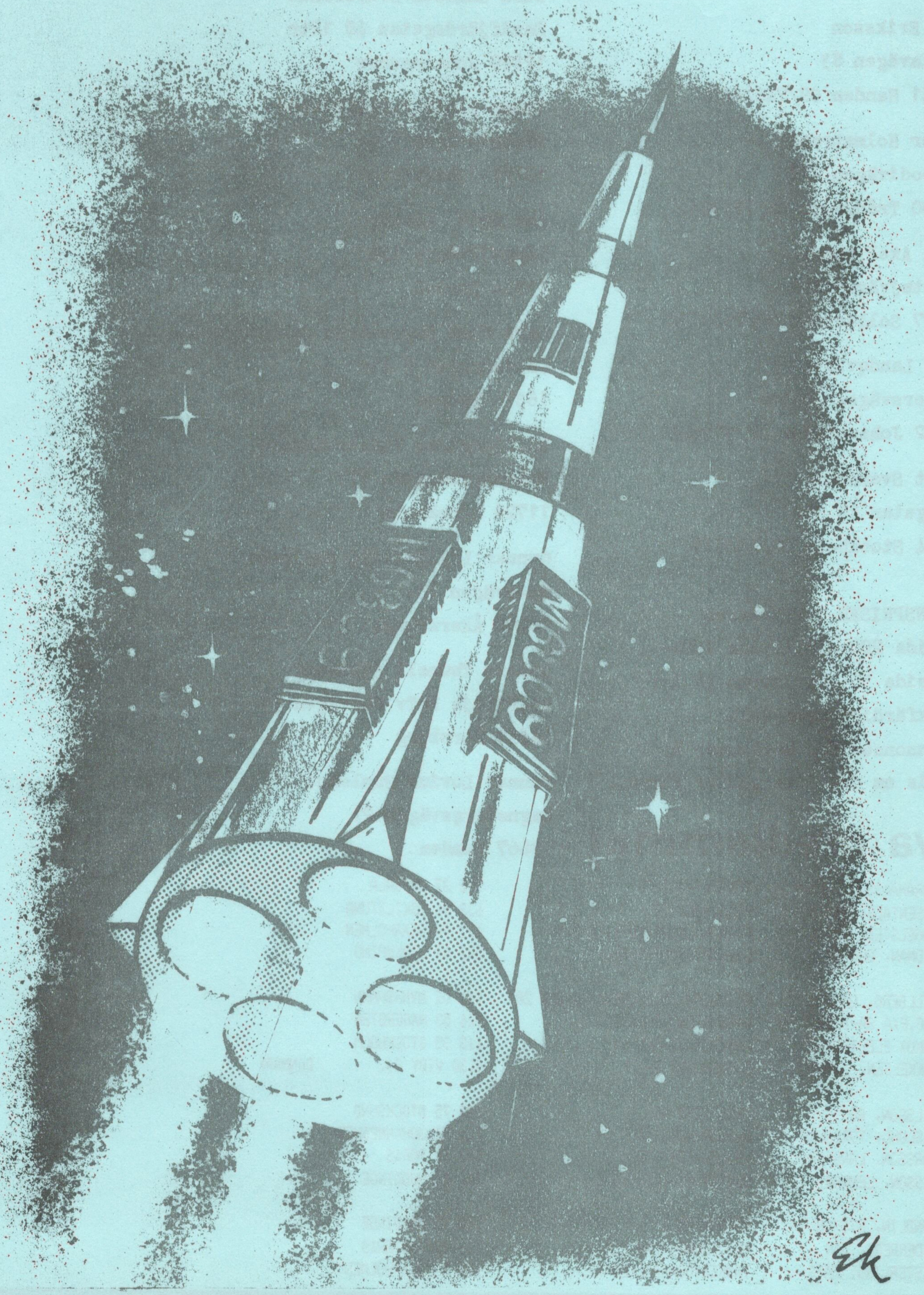
MPU-laren nr 2·1979

Tidskrift för Privatdataklubben PD68 · Årgång 2



Redaktionen

MPU-larenredaktionen består av
följande medlemmar.

Alf Eriksson
Vallavägen 83
13641 Handen 08/7452667

Roger Holmstrand
Rapsodivägen 128
14200 Trångsund 08/7713852

Carl Axel Bruno
Timotejvägen 116
19177 Sollentuna 08/7543788

Lars Landqvist
Skagersvägen 20 1tr
12142 Johanneshov 08/912625

Bernt Svensson
Ploggatan 16 3tr
11634 Stockholm 08/403517

ANNONSPRISER I MPU-laren
Helsida 200:- halvsida 100:-
kvarsida 50:- priserna gäller
tryckfärdigt material.
Radannonser för medlemmar är
gratis om det inte gäller firma..

Nya medlemmar

Styrelsen

FD68 nuvarande styrelse

Nils Ohlsson Ordförande
Sandfjärdsgatan 68 12tr
12169 Johanneshov

Tommy Bladh V.Ordförande
Båstadsvägen 7
12151 Johanneshov

Åke Holm Kassörr
Långsjövägen 15B
13554 Tyresö

Lars Olof Fagerström Sekreterare
Gärdebyplan 6 4tr
16374 Spånga

Bo Ljungblad V.Sekreterare
Brännkyrkagatan 117
11728 Stockholm

Gunnar Lilliesköld Supleant
Mossvägen 1C
18400 Åkersberga

Anders Thunell Supleant
co,Fittja R&Tv Forvägen 15
14551 Norsborg

Gunnar Lovius Supleant
Ragnebergsvägen 21
13667 Handen

Vi växer och måste flytta till

Strandbergsgatan 47
i Hornsberg Stockholm

Vårt nya telefonnummer
blir f.o.m. 19 februari

08/13 21 60

Vår postadress förblir den samma
d.v.s. Box 32 122 21 Enskede

Lagret blir kvar på den gamla adressen, Sandsborgsvägen 55
i Enskede.



DISTRIBUTÖREN

08/13 21 60

INTERELKO AB
Box 32
122 21 Enskede

ORDFÖRANDE HAR ORDET

Klubbens första årsmöte har hållits.Vad som beslutades på mötet finns beskrivet på annan plats i tidningen, så om det skall jag inte "orda" så mycket. Tänkvärt är dock att man tydligen inte är intresserad av samordnade aktiviteter inom klubben. Jag tänker då exempelvis på att inte någon har anmält sig till projektet "Mikrodator på Tekniska museet". Här fanns möjligheten att med både kapital och utrymme inför offentligheten demonstrera de sidor av livet som kan påverkas av den snabba utvecklingen inom elektroniken. Dessa tankar är dock inte riktade som kritik mot medlemmarna. Redan innan diskussionen på årsmötet hade jag mina aningar om att den hobby vi sysslar med kräver så mycket tid att det blir föga eller ingen tid över för andra projekt än de egna. Det som dock skall noteras är att vi inom klubben inte skall satsa på denna typ av samordnad verksamhet, utan arbeta vidare med tidning, utbyte av erfarenheter inom mjuk- och hårdvara och våra månatliga möten. En breddning inom dessa områden torde vara av godo för klubben. Den eller de som inte samtycker med detta bör ta kontakt med styrelsen för att isåfall undersöka möjligheterna till en kursändring i dessa frågor.

I ett cirkulärbrev har SEMKO (Svenska Elektriska Materielkontrollanstalten) aktualiserat kravet på s-märkning av datorutrustning använda i hemmet. De regler som tillämpas är synnerligen stränga, men får för den skull inte uppfattas som överambitiösa. Vi har i Sverige en mycket låg frekvens av olyckor i samband med utnyttjandet av elektrisitet. Detta får till stor del tillskrivas SEMKO som övervakar genom tester och stickprovskontroller. Läs därför det på sidan publicerade brevet och beställ gärna informationsbroschyren SEMKO-900.

N.O.

P S Räkna med en kostnad av
ca 15 kronor.

Information om provningsplikt för hemdator-utrustningar och -terminaler

Då det synes råda en viss oklarhet beträffande omfattningen av kontrollplikten avseende rubr materiel, meddelas att nätanslutna datorutrustningar och terminaler för användning i hem är provningspliktiga fr o m 1979-01-01 enligt Statens Industriverks kungörelser om kontroll av viss teleteknisk och elektronisk materiel, SIND-FS 1977:2, § 1, pkt 9, och därför måste vara typprovade och godkända av SEMKO (S-märkta) för att få säljas och användas i Sverige.

Undantag: Apparater för högst 24V spänning och högst 15W effektförbrukning, som ansluts till starkströmsnätet över godkänd separat skyddstransformator eller annat godkänt separat strömförsörjningsdon är dock, enligt § 2, pkt 4 i kungörelsen, undantagna från provningstvånget.

Godkännande från SEMKO erfordras dessutom ej, enligt § 2, pkt 6, för sådan apparat som framställts genom att ägaren själv monterat en större del av de ingående komponenterna /självbygge/, under förutsättning att:

a) apparaten uppfyller gällande SEMKO-bestämmelser

b) ingående komponenter är godkända i den utsträckning provningsplikt föreligger

c) apparaten förblir i självbyggarens besittning

Samtidigt meddelas att skriften SEMKO 900, Råd för självbyggare av elektronikapparater - Elsäkerhetskrav, för närvarande är under revision. En ny upplaga beräknas utkomma inom kort. Rekvisition kan ske redan nu hos SEMKOs kassaavdelning, tfn 08/ 13 07 00.

Med vänlig hälsning

SVENSKA ELEKTRISKA
MATERIELKONTROLLANSTALTEN AB

Vårt stora problem i redaktionen är att vi får så litet bidrag från övriga medlemmar i klubben. Därför uppmanar jag alla medlemmar att sända oss bidrag till tidningen i form av mjukvara,hårdvara,nya ideer eller problem av intresse för andra.
A.E..

Nästa nummer

Fullständigt medlemsregister.
Ett program för sändning och mottagning av morse via tangentbord och som klarar från ½ till
250 ord i minuten.
"Födelsedagsprogram" i Basic
som talar om veckodag m m.
En utvidgad Mikbug.

Verksamhetsberättelse 1978 för PD68

Klubben bildades genom att 12 dataintresserade träffades en kväll för att diskutera eventuellt bildande av en dataklubb. I och med detta möte bildades PD68 från och med årsskiftet 77/78. Styrelsen har under året haft 11 sammanträden för att utforma klubbens ideela aktiviteter samt utarbeta rutiner för att administrera verksamheten. Styrelsen har dessutom besvarat en mängd tekniska problem från medlemmar. Efter en inbjudan från Elemtel besökte styrelsen företaget, vilket var mycket intressant och givande. Under året har MPU-laren, vår klubbtidning, utgivits med 3 nummer. God hjälp har erhållits från annonsörer och medlemmarna har bidragit med intressant matrial. Klubbmöten har hållits varje månad med undantag för juni,juli och december. Aktiviteter på mötena har varit föredrag av medlemmar, företag inom data och komponentbranchen samt många demonstrationer av datautrustningar både av proffs och hembyggda. Klubben deltog på IM Älvsjö med en dator i Interelko´s monter. Denna skötte klubbens reklam under hela mässan. Klubbens service mot medlemmarna har varit att gratis dela ut kassettband med program införda i MPU-laren efter beställning. Dessutom har klubben sammanställt en samlingsbok över 6800 familjens kretsar för Interelkos räkning. Som ersättning för detta erhöll samtliga medlemmar utan kostnad ett exemplar av detsamma. Medlemmsantalet var vid årsskiftet cirka 120 själar. Medlemmar finns förutom i Sverige även i Frankrike och Danmark. En bra start för klubbens första år med en önskan om en lika bra fortsättning. Åsikter och ideer till fortsatt förbättring önskas i form av brev eller samtal till klubben.
Med önskan om ett nytt gott privatdataår.

