig, den som vet bättre !)

(Text 1230) Kristoffer Eriksson <5357>
Ärende: Generera BREAK från TERM10
Break brukar vara längre än ett NULL-tecken, och ingår inte i ASCII-alfabetet. Man får inte break genom att trycka flera gånger på CTRL-E (E-FNUTT) heller, eftersom start- och stopp-bitarna mellan NULL-tecknen skiljer det från break. Men försöka duger ju.

(Text 1275) Kjell Svensson <5318>
Ärende: ASS
En medlem (icke modemägare) ville att jag skulle påpeka ett fel i ASS. Han påstod att instruktionen RLC B gav resultatet 203D istället för 203,0.
Han saknade också möjlighet att ändra en rad som gett syntaxfel. Jag har själv inte provat programmet i fråga.
Kjell.

(Text 1286) Anders Franzen <5258>
Ärende: ASS
Medlemen utan modem har helt rätt, ASS klarar inte av assemblerinstruktionen RLC B. Hela familjen RLC utom RLC A blir felaktig. Det beror på ett fatalt fel. Data för alla assemblerinstruktioner ligger i en tabell och datavärdena för RLC-instruktionen är helt riktiga, det är bara det att programmet som läser tabellen tolkar värdet fel! Sorry, jag får väl försöka rätta felet.
Det går inte att med ASS ändra en rad som gett syntaxfel. Använd istället editorn TED (eller någon annan) för att skriva källkod med, spara texten på fil och kör sedan programmet ASS för att assemblera filen. För att rätta syntaxfel använder man sin editor igen.
Tills felet med RLC är rättat, var god undvik RLC s där s inte är A.

(Text 1293) Kjell Svensson <5318>
Ärende: JP 197 resp JP 799
JP 197 =NEW
JP 799 =T-bordsinterruptrutinen
Kjell.

Inlägg 1294 är raderat.

(Text 1295) Kent Berggren <6019>
Ärende: BASICERR.SYS
Jag har inte hittat någon som kan förklara för mig hur error texten kommer fram automatiskt vid fel. Igentligen skulle jag vilja använda detta till att åter starta mitt prg istället. Jag har sätt någon som lyckats med det, men jag har inte riktigt fattat hur. Igentligen skulle det räcka med att ABC80 gjorde Z=call(0). Är det någon som har gräft ner sig i BASICERR.SYS?
Mycket tacksam om någon kan hjälpa mig.

(Text 1298) Anders Franzen <5258>
Ärende: BASICERR.SYS
BASIC.tolken är så finurligt skriven att den känner av om BASICERR.SYS finns på dina skivor. Det görs när du trycker RESET. När du får ett fel i BASIC kontrollerar tolken om BASICERR.SYS finns (kollar den flagga som sätts av det-som görs-vid-reset-rutinen) och anropar isåfall en subrutin i DOS. Vad den gör vet jag inte men man kan ju gissa! Den läser väl in aktuellt felmeddelande beroende på felkod och visar på skärmen...

(Text 1327) Kjell Svensson <5318>
Ärende: TKN 40/80 - jag blir galen!
Det finns många fler ställen man kan kolla radbredd på,PEEK(590) eller PEEK(472) bör vara de mest korrekta ställena att titta på. Där får du ju 40/80 som svar.
PEEK(884) kan variera beroende på vilken 80tkn-ombyggnad man har.
Kjell S.

(Text 1332) Bengt Österholm <2077>
Ärende: TKN 40/80 - jag blir galen!
Jag tycker att peek (900) är mycket bra för detta ändamål. Det är ju lätt att komma ihåg och resultatet är också 40 respektive 80 beroende på vilket mode skärmen är i.

(Text 1350) Peter Öhlen <4659>
Ärende: Odokumenterad Z80-instruktion
Instruktionen vars kod är CB,37 hex är inte dokumenterad men finns i Z80n. Den ska ha fäljande funktion, "Shift left, insert 1". Är det någon som vet varför den inte är dokumenterad och om det finns fler sådana instruktioner.
/nyfiken

(Text 1360) Kjell Svensson <5318>
Ärende: SEKTOR 1-5 PÅ SKIVAN
Används sektor 1-5 på skivorna till något?
mvh Kjell S.

(Text 1362) Peter Öhlen <4659>
Ärende: SEKTOR 1-5 PÅ SKIVAN
Nej!

(Text 1365) Kristoffer Eriksson <5357>
Ärende: Odokumenterad Z80-instruktion
Om instruktionen inte finns dokumenterad,hur kan då veta att den finns? Jag har för mig att någon för en tid sedan skrev att de här "gluggarna" i instruktionsuppsättningen berodde på instruktionerna inte fungerade som de skulle.

(Text 1369) Kent Berggren <6019>
Ärende: Odokumenterad Z80-instruktion
Det finns någoneller några instruktioner som inte finns dokumeterade. Jag mins att det var någon på Metric (troligen Björn Damm) som skrev ett prg där han utnyttjade sig av en av dessa rutiner. Sedan märkte man att prg inte fungerade på alla Z80 och han fick ringa Zilog i staterna och kolla. Det fanns då flera "masker" av Z80 och det skilde sig mella olika fabrikat. (Masker = layout på kretsen)
Jag tror att ni skall fråga Mats Petterson 1082 om han mins det hela. Han brukar logga in i bland här.

(Text 1374) Sven Wickberg <1384>
Ärende: Mina filer sabbas
Jag har en skiva med autostart och TERM100 som alltid fixar till en massa saker vid inkoppling (fixar printerformat och ger viss valfrihet till TED, TERM100 m m)
Plötsligt började det uppträda ERR 35, 37 och framför allt 48 på diverse filer.
Jag tog Magnus Lundbergs DDISP för att titta på skivan: biblioteket på spår 2 såg rätt ut, men när jag tittade på en av de aktuella filerna fann jag att headern var borta! Sabbad av en annan fil som skrivit över den sektorn.
Jag har hört att det felet kan uppkomma, men jag har också ML:s justerade DOS som lär ska ha rättat till det felet. Nu är min fråga: är detta vanligt förekommande? Kan man göra något åt det? (Jag hade BACKUP /duktig gosse!/ så det gäller inte filerna, utan själva systemet.)
I annat fall är det tydligen lugnast att aldrig ta bort en fil som en gång kommit in på skivan. Det tycks gå bra så länge man lägger till alla filerna bak, men när man börjar leta efter lediga sektorer inne bland de andra händer det här.

(Text 1376) Magnus Stråle <2504>
Ärende: Odokumenterad Z80-instruktion
De icke-dokumenterade instruktioner som jag känner till berör IX & IY registrena. Man kan t.ex addera A med låg eller hög del av IX resp. IY. Jag har tyvärr inte mina papper tillgängliga nu, men det skall nog inte vara så svårt att lista ut objektkoden för dessa instruktioner.
M.v.h MS

(Text 1377) Bengt Österholm <2077>
Ärende: Mina filer sabbas
Jo, jag har varit med om det där också. Det är tydligen så att headern aldrig kan ligga ensam, utan när den skrivs in så skrivs även nästa sektor in oavsett om den sektorn är ledig eller ej.
Det enda botmedlet jag kunde komma på var att när man har brukat en diskett en längre tid så att den är ordentligt segmenterad så kan det vara lämpligt att kopiera den fil för fil så att alla lediga sektorer hamnar längst bak tillsammans.

(Text 1382) Kenneth Borg <1738>
Ärende: Likriktare
Efter att ha servat ca 25 abc80 i 5 år vill jag säga att det i särklass vanligaste felet på abc80 är att den stora likriktarbryggan till 5v lämnar in. Skumma fel som rensningar av minnet och svajig bild o.dyl. kan bero på likriktaren. Kontroll göres med oscilloskop, och man mäter spänningen över den stora elektrolytkondensatorn. Om frekvensen på "ripplet" är 50 Hz är likriktaren trasig.

(Text 1411) Sven Wickberg <1384>
Ärende: Sändfel med TERM100 ?
Jag har TERM100 (ett fantastiskt bra program !!) som har fungerat till belåtenhet nästan ett år. Men nu har jag fått problem: När jag försöker sända en fil, huggs de 3-4 vänstersta tecknen bort, dvs i stort sett radnumret.
Vad kan det bero på? Jag försökte skicka filen till Norrköpingsmonitorn, och det blev likadant varje gång.

(Text 1413) Arne Hellström <4838>
Ärende: TERM100 Ver 1.4
Kan någon hjälpa mig att ändra på antalet sektorer som reserveras på disketten med Y1 i TERM100. Antalet sektorer som nu reserveras(prepareras?) är 32. När dom är använda så reserveras 32 nya. Det är i den skarven jag får skräp i mina filer. Om man kunde välja antal sektorer själv så skulle det inte behöva hända. Detta kanske inte händer på en modern drive Men jag har en gammal FD2.

(Text 1418) David Andersson <5201>
Ärende: Sändfel med TERM100 ?
TERM100 sänder nog alla tecken korrekt, men Norrköpingsmonitorn kan nog inte ta emot tecken som kommer direkt efter en carriage-return, altså de första tecknena på en rad.

(Text 1419) David Andersson <5201>
Ärende: TERM100 och diskallokering
TERM100 reserverar inga sektorer på disketten. Det gör DOSet när man skapar en fil och skriver till filen.
Doset ska deallokera överflödiga sektorer när filen stängs.

(Text 1421) Johan Struwe <3338>
Ärende: BASIC-problem ?!
Jag satt och höll på med ett BASIC-program till min ABC80 där jag använda listor och matriser. I detta fall strängvariabler A$(x) och A$(y,z). I början av programmet hade jag en DIM-sats:
10 DIM A$(100)
och lite längre ner hade använde jag stränglistan med samma variabelnamn.
100 ; A$(1,4)
Detta ville inte datorn gå med på. Jag fick ERR 1 LINE 100, "Fel antal index". Inte förrän jag bytte variabelnamnet på rad 100 till 100 ; B$(1,4) så fungerade det. Ska det vara så här? Jag har hittills haft för mig att ABC80 kan använda alla variabeltyper samtidigt och oberoende av varandra.
Har jag haft fel?
(Hoppas jag har presenterat frågan korrekt - det var en tid sedan problemet uppstod - minnet kanske sviker...)