PRIVATDATAKLUBBEN PD68 SALDO 1977-78

Ingående balans 0,-

Intäkter		Utgifter	
Medlemsavgifter	9.019,-	Lokalhyra	200,-
Annonser	850,-	Frimärken	964,30
		Tryckkostnader	2.336,55
		Övrigt	574,15
	9.869,-		4.075,-

Utgående balans 78-12-31

Tillgångar postgiro	5.794,-	Skulder	0,-

Indexregister

INDEXREGISTER MPU-laren 1978.

Programvara.

Gubben Noak	nr 1
Månlandning	" 1
Master Mind	" 1
Personnummer	" 1
Telefonregister	" 2
Speed	" 2
Input	" 2
THMPLSTK V2.0	" 2
MACSS	" 3
Textrutin för D2	" 3
Schack	" 3

Byggen.

Videomodulator	" 1
Hardware X-reg stack	" 3
Videoanpassning	" 3

Diverse..

Felorsaker vid användande av mjukvara	" 1
Provbygge epromprog EPB2	" 1
Enkäten	" 2
Hoppsan	" 2
Modem	" 2
Medlemsreg	2,3
Lusen	2,3

ORDLISTAN

Assign	Tilldela
Clear	Nollställa
Debug	Testa,avlusa
Entry	Ingång
Integer	Heltal
Label	Etikett
Pointer	Visare
String	Sträng

VIDEOKORT FÖR CRTKRETS ESM364

Färg/Svart-vit/ VIDEOTERMINAL

64 tecken per rad, 16 rader,
6 färger, DATORSTYRD CURSOR
scrolling.
Videoutgång vid svart-vit,
RGB vid färg, RS 232 in-ut
samt ASCII parallell för
tangentbord.

KRETSKORT OCH SCHEMA:

Pris: 240:—

Tel: 08/842984 eft. 18.00

SPELHÅLAN

Här följer ännu ett bidrag till vår flora av spel i Basic. Vem författaren RJC är vet vi inte. Programmet kommer från en dataanläggning hos ett av våra största företag. Förmodligen har det komponerats av en operatör borta i USA för 10 år sedan.

Koden har fixats till så att den går att köra på vår vanliga 8k Basic. Som synes innehåller koden regler för att kunna spela. Runt ett hål "O" står fyra st pjäser på vardera sidan. Det gäller att flytta alla "x" till vänster sida, och alla "*" till högra sidan. Man får hoppa till hålet eller hoppa över en pjäs. Endast frammåt hopp är tillåtna.

TBL

forts.nästa sida

```
0003 REM ARKABLE PROGRAM WRITTEN BY RJC 4-10-69
0005 REM VERSION 3 4/18/69
0006 REM FIXAT TILL M6800 AV TBL  1978-10-27
0008 DATA *,*,*,*,O,X,X,X,X
0009 DATA X,X,X,X,O,*,*,*,*
0010 DIM Z$(9),Y$(9)
0014 PRINT
0015 PRINT
0016 PRINT
0017 PRINT "* * *   E I G H T  -  M E N   * * *"
0018 PRINT
0020 PRINT "DO YOU KNOW HOW TO PLAY";
0030 GOSUB 680
0040 ON F1 GO TO 240,70,50
0050 PRINT "ANSWER EITHER YES OR NO"
0060 GOTO 30
0070 PRINT " ","POSITION NUMBERING-"
0080 PRINT " ","123456789"
0090 PRINT " ","****OXXXX"
0100 PRINT
0110 PRINT " ","OBJECT-"
0120 PRINT "TO REVERSE THE POSITION OF THE *▫S AND THE X▫S."
0130 PRINT "THE O IS AN EMPTY SPACE."
0140 PRINT
0150 PRINT " ","PUZZLE-"
0160 PRINT "THE CENTER HOLE IS VACANT,"
0165 PRINT "AND FOUR MEN OF EACH KIND ARE ON EACH "
0170 PRINT "SIDE OF THE VACANT HOLE."
0175 PRINT "YOU MOVE BY EITHER JUMPING TO THE NEXT HOLE"
0180 PRINT "OR JUMPING OVER ONE MAN."
0185 PRINT "BACKWARD MOVES ARE NOT PERMITTED."
0190 PRINT "A SAMPLE MOVE WOULD BE 4,5 OR 3,5"
0200 PRINT "YOU PLAY AGAINST YOURSELF,"
0205 PRINT "M6800 JUST KEEPS YOU HONEST."
0210 PRINT "TO CONCEDE AT ANY TIME TYPE 9,9 "
0220 PRINT " ","GOOD LUCK"
0225 INPUT "HIT ▫CR▫ WHEN READY TO GO.",R$
0230 REM RESTART
0240 FOR I=1 TO 9
0250 READ Z$(I)
0260 NEXT I
0270 FOR I=1 TO 9
0280 READ Y$(I)
0290 NEXT I
0320 RESTORE
0330 REM PLAYS THE GAME
0340 PRINT "123456789"
0341 FOR I=1 TO 9
0342 PRINT Z$(I);
0344 NEXT I
0345 FOR I=1 TO 9
0350 PRINT "    YOUR MOVE";
0360 INPUT A,B
0370 REM CHECKS FOR VALID MOVE
0380 IF A=B GO TO 610
0390 IF A+B>17 GO TO 580
0400 IF Z$(B)<> "O" GO TO 580
0410 IF Z$(A) = "X" GO TO 460
0420 IF A>B GO TO 580
```

```
0430 IF B-A>2 GO TO 580
0440 LET Z$(B)="*"
0450 GOTO 490
0460 IF B>A GO TO 580
0470 IF A-B>2 GO TO 580
0480 LET Z$(B)="X"
0490 LET Z$(A)="O"
0500 REM CHECKS FOR A WIN
0510 FOR I=1 TO 9 STEP 1
0520 IF Z$(I) <> Y$(I) THEN 340
0530 NEXT I
0540 FOR I=1 TO 9
0542 PRINT Z$(I);
0544 NEXT I
0550 PRINT "       ******* YOU DID IT *******■■■"
0560 LET H=H+1
0570 GOTO 630
0580 PRINT "INVALID MOVE"
0590 LET K=K+1
0600 GOTO 350
0610 PRINT "YOU CONCEDE"
0620 LET G=G+1
0630 PRINT "DO YOU WANT TO TRY AGAIN":
0640 GOSUB 680
0650 ON F1 GO TO 240,750,660
0660 PRINT "JUST A YES OR NO PLEASE"
0670 GOTO 640
0680 INPUT R$
0690 LET F1=1
0700 IF R$="YES" THEN 740
0710 LET F1=2
0720 IF R$="NO" THEN 740
0730 LET F1=3
0740 RETURN
0750 IF H>0 THEN 840
0760 PRINT "YOU HAVEN■T SOLVED THE PUZZLE YET,"
0765 PRINT " SO LET■S PLAY AGAIN ANYWAY"
0770 PRINT "WHAT DO YOU SAY";
0780 INPUT J$
0790 IF J$ = "YES" THEN 240
0800 IF J$ = "NO" THEN 830
0810 PRINT "A YES OR NO WILL DO"
0820 GOTO 780
0830 PRINT "QUITTER"
0840 PRINT "YOUR SCORE IS :"
0845 PRINT H;" WINS,"
0846 PRINT G;" LOSSES AND"
0847 PRINT K;" INVALID MOVES."
0850 PRINT "GOODBYE"
0860 END
```

S1 FORMATET

När man har ett program i maskinkod lagrat på ett kassettband, brukar lagringsformatet vara S1. Detta är ett ASCII format vilket gör att det är helt printningsbart. Då vi i vår tidning visar en assemblerlista brukar vi även trycka S1 dumpen.

I bland har vi ingen källkod att producera en assemblerad listning ifrån. Vi kan emellertid alltid trycka S1 dumpen. Skälet till detta är att man ibland råkar ut för det förargliga att ett record inte går att ladda. Det kan bero på drop-out eller skada på magnetskiktet. Har man då en tryckt S1 dump inom synhåll är det i regel en smal sak att handjaga det falerande recordet.

Hur är då ett S1 record uppbyggt. Först kan vi konstatera att det innom "S" familjen även finns S0 och S9. S0 innehåller inte någon laddningsbar data, utan istället ett filnamn. Namnet genereras bl.a. av en del assemblerprogram från NAM direktivet. I EXBUG kunde man med ett kommando söka efter detta namn i en tape som innehöll flera filer. För hobbybruk har detta ej ansetts som något man bör gå och längta efter, enär inga (?) av de BUG:s vi sett har detta.

S9 används däremot som filslutmarkering. När laddningsrutinen fått i sig ett S9 record återgår i regel kontrollen till monitorprogrammet.

Uppbyggnaden av de tre olika recorden är ursprungligen lika. I hobbysammanhang förekommer dock S9 alltid i strippad form. Ett S1 record består därför av följande delar:

S1 identifikator av data - record. Inleder alla data bärande record.

bb byte räknare. Innehåller antalsuppgift för de därefter följande bytes som ingår i detta record.

aaaa adress till startpunkten i minnet dit de efter - kommande data skall lagras.

dd data. Varje byte är uppbyggd av två ASCII tecken. Antalet byte som skall in i minnet är angivet av den tidigare byte räknaren. Frådraget är då adressens två bytes och checksummans byte.

cs checksumma. Denna är en kontrollsumma som genereras av PUNCH programmet. Man adderar helt enkelt byte räknaren, de båda adress byten samt alla de ingående dataorden. Om spill uppstår genom carry från bit 8 ignoreras detta. Slutligen inverteras den adderade summan och utläggs sist i recordet. Vid laddningen i LOAD sker omvänt ett liknande adderande av samma komponenter. Även checksumman kan adderas till de övriga av det skälet att den utlades inverterat. Tack vare detta räcker det nu att öka den bildade summan med ett. Är då checksumman riktig , dvs är recordet rätt emottaget, skall summan vara noll. En bra monitors laddrutin har nu temporärt undanlagrat adress och data tills den har fått till - fälle att kontrollera mot checksumman. Därigenom garderande mot exempelvis ett fel i adressen.

Verkar det oklart? Låt oss ta ett exempel. För att inte begränsa överskådligheten nöjer vi oss med att dumpa två bytes från vårt minne. I detta fall från adress $1000. Antag att minnesinnehållet ser ut som följer:

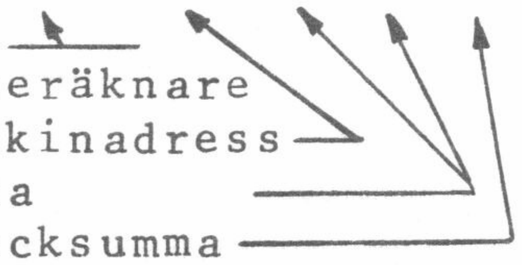
$1000 06 FA

Punchar vi ett S1 record med dessa data får det följande utseende:

S105100006FAEA

För tydlighetes skull kan vi sära lite på de skilda storheterna:

S1 05 1000 06 FA EA



Vi ser att byteräknaren fick värdet fem. Detta är riktigt enär:

adress	2
data	2
checks.	1
	05

Stämmer då checksumman? Lätt att kontrollera:

byteräknaren	05
adress, ms	10
adress, ls	00
data	06
data	FA
	115

Ignorera spillet (ettan), invertera $15.

$15 = 0001 0101
$EA = 1110 1010

I verkligheten är datarecorden längre. Vanligt är $10 som data antal. Byteräknaren syns då som $13. Formatet medger dock stor valfrihet.