(Text 1423) Bengt Österholm <2077>
Ärende: TERM100 Ver 1.4
Det låter som om du inte har satt protokoll til YES i keypad-mode R4. När din dator har fyllt 32 sektorer så lägger den upp 32 sektorer till. Detta tar lite tid och om du då har satt protokollet så sänder din dator XOFF när den inte hinner med att lagra alla tecken som kommer. Då slutar monitorn att sända tills du sänder XON igen. Det görs automatiskt när din FD2:a har skrivit klart i header-sektorn och är redo att ta emot mer data.

(Text 1436) Anders Franzen <5258>
Ärende: BASIC-problem ?!
Pga en bugg i tolken kan man inte ens skriva 10 DIM A$(1),A$(1,1)=1
När man skriver RUN får man ERR 1 (fel antal index). Det är alltså inte möjligt att ha en vektor och en matris med samma namn. Fast det går ju att kalla dem för A1$ och A2$. Samma underlighet gäller heltals- och flyttalsvariabler.

(Text 1437) Gerry Eriksson <4842>
Ärende: BASIC-problem ?!
Som jag ser det är det inte frågan om en bugg i tolken. Det finns två huvudtyper av variabler enkla resp. indexerade variabler (skalära o matrisvar.). Och du bara kan använda samma varabelnamn en gång för resp. variabeltyp. Hur många index den innehåller spelar då ingen roll utan namnet är upptaget för typen indexerade variabler. Däremot kan man ha samma namn men olika antal index för heltal, flyttal och strängar. T.ex. DIM A%(20),A(5,10),A$(4,4,4)
Samma regler gäller för övrigt även 800:an.
Mvh Gerry

(Text 1438) Arne Hartelius * <2731>
Ärende: Matriser på ABC80
A$(4,4,4) var väl ändå att ta i ?

(Text 1448) Anders Olsson <1019>
Ärende: Reparationer av flexskivenheter
Om det kostar mellan 1000 o 2000 kronor att få en FD reparerad borde väl den lite händige hellre än att lämna FDn på rep köpa en ny drive o sätta in. Det går att få nya slimline drivar för under 1500 kronor. De är dessutom mycket driftsäkrare än de gamla fullhöjdarna. (Direktdrift utan rem, stegmotor med stålband i stället för skruv m m).
Man måste bli mycket misstänksam om man får tillbaka sin floppy efter rep och "det enda som behövdes var justering". Justeringen *ändrar* sig normalt inte (om inte den som justerat sist glömt att dra åt skruvarna), men däremot kan olika detaljer slitas ner så att justeringen inte stämmer. Då måste man först byta den slitna detaljen och först därefter justera.

(Text 1455) Kent Berggren <6019>
Ärende: RTS
Anders F. frågade mig varför det är så viktigt att kunna hantera det. Jo det använder man när man t ex "pratar" med en printer. Den talar då om att den är klar att ta imot data. RTS o CTS. Kan man använda sig av när man inte har XON/-XOFF. och inte kör med modem. Ta till ex om du vill föra över en file från en anna dator som inte har det protokollet. Du tappar ut datan ur dess printer utgång. Och för att stoppa data använder du dig då av RTS o CTS. Mycket enkelt. Tag Metrics prg T80RTS som finns som är det enda prg till ABC80 som jag vet som hanterar dessa signaler okey. Jag kan oxo tänka mig en plotter som man kan köra 2vägs. Både skriva till och hämta från. Att där syslar man inte med XON/XOFF.
I alla fall finns det massa olika uppfatningar om hur man skall hantera data vi v24:

(Text 1466) Bengt Holgersson <559>
Ärende: Extraminne till ABC80 med 64k enligt MIKRODATORN
Kort beskrivning hur man kan bygga ut sin ABC80 med 64k till multiplar av 64k. (128k, 192k,256k o.s.v)
Beskrivning av grundutbyggnaden finns i tidningen MIKRODATORN.(Nr9,10 -83 och Nr1 -84)
OBS !!! DETTA ÄR INGET FÖR PERSONER UTAN PRAKTISKA KUNSKAPER OM HUR EN MIKROPROCESSOR ARBETAR.
Principen för expanderingen är att löda på fler RAM-kapslar ovanpå de redan existerande. (Piggybacking)
De enda stift på RAM-kapslarna som som skall kopplas annorlunda är:

RAS (Row Adress Strobe)
Klockar in ena halvan av adressen

$\overline{CAS}$ (Column Adress Strobe)
Klockar in andra halvan av adressen

För varje ny bank med 64k behövs:
1st 74LS10
8st 4164 (Obs, måste vara varianten med 7-bit refreshadress)

Se bild längre ned.

Adressavkodningen av de nya bankerna görs lämpligen av ett bipolärt prom i likhet med tidigare. Man får då stor frihet att skräddarsy vissa banker för speciella funktioner. (CP/M,BASICII,RAMDISK el.dyl) Att bygga en programmeringsenhet för bipolära prom kan man göra ganska enkelt med hjälp av databladet över prommen. I övrigt är det kanske inte lämpligt att använda OUT7 som utport för bankswitchningen eftersom denna ej är fullständigt avkodad.
M.v.h Bengt H

(Text 1472) Anders Olsson <1019>
Ärende: Extraminne till ABC80 med 64k enligt MIKRODATORN
Tack, Bengt! Du har utan att veta om det gett mig lösningen på något som jag har grunnat på i ett par år, nämligen:
4164 MÅSTE VARA VARIANTEN MED 7-BIT REFRESH
Jag har klurat o klurat o klurat på den där beskrivningen i Mikrodatorn och det enda jag har kommit på är att den rent teoretiskt aldrig skulle kunna fungera. Mina 4164or har 8-bits refresh. Jag gjorde en egen konstruktion med ett litet kretskort som lade till den 8:e biten.
Vad har 4164:or med 7-bits refresh för fabrikat / beteckning?

(Text 1476) Leif Andersson <155>
Ärende: SNABBFORMATTERING
Kan någon tipsa om ett prg som formatterar disketter utan att man varje gång behöver svara på alla frågorna.Har ett svagt minne att det skall finnas någonstans,vad heter det i så fall?

(Text 1479) Jörgen Gustavsson <3528>
Ärende: SNABBFORMATTERING
Det finns ett program på sista eller näst sista kassetten som heter FORM (tror jag). Visserligen frågar programmet, men om du inte vill svara på frågor är det lätt att ändra programmet. Att programmet är lätt att ändra beror på att det är gjort i basic och inte körs under CMDINT. Börje Gustavsson har visst svängt ihop programmet.

(Text 1484) Gunnar Forssell <1631>
Ärende: Ram-disk.
För alla er glada som har byggt ut ABC'n med 64k med hjälp av Mikrodatorn så har jag modifierat den ram-diskrutin som har funnits här på monitorn lite. Den funkar som ABC802's ram-disk.
Jag ska lägga in den nu, få se om jag klarar av det med TERM100...

(Text 1489) Gunnar Forssell <1631>
Ärende: Extraminne till ABC80 med 64k enligt MIKRODATORN
Bra beskrivning! Jag har ett par frågor dock.
Kan du göra prommar enligt specar? jag skulle vilja ha bankar om 32k vardera där det är de nedre 32k'na som berörs.
Hur fixar man CS-signalen? Via prommet? Kan inte du skriva lite mer utförligt hur man gör? Visa hur man gör en komplett 64k extra sats alltså?

(Text 1494) Bengt Holgersson <559>
Ärende: Extraminne till ABC80 med 64k enligt MIKRODATORN
De flesta japanska varianter av 4164 har 7 bit refreshadress. Det enda jag har stött på med 8-bit adress är Siemens. Men det finns säkert flera. Det säkraste är att fråga INNAN man köper dem. Man kan få 150ns 4164:or för c:a 10 kr. st:
M.v.h Bengt H

(Text 1496) Bengt Holgersson <559>
Ärende: ADRESSAVKODNING MED BIPOLÄRA PROM
Vi kan ta ett exempel som gäller alla nämligen ABC80:s ursprungliga adressavkodning. Detta görs i ett bipolärt PROM som sitter i position E7 och har beteckningen ABC80/13
Man bör nog ha ett schema över datorn för att hänga med riktigt. (Sno er och skaffa en servicemanual innan de tar slut.) Nåväl, i denna kan man se att detta prom har storleken 256 x 4 vilket betyder att man har 256 celler om 4 bitar vardera. För att kunna adressera dessa 256 celler behövs 8 ledningar varav 6 stycken är kopplade till CPU:ns 6 övre adressbitar. Detta betyder att man kan få ut en valfri 4-bit kod för varje 1k-block av processorns adressområde.
De resterande 2 inbitarna är kopplade till två byglingar för att kunna välja en av 4 minneskartor.(0-3) Normalt har dessa värdena 1 resp. 0 vilket medför att minneskarta 1 är vald. De 4 utbitarna används sedan för att välja 1 av 4 typer av minne:

BIT 0	Externt minne	Aktiv låg
BIT 1	Internt ROM	Aktiv låg
BIT 2	Bildminne	Aktiv hög
BIT 3	Internt RAM	Aktiv låg

Eftersom man bara vill ha 1 typ av minne inkopplat åt gången så får vi följande koder för de olika typerna av minne.