Nu kan väl alla tolka våra S1 dumpar ?

TBL

TILL SALU

Tetetype ASR 33 fabr, Westrex
gångtid under 400 tim
med full dokumentation
Pris 2 400:- eller
överenskommelse
Tel: 08/84 29 84 eft 18,30
Bo Ljungblad

Kretskort 4 K statiskt 2102

med fel i layouten måste byglas på baksidan.75:-per st
08.7452667 A.Eriksson

DIGITALT Kassettdäck fabr,U.S.A

beg 300:- A.Eriksson 08/7452667

1 St MP-M 4K statiskt minneskort för SWTPc bussen. Utan 2102, men med avkod. & buffertar. Socklar för alla kretsar. 250:-
1 St. 40 kolumn. printer, lika SWTPc PR-40. 1000:-
Lars Landqvist
Skagersvägen 20/1
121 42 JOHANNESHOV
Tel. Hem 08/912625
Tel. arb. 08/990657

BILLIGA RAM..

107 st 2102 650 ns à 6:-
25 st 2102 450 ns à 10:-

MÄNGDRABATT.

Fredrik Ströberg 08/852555

Denna sida är upplåten för alla medlemmar

Endast privat annonsering är tillåten.

Helt utan kostnad för medlemmen.

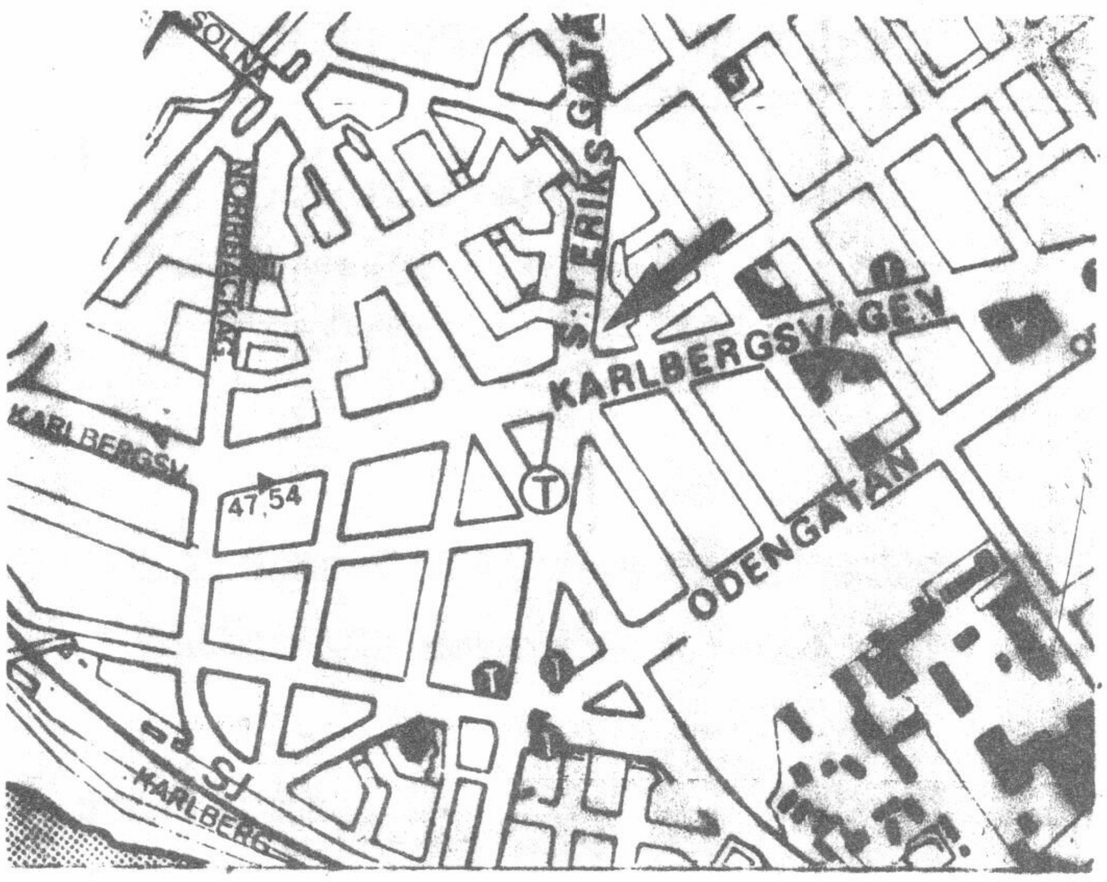
Adressen är

PD 68

Box 98

122 21 Enskede 1

INKO'X
ELECTRONIC AB



SÅ HÄR HITTAR DU



HÄR ÄR DET

INKO'X HAR NU FÅTT STÖRRE LAGER OCH BUTIK VI FINNS PÅ KARLBERGSV 44 – S:t ERIKSG 96

INKO`X HAR MIKRODATOR - KOMPONENTEN DU BEHÖVER.

INKO`X HAR Z80,6800,6502,mm,mm.

MINNS GÖRMAN PÅ INKO`X,med 2102 à 11.50,21L14 à 74:-,4060 à 49.75,4116 à 97:-

VI HAR FÅTT HEM S - 100 Bussmkort från USA.

INKO`X SÄLJER OCKSÅ 6800 Bussen.Lagret skall innefatta,Minnen 4K,16K,32K RAM

Moderkort med 11 kontakter kommer,I/O kommer,

CRT - Kort för 96364,enl beskrivning i RT.

FLOPPY Interface kommer med Operativsystem.

Postadress	Butiksadress	Telefon		Bankgiro	Postgiro	Postgiro
Box 21038	Karlbergsv 44	Kontor	08/315115	414-8425	Sverige	Norge
		"	08/307515			
S-10031 Stockholm	S:t Eriksg 96	Butik	08/322202		554687-4	1992783

PROGRAM FÖR HOPP TILL EN AV
ÅTTA NIVÅER

Ibland behöver man hoppa till en av flera adresser, beroende på en variabels värde.
På dessa adresser (NIVÅER) kan olika (sub)rutiner ligga.

NIVÅ 0 till NIVÅ 7 är åtta adresser vilka programmet hoppar till beroende på N som kan ha värden från 0 till 8.

```
0000 B6 0100 SWITCH LDA A  N
0003 06             TAP
0004 25 08          BCS    XX1
0006 2E 0E   XX0    BGT    NIVÅ   0
0008 26 2C          BNE    NIVÅ   2
000A 28 4A          BVC    NIVÅ   4
000C 20 68          BRA    NIVÅ   6
000E 2E 16   XX1    BGT    NIVÅ   1
0010 26 34          BNE    NIVÅ   3
0012 28 52          BVC    NIVÅ   5
0014 20 70          BRA    NIVÅ   7
```

I detta exempel ligger adresserna på HEX 10 bytes mellan rum med början HEX 0016 och nästa på HEX 0026, upp till 7 nivåer. B.S.

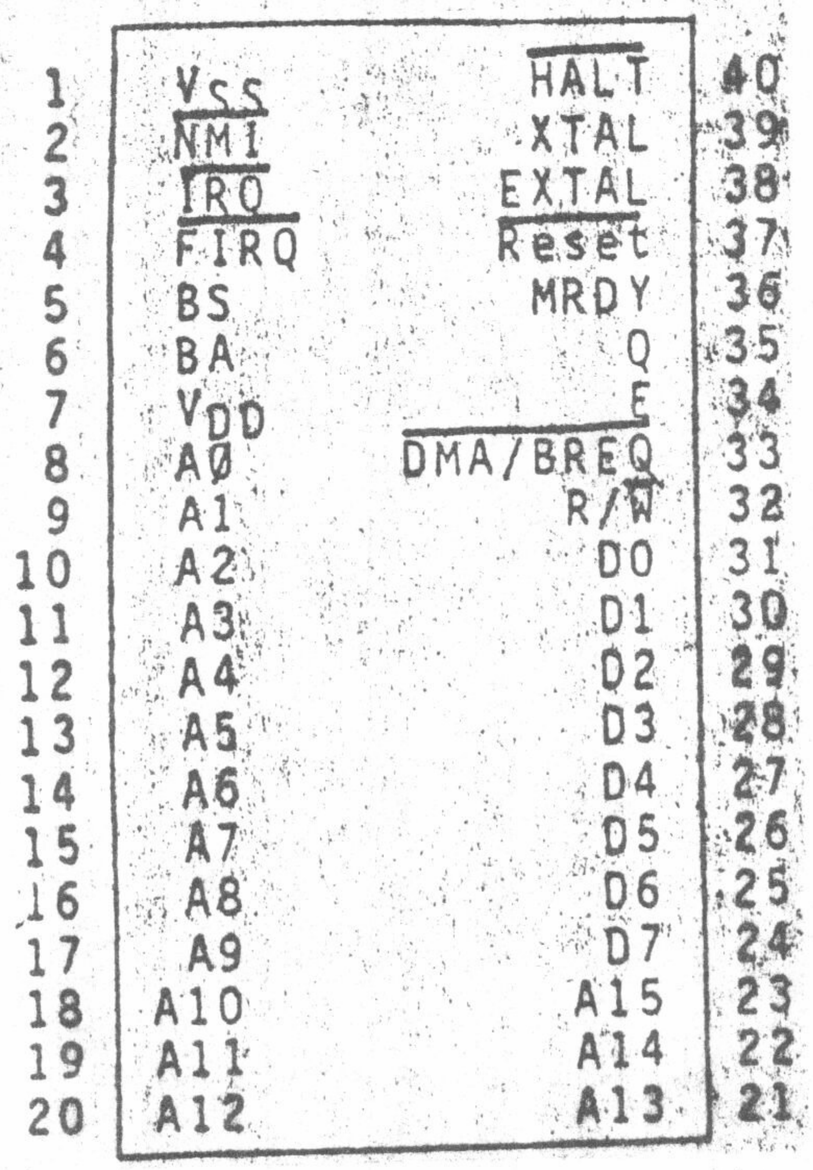
KLIPP TILL KASSETTEN

Har ni någongång råkat ut för att en kassett omöjligt går att ladda. Den bara missar. Är det en billig kassett kan det löna sig att slå den mot handflatan. Med kassettens plattsida mot handen. Vad som händer då är att rullarna lägger sig tillrätta. För att få ned priset på kassetter ger man ibland avkall på kvaliteten hos mekaniken. Som sagt.., trilskas en kassett för dig kostar det inte något att klappa till den.

TBL

6809

MC6809 Pinout



Pinningen släppt! Mera information i nästa nummer.

PROGRAMENKATEN

Här kommer resultatet av programenkäten i MPU-laren nr 1-79. Ett speciellt tack till Ted Johansson och Allan Strömstedt som möjliggjorde presentationen av detta resultat.

forts.nästa sida

TITEL	PUBLIKATION	NR	TRYCK ÅR	HARDWARE PIA	HARDWARE ACIA	ADDRESS RAM	ADDRESS ROM	MONITOR	SA KÄLLKOD	LX HEX	STORLEK	ANMÄRKNING
EM 8080 t. 9080 Sm for	DDJ	28	1978	×	×	6000 - 61F0		Mikbug			≈ 3 K	Hastighetsmätning
6800 Systems												20 gpr
LISP for the M6800	DDJ	28	1978									Endast uppg. om hur man kör. Program kan köpas från särf.
Resident Mem. Test Systems.	DDJ	25			×	6000 - 07BC					≈ 2 K	
With ex. for the 6800												
The mystery of Ackermann's function	DDJ	31	1979									Matematik
Cut 6800 Programming time	Kilobaud	2	1977	×		1FØØ		Mikbug			112 bytes	"Hexload"
A6800 single stepper	Kilobaud	2	1977	×		AØØ6		Mikbug			128 bytes	
Automatic Memory Dumper	Kilobaud	4	1977	×		Ø7ØØ		Mikbug			53 bytes	
Disassembler	Kilobaud	7	1977	×		1BØØ		Mikbug			<1,2 kbytes	Bra
Customized Mikbug	Kilobaud	3	1978	×		AØXX		Mikbug				Extended use of Mikbug-Pia
Temperature Sensing	Kilobaud	3	1978	-	-	ØØØØ		Mikbug			348 bytes	Ext. I/O
Utility Routines	Kilobaud	4	1978			ØØØØ					<1k bytes	Subrutiner
Cores time	Kilobaud	10	1978	×		7Ø84		Mikbug			272 bytes	Prints Time
Hexadecimal Memory Dump	Kilobaud	11	1978	×		3F2Ø		Mikbug			195 bytes	
IC Testing	Kilobaud	11	1978	×		ØØ8Ø		Mikbug			2k bytes	
Raster Scan Graphics	Kilobaud	11-12	1978			ØØFØ						Ext. I/O
Editor	Kilobaud	1	1979	×		Ø1ØØ		Mikbug			791 bytes	

....LUSEN....