Internt RAM	0011 = 3	(Hexadecimalt)
Bildminne	1111 = F	- " -
Externt minne	1010 = A	- " -
Internt ROM	1001 = 9	- " -

Om vi sedan tittar på innehållet i prommet (ABC80/13) så kan ni försöka gissa vad konstruktören har haft i tankarna.

```
000 AAAAAAAAAAAAAAAA !MINNESKARTA 0
010 AAAAAAAAAAAAAAAF !
020 AAAAAAAAAAAAAAAA !
030 3333333333333333 !
040 9999999999999999  !MINNESKARTA 1
050 AAAAAAAAAAAAAAAF  !
060 AAAAAAAAAAAAAAAA  !
070 3333333333333333  !
080 9999999999999999   !MINNESKARTA 2
090 99999999AAAAAAAF   !
0A0 AAAAAAAAAAAAAAAA   !
0B0 3333333333333333   !
0C0 9999AAAAAAAAAAAA    !MINNESKARTA 3
0D0 AAAAAAAAAAAAAAAF    !
0E0 AAAAAAAAAAAAAAAA    !
0F0 3333333333333333    !
```

M.v.h Bengt H

```
SCHEMA ÖVER INKOPPLINGEN AV RAS OCH CAS
=======================================

                +-----+         +-----+
CS (NY) ---+--!     !     +--!     !
           ! !      !     !  !     !
           +--!  &  !O--+--+--!  &  !O--- RAS (NY)
              !     !  !     !     !
IC F7 pin 10 -!     !  ! +--!     !
              +-----+  ! !  +-----+
                       ! !
                       ! !  +-----+
IC E6 pin 10 ----------(-+--!     !
                       !  !     !
                       +--!  &  !O--- CAS (NY)
                          !     !
IC E3 pin 6 -------------------!     !
                             +-----+
```

(Text 1507) Lennart Einarsson <6554>
Ärende: FÖR LÅNG RAD....
Jag har ett program som är sparat på kassett, det är inkört med 'LIST' kommando. Och det är en rad i programmet som har mer än 120tkn. Jag är väldigtntresserad av att få igång den filen igen men jag lyckas inget vidare. Det enda jag kan åstakomma är ERR 20. Men jag har för mig att man kan poka maskinen till att inte bry sig om sådana bagateller. Problemet är, vilken adress är det.
H E L P ... M E !!

(Text 1509) Gunnar Forssell <1631>
Ärende: FÖR LÅNG RAD....
Jag har för mig att det finns ett pgm, 'ERR20FIX' eller något ditåt, som gör det du vill. Det är inte bara att poka!

(Text 1511) Kent Berggren <6019>
Ärende: LÅNGRAD.BAC
Ligger i KAS/K17
Den räddar för långa rader.

(Text 1517) Arne Hartelius * <2731>
Ärende: FÖR LÅNG RAD....
Använd FILOMV .BAC - CAS 2
eller LÅNGRAD .BAC - CAS 17
eller TED .BAC - ABC/EDITORER

(Text 1542) Bo Kullmar * <1789>
Ärende: Tangenter till ABC80
Enligt Kjell Svensson (som fn saknar modem) så säljer Björnhems (återförsäljare i Sthlm) lösa tangenter till ABC80. Luxor säljer bara hela tangentbord och det blir dyrare.

(Text 1555) Anders Johnson <4001>
Ärende: 16K extra RAM som självbygge
Det finns en beskrivning för utbyggnad av ABC80:s RAM till 64K i Mikrodatorn nr 9 1983. Finns det någonstans en motsvarande beskrivning för utbyggnad med bara 16K extra, dvs motsvarande div. tillsatser soom finns till salu hos olika firmor? Jag tycker det är för dyrt med ca 600:- plus moms för en inbyggnadssats för 16K extra. Det är, tror jag, ungefär vad den billigaste kostar.

(Text 1561) Gunnar Faith-Ell * <2733>
Ärende: 16K extra RAM som självbygge
Ja, det finns det i nummer 7,1982, den beskrivningen lär faktiskt fungera utan ett antal uppdateringar i senare nummer.

(Text 1562) Kent Berggren <6019>
Ärende: 16K extra RAM som självbygge
Jag tror att det är den jag har byggt efter Funkar ganska bra, men inte om du har GEJE:s ramfloppa.

Inlägg 1563 är raderat.
Inlägg 1564 är raderat.

(Text 1568) Anders Johnson <4001>
Ärende: Bildskärm/TV/TKN80
Skall skaffa en bildskärm för 80-teckens ABC80.
Duger en svartvit TV till detta? Jag syftar på billiga 12-tumsapparater som finns till salu lite varstans. Om videobandbredden är för liten, brukar det då vara enkelt att öka den genom något smärre ingrepp (såsom är fallet med originalskärmen för ABC80)? Det vore intressant att se åsikten hos någon som har personlig erfarenhet.

(Text 1569) Arne Hartelius * <2731>
Ärende: Bildskärm/TV/TKN80
Jag borde vara den sista att yttra mig om hårdvarusidan, men vad jag vet ligger det till så här:
En vanlig TV ritar 625 linjer och börjar sen om med första linjen. Detta görs under en viss tidsenhet. (25 ggr / sek) ?
En monitor ritar 312 linjer under samma tidsehet, men gör det i gengäld 2 ggr. Detta för att få en flimmerfri bild.
Så en monitor borde väl alltid vara att föredra. Men jag kanske har missuppfattat det hela.

Möte Medforum

(Text 1239) Stefan Persson <1980>
Ärende: VAD KÖR NI PÅ?!!
För dem som har möjlighet och intresse/-fallenhet att utveckla program till ABC80 med 32 K och 40/80 tecken, så får man väl utgå från att en större del av klubbmedlemmarna fortfarande nyttjar denna utrustning? Sven Wickberg har ju t ex sjungit trotjänarens lov flera gånger, och jag har ingen tanke på att byta. Hur är det med de andra i klubben? Av inlägg i bladet att döma finns det fortfarande många "enstusiaster" kvar.

(Text 1240) Göran Westlund <5976>
Ärende: VAD KÖR NI PÅ?!!
Kanske kan ABC800/802/806 så småningom bli "hobbymaskin"? Det finns ju väldigt många maskiner ute (jag har hört om kring 50 000, kan det vara sant?). De flest av dessa har väl anskaffats för "yrkesmässigt" bruk. Då kommer de också antagligen att fungera länge än efter det att företaget bytt. Jag vet t ex att ett stort antal statliga myndigheter är på väg att byta till större maskiner. Så det borde bli en marknad för billiga begagnade 800-system. Kanske kunde klubben med hjälp av Luxor bevaka detta? Hellre borde ju dessa dyrgripar spridas till glada amatörer (t ex f d ABC80-ägare) än hamna i sopen.
Sen är man ju väldigt spänd på hur 800an ska vidareutvecklas. "Den får väl självdö" hävdade en "Nokia närstående person" som jag talade med nyligen. Han menade att Nokia inte kunde ha plats för två olika PC-system inom sin produktion. Luxor kanske i framtiden får ägna sig åt att tillverka Nokias PC? Men Nokia-PCn är ju en vanlig, charmlös MS-DOS-maskin. Vad jag ser framemot är några nya 800-modeller, t ex "804" - en prisbillig 802-variant med 3" disketter (paketpris 8000 med ORDIII, Kalkyl och registerprogram i paketet, åssså en hyfsad skrivare för 1500 därtill, om man vill ha), samt en "808", en 806a med monokrom grafik samt 20 Mbyte halvhöjds winchester inbyggd, i kompakt utförande. Förslagsvis till ett pris av ca 12000:-. Och en "810", lika 808 men med nätmöjlighet integrerad. Är det någon som vet Luxors policy? Tar 1600 och 9000 allt intresse nu, eller kommer 800an att leva vidare?

(Text 1347) Patric Ljung <5455>
Ärende: Kopieringskydd
I Bladet nr 3 1985 står det på sid 50 om kopieringskydd. Man skulle alltså sticka hål på skivan och sedan kolla om det går att läsa sabbad sektor och skriva på den. Detta är inte allt för bra, då läshuvudet kan skadas om skivan är deformerad. Vidare kan man väl sätta på skyddstejpen vilket innebär att man inte kan skriva, eller?
Om man då sätter på en skyddstejp, kan man inte skriva och skyddet blir meningslöst om man sätter tejp på skivan (även orginal). Stämmer inte detta???

(Text 1348) Kent Berggren <6019>
Ärende: Kopieringskydd
Du har missuppfattat det hela. Det är frågan om ilegalkopiering.

(Text 1350) Benny Löfgren * <2615>
Ärende: Kopieringskydd
Jag tror inte att förslaget innebär någon skrivning, det räcker väl att konstatera ifall skivan går att läsa, eller hur?
Vad beträffar skador på läshuvudet så finns det bot för det också, dock inte med så enkla medel som ett nålstick... Det har nämligen i USA utvecklats utrustning för att 'brännmärka' skivor med laser, man bränner ett litet, litet hål i skivan, och detta hål ger samma effekt som ovan nämnda nålstick.

(Text 1353) Patric Ljung <5455>
Ärende: Kopieringskydd
Att man brännmärker skivor med laser visste jag, och att nålmärkning är litet riskabelt med tanke på läshuvudet. Dessutom syns det ju, borde i alla fall synas, att man petat hål. Så det borde ju inte vara en omöjlighet att "pricka" in samma sektor, eller? Om man är litet exakt av sig så.
Moss / Patric

(Text 1379) Bo Kullmar * <1789>
Ärende: Försäljning av adresser ur medlemsregistret
Vi säljer i bland adresser ur registret och enligt styrelsebeslut 1984 så är kostnaden 500 kronor i engångsavgift och 2 kronor per etikett. Detta har Berit Gustavii själv rätt att avtala med "kunder" om. I tvekssamma fall hänvisar hon till mig eller styrelsen enligt nämnad beslut.

Naturligtvis så blir det inga etiketter till de som har meddelat oss att de inte vill ha direktreklam. Sen är det ju självklart att vi säljer bara för direktreklam för dataprodukter. Vi har inte sålt så många adressetiketter hitintills.
Nu finns det dock en framställan från P--Data. De var inte nöjd med priset ovan, så de har kontaktat Stig Löfgren. De skall få etiketter men jag vet inte vad Stig har gjort för överenskommelse vad gäller priset. P-Data vill visst erbjuda medlemmarna att köpa program till 50% rabatt. Jag känner dock inte till några närmare detaljer, så fråga mig inte!
Annars vill vi helst att de skall annonsera i ABC-Bladet i stället. Datainspektionen har inget emot att vi säljer adresser. Vi får göra vad vi vill med vårt register. Vi har licens. Däremot är det ett brott mot datalagen om någon medlem skriver av matrikeln och upprättar ett dataregister. Det spelar då ingen roll hur den tekniska överförigen sker.

(Text 1428) Benny Löfgren * <2615>
Ärende: Kopieringskydd
Ja, du får för det första CRC-fel när du försöker läsa in sektorn. Dock är det då så att sektorn ligger kvar i controllern, och kan enkelt läsas därifrån med ett enkelt controllerkommando.

(Text 1462) Gunnar Faith-Ell * <2733>
Ärende: Kopieringskydd
Vissa datorer kan få driven att strunta i tapen (driven måste då naturligtvis vara gjord för det oxo). Anledningen är att de då, avsiktligt, kan skriva sönder en piratkopia.

(Text 1466) Stefan Berg <216>
Ärende: Kopieringskydd
Hmm, det är väl bergis inte så bra för företagets image ? Vad blir nästa steg - att slå ut själva burken om det ligger en piratkopia i minnet ? (Borde gå om man använder en gammal PET, har jag hört nånstans...)

(Text 1473) Gunnar Faith-Ell * <2733>
Ärende: Kopieringskydd
Har inte Apples Macintosk vissa möjligheter till att kvadda diskar med skyddsremsan på ?

(Text 1481) Arne Hartelius * <2731>
Ärende: Styrelse presentation.
Mitt telefon nummer står varken i gamla matrikeln eller på något annat ställe i ABC-sammanhang. Ändå blir jag dagligen uppringd av medlemmar som vill ställa frågor och få tjänster utförda. Att publicera våra telefonnummer på framträdande plats vore ju en uppmaning till medlemmar att ringa i tid och otid.
Kent, du måste förstå att vi redan lägger ner större delen av vår fritid på ett obetalt arbete, och att vi vill ha en privat del av tillvaron också.
Våra telefonnummer kommer att finnas i den nya matrikeln, om än inte speciellt duplicerade på baksidan.

Möte MONITOR

(Text 2449) Bo Kullmar * <1789>
Ärende: 020-linjen
Som jag skriver i ABC-Bladet som kommer ut nu är det fn inte möjligt att införa 020 tjänsten, tyvärr. För det första så tillåter Lurverket bara det för Datavisonssystem och då skall det vara temporärt fram till att man kopplar ihop sig med Televerkets Datavisonssystem.
För det andra så är intresset för det från medlemmarna för litet. Den förfrågan som jag riktade i ABC-Bladet nr 2 har gett ett mycket litet antal svar. Något 10 tal, bara.
Detta tycks beror på att många som kör riks, kör det på ett företags bekostnad. Sen är det ju många som riner från 08-området, vi har ju 1/4 av medlemmarna här uppe.
Televerket skall starta ett brevsystem och till det kommer man att använda 020. Kanske kan Televerket mjukna med tiden och då kanske kan vi skaffa oss det. QZ är i samma situation som vi. Vanlig 020 tjänst som erbjuds till vanliga abonnenter är för dryt för där får man betala **varje** samtal till högsta rikstaxan.

(Text 2470) Bo Kullmar * <1789>
Ärende: 1 år!
Nu om några timmar är det snart ett år sedan vi öppnade sysemet. Oj vilken skillnad det är nu på kapacitet vad gäller antalet linjer. Vi har ju nu dessutom två nya 802:or som bara väntar på att kopplas in. Winchester till LUX-NET kommer nu att leveras vilken dag som helst. Klubben blir den första kund som köper den. Den finns avbildad i Tranfors annonsbilaga på bild nr 9. Den innehåller en 52 MB winchester, en 640 KB floppy och en streamer för bandbackup samt 15 kortplatser. Jag kommer ihåg när jag tog ner den gamla monitorn och startade upp den nya för alla medlemmarna! Vi hade problem med CAT-NET:s centrala ISAM då. Det vara bara en prerelease, dvs förutgåva. Den klarade inte index på svensk sortering, men när vi körde med ASCII sotering så gick det. Detta innebär faktiskt att vi fortfarande kör med ASCII sotering och inte svensk ordning på ÅÄÖ! Fast det har väl ingen märkt!
PS annonsbilaga i ABC-Bladet nr 3, som just nu är på väg ut med posten.

(Text 2522) Sven Wickberg <1384>
Ärende: Har det hänt något med PB?
Ursäkta att jag stör i ABC-monitorn, men hur ska man klaga i PB när man inte kommer inte? Jag har ringt 20 samtal i dag på eftermiddagen. Försökt alla numren vart för sig. Antingen blir det inget svar, eller är det LINE FAILURE (jag ha automatmodem) eller kommer jag fram men får inte inloggningssekvensen. EN GÅNG kom jag in, men blev efter en stund abrupt utkastad. Någon som vet eller kan ge beskedet vidare?

(Text 2523) Bo Kullmar * <1789>
Ärende: Har det hänt något med PB?
Ja, jag brukar köra på PB någon gång i veckan bara. Det händer att jag får problem att komma in också. Fast du verkar ha ovanligt mycket problem.
Ett tips kan vara att du kanske skall felanmäla din telefon och säga att det går dåligt när du ringer Åkersberga. Jag hade liknande problem när jag ringde hit och ringde då felanmälan för Data för 80 64 40. Ibland kom jag in på en dålig linje som gjorde att jag **alltid** tappade bärvågen efter några minuter. Televerket sa att jag skulle felanmäla min igen telefon i stället, vilket jag har gjort. Får se om problemen upphör.

(Text 2528) Kent Berggren <6019>
Ärende: XOFF
Jag har kommit under fund med att det efter ett Xoff kommer en del tecken innan det hinner stoppa på andra sida. Ca 7 st i ABC-monitorfallet. Hur är det med andra stora monitor behöver skriva av sig fler tecken innan de stoppar. Jag skall göra en buffert där de tecken som kommer efter ett XOFF skall hamna och då vill jag veta hur stor man måste göra den.

(Text 2542) Benny Löfgren * <2615>
Ärende: XOFF
Monitorn reagerar blixtsnabbt på XOFF, så jag tror inte på att 40 tecken inte räcker. Är det inte ditt BASIC-program som inte hinner med? Längst tid tar det i 1200/75, då XOFF-tecknet givetvis skickas med 75 baud, dvs 1/16-del av den mottagande hastigheten. Då får man ett teoretiskt glapp på 16 tecken innan XOFF ger effekt.

(Text 2570) Bo Kullmar * <1789>
Ärende: Sökning i MSG
MSG programmet i CAT-NET är så stort som det kan vara. Det är länk squezat, eftersom man inte kan ta in ett så stort program i minnet. I LUX-NET består MSG av två program, varav nr 2 är lite mindre. Som sagt är det inte programstorleken som hindrar det för det går att göra ett 3:e MSG program för sökning. Nej, det är alltså det faktum att disken skulle bli alltför slö som hindrar det. Vi får räkna med att vi måste bygga ut systemet med fler noder i stället!

(Text 2648) Bo Larsson <5630>
Ärende: Koppla V.24/RS-232
BOK att REKOMENDERA. Igår köpte jag boken Koppla V.24/RS-232 av Joe Campbell, en mycket lärorik och faktiskt smått underhållande bok.
Boken beskriver de problem man kan stöta på när man skall försöka koppla ihop två seriella enheter, hur man lurar gränssnittet samt mycket mera. Detta är en bok som borde ha funnits hemma hos de datorpulare som blivit gråhåriga vid tretton års ålder när de i sitt anletes svett försökt få sin nyinköpta printer handskaka med sin dator. För att underlätta interfacing beskriver boken vidstående koppling, som är enkelt testverktyg för att anlysera spänningar och polaritet i sitt gränssnitt. För att rätt lysdiod skall reagera enligt ovanstående har jag märkt den flata sidan (och eller) det korta benet med (o. Resistansen på motståndet på kan variera med 20%. Med denna koppling i bakhuvudet kan man ju för en billig penning lätt tillverka sin egen Breakout Box, som i handeln kostar skjortan.

(Text 2652) Kent Berggren <6019>
Ärende: KERMIT
Hur går det med att implimentera den till monitorn? Jag har för mig att jag någon stans har sett att man hade för avsikt att göra det. Det ligger ett program som heter Rkermit under ABC800/kommunik. Vad jag förstår så är den ansed för att användas i monitor system.

(Text 2655) Bo Kullmar * <1789>
Ärende: KERMIT
Inget är gjort ännu. LG skall görat, men nu skall vi i första hande köra igång LUX-NET systemet. Vi får se när LG har tid annars får jag väl göra en insats sen när jag har tid.
Det som heter RKERMIT måste köras under en REMOTE programvara och det kan vi inte använda.

(Text 2656) Bo Kullmar * <1789>
Ärende: 1200/1200
Ingen 802:a kommer att finnas ute i lokalen och det är inte meningen att någon 802:a skall tas i anspråk för lokala arbeten normalt. Måste det ändå ske, så skall det vara en på gruppnummret. För lokala körningar mm kommer den 806:a som finns i lokalen att användas.
Min **ambition** är att vi skall kunna byta ut modemen till modem som klara såväl 1200/1200, 1200/75 och 300/300 och i samband med detta köra allt på gruppnummret. Detta kan tänkas ske någon gång under 1986.