PROMPROGRAMERAREN i nr 1 79 är insänd av ARNE TILLBOM

SWEDISH ELECTRONICS annons i nr 1 otydlig sida 9.
Den rätta är Box 2065
75002 Uppsala 018/100100

FIRMANYTT

Interelko AB utvidgar på personalsidan.Servicesidan byggsut med 1 man från 1 april som heter Åke Svanfeldt jobbat med service 10 -15 år.1 nyanställd till microsystem heter Bengt Sjöholm jobbat tidigare med konsultuppdrag främst för minidatorer.börjar 1 juni.

MODEM NYTTA

Runt om i världen finns det en massa stora dataanläggningar som körs i skift. Framåt småtimmarna när de stora jobben ligger och stampar i kärrorna, verkar det som om en del operatörer drabbas av en oemotståndlig lust att komponera spelprogram i Basic. Den slutsatsen kan man dra efter att ha sett en del imponerande spelfiler. Många av programmen är verkligt stora. Det måste ha tagit åtskilliga nattskift att skriva och debugga dem.

Eftersom man är en sann mikrodator entusiast, får man infallet att försöka plocka över några program och köra dem under den kära 8k Basicen i den egna lilla snurran.

Nu ser man med en gång att den Basic som finns på de stora datorerna innehåller en del extra "finesser". En del går att med tämligen enkla medel komma runt. Andra fordrar ganska mycket av pannskrynkel och testning innan ett korrekt resultat kan emot - ses. OK, nu har man valt ut ett program som verkar bra. Hur gör man för att få över det till en mikrodatoranläggning?

Dels måste man få över koden, dels måste Basic kunna läsa in den på ett sätt som den gillar. Det senare innebär bl.a att en tidsfördröjning måste finnas efter varje radslut för att raden skall hinna lagras i arbetsarean.

Den metod som skall beskrivas här förutsätter att man har:

- en ACIA förutom terminalens.
- kan låna hem ett modem.
- har ett KCS interface på samma ACIA som terminalen.
- har det kommunikationsprogram som visas här instoppat i sin dators minne.

Kopiera till mikrodatorminnet

Vad som utförs med bl.a. detta kommunikationsprogram och den stora datorns editor, är att få en textmassa (Basic programmet) överförd och inlagrad i den egna datorns minne. Därefter dumpar man den lagrade texten på kassett med tillägg av delay efter radbyte.

När detta är klart kan man köra igång sin Basic tolkare. För att ladda in bandet skall inte LOAD kommandot användas. Man ska istället stå kvar i kommandomod.

forts.nästa sida

Det är därför som KCS modemet ska sitta parallellt med terminalen. Slår man nu på bandspelaren tror datorn att man sitter och skriver in ett program på tangentbordet. Det betyder att den inte checkar efter speciella koder för början av ny rad t.ex. Under LOAD gör en del versioner av Basic detta.

Den av kommunikationsprogrammet inlagda fördröjningstiden mellan varje rad gör att datorn har god tid att både lagra undan raden och att skriva ut sin promt på skärmen.

Programmets funktion

Programmet är lagrat på adress $2F00 och uppåt. Det använder till lagring allt minne mellan $10 och $2EFF. För att inte fylla med onödig bråte i minnet, lagras ej LF,NULL,RUBOUT. Mellanslagskomprimering har ej ansetts nödvändig att göra. Räcker inte minnet till vid denna primära lagring, kommer det inte att räcka när Basic har slickat i sig programmet.

Efter inhoppet på läge ENTRY och initieringen av modemacian hamnar man i en loop. X-reg har initierats att peka i databanken. Finns tecken från modemet kollar man om det skall lagras. Sedan skrivs det ut på skärmen. Därefter är det dags att titta om något har hänt på det egna tangentbordet. Har ett tecken inkommit gör man en koll om det var CTRL-C (03). Var det inte en BREAK så sänds det bara vidare ut på modemet. Om det nu var CTRL-C så lägger programmet sig och inväntar ett tecken till.

Är nu nästa tecken ett 'X' går man tillbaka till monitorn. Samtidigt passar man på att ge utskrift om sista nyttjade position i minnet.

Var det i stället ett 'C' sker en återställning av X-reg till starten av minnet. Det är nämligen så att allt som inkommit från modemet har lagrat sig i minnet. Man måste ju först logga in i stordatorn innan man kan plocka ut något. Varje dator har en editor. Med editorn kan en utprintning alltid göras av en textmängd utan att värddatorfilens egna radnummer kommer ut. Omedelbart när man givit kommando om att printa allt ur filen, men innan man tryckt på den verkställande RETUR:en, ska CTRL-C och C nedtryckas. Då blir pekaren återställd till sitt startläge. Dessa tecken går ej vidare ut över modemet. Därefter verkställer man överföringsutprintningen med RETUR. Den text som nu kommer upp på skärmen lagras samtidigt i minnet. Om minnesbanken blir full, sker ingen lagring efter EOMEM.

Var tecknet i stället ett 'P' betyder det att allt lagrat i minnet skall ut på kassetten. Efter varje lagrad CR ges en LF. Utmatningen pågår tills ett EOF påträffas. EOF är ett tecken med bit 8 hög (FF). Detta stoppas in efter varje lagrat tecken.

Vid 'X' och 'P' resettar man modemacian för undvikande av inkommande felaktiga tecken. Den initieras vid CTRL-C förutom vid kallingången ENTRY. Gör man en sejour i monitorn via 'X' och går tillbaka, kan man hoppa in på en label VARMST. Då förblir modemets acia nollställd tills man självt via CTRL-C och C aktiverar den.

Modeminkoppling

I figur 1. kan man se hur ett 300 bauds modem kopplats till vår acia. Modemet har en 25-polig CANNON kontakt enligt standard. Alla de ledare som finns till - gängliga är det inte nödvändigt att använda. Vissa måste dock läggas till fast nivå i detta fallet. I figuren visas en trådminimerad koppling.

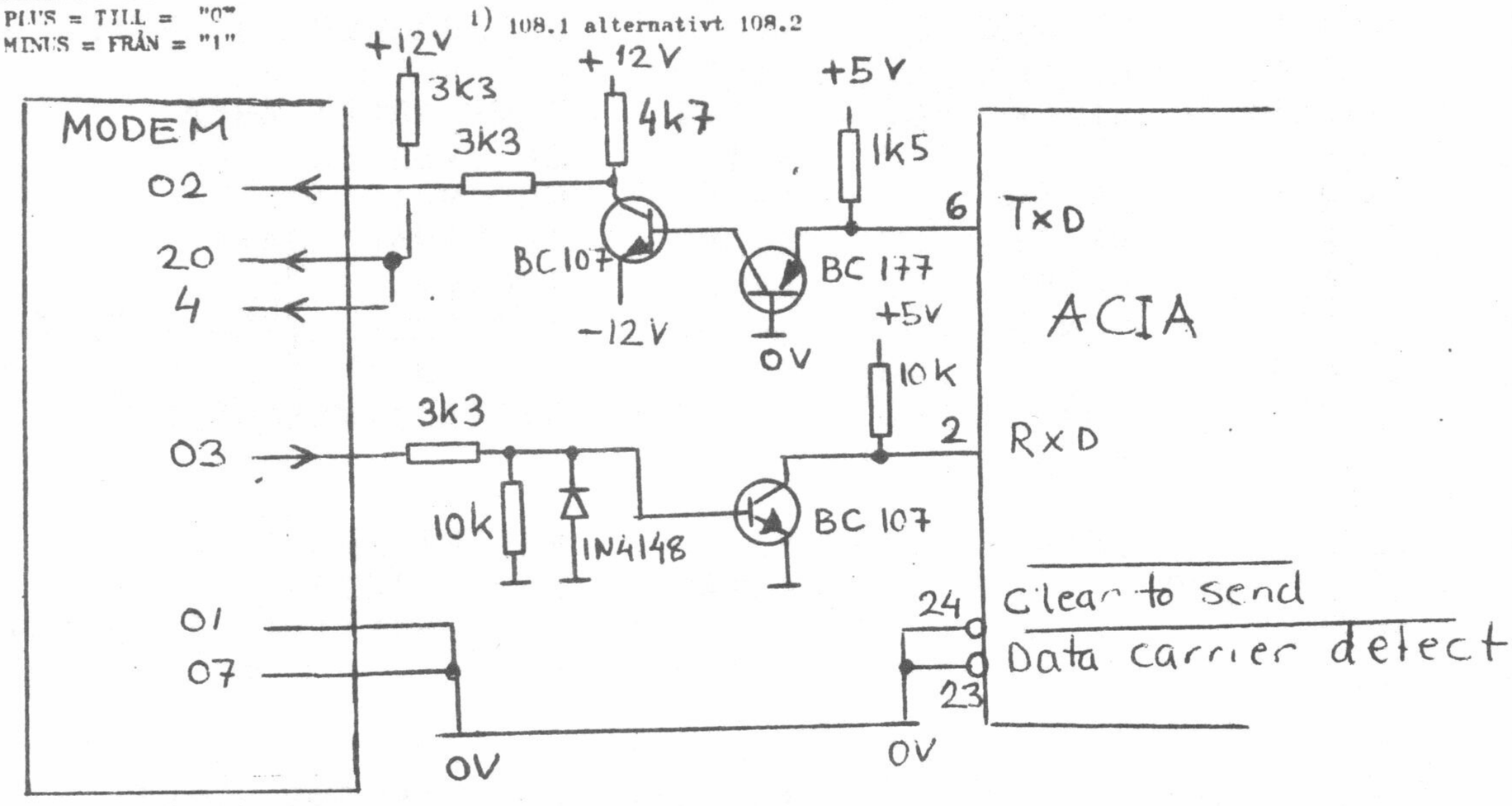
Ett Basic spelprogram som har överförts på detta sätt visas på annan plats i tidningen. Det heter EIGHT-MEN.

TBL

Ledare nr.	J1 stift	Riktning Term - Mod	Benämning	Betydelse
101	01		Skyddsjord	Ansl. till skyddsjord i nätsladd och chassi. Ansl. via strapp.
102	07		Signaljord	Gemensam ref.potential för alla gränssnittsledare exkl. 101.
103	02	——→	Sänddata	Datasign. från dataterminalutrustningen att överföras via linjen.
104	03	←——	Mottaget data	Datasign. till dataterminalutrustningen överförda via linjen.
105	04	——→	Sändningsbegäran	PLUS medför sändning.
106	05	←——	Datakanalen klar	PLUS anger att data kan sändas.
107	06	←——	Klar för drift	PLUS anger att modemen är klar för drift. Indikering på fronten.
1) 108.1	20	——→	Anslut fjärrförbindelsen	PLUS anger att modemen skall anslutas till linjen.
1) 108.2		——→	Dataterminalen klar	PLUS anger att dataterminalen är redo att sända eller mottaga data.
109	08	←——	Bärvågsindikering	PLUS anger acceptabel bärvågsnivå. Indikering på fronten.
125	22	←——	Anropsindikering	PLUS anger att ringsignal mottages.
126	11	——→	Frekvensval	PLUS anger att den högre frekvensen utsändes och den lägre mottages (B-station).

PLUS = TILL = "0"
MINUS = FRÅN = "1"

1) 108.1 alternativt 108.2



Figurtext till modemkontakten
och bilden ACIA-modemet.

Fig 1. Överst visas stiftdispositionen i modemet ZAT 300's Cannondon.

Underst visas nödvändig spänningsomvandling för datatrådarna mellan ACIA och modem. Endast fyra trådar behöver gå mellan ACIA och modem om bygling göres i kontakten.

forts.nästa sida

```
SIDA  001  GETINF



0001                        NAM   GETINF
0002                        OPT   NOP
0003                        OPT   O
0004 2F00                   ORG   $2F00
0005                *
0006                * PROGRAM FÖR ATT KONVERSERA
0007                * TRANSPARENT MED STORDATOR, LAGRA
0008                * INPUT FRÅN DATORN I MINNET,
0009                * PUNCHA UT TILL KASSETT MED
0010                * DELAY VID CR/LF
0011                * FÖRE NY RAD.
0012                *
0013                * 1979-04-04 TBL
0014                *
0015      0010      STARTM EQU   $10       DATABANKENS STARTADRESS
0016      E0C8      OUT4HS EQU   $E0C8     4 TECKEN + SP
0017      A002      SLASK  EQU   $A002     TEMP ARBET AREA
0018      E131      CRLF   EQU   $E131     SKRIVER CR + LF
0019      800C      ACIAM  EQU   $800C     MODEM ACIA
0020      8008      ACIAT  EQU   $8008     TERMINAL ACAIA
0021      E1D1      OUTEEE EQU   $E1D1     TECKEN UT
0022      E0D5      CONTRL EQU   $E0D5     KONTROLLSLINGA I TBUG
0023      E1AC      INEEE  EQU   $E1AC     TECKEN IN
0024      2F00      EOMEM  EQU   $2F00     SISTA MINNESCELL
0025                *

0026 2F00 86 03   ENTRY  LDAA  #3         MASTER RESET
0027 2F02 B7 800C        STAA  ACIAM      MODEM ACIAN
0028 2F05 86 09          LDAA  #9         JÄMN PARITET, 1 STOPBIT
0029 2F07 B7 800C        STAA  ACIAM

0030                *
0031 2F0A CE 0010 VARMST LDX   #STARTM    DATABANKENS START
0032                *

0033                * HUVUDSLINGA
0034                *
0035 2F0D F6 800C START  LDAB  ACIAM      TECKEN IN/MODEM
0036 2F10 57             ASRB
0037 2F11 24 1F          BCC   L200

0038 2F13 B6 800D        LDAA  ACIAM+1
0039 2F16 84 7F          ANDA  #$7F       NO PAR
0040 2F18 81 7F          CMPA  #$7F       LAGRA EJ RUB OUTS
0041 2F1A 27 13          BEQ   L101

0042 2F1C 81 0A          CMPA  #$A        LAGRA EJ LF
0043 2F1E 27 0F          BEQ   L101

0044 2F20 4D             TSTA             LAGRA EJ NULLS
0045 2F21 27 0C          BEQ   L101

0046                *
```

```
0047 2F23 A7 00          STAA  X          LAGRA I MINNES BANKEN
0048 2F25 C6 FF          LDAB  #$FF       EOF FILSLUT MARKERING
0049 2F27 E7 01          STAB  1,X

0050 2F29 8C 2EFE        CPX   #EOMEM-2 SISTA LAGRINGS CELL
0051 2F2C 27 01          BEQ   L101

0052 2F2E 08             INX
0053 2F2F BD E1D1 L101   JSR   OUTEEE    EKA TILL SKÄRMEN
0054                     *

0055                     *
0056                     * KOLLA OM TECKEN FRÅN TANGENTBORDET.
0057                     * OM CTRL-C OCH:
0058                     *    C ÅTERSTÄLL DATABANKSADRESSEN
0059                     *    X EXIT TILL TBUG
0060                     *    P PUNCHA TILL KASSETT
0061                     *
0062                     *
0063 2F32 F6 8008 L200   LDAB  ACIAT
0064 2F35 57             ASRB
0065 2F36 24 D5          BCC   START     OPERATÖREN VAKEN
0066 2F38 BD E1AC        JSR   INEEE

0067                     *
0068                     * CHECK IF ▪BREAK▪ + KOMMANDO
0069                     *
0070 2F3B 81 03          CMPA  #$3        CTRL-C
0071 2F3D 26 4F          BNE   L500

0072                     *
0073 2F3F BD E1AC L210   JSR   INEEE
0074 2F42 81 43          CMPA  #▪C
0075 2F44 27 BA          BEQ   ENTRY     OMLADDA ADRESSEN
0076                     *

0077 2F46 81 58   L300   CMPA  #▪X        EXIT
0078 2F48 26 14          BNE   L400

0079 2F4A 86 03          LDAA  #3        RESET
0080 2F4C B7 800C        STAA  ACIAM     MODEMET
0081 2F4F FF A002        STX   SLASK

0082 2F52 BD E131        JSR   CRLF
0083 2F55 CE A002        LDX   #SLASK
0084 2F58 BD E0C8        JSR   OUT4HS    SKRIV MAX ADRESS
0085 2F5B 7E E0D5        JMP   CONTRL    TILL TBUG
0086                     *

0087 2F5E 81 50   L400   CMPA  #▪P        PUNCHA
0088 2F60 26 AB          BNE   START

0089 2F62 86 03          LDAA  #3        RESET
0090 2F64 B7 800C        STAA  ACIAM     MODEMET
```

forts.nästa sida

```
0091                    *
0092                    * PUNCHA TILL KASSETT
0093                    *
0094 2F67 CE 0010               LDX   #STARTM
0095 2F6A 86 0D                 LDAA  #$D       CR
0096 2F6C 20 05                 BRA   L401

0097                    *
0098 2F6E A6 00    L410         LDAA  X
0099 2F70 2B CD                 BMI   L210      SLUT
0100 2F72 08                    INX
0101 2F73 BD E1D1  L401         JSR   OUTEEE

0102 2F76 81 0D                 CMPA  #$D
0103 2F78 26 F4                 BNE   L410
0104 2F7A 86 0A                 LDAA  #$A       LF
0105 2F7C BD E1D1               JSR   OUTEEE

0106                    *
0107                    * DELAY
0108                    *
0109 2F7F 86 FF                 LDAA  #$FF
0110 2F81 C6 FF    L411         LDAB  #$FF
0111 2F83 8C 2F00  L420         CPX   #EOMEM    EXTRA DELAY INSTR
0112 2F86 5A                    DECB
0113 2F87 26 FA                 BNE   L420

0114 2F89 4A                    DECA
0115 2F8A 26 F5                 BNE   L411

0116 2F8C 20 E0                 BRA   L410
0117                    *
0118                    * VANLIG UTMATNING
0119                    *
0120 2F8E F6 800C  L500         LDAB  ACIAM
0121 2F91 57                    ASRB
0122 2F92 57                    ASRB
0123 2F93 24 F9                 BCC   L500      MATA UT TECKNET
0124 2F95 B7 800D               STAA  ACIAM+1

0125 2F98 7E 2F0D               JMP   START
0126                    *
0127                          END

TOTALA FEL   00000
```

••••••••• MJUKVARUNYTT •••••••

Pascal kommer för 6800,01 och
02 i december 79.

DE VANLIGASTE INSTRUKTIONERNA I BASIC

Instruktion	Exempel	Beskrivning
Tilldelning		
LET	`10 LET A=5`	Variabeln A tilldelas värdet 5.
Inmatning		
READ-DATA	`10 READ A,B` `40 DATA 50,60`	Inläsningen från DATA-satsen ger A = 50 och B = 60.
RESTORE	`30 RESTORE`	Inläsningen sker åter från DATA-satsens början.
INPUT	`10 INPUT A,B`	Datorn skriver ? och användaren svarar med att skriva in sina värden på A och B från terminalen.
Utmatning		
PRINT	`10 PRINT "ANGE A OCH B"`	Texten ANGE A OCH B skrivs ut.
	`20 PRINT`	Ger en blank rad.
	`30 PRINT X`	Ger en utskrift per rad.
	`40 PRINT X,`	Standardformat, 5 utskrifter per rad.
	`50 PRINT X;`	Ger tätare utskrift.
	`60 PRINT TAB(10);"EXEMPEL"`	EXEMPEL skrivs ut med början i position 11.
Hopp		
GO TO	`40 GO TO 10`	Hopp sker till sats 10.
IF – THEN	`50 IF A+B<13 THEN 20`	Om A + B < 13 sker hopp till sats 20.

GOSUB - RETURN	`10 GOSUB 400`
	`20 LET S=S+1`
	`...`
	`400 LET S=0`
	`410 RETURN`
		Vid sats 10 sker uthopp till subrutinen, som nollställer S.
		Vid RETURN sker återgång till sats 20.