Möte Lokalavd

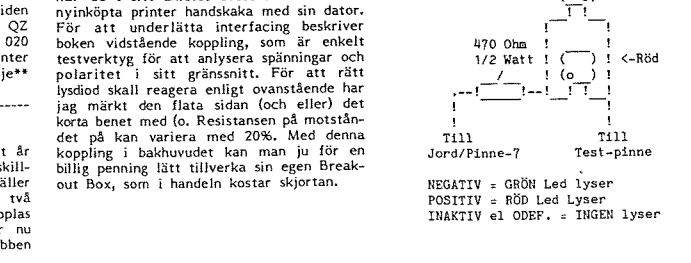
(Text 46) Linus Thrybom <1848>
Ärende: Göteborgsmonitorn
Va bra med en Göteborgsmonitor tänkte man när senaste bladet kom!
Men icke!
I luren hörs bara en evig upptagetton. Vadan detta? Har det kraschat (eller är det bara så att alla de ca 25 gånger jag ringt dit har någon varit där före mig?) Någon som vet?
Mvh Linus

(Text 47) Bo Kullmar * <1789>
Ärende: Göteborgsmonitorn
Ring Anders Lundberg och fråga. Telefonnummret finns i en fil i TEXT/DIV. Den innehåller adresser och telefonnummre till alla lokalavdelningarna.
Jag hörde visst att det vara några problem, bl a med telefonledningarna.

(Text 67) Bo Kullmar * <1789>
Ärende: Lokalavd
Nja, intesset för detta möte är inte så stort från Sthlm-are iheller så det lönar sig knappast att starta ett nytt möte.
Stöd utgår från ABC-Klubben i form av pengar, 5 kronor per medlem inom regionen och år. Vi kan ordna med pengar till mötes-lokal och sådant. Dessutom har jag en del lappar från ett 20-tal personer i Skåne som vill vara med i en lokalförening i Skåne. Naturligtvis kan vi också köra ut etiketter från medlemsregistert på alla som har 2 som första siffra i postnummret och bekosta utskicket.

(Text 68) Bengt Holgersson <559>
Ärende: Lokalavd
Jag undrar även om det kanske kan utgå någon form av bidrag från tillverkaren av datorerna. Rabatter och dylikt.
Om man skulle starta något så krävs det ju lite hårdvara även om man kanske inte behöver nån 40Mbyte monitor med en gång. Vidare kanske man behöver lite tips om hur allting bör organiseras. Mötet jag föreslog kanske kunde hjälpa till att sammanföra de som är intresserade så att något händer.
M.v.h Bengt H

Bo Larsson <5630>



NEGATIV = GRÖN Led lyser
POSITIV = RÖD Led Lyser
INAKTIV el ODEF. = INGEN lyser

(Text 69) Gunnar Faith-Ell * <2733>
Ärende: Lokalavd
Varför inte göra ett försök med ett utskick inom området ?

(Text 70) Gunnar Faith-Ell * <2733>
Ärende: Lokalavd
Jag tror nog att Luxor m.fl intressenter gärna vill se att en förening kommit igång innan de beslutar sig för att bidraga med utrustning. ABC-Klubben har viss rabatt som kanske kan utnyttjas. För det första måste ni ha ett möte där en styrelse väljs, lämpligen skickas en kallelse till alla medlemmar inom regionen. Och då ser ni även hur stort gensvar ni får.

(Text 71) Bengt Holgersson <559>
Ärende: Lokalavd
Jag vet att Mats Knuts kom med en liten trevare för några år sedan men det rann väl ut i sanden.
Det kanske kunde hjälpa om moderklubben organiserade nån form av enkät för att se hur många som är inresserade att jobba aktivt.

(Text 73) Gunnar Faith-Ell * <2733>
Ärende: Lokalavd
Det bästa är väl om du fick fram en förlaga och sedan bad klubben om bidrag till porto och kopieringskostnader.

LOKALAVD.

Möte MSG

(Text 9) Sven Wickberg <1384>
Ärende: INMATNINGSMINNE
Vad är det som gör att ABC och PB kan ta emot en lång rad tecken från tangentbordet i sitt minne, medan många andra system bara minns ett enda tecken. (Jag tänker på att man kan köra sin inloggningssekvens medan datorn bladar på med Välkommen (2 gånger) och en massa annan spaltfyllnad.)

Möte FORTH

(Text 63) Ulf Dahlen <2422>
Ärende: UFORTH
Jag har knåpat ihop en FORTH som jag fyndigt döpt till UFORTH (Ulfs FORTH är en vettig gissning). Den hanterar textfiler istället för screener och går således att köra även på ett kassettsystem. Den har ord som understödjer grafiken, random access till skiva, många strukturord etc.
Är någon intresserad, så ta kontakt med mig! Antagligen kommer jag att ge alltihop till ABC-klubben.

(Text 66) Magnus Bodin <4304>
UFORTH: Intresse!
Källkoden är av stort intresse, lägger du in den i FORTH-biblioteket??
Magiska hälsningar Magnus :-)

(Text 67) Magnus Bodin <4304>
UFORTH: Intresse!
Ja, källkoden vore av intresse, liksom källkoden till FORTH2, men UFORTH kanske ersätter denna? Har du förresten bytt FORGET enligt FORGET.FIG, i kärnan alltså?
Random-access? Vad lägger du för betydelse i det uttrycket? Kan man spara tolken med nya definitioner a la FORTH18?
Ja, fler frågor finnes, och dessa skulle besvaras å det rappaste om du "la in" källkoden på FORTHbiblioteket.
Förresten, har du en CROSScompiler?
Håller på att skriva en till ABC-FORTHen, kan kanske vara av intresse till klubben också???
Har man en CROSScompiler, så kan man ju skriva program i FORTH som blir körbara utan att man behöver ladda in kärnan, skärmfiler etc..
Magiska hälsningar Magnus

(Text 68) Ulf Dahlen <2422>
UFORTH: Intresse!
FORTH2 är ju bara den gamla vanliga FORTHen med några extraord. Jag skickade in den för ganska länge sedan och visste inte att den hade lagts in i programbanken. Extraorden är för grafik (SETDOT etc) och strukturering (CASE-typ). Min dokumentation borde också finnas. Annars ska jag försöka lägga ind den (tar ett tag, nu bor jag i Linköping och ABC:n finns i Båstad).
UFORTHen är en alldeles egen skapelse gjord helt från scratch. Den gör allt som BASICen gör och lite där till.
Med random access menade jag helt enkelt motsvarande CALL(28666) osv i BASICen.
UFORTH hanterar textfiler precis som

BASICen. Du kan skriva ditt UFORTHprogram med en vanlig texteditor (TV t ex) och sedan ladda in den i UFORTHen eller skriva den direkt i UFORTHen.
Ord för hantering av textfiler (OPEN, PREPARE etc) finns naturligtvis. UFORTHen klarar att göra lite grafikanimation om än inte så snabbt som SPRITE-paketet som kom på någon kassett.
Om jag ska lägga in UFORTHen vill jag också lägga in hela dokumentationen och dessa filer är inte riktigt klara. En manual (25 sidor) finns dock.

(Text 76) Magnus Bodin <4304>
Ärende: FORTH-83
Sven Östberg gick tyvärr bort i Augusti, vilket innebar att en av våra mesta FORTH-programmerare i klubben även lämnade oss. Som alltid måste man ta nya tag då något sorgligt händer, och trots det låta ens vanliga liv gå sin gilla gång.
FORTH-83 var ju ett av Svens stora projekt, att implementera, göra om skärmar (anpassning) till FIG-FORTH.
Jag skulle gärna vilja återuppta Svens arbete där han slutade, då på en ABC-maskin, helst ABC806. Anpassning skulle då ske till '800-doset så att man inte skulle vara bunden till UFD-dos. Man skulle då även ha möjligheten att köra på nätverk typ CATNET el ("LYX-NET"). Är det någon som har tillgång till en FORTH-83, eller vet var Svens arbete tog vägen?
Tacksam för svar här, inom kort.
PS. FORTH är något som angår oss alla!
Magiska hälsningar Magnus

(Text 84) Ulf Dahlen <2422>
Ärende: FORTh med kassett? Jodå, "on it's way"...
Mitt lilla skötebarn UFORTH går hur bra som helst att köra på ett kassettsystem eftersom begreppet "screener" inte finns (ställer annars till problem då det kräver direktåtkomst av enskilda block på "lagringsmediat").
Förhoppningsvis lagom till jul eller nyår ska jag ta mig samman och skicka in eländet så får vi se vad du tycker om den.
Hälsningar Ulf

(Text 86) Gerry Eriksson <4842>
Ärende: CASE definition
Är det någon som vet var man kan få tag på CASE screenen som används i editorn som finns i FORTH-bibl. Den har varit publicerad i ABC-bladet , men saknar det nummret. Finns den på någon kasset eller så ?
Hälsningar Gerry

(Text 87) Magnus Bodin <4304>
Ärende: CASE definition
Nej, inte på någon kassett tyvärr.

```
: CASE ?COMP CSP ´ !CSP 4 ; IMMEDIATE
: OF 4 ?PAIRS COMPILE OVER COMPILE =
     COMPILE OBRANCH HERE 0 , COMPILE
     DROP 5 ; IMMEDIATE
: ENDOF 5 ?PAIRS COMPILE BRANCH HERE 0 ,
        SWAP 2 ÄCOMPILEA THEN 4 ;
        IMMEDIATE
: ENDCASE 4 ?PAIRS COMPILE DROP BEGIN
          SP´ CSP ´ = 0= WHILE 2
          ÄCOMPILEA THEN REPEAT CSP ! ;
          IMMEDIATE
```

Denna rutin hittad i Discover FORTH

(Text 89) Magnus Bodin <4304>
Ärende: CASE, igen...
Ooppps, igen...
Denna gången lite rejälare..
(Det var en stund sedan jag använde ON:)
Snackade helt i nattmössan för en kvart se'n!
Så här används ON som förresten heter ON: (med kolon).
ON: <namn> <ord0> <ord1> <ord2> .. <ordx> ;

Då definieras ordet <namn>.
När man sedan utför <namn> så här:
<n> <namn>
så utförs <ord0> om n=0, <ord1> om n=1 osv.
Så ja, denna gången var det helt rätt, jag kollade just.
Skillnaden mellan CASE och ON: blev nu ytterligare en: ON: används för att definiera ett ord som i sin tur är "vektoriserat". CASE väljer inne i ett ord vad som skall ske. Föredrar CASE själv, faktiskt.
Magiska hälsningar Magnus :-)

(Text 90) Bo Kullmar * <1789>
Fråga: MVPFORTH.DOC's ursprung
Det är Robert Claeson,- Boden. Han sökte och fick projektbidrag för att utveckla en MVPFORHT till ABC80. Själva Forthen var inte i textformat så jag lade inte in den. Sen har inget hänt med programvaran alls. Sven Östberg skulle låna en ABC80 och titta på den, men han gick som bekant bort och sedan dess är det ingen som har intresserat sig för den eller Forth som sådant.