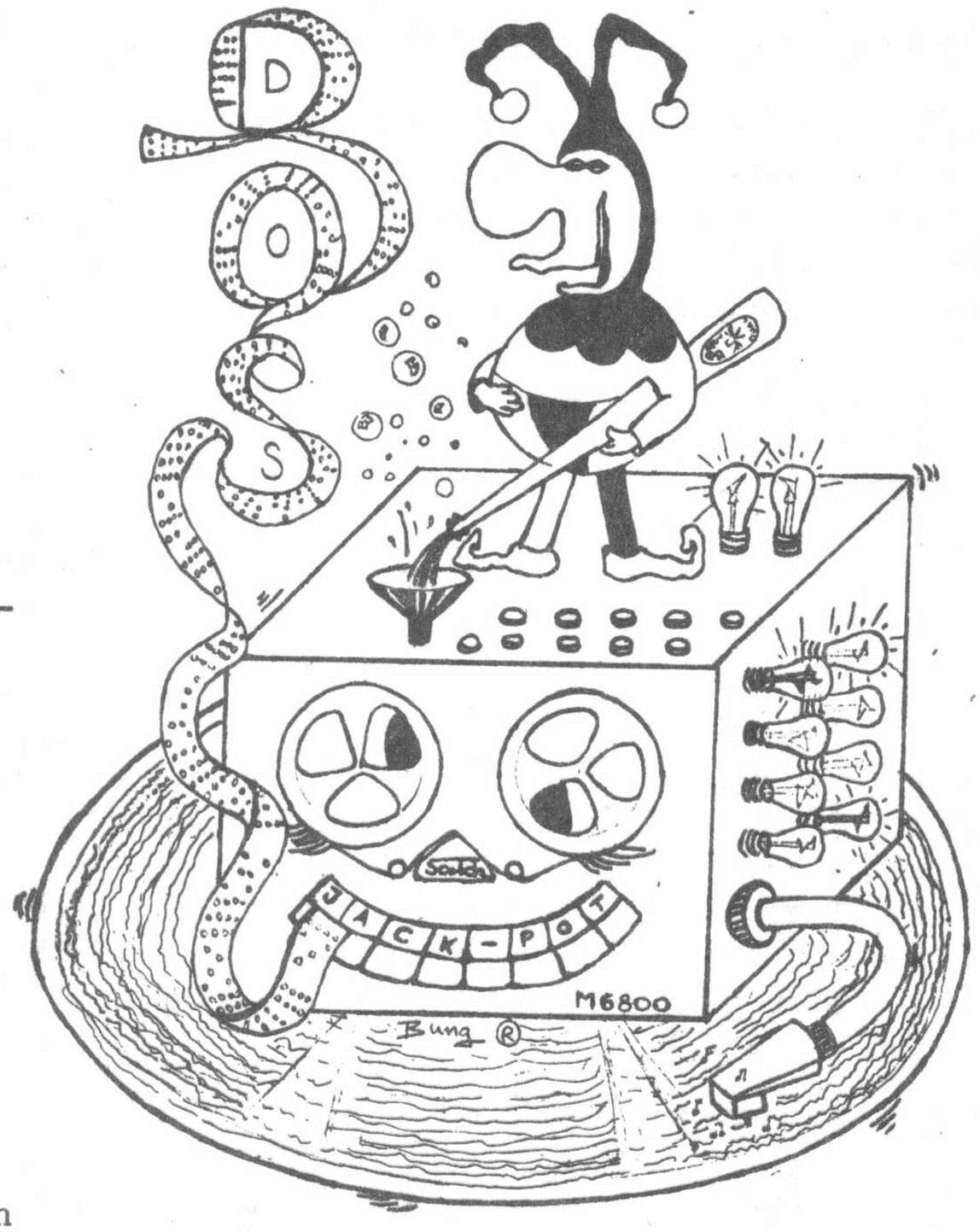
Loopar		

FOR - NEXT	`10 FOR I=2 TO 11 STEP 3`
	`...`
	`50 NEXT I`
		Mellanliggande satser utförs för I = 2, 5, 8, 11.
		STEP 1 kan utelämnas.

Övrigt		
DEF FN bokstav	`10 DEF FNB(X)=X^2`	Av användaren definierad funktion. Godtycklig bokstav kan användas.
DIM	`10 DIM A(25)`	Plats reserveras för 25 variabler i fältet A. A(1), A(2), . . ., A(25).
REM	`10 REM NU STARTAR VI`	Kommentarer kan infogas i programmet, ty datorn förbigår den information som följer efter REM.
STOP	`90 STOP`	Avslutar programutförandet.
END	`99 END`	Markerar programslut.

FLEX 2.0

en rejäl DOS till det lilla systemet.



Efter ett tag när man börjar finna tiden att ladda in ett av älsklingsprogrammen som besvärande, uppstår tanken på anskaffande av ett disksystem. Det är ett stort steg. Inte minst ekonomiskt. Även om man väljer ett mini-flexskivesystem blir ut - giften stor. Förutom drivenheterna (en eller två) tillkommer kontrollkort och systemprogramvara. Disketterna kostar ju även en slant eftersom man har dem i större antal.

Efter att ha tagit det stora steget att inhandla ett disk - system öppnar sig helt andra och intressantare tillämpningar. Anledningarna är främst den för - bättrade åtkomsten. Både i tid och storlek. Det finns saker som man behöver uppdatera dagligen och som tidigare varit opraktisk pga den långa laddningstiden. Till detta kommer även de för - tretliga laddningsfelen som uppträder icke helt sällan med audio kassetter.

Tack vare disksystemet kan man kasta in en skiva, riva ned ett program i minnet och uppdatera ett register e dyl. på nästan nolltid.

Det system som vi skall titta lite närmare på här kommer från TECHNICAL SYSTEMS CONSULTANTS INC. Namnet är FLEX. Tidigare sålde SWTPc ett system med namnet 'mini FLEX'. Nu finns FLEX 1.0 och FLEX 2.0. Versionen 1.0 jobbar med de större skivorna medan 2.0 är avsedd för miniskivor. Version 2.0 passar till SWTPc MF-68 och till Smoke Signal Broadcasting's BFD-68. Eller snarare, det finns versioner med olika diskdrivers för resp. systems olika hårdvara.

Uppgraderingen till version 2.0 från mini FLEX innebär bl.a. :

- 32k minne tillgängligt
- Printer spooling (utmatning av listor kan ske samtidigt med och oberoende av andra exekveringar).
- Random Access filer. Medger adressering direkt till önskad data.
- Uppdateringsmod. En fil kan öppnas för både skrivning och läsning samtidigt.
- Filer kan skyddas mot radering genom mjukvarumässig märkning.
- Filer kan datummärkas.
- Mera data per diskett genom större sektorer.

forts.nästa sida

Hårdvaruförutsättningar

FLEX 2.0 behöver RAM från 0000 och kontinuerligt uppåt. Minst 12k krävs. Dessutom behövs 8k i området $A000 till $BFFF. Vidare krävs en monitor liknande SWTBUG. Det är främst in/ut tecken rutinerna som efterfrågas. MIKBUG kompatibilitet duger således (TBUG) för att det skall fungera. I de fall man använder spoolingen behövs förutom timerhårdvara även interrupt vektorerna på rätt RAM ställen.

Det här systemet som rapporten handlar om består av en 6800 dator från CÅ-Elektronik enligt den byggserie som gått i Radio & Television. Kontrollenheten för de använda SHUGART drivenheterna är likaså från CÅ. Det har beteckning CÅ 6817.

Dokumentationen

FLEX 2.0 beställdes från TSC i USA och kostar $75. För detta får man en systemdiskett samt en manual över medsänd systemprogramvara. Det skall på en gång sägas att all dokumentation är ypperlig. Med mycket förklarande text och exempel beskrivs diskoperativsysytemets ingående komponenter.

I den tjocka manualen (200 sid, ungefär) finns följande delar:

- The FLEX disk operating system. Innehåller allmänna förutsättningar för att komma i gång.Kompatibilitet mot det äldre systemet mm.
- FLEX User's Manual. Beskriver de olika filtyperna, kommando hanteringen,utility rutinerna,felmeddelanden,FLEX i/o subrutiner,bootningen mm.
- FLEX Editor. Användarbeskrivning av texteditorn.
- FLEX Assembler. Användarbeskrivning av TSC Mnemonic Assembler för 6800.
- FLEX Programmer's Manual. Beskriver disk operativsystemet, filhanteringssystemet, diskett struktur, printer rutiner, interrupt i FLEX mm.

Operativsystemets beståndsdelar

FLEX operativsystemet består huvudsakligen av tre delar:

1 The File Management System. Förkortat till FMS.

2 The Disk Operating System. Även kallat DOS.

3 The Utility Command Set. Icke oväntat kallat UCS.

Kortfattat kan man säga att DOS processerar kommandon, FMS tar hand om och administrerar filer samt att UCS är användar anropbara rutiner. Exempel på detta sista kan vara en BASIC intepretator.

Ett diskoperativsystem skall vara så konstruerat att det helt sköter hanteringen av dataflödet utan att användaren behöver bekymra sig om vart bitarn slut - ligen hamnar.

Filer i FLEX

Alla filer på disketten lagras i sektorer. Varje sektor består i version 2.0 av 256 bytes. Varje byte kan motsvaras av ett tecken i det fallet att filen innehåller text. Det kan även vara en byte av maskin information. Maximalt ryms 340 av användaren tillgängliga sektorer på en enkelsidig minidiskett. En dubbelsidig har dubbelt så många vilket systemet också håller reda på.

En fil är alltid minst en sektor lång men kan expanderas till så många sektorer som finns på skivan. Sektorerna som en fil är uppbyggd av behöver inte ligga i kontinuerlig ordning. Behövs mer än en sektor i en fil lägger FMS ut en adresslänk till den nya sektorn. På detta sätt kan en fil som skapas i ett läge då det uppstått luckor i sektorerna, bli utspridd över hela skivan. Luckor blir det normalt när man tar bort en fil. Detta märker inte användaren eftersom FLEX sköter detta automatiskt. Att en fil är utspridd resulterar dock i längre accesstider. Huvudet måste då fara längre över skivan vilket tar sin tid.

Filnamn och suffix

Alla filer har ett namn. Ex:
REGISTER
TIC-TAC
GLUFFS

Varje gång man refererar till en fil användar man sig av dess namn. Det kan röra sig om en editering, skapning eller borttagning. Maximalt åtta tecken, varav det första skall vara en bokstav, kan namnet bestå av.

En fil måste även ha ett s.k. suffix. Suffixet definierar typ av information som lagras i filen. Exempel är TXT för textfiler, BIN för maskinkodsfiler,CMD för utility kommando rutiner och BAS för Basic källprogram. Ett suffix kan bestå av upp till tre st tecken. Det första måste vara en bokstav.

De flesta av FLEX utilityprogrammen förutsätter ett standardvärde på suffixet. Det gör att man inte behöver speciellt ange ett sådant. Att editorn förutsätter en TXT fil är inte att undra över. Likaså att SAVE som flyttar minnesinnehåll till en diskfil vill ha suffixet BIN.

Exempel på fullständiga filnamn kan vara:

REGISTER.TXT
TIC-TAC.BAS
GLUFFS.BIN

Genom att suffix existerar och att utility programmen förut - sätter vissa default (standardvärden), elimineras en massa tråkigheter. Ett program som man jobbar med kan således samtidigt existera som TXT källkod, BAK spard back-up kopia, BIN som assemblerat binärformat, CMD om det försetts med transfer - adress (att starta automatisk exekvering på). Det finns helt säkert fler varianter men dem kan vi spara till en annan gång.

Disk drive nummer

Förutom att en fil kan förses med namn och suffix för identifikation kan även ett drivenummer ges. Detta gör att en fil med samma namn och suffix kan existera samtidigt på olika drivar.

Anges inget nummer antas som default 'system' driven eller 'arbets' driven. Numret kan ges antingen före eller efter filspecifikationen. Exempel:
REGISTER.TXT.1
0.GLUFFS.BAS

Beteckningarna system och arbets drive behöver kanske et ord eller två till förklaring. Ett systemkommando med suffix CMD har som default hemvist på drive nummer noll. Det betyder att systemet försöker hämta den kommandofilen från drive noll. En textfil som man jobbar med antas ligga på 'arbets' driven.

När systemet initieras den första gången är både system och arbets drive satta med default till 0. Genom ett kommando kan arbetsstatusen tilldelas drive 1. Härigenom behöver man inte självt ange ett drivenummer efter de filer man jobbar med. Allt som inte har suffixet CMD kommer då att sökas efter på drive 1.

Version 2.0 av FLEX har även begåvats med en annan finess, automatisk sökning. Om systemet är satt i denna auto mod, kommer en filsökning att börja på drive 0, sedan automatiskt fortsätta på de övriga drivenheterna. Då det tar en tid innan drivmotorn nått sitt stabila varvtal (den går ju bara vid accesser till skivan), är det förståss att föredra att rätt drive utpekas från början.

forts.nästa sida

Inmatning av kommandon

FLEX startas upp i flera steg. Det första som skall ske är att man från sin monitor anropar en boot-strap loader. Denna är ett mycket litet program som kan ligga i PROM eller knappas in i RAM. Det är naturligtvis att föredra ROM basering. Denna lilla laddares enda uppgift är att från ett förutbestämt ställe på skivan draga ned en större laddare i minnet. Den större laddaren tar sedan hand om uppstartningen av resten av operativsystemet. När allting så småningom är klart dyker tre promt tecken upp på skärmen (+++). Då vet man att FLEX är redo att börja jobba åt en.

Det man skriver från sitt tangentbord hamnar först i en in - buffert. Felaktiga nedslag kan därför korrigeras innan operativsystemet sätts att analysera vad man skrivit. Exekvering av en rad sker först efter tryckning av RETUR knappen.