(Text 93) Magnus Bodin <4304>
Ärende: multi-WHILE
Ibland kan det vara praktiskt att använda flera WHILE inom samma 'BEGIN ... REPEAT'-slinga. Om nu inte Din FORTH klarar detta (Den FORTH som klubben distribuerat klarar det INTE.) så kommer här en omdefiniering av REPEAT och WHILE. OBS! REPEAT måste omdefinieras före WHILE, eftersom REPEAT använder sig av WHILE vid omdefinieringen.

```
( MultiWHILE  3 Dec 85 Magnus Bodin )
: REPEAT ÄCOMPILEA AGAIN DUP 4
         ?PAIRS BEGIN DUP 4 =
         WHILE 2 - ÄCOMPILEA ENDIF
         REPEAT ; IMMEDIATE
: WHILE  ÄCOMPILEA IF 2+ 2SWAP ;
         IMMEDIATE
```

Så här använder man flera WHILE:

```
BEGIN
  <huvudföljd>
  <flagga§1 lämnas på stacken>
 WHILE
  <följd§1>
  <flagga§2 lämnas på stacken>
 WHILE
  <följd§2>
  .
  .
  <flagga§x lämnas på stacken>
 WHILE
  <följd§x>
REPEAT
```

Först utförs <huvudföld>.

Om flagga§1 är sann, utförs <följd§1>, annars fortsätter exekvering efter REPEAT.
Om flagga§2 är sann, utförs <följd§2>, annars fortsätter exekvering efter REPEAT.
Om flagga§x är sann, utförs <följd§x>, annars fortsätter exekvering efter REPEAT.
Lycka till, FORTH är något som angår oss alla!
Magiska hälsningar Magnus

(Text 98) Magnus Bodin <4304>
Ärende: DO .. WHILE .. LOOP
WHILE kunde behövas i en loop också, eftersom man ofta gör test i loopar, typ <flagga> IF LEAVE THEN etc.
Med denna omdefinering av LOOP, kan WHILE användas inom DO .. LOOP, och kommer då att låta "LOOP" motsvara "REPEAT", se här:

```
10 0 DO
      ." Tryck <CR>"
      KEY 13 = 0=
     WHILE
      ." ... fel!"
     LOOP
     ." Tio felslag, eller rätt."
```

Ovanstående (om det kompileras i en kolondefinition) kommer att skriva ut texten "Tryck <CR>", testa en tangent, om det är ett <CR> skrivs texten "Tio ." ut, annars skrivs "... fel!" ut. Denna loop genomlöps ånyo om sann flagg lämnats före WHILE, tills loopindexet=10 förstås.
Här kommer omdefinitionen:
(OBS! Omdefinieringen av REPEAT & WHILE i inlägg 93, krävs. Givetvis medför detta att man även kan använda flera WHILE inom samma DO .. LOOP, snurrigt?)

```
( DO - WHILE - LOOP          4 Dec 85 )
 : (LOOP2)  R> R> R> 2DROP DROP ;
 : LOOP     ÄCOMPILEA LOOP
            BEGIN DUP 4 =
            WHILE 2 - ÄCOMPILEA ENDIF
            REPEAT COMPILE (LOOP2) ;
            IMMEDIATE
```

Magiska hälsningar Magnus

(Text 107) Magnus Bodin <4304>
Ärende: multiWHILE
Nu kan man ju snabbt och enkelt definiera två ord till, nu när loop är omdefinierad.
IFLOOP & LOOPUNTIL
IFLOOP används sålunda:

```
10 0 DO
      <följd>
      .
      .
      <flagga lämnas på stacken>
     IFLOOP
```

Om flaggan=true, och loopindexet<10 så genomlöps loopen igen.
LOOPUNTIL gör motsatsen:

```
10 0 DO
      <följd>
      .
      <flagga lämnas på stacken>
     LOOPUNTIL
```

Om flaggan=false, och loopindex<10 så genomlöps loopen igen.
Simpla ord, som alltså kräver att WHILE och LOOP omdefinierats, enligt tidigare inlägg:

```
: IFLOOP  ÄCOMPILEA WHILE
          ÄCOMPILEA LOOP ; IMMEDIATE
: LOOPUNTIL COMPILE 0=
            ÄCOMPILEA IFLOOP ;
            IMMEDIATE
```

(OBS. FIG-FORTH)
Magiska hälsningar Magnus :-)

(Text 122) Magnus Bodin <4304>
Ärende: FORTH vs ALGOL-språk
Man kan se skillnaden i den här liknelsen:
Om vi ser språken som restauranter så kan följande skillnader visas:
När vi befinner oss i Pascal-restauranten, behandlas vi som en gäst. Det finns ett antal reserverade (b)ord som vi ABSOLUT INTE får sätta oss vid eller ändra på. Vi får däremot lägga till nya bord, göra långbord, om vi skulle ha med oss ett stort sällskap (program). Menyn (kommandolistan) är bestämd av köksmästare och tillika restaurantchefen, Niklaus Wirth. Det finns givetvis a la carte, där man kan välja egna rätter (procedurer), men dock inte hela menyer (kontrollstrukturer, typ WHILE, REPEAT,FOR, osv)...
I FORTH-restauranten är vi själva restaruantchefer. Bjuder vi in ett stort sällskap så kan vi givetvis anpassa de reserverade (b)orden efter detta. Skulle vi vilja komponera någon rätt med nytt recept, så att förändringar måste ske under själva tillagningen (kompileringen) kan vi givetvis göra detta. Här har vi ingen köksmästare som styr oss. Charles Moore, mannen som grundade restaurant-kedjan lämnade bara en grund att stå på. Om sedan restauranten skall kunna servera fisk, skaldjur, sniglar (flyttal, strängar etc.) skall kunnas läggas till samt dras ifrån allt efter restaurantchefens behov. Om restaurantchefen tycker att han vill utvidga sin restaurant, så att den kan ta emot flera sällskap på en gång, även fast Moore inte tänkte på det i början, (multitasking) så kan han givetvis göra detta. Vad lär vi oss av detta?
Jo att FORTH-restauranterna borde vara mer utbredda än Pascal-dito. Varför äter inte en vanlig programmerare på FORTH-kedjans restauranger vanligtvis?

* FORTH-kedjans restauranger verkar ha trögare entredörrar, och lata programmerare stannar ofta i vestibulen eftersom entredörren tröttar ut dem.
* Man måste tänka på ett annat sätt när man äter hos FORTH's restauranter, (postfix, omvänd polsk not. (RPN) detta skrämmer flera programmerare. - Va? Skall jag behöva tänka annorlunda för att kunna äta?
* Inte tillräckligt många FORTH-gourmeer har funnits som kan visa upp vad FORTH

Så: egentligen har att erbjuda.
Nästa gång du är hungrig, testa en FORTH-restaurant i stället för en Pascal-dito!
En nöjd köksmästare

(Text 123) Magnus Bodin <4304>
Ärende: Mikrodatorn 5/85 s.53
Citerar här en artikel av Lars-Erik Svahn, för att ytterligare klargöra vad FORTH är för något och varför man bör ta sig en extra titt på språket.
"Forth är komplett som system. Det ger full kontroll över datorn och dess perifierienheter, samtidigt som det rymmer alla de möjligheter till strukturerad programmering som krävs av ett modern högnivåspråk."
"FORTH är generellt. Det som gör FORTH så generellt, är dess många och lättanvända utbyggnadsmekanismer."
"Han kan t o m ändra syntaxen i redan färdiga konstruktioner"
"Som framgått är FORTH ett kompilerande språk, men också ett tolkande."
"Även kompileringsprocessen är dialogbetonad och helt avdramatiserad."
" ... kompilerad Pascalkod är åtminstone dubbelt så långsam."
"Har då medaljen ingen baksida?"
"Sant är att FORTH kan vara motsträvigt i början."
"Efter en rimlig tids träning inser man att 'allt' kan göras i FORTH, och i regel på ett elegant sätt."
"I datormiljö är det tvärtom postfix notation som är enkel och naturlig."
"Dessutom är postfix notation smart, man behöver t ex aldrig använda några parenteser."
"Den som vill lära sig FORTH skall absolut läsa de böcker som Leo Brodie på FORTH Inc. har skrivit:
'Starting FORTH' och 'Thinking FORTH'."
"Programspråket FORTH tillhör inte något av de enklare språken att lära sig, men det är resurssnålt och har den fördelen att när man väl kan det så går det mycket snabbt att programmera med FORTH."
"FORTH är framtidens språk, så GO FORTH!"
En nöjd köksmästare lagade ännu en måltid..

En billig talsyntes

Jag tänker berätta lite hur man i ABC80 kan digitalisera ljud med en mycket enkel och billig hårdvara. Återgivning av ljudet kan sedan ske helt utan extra hårdvara med denna metod.

Bakgrund

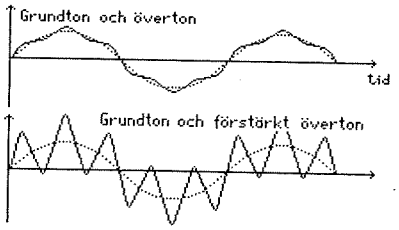
I begynnelsen upptäcktes att om man spelar en vanlig ljudkassett i ABC80s bandspelare så hördes ljudet på bandet, visserligen med stark distorition, men hörbart. Då det är möjligt att från programvara känna av kassettporten samt styra högtalaren så infann sig tanken att lagra ljudet digitalt och efteråt spela upp det. Försök visade att det inte gick så bra. På kassettporten kan programmet bara känna om signalen skiftar från plus till minus eller tvärtom (flankdetektering). Polariteten, om signalen ligger på plus eller minus, kan inte kännas av. V24 porten däremot kan känna polariteten.