En kommandorad består vanligtvis av ett inledande kommando samt en lista av namn och parametrar. Flera än ett kommando kan skrivas på samma rad. Man separerar dem bara med ett ':' tecken. Gränsen är dock maximala 128 tecken i inbufferten. Exempel på kommandorader kan vara:

```
+++CAT 1
+++LIST REGISTER
```

Den första raden ger en ut - skrift över filkatalogen på drive nummer ett. Den andra raden ger en listning av textfilen REGISTER på bildskärmen.

Styrning mot radskrivare

I många fall önskar man få de av utility programmen producerade utskrifterna mot en printer i stället för på bildskärmen. Till hjälp för detta finns en speciell utility som heter P. Denna ser till att de utmatningar som skall till printern hamnar där, samt att det som fortfarande skall till skärmen också kommer dit. Felmeddelanden och liknande som fordrar operatörens uppmärksamhet styrs till skärmen. Dokumentationen är så utformad att man kan skriva en egen printerrutin som passar just den häck man kommit över billigt på överskott. Exempel på ett kommando med radskrivarutmatning:

```
+++P LIST REGISTER
```

Finesser vid uppstartningen

När FLEX har bootat upp och är redo kanske man har önskemål om att vissa saker skall utföras automatiskt. Det kan t.ex. vara definition av systemparametrar, egenskaper hos ens bildskärm, utskriftsformat,specialtecken definitioner etc. Man kanske även önskar dra igång Basic interpretatorn med engång. Bara lugn, de har tänkt på den saken också!

Det första FLEX gör efter att ha initierat sig är att söka i diskens directory efter en fil med namnet STARTUP.TXT. Om inte denna fil kan hittas så kastar FLEX ut de tre promtarna och är redo för användarkommandon. Om däremot filen finns, kommer den att läsas och behandlas som en enstaka kommandorad. Vill man promt ha sin Basic snurrande helt automatiskt, räcker det således att bygga upp en fil med namnet STARTUP.TXT som endast innehåller kommandot BASIC.

Det kan tyckas vara en begränsning att bara kunna fördefiniera ett kommando. Det verkar ju som om man inte både kunde anropa TTYSET för definition av systemparametrar och sedan dra igång Basic. Ingen panik, det finns medicin mot detta med!!

Hjälpen är ett utility program som heter EXEC. Denna rutin gör ett anrop av en TXT fil som kan innehålla hur många kommandon som helst. EXEC rutinen "lurar" operativsystemet att tro att de kommandon som det utför helt enkelt kommer från operatörsterminalen.

Vad man behöver göra är nu i stället att låta den enradiga STARTUP.TXT innehålla ett an - rop på EXEC med tillhörande fil.

Utilitys medföljande FLEX 2.0

Vid beställningen från TSC i USA fick man som sagt tidigare en systemdisk. Denna innehåller inte enbart diskoperativsystemet utan även en rad hjälprutiner, s.k. utility program. I fig 1. ser vi en directory utskrift från denna systemdiskett. Utskriften är begärd med kommandot CAT.
En liten kort genomgång av samtliga utilitys kanske faller i smaken ?

APPEND.CMD Detta kommando nyttjas för att sammanfoga, konkatenera, filer. Två eller flera. Alla typer av filer kan smäckas ihop. Normalt är det bara vettigt att sammanfoga filer av samma typ, exempelvis binärfiler. En ny fil skapas för resultatet och alla ingående filer förblir intakta.

ASN.CMD Kommandot används för att assignera de tidigare omtalde 'system' och 'arbets' numren. Ger även utskrift av aktuell status.

BUILD.CMD Används för att snabbt bygga små textfiler. Exempelvis STARTUP filen. Är en mycket anspråkslös editor för nya filer.

CAT.CMD Katalog kommando för utskrift av diskettens directory. Ger den utskrift som finns i fig 1. Visar filens namn och suffix samt skyddsflaggor. De senare är delete resp. write protect för filen.

COPY.CMD Används för kopiering av filer. Antingen enstaka filer eller filer med vissa suffix. Vidare kan hela disketter kopieras mm. Om en fil haft sina ingående sektorer spridda, kommer de att försökas omgrupperas för att bli mera sammanhållna.

DATE.CMD Används för att visa och/eller ändra FLEX interna datumregister. För framtida bruk.

DELETE.CMD Kommandot används för att plocka bort, deleta, en fil från disketten. Rår inte på en fil som man självt satt delete protect på.

EXEC.CMD Som vi sett tidigare används detta kommando för att starta kommandoströmmar, Hämtar upplagda kommandorader från en fil och exekverar dem i tur och ordning.

GET Detta är inte en utility utan en funktion i DOS. Detta kommando ligger resident i det av DOS använda minnet. Med GET kan binärfiler laddas ned till minnet.

I.CMD Medger att hämta input tecken från en fil i stället från terminalen. Kan användas i en STARTUP rad som anropar en utility vilken ställer en fråga till operatören och vill ha svar innan den går vidare. Svaret, exempelvis "YES" kan då automatiskt hämtas från en diskfil.

JUMP.CMD Används för att starta exekvering av ett program som redan ligger i datorns minne. Man ger som enda parameter den hexadecimala adress där exekveringen skall börja.

LINK.CMD Används för att informera laddprogrammet vart själva diskoperativsystemet ligger på skivan. Laddprogrammet är ju en "dum" rutin som bara vet att det skall läsa ned från en viss sektor. Flyttar man FLEX2.SYS genom kopiering e.dyl. måste en LINK utföras på den nya skivan.

LIST.CMD Används för listning av Basic eller text filer på terminal eller printer. Man kan även speca listning mellan vissa rader eller från viss rad.

MON Detta kommando är i likhet med GET ingen utility. Det är ett resident kommando. Dess enda uppgift är att hoppa till ens monitors kontrollslinga. Med monitor avses ex.vis MIKBUG, T-BUG eller dyl. **forts.nästa sida**

NEWDISK.CMD Sätter upp och formatterar en ny diskett. Lägger upp viss systeminformation samt kollar att inga taskiga sektorer finns på skivan. Sektorer som ej går att använda märks som felaktiga. De kommer sedan att automatiskt uteslutas av operativsystemet. Har man otur kan förståss en sektor som systemet själv behöver vara paj.

O.CMD Används för att styra om utmatning från skärmen som ett utility program producerar ut på en fil.

P.CMD Detta är systemets printer rutin. Medger omstyrning av utskrift från alla kommandon till en radskrivare.

PRINT.CMD Används för att mata ut tidigare lagrade utskrifter samtidigt som datorn utför andra uppgifter. Kräver hårdvara för att fungera. Bl.a. en timer. Medger att man spelar STARTREK samtidigt som 100 sidor rasslar ut på printern.

PROT.CMD Används för att ändra och skriva ut eventuella delete och write protect markeringar på en fil.

QCHECK.CMD Administrerar den kö av utskrifter som kan finnas då man nyttjar printer spoolingen som FLEX tillåter.

RENAME.CMD Används för omdöpning av filer och suffix.

SAVE.CMD Flyttar minnesinnehåll upp på en diskfil. Man specar filnamn, start/stopp adress samt en s.k. transferadress (option). Transferadressen används för en automatisk exekveringsstart av ett program. Alla kommandorutiner har en sådan.

SAVE.LOW Samma som SAVE.CMD utom att den ligger på lägre minnesadress. Medger undanlagring av program i själva utility arean.

STARTUP.TXT Detta är inte en utility utan en finess i FLEX. Tidigare har vi beskrivit hur FLEX efter initieringen kollar i directoryt efter en fil med detta namn. Finns den så kommer dess enda tillåtna rad att tas för ett kommando och utföras.

TTYSET.CMD Med detta kommando kan man sätta en rad parametrar som kontrollerar egenskaperna hos terminalen. Ex: speca ant. rader/sida, tecken/rad, delete tecken, antal nulls vid CR mm. Denna rutin bör inkallas automatiskt genom en STARTUP.

VERIFY.CMD Används för att visa och/eller ändra verify flaggan. Med flaggan satt kommer alla de sektorer som skrivs till skivan att automatiskt läsas och jämföras med indatat.

VERSION.CMD Printar versionsnam för ett program. Fordrar att amn följer en programmeringsregel och lägger en FCB nr på visst ställe i ens program.

XOUT.CMD Deletar de filer av typ OUT som innehåller lagrade ut - skrifter. Tar bort alla som ej är protected.

Övrigt i katalogen

I listan i fig 1. finns ytterligare några filer. Dessa är ej utility program.

ERRORS.SYS Innehåller felmeddelanden av fix längd. Filen är av random access typ. Det gör att en systemrutin som vill ha en felutskrift gjord och har ett felnummer i handen, kan beräkna en sektoradress och direkt gå in och hämta en textsträng. Sparar primärminnesutrymme och medger texter på olika språk som enkel förlängning.

FLEX2.SYS Diskoperativsystemet. Komplett med I/O rutiner. Finns även i versionen FLEX.COR utan dessa rutiner. Dokumentationen visar hur man för ett eget system skräddarsyr sådana om de som finns inte passar.

Egna program som nyttjar FLEX

Medföljande FLEX Programmer's Manual innehåller all dokumentation för att man skall kunna skriva egna program som använder de möjligheter FLEX ger.

Först finns en förteckning med de minnesceller inom DOS delen av FLEX vilka kan vara intressanta. Det är bl.a. alla parametrar som styr utmatning och inmatnings formaten. Vidare de pekare associerade med kommandobuffert hanteringen, transfer - adress, feltyp och en jäkla massa andra dataceller som skulle ta för lång tid att räkna upp.

Sedan kommer en lista över de användare anropbara rutinerna i systemet. Det är bl.a. kalla och varma entrypunkter i DOS. Teckenrutiner av alla de slag för in/utmatning mot terminal och radbuffert. Rutin för att extrahera filspecifikationer ur inbuffert-en och lagra dessa i ett FCB, filkontrollblock. Rutin för laddning av binärfil. Felmeddelande rapportering, talkonvertering mm. Vidare finns möjlighet att anropa DOS som en subrutin.

Hur man lägger upp egna kommandon visas i manualens efterföljande stycke. Dessa kan vara både minnesresidenta och diskresidenta. Variabel läng tillåts.

Filhanteringssystemet FMS

Filhanteringssystemet är en länk mellan DOS och hårdvaran för diskarna. FMS utför all allokering av filern på disketten. Likaså borttagning av filer och frigöring av använt utrymme. Detta sköts dynamiskt vilket gör att man som användare inte behöver bry sig om var en fil rent fysiskt ligger.

Kommunikationen med FMS sker genom filkontrollblock FCB:n. Dessa innehåller nödvändig information om filen. Disk I/O utförd genom FMS är av "ett tecken i taget" karraktär. Ett program behöver därför bara sända eller begära ett tecken i taget. Likheten med arbete mot en vanlig terminal är därmed slående.

FMS är ett kommando språk som styrs med hjälp av olika funktionskoder. Varje kod informerar FMS vad den skall göra. Det kan vara att öppna en fil för läsning eller att deleta en fil. I praktiken är det rätt enkelt. X-reg får peka på det FCB som skall användas. Funktionskoden lagras i första ordet i FCB:n och sedan anropas FMS som en subrutin. Efteråt kan man testa på en returkod för att se om allt gick rätt. Man behöver således aldrig bekymra sig om vart all information hämtas. Det utför FMS helt på egen hand.

Ett obestämt antal filer kan hållas i luften samtidigt. Det krävs bara att varje fil har en egen FCB.