I början av projektet tänkte jag att det skulle krävas lite hårdvara både för att digitalisera ljudet och för att bilda en analog signal när ljudet återges, men när jag läst en artikel i BYTE (februari 1981 sid 36-43) så tänkte jag att det borde vara möjligt att återge ljudet i ABC80s inbyggda högtalare. Det visade sig vara riktigt, hårdvara behövs bara vid 'inspelningen'. Att åtege ljudet utan extra hårdvara ger möjlighet att distribuera ljudet till alla som har en ABC80. (t ex RULLE.BAS på kassett §17).

Digitalisera ljudet

Ett ljud (t ex ett vokal-ljud eller ett instrument) kan beskrivas som ett antal överlagrade sinusvågor (grundton och övertoner). För att tal ska kunna återges så måste både grundtoner och övertoner i ljudet återskapas, annars blir inte talet förståeligt.

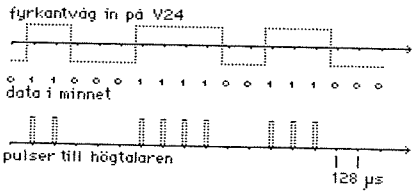
För att digitaliserar ljudet så samplar man (känner av) ljudet, i det här fallet, 7812 gånger per sekund. Vid varje sampel lagras bara en bit information; 1 om signalen är positiv och 0 om den är negativ. Den här metoden gör i praktiken fyrkantvåg av ljudet. Grundtonen i ljudet ger den huvudsakliga formen på fyrkantvågen och information om övertonerna försvinner. Detta åtgärdas genom att förstärka övertonerna mer än grundtonen och låta övertonerna göra att signalen växlar polaritet flera gånger under en period. Ett filter får derivera (eller differentiera) den ursprungliga signalen.



Återge ljudet

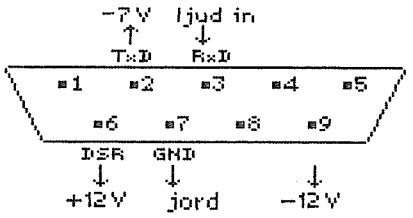
När ljudet ska återges är det viktigt att bitarna som representerar ljudet matas ut i precis samma takt som de samplats, även nu 7812 gånger/s. Om man låter en etta ge en fast positiv spänning och noll en negativ så borde utsignalen, som nu liknar fyrkantvåg, integreras i ett filter innan den går till högtalaren (för att kompensera deriveringen vid digitaliseringen). Detta är dock inte nödvändigt för att förstå talet och vi kommer inte heller göra det. På grund av egenheter i ABC80s ljudkrets så kan man inte mata ut den fyrkantvåg som bitarna representerar utan ljudet måste huggas upp i korta pulser, egentligen en puls (eller spik) per bit som är ettställd. I nyare ABC80 med högtalaren i separat nätdel blir ljudet ganska svagt och dåligt, ev. kan det förbättras med utmatning av längre pulser.

Anledningen till att jag valt samplingsfrekvensen 7812 är att PIO:n kan programmeras att ge interupt så ofta, det ger en tillräckligt exakt tidbas.



Hårdvara

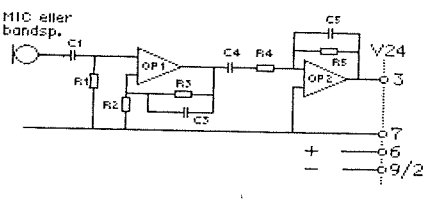
Filtret som används vid 'inspelning' består av två Op-förstärkare och några motstånd och kondensatorer. Op-förtärkarna tar matningsspänning från V24 kontakten pinne 6 och 6 (eller 9) och den förstärkta och filtrerade signalen går in på V24 kontakten pinne 3. OBS! För att matningspänningen ska bli rätt måste man först göra OUT 58,INP(58) OR 8 (sätta pinne 2 låg=-7V) innan man kopplar in förstärkaren/filtret. V24an ger med denna koppling -7V och +12V, inte direkt balanserat men det gör nog inget. Alternativt kan minusspänning tas från pinne 9 (-12V) vilket borde minst lika bra (men jag har ingen V24 sladd med pinne 9 ansluten).



Förutom att derivera signalen så måste filtret ta bort frekvenskomponenter över ca 3500 Hz annars kommer höga frekvenser att återges felaktigt. De skulle samplas mindre än 2 gånger per period och data kan misstolkas som en annan, lägre, frekvens.

Kondensatorn före första steget fungerar både som isolator för likström och som filter för lågfrekvent brum.

Förstärkningen i filtret kan få vara så stor att förstärkarna bottnar. Efter förstärkaren/filtret är vi bara intresserade av om signalen är positiv eller negativ, inte amplituden på signalen. ABC80 tillåter att signaler till V24an är +/- 30V (volt). En signal som är över 3V registreras som en nolla (0) och signaler under 0V registreras som etta (1). Mellan 0V och 3V är hysteres, dvs. en nolla eller etta ligger kvar tills signalen går över endera gränsen. V24an blir alltså en nollgenomgångsdetektor och hysteresen gör att bakgrundsbrus inte behöver höras, små variationer kring 0V ger inget utslag.



R1=150 kohm
R2=1.2 kohm
R3=56 kohm
R4=1.8 kohm
R5=39 kohm
C3=1.0 nF
C4=47 nF
C5=1.0 nF

LJUD2.TXT

```
;--------------------------------
;86-02-11
;DAVID ANDERSSON <5201>
;ASSEMBLER RUTINER FÖR ATT SPELA IN
;OCH ÅTERGE LJUD.
;--------------------------------
PRGADR: EQU 61440      ;PROGRAMMET PLACERAS UNDER STACKEN -
                       ;DET MÅSTE VARA PÅ EN JÄMN PAGE-ADRESS
HI:     EQU .HIGH.PRGADR ;HÖGA BYTEN AV ADRESSEN
SADDR:  EQU 51200      ;STARTADRESS FÖR LJUD-DATA (16K RAM)
EADDR:  EQU PRGADR-1   ;SISTA ADRESS FÖR LJUD-DATA
;--------------------------------
; PROGRAMMET BÖRJAR HÄR
        ORG PRGADR
        JP START       ;CALL(PRGADR)   SPELA IN LJUD
        JP PLAY        ;CALL(PRGADR+3) SPELA UP LJUD
; VARIABLER
BYTE:   DEFB 0         ;SHIFTA IN/UT EN BYTE
COUNT:  DEFB 8         ;RÄKNAR 8 BITAR I EN BYTE
ADDR:   DEFW 0         ;ADRESS TILL LJUD-DATA
; INTERUPT VEKTORER
        ORG PRGADR+52  ;NORMAL I-VEKTOR I PIO:N
SINT:   DEFW SINT1     ;STROBE INTERUPT (SÄTTS VID INITIERING)
        DEFW 1428      ;CASSETT INTERUPT
;--------------------------------
; RUTINER SOM ANROPAS FRÅN BASIC
; SPELA IN LJUD
; ANROPAS FRÅN BASIC, AVBRYTS MED <RETURN>
; <MELLANSLAG> ÅTERSTARTAR INSPELNING
START:  LD A,8
        LD (COUNT),A   ;INITIERA VARIABEL
        LD HL,EADDR
        LD (ADDR),HL   ;SÄTT SLUTADRESS FÖR INSPELNING -
                       ;INSPELNING SKER DÅ INTE FÖRENS -
                       ;<SPACE> TRYCKS NER
        LD HL,SINT1    ;INTERUPT FÖR INSPELNING
        LD (SINT),HL   ;SÄTT INTERUPTRUTIN 1
        CALL INITI     ;INITIERA PIO-INTERUPT
RECORD: IN A,(56)      ;TESTA TANGENTBORDET
        CP 13+128      ;<RETURN>
        JR Z,EXITI     ;TERMINERA INTERUPT OM RETURN
        CP 32+128      ;<SPACE>
        JR NZ,RECORD   ;SPELA IN LJUD OM EJ SPACE, LOOPA
        LD HL,SADDR    ;SPACE ÄR INTRYCKT -
        LD (ADDR),HL   ;SÄTT STARTADRESS FÖR DATA
        JR RECORD      ;LOOPA
; SPELA UPP LJUD
; ANROPAS FRÅN BASIC, AVBRYTS MED <RETURN>
PLAY:   LD A,1
        LD (COUNT),A   ;INITIERA VARIABEL
        LD HL,SADDR
        LD (ADDR),HL   ;SÄTT STARTADRESS FÖR DATA
        LD HL,SINT2    ;INTERUPT FÖR ÅTEGIVNING
        LD (SINT),HL   ;SÄTT INTERUPTRUTIN 2
        CALL INITI     ;INITIERA PIO-INTERUPT
PLAY1:  LD HL,(ADDR)   ;TESTA OM EFTER SLUTADRESS
        LD A,H
        CP HI
        JP NC,EXITI    ;OM SÅ, TERMINERA INTERUPT
        IN A,(56)      ;TESTA TANGENTBORDET
        CP 13+128      ;<RETURN>
        JR NZ,PLAY1    ;LOOPA OM INTE RETURN
        JP EXITI       ;TERMINERA INTERUPT

;--------------------------------
; INITIERINGS RUTINER
; INITIERA STROB INTERUPT
INITI:  LD A,I         ;TESTA OM I-REGISTRET REDAN ÄR SPARAD
        CP HI
        JR Z,INITI3
        LD (SAVEI),A   ;OM INTE, SPARA SIST I LISTAN
INITI3: LD HL,ILIST    ;PIO DATA FÖR STROBE INTERUPT
        JR SETPIO
; TERMINERA STROB INTERUPT
EXITI:  LD HL,ULIST    ;PIO DATA FÖR NORMAL INTERUPT
    ;JR SETPIO
; PROGRAMERA PIO OCH ENABLE INTERUPT
; HL SKA PEKA PÅ LISTA ,5 BYTES LÅNG
SETPIO: PUSH BC
        LD BC,4*256+57;B=4 BYTES, C=PORT 57
        DI
        OTIR
        LD A,(HL)      ;A=SISTA BYTEN I LISTAN -
        LD I,A         ;TILL I-REGISTRET
        EI
        POP BC
        RET
; DATA FÖR PROGRAMERING AV PIO PORT A (KEYBOARD)
ILIST:  DEFB 04FH      ;MODE 1 (ALL INPUTS)
        DEFB 087H      ;ENABLE INTERUPT
        DEFB 087H      ; -"-
        DEFB 087H      ; -"-
        DEFB HI        ;HIGH VEKTOR
ULIST:  DEFB 0CFH      ;MODE 3 (CONTROL)
        DEFB 0FFH      ;DIRECTION ALL INPUT
        DEFB 0B7H      ;ENABLE,OR_BITS,GO_HIGH, MASK_FOLLOWS
        DEFB 07FH      ;BIT 7 IS MASKED
SAVEI:  DEFB 0         ;OLD HIGH VEKTOR (SÄTTS VID INITIERING)
;--------------------------------
; INTERUPT RUTINER
; STROB INTERUPT, 7812 PERIODER/S
; INTERUPT RUTIN FÖR INSPELNING AV LJUD
SINT1:  PUSH AF
        PUSH HL
        IN A,(58)      ;KÄNN AV V24-PORTEN
        AND 1          ;TESTA OM LJUDSIGNALEN, CLEAR CARRY
        JR NZ,NOOUT    ;HOPPA OM 1 (AKTIV LÅG)
        XOR A          ;GE PULS I HÖGTALAREN
        OUT (6),A
        LD A,121
        OUT (6),A
        SCF            ;SÄTT CARRY CARRY ÄR SAMPLAD BIT
NOOUT:  LD HL,BYTE
        RL (HL)        ;CARRY TILL BIT 0 AV (BYTE)
        LD A,(HL)      ;A=BYTE
        INC HL         ;HL=ADRESS TILL COUNT
        DEC (HL)       ;RÄKNA NER COUNT
        JR NZ,NONEX2   ;HOPPA OM COUNT>0
        LD (HL),8      ;COUNT=8, FÖR NÄSTA BYTE
        LD HL,(ADDR)   ;ADRESS TILL LJUD-DATA
        LD (HL),A      ;SPARA BYTE I LJUD-DATA
        JP NXTADR      ;SÄTT NY ADRESS, RETURN
                       ;FROM INTERUPT
; INTERUPT RUTIN FÖR ÅTERGIVNING AV LJUD
SINT2:  PUSH AF
        PUSH HL
        LD HL,BYTE
        RL (HL)        ;BIT 7 AV (BYTE) TILL CARRY
        JR NC,NOOUT2   ;HOPPA OM BIT=0
        XOR A          ;BIT=1, GE PULS I HÖGTALAREN
        OUT (6),A
        LD A,121
        OUT (6),A
NOOUT2: INC HL         ;HL=ADRESS TILL COUNT
        DEC (HL)       ;RÄKNA NER COUNT
        JR NZ,NONEX2   ;HOPPA OM COUNT>0
        LD (HL),8      ;COUNT=8 FÖR NÄSTA BYTE
        LD HL,(ADDR)   ;ADRESS TILL LJUD-DATA
        LD A,(HL)      ;A=LJUD-DATA BYTE
        LD (BYTE),A    ;SPARA BYTE FÖR ATT SHIFTA UT
NXTADR: INC HL         ;NÄSTA BYTE
        LD A,H         ;TESTA OM EFTER SISTA ADRESS
        CP HI
        JR NC,NONEX2   ;JA, SPARA INTE NY ADRESS
        LD (ADDR),HL   ;SPARA NÄSTA ADRESS
NONEX2: POP HL         ;RETURN FROM INTERUPT
        POP AF
        EI
        RETI
;--------------------------------
        END
```

Vad Op-förstärkarna heter vet jag inte men användningen är inte krävande och det går nog lika bra med selleri (alltså vanliga 741:or). Uppkopplingen har jag gjort på några IC-hållare och virat trådar mellan stiften.

Ungefärlig totala förstärkningen ges av R3/R2*R5/R4. Deriveringen sker i C4 och R4 (egentligen ett högpassfilter för 1900 Hz). C3,R3 samt C5,R5 står för lågpass-filtren (2800 resp. 4000 Hz). Experimentera gärna med andra värden på komponenterna för att testa om ljudet kan bli bättre.

Programmet

När en bit ska lagras känner programmet av signalen på port 58 bit 0. Då 7812 bitar för varje sekunden ljud ska sparas så måste minnet utnytjas effektivt genom att lagra 8 bitar i varje byte. Ändå så kommer en sekund att ta cirka 1 Kbyte i minnet. Programmen utnytjar 10K minne som räcker till 10 sekunder ljud.

Programmet fungerar så att PIO:n initieras att ge interupt med 128 us intervall (7812 per sekund) och en av två interuptrutiner aktiveras. Interuptrutinen för inspelning, SINT1, sparar en bit i minnet och ekar ljudet i högtalaren. Interuptrutinen för återgivning, SINT2, hämtar bitar från minnet och skickar bitar till högtalaren.

Med BASIC programmet som listas sist behöver man inte skriva ;CALL(adrs) för hand medans man testar och spelar in olika ljud. Bakåtpil börjar digitalisera ljud och mellanslag startar inspelning av ljudet. Framåtpil spelar upp ljudet, och RETURN avbryter allting. Med knappen P visas minnet bit för bit och med R får man tillfälle att räkna ut gränsfrekvenser för RC-filter.

```
 1 POKE 61440%,195%,56%,240%,195%,94%,24
   0%
 2 POKE 61492%,172%,240%,148%,5%,62%,8%,
   50%,7%,240%,33%,255%,239%,34%,8%,240%
   ,33%
 3 POKE 61508%,172%,240%,34%,52%,240%,20
   5%,132%,240%,219%,56%,254%,141%,40%,6
   4%,254%,160%
 4 POKE 61524%,32%,246%,33%,0%,200%,34%,
   8%,240%,24%,238%,62%,1%,50%,7%,240%,3
   3%
 5 POKE 61540%,0%,200%,34%,8%,240%,33%,2
   07%,240%,34%,52%,240%,205%,132%,240%,
   42%,8%
 6 POKE 61556%,240%,124%,254%,240%,210%,
   146%,240%,219%,56%,254%,141%,32%,241%
   ,195%,146%,240%
 7 POKE 61572%,237%,87%,254%,240%,40%,3%
   ,50%,171%,240%,33%,162%,240%,24%,3%,3
   3%,167%
 8 POKE 61588%,240%,197%,1%,57%,4%,243%,
   237%,179%,126%,237%,71%,251%,193%,201
   %,79%,135%
 9 POKE 61604%,135%,135%,240%,207%,255%,
   183%,127%,0%,245%,229%,219%,58%,230%,
   1%,32%,8%
10 POKE 61620%,175%,211%,6%,62%,121%,211
   %,6%,55%,33%,6%,240%,203%,22%,126%,35
   %,53%
11 POKE 61636%,32%,47%,54%,8%,42%,8%,240
   %,119%,195%,236%,240%,245%,229%,33%,6
   %,240%
12 POKE 61652%,203%,22%,48%,7%,175%,211%
   ,6%,62%,121%,211%,6%,35%,53%,32%,18%,
   54%
13 POKE 61668%,8%,42%,8%,240%,126%,50%,6
   %,240%,35%,124%,254%,240%,48%,3%,34%,
   8%
14 POKE 61684%,240%,225%,241%,251%,237%,
   77%

20 REM ------------------------------
30 REM 86-02-24
40 REM DAVID ANDERSSON <5201>
50 REM
60 REM PROGRAMMET ANROPAR MASKINKODS-
70 REM RUTINER FÖR ATT SPELA IN OCH
80 REM ÅTERGE LJUD.
90 REM ------------------------------
100 REM
110 A1%=51200% : REM ADRESS FÖR LJUD-DA
    TA
120 OUT 58%,INP(58%) OR 8% : REM PINNE
    2=-7V
130 ;
140 ; 'Spela in med BAKÅTPIL + MELLANSL
    AG'
150 ; 'Återge med FRAMÅTPIL'
160 ; 'Titta på lagrade bitar med P'
170 ; 'Beräkna filterfrekvenser med R'
180 ; 'Avbryt funktionerna med RETURN'
190 REM MAIN LOOP
200 GET A$
210 IF ASC(A$)=8% GOSUB 1000
220 IF ASC(A$)=9% GOSUB 2000
230 IF A$='P' OR A$='p' GOSUB 3000
240 IF A$='R' OR A$='r' GOSUB 4000
250 GOTO 190
1000 REM SPELA IN
1010 ; '<-- In';
1020 Z%=CALL(61440%)
1030 ; CHR$(13%)SPACE$(10%)CHR$(13%);
1040 RETURN
2000 REM ÅTERGE
2010 ; '--> Ut';
2020 Z%=CALL(61443%)
2030 ; CHR$(13%)SPACE$(10%)CHR$(13%);
2040 RETURN
3000 REM  BITPEEK
3020 FOR A%=A1% TO 61439%
3030 FOR I%=7% TO 0% STEP -1%
3040 IF PEEK(A%) AND 2%^I% ; '1'; ELSE ;
     '0';
3050 NEXT I%
3060 IF INP(56%)<128% GET A$
3070 IF A$<>CHR$(13%) NEXT A%
3080 ;
3090 RETURN
4000 REM BERÄKNA RCNÄT
4010 ONERRORGOTO 4080
4020 ; 'C (nF)    ='; : INPUT C
4030 ; 'R (kohm) ='; : INPUT R
4040 C=C*1E-9
4050 R=R*1000
4060 ; 'f (Hz)    =';1/R/C/2/PI : ;
4070 GOTO 4010
4080 RETURN
```