Funktionskoder

Koden består av en byte som lagras först i en FCB. För närvarande finns 20 st använda. Att se dem listade ger bra grepp om vad man kan göra:

00 Read/Write Next Byte/Char
01 Open for Read.
02 Open for Write.
03 Open for Update.
04 Close File.
05 Rewind File.
06 Open Directory.
07 Get Information Record.
08 Put Information Record.
09 Read Single Sector.
10 Write Single Sector.
11 -
12 Delete File.
13 Rename File.
14 -
15 Next Sequential Sector.
16 Open System Information Rec.
17 Get Random Byte from Sector.
18 Put Random Byte in Sector.
19 -
20 Find Next Drive.
21 Position to Record N.
22 Backup One Record.

forts.nästa sida

Filer: Random,Binär,Text

FLEX 2.0 medger bruk av random access filer. Med detta förstås att man kan nå information på ett slumpmässigt ställe i filen utan att behöva läsa den från början i sekvensiell mod. För att få denna finess måste man offra utrymme. En random fil måste skapas som random. Det är en egenskap som ligger i filens struktur. Systemet skapar och hänger på filen en File Sector Map. Denna har alltid längden två sektorer. I denna FSM kommer nu systemet att lagra den fysiska adressen för varje logisk sektor.

Med logisk sektor skall förstås ett ordningsnummer. En fil består ju av flera sektorer som kan vara utspridda på skivan. De logiska numren är dock alltid desamma, ex 0,1,2..n. Genom att man vet antalet användbara bytes inom en sektor (252) och längden på den informationsenhet man processar, kan den önskade logiska sektorns nummer framräknas. Att få fram rätt information kräver sedan endast två filaccesser. Gör man mycket accesser kan det löna sig att ha FSM:en nere i primärminnet. Då behöver denna länkinformation inte accessas varje gång på disken.

En binärfil kan innehålla allt som data. En binärfil består av ett eller flera binärrecords. Ett sådant record kan se ut som följer:

Byte 0 Start of record indicator ($02, ASCII STX).
Byte 1 Mest signifikant byte i laddadressen.
Byte 2 Minst signifikant byte i laddadressen.
Byte 3 Antal bytes data i rec.
Byte 4-n Binära data bytes.

En binärfil kan även innehålla ett transfer adress record. Det används för att kunna starta en automatisk exekvering. Format:

Byte 0 Transfer address indicator ($16, ASCII ACK).
Byte 1 Mest signifikant byte i transferadressen.
Byte 2 Minst signifikant byte i transferadressen.

En textfil innehåller bara de ASCII tecken som är printbara. Plus ett par specialare.

$0D (ASCII retur). Markerar radslut.

$09 (ASCII HT). Denna är en flagga som säger att en sträng av mellanslag har tagits bort och komprimerats i filen. Den byte som följer omedelbart efter flaggan anger antalet borttagna blanka (2-127). När man har öppnat en text fil för läsning eller skrivning, kommer FMS att på egen hand lägga till resp ta bort dessa blanka. Detta märker man inget av.

Exempel på hur det kan se ut

Avslutningsvis tittar vi på ett verkligt exempel. Det är samma LIST utility som finns i FLEX. Programmet hämtar ett tecken i taget från en fil och skickar ut det till terminalen. Listan är självdokumenterande, men ett par ord kan vara på sin plats.

Observera hur man anger ett versionsnummer på adress $A102. Programmet börjar på $A100 och hoppar över detta nummer. Nåväl, på $A103 pekar man mot den använda FCB:n. Därefter anropas rutinen GETFIL som extraherar filnamn och suffix och ser till att de stoppas in på rätt plats i filkontrollblocket. Sedan är det dags att lägga in funktionskoden för öppning av filen i läsmod. Innan FMS anropas stoppar man in ett defaultvärde för TXT ifall detta inte kommit via den av operatören givna kommandoraden.

På adress $A117 läser man sedan en byte. Är det en CR så bifogas även en LF.

Hur får man då hejd på slingan? Jo genom att FMS återkommer med en felkod. När den gör det hoppar programmet till rad A12D. Där kollar man om felkoden betydde filslut. Var det filslut stänger man filen och går tillbaka till FLEX. Var det inte filslut utan ett riktigt fel, anropas en rutin som raporterar felet. Därefter stänger man filen och återgår till FLEX.

Det verkar vara ganska enkelt att kunna använda FLEX för egna ändamål. I alla fall motsäger inte detta programexempel en sådant påstående.

Summering

Att ta fram ett diskoperativsystem är förmodligen en ganska grannlaga affär. Det är inget man snyter ur näsan på en kafferast. Vad som framkommit vid körning av FLEX, studier av manual och erfarenheter av andra system förstärker intrycket att FLEX 2.0 är ett mycket kraftfullt diskoperativsystem.

Till FLEX finns en del programvara, exempelvis Textprocessor, Basic (intepretativ och kompi - lerande) samt SORT/MERGE paket. Vi kanske kan återkomma med en rapport om dessa vid senare tillfälle.

TBL

Fig 1. Utskrift av systemdiskettens directory.

```
CAT

CATALOG OF DRIVE NUMBER 0
DISK:    #0

 NAME    TYPE  SIZE  PRT

ERRORS   .SYS     9
FLEX2    .SYS    25
PRINT    .SYS     1
CAT      .CMD     3
LIST     .CMD     3
COPY     .CMD     5
TTYSET   .CMD     2
RENAME   .CMD     1
DELETE   .CMD     2
ASN      .CMD     1
APPEND   .CMD     3
EDIT     .CMD    24
ASMB     .CMD    25
BUILD    .CMD     1
P        .CMD     1
JUMP     .CMD     1
SAVE     .CMD     2
PROT     .CMD     1
EXEC     .CMD     1
DATE     .CMD     2
I        .CMD     1
LINK     .CMD     1
O        .CMD     2
QCHECK   .CMD     3
VERIFY   .CMD     1
VERSION  .CMD     1
XOUT     .CMD     2
NEWDISK  .CMD     7
PRINT    .CMD     2
SAVE     .LOW     2
FLEX     .COR    24

SECTORS LEFT = 181
```

forts.nästa sida

SCHACK-PIIP

I schackprogrammet i MPU-laren nr 3/78 kan en liten "FIX" göras så att datorn ger BELL till terminalen när den ger sitt svarsdrag. Denna ändring samt de angivna vektorerna är införda i de versioner som vi sänder på KCS-kassetter. Programmet går således direkt mot MIKBUG/TBUG etc.

För att få signalen vid svar från datorn görs följande änd - ringar:

```
0427   BD 0700  JSR EXTRA

0700   86 07    LDAA # 7  BELL
0702   BD 048E  JSR UTTECK
0705   BD 0649  JSR OUTDAT
0708   39       RTS
```

TBL

Exempel på uppgörande av egna
rutiner med hjälp av FLEX:ens
system komponenter.

FLEX Advanced Programmer's Guide

```
                    *
                    * SIMPLE TEXT FILE LIST UTILITY
                    *

                    * COPYRIGHT (C) 1978 BY
                    *
                    * TECHNICAL SYSTEMS CONSULTANTS, INC.

                    * DOS EQUATES

AD03                WARMS   EQU     $AD03       DOS WARMS START ENTRY
AD2D                GETFIL  EQU     $AD2D       GET FILE SPECIFICATION
AD18                PUTCHR  EQU     $AD18       PUT CHARACTER ROUTINE
AD24                PCRLF   EQU     $AD24       PRINT CR & LF
AD33                SETEXT  EQU     $AD33       SET DEFAULT NAME EXT
AD3F                RPTERR  EQU     $AD3F       REPORT DISK ERROR

                    * FMS EQUATES

B406                FMS     EQU     $B406
B403                FMSCLS  EQU     $B403

                    * SYSTEM EQUATES

A840                FCB     EQU     $A840       SYSTEM FCB

                    * LIST UTILITY STARTS HERE

A100                        ORG     $A100

A100 20 01          LIST    BRA     LIST2       GET AROUND TEMPS

A102 01             VN      FCB     1           VERSION NUMBER

A103 CE A8 40       LIST2   LDX     #FCB        POINT TO FCB
A106 BD AD 2D               JSR     GETFIL      GET FILE SPEC
A109 25 34                  BCS     LIST9       ANY ERRORS?
A10B 86 01                  LDA A   #1          SET UP CODE
A10D A7 00                  STA A   0,X         SAVE FOR READ OPEN
A10F BD AD 33               JSR     SETEXT      SET TXT EXTENSION
A112 BD B4 06               JSR     FMS         CALL FMS - DO OPEN
A115 26 28                  BNE     LIST9       CHECK FOR ERROR
A117 CE A8 40       LIST4   LDX     #FCB        POINT TO FCB
A11A BD B4 06               JSR     FMS         CALL FMS - GET CHAR
A11D 26 0E                  BNE     LIST6       ERRORS?
A11F 81 0D                  CMP A   #$D         IS CHAR A CR?
A121 26 05                  BNE     LIST5
A123 BD AD 24               JSR     PCRLF       OUTPUT CR & LF
A126 20 EF                  BRA     LIST4       REPEAT
A128 BD AD 18       LIST5   JSR     PUTCHR      OUTPUT THE CHARACTER
A12B 20 EA                  BRA     LIST4       REPEAT SEQUENCE
```

```
A12D A6 01     LIST6   LDA A   1,X       GET ERROR STATUS
A12F 81 08             CMP A   #8        IS IT EOF ERROR?
A131 26 0C             BNE     LIST9
A133 86 04             LDA A   #4        CLOSE FILE CODE
A135 A7 00             STA A   0,X       STORE IN FCB
A137 BD B4 06          JSR     FMS       CALL FMS - CLOSE FILE
A13A 26 03             BNE     LIST9     ERRORS?
A13C 7E AD 03          JMP     WARMS     RETURN TO FLEX

A13F BD AD 3F  LIST9   JSR     RPTERR    REPORT ERROR
A142 BD B4 03          JSR     FMSCLS    CLOSE ALL FILES
A145 7E AD 03          JMP     WARMS     RETURN TO FLEX

                       END     LIST
```

SLÄPP STOPP-BITARNE LOSS

När man skickar en kassett med ett Basic program lagrat i KCS kod till någon, vill man vara säker att denne kan ladda in det. Någonting som skulle kunna sätta en käpp i hjulet är att de båda systemen har olika antal stoppbitar på tecknen. Det är inget problem att ladda in en dump där data lagrats med två stoppbitar i ett system med en ACIA pro - grammerad för en stoppbit. Nej, det är det omvända fallet som ger problem.

Den som dumpar bör därför se till att två stoppbitar används. Nu är det inte säkert att det hjälper att exempelvis med sin monitors memory change rutin för hand lägga in $11. Det första som Basic interpretatorn gör när man slagit ett kommando, verkar vara att initiera porten. Bläh!

I de Basic versioner som SWTPc har för M6800 kan man starta en exekvering både med RUN och med GOTO. Om man då knappar in det lilla program som visas i fig 1. så fixar det sig. Det kan läggas någonstans där det inte stör det ordinarie programmet. Med hjälp av POKE stoppar man in det nya kontrollordet i ACIA:n. I exemplet ligger den på $8008. Kon - trollordet ($11) blir 17 i decimal form. FOR-NEXT loopen är till för att ge tid att slå på bandspelaren. När sedan vår Basic når satsen 9400 SAVE kommer allt ut på tapen som vanligt. Fast med två stoppbitar...!

TBL

```
9000 REM
9100 FOR I=1 TO 200
9200 NEXT I
9300 POKE( 8*4096+8,17 )
9400 SAVE
9500 STOP
```

Fig 1.

SPECIALERBJUDANDEN FRÅN CA-ELEKTRONIKS DATAAVDELNING :

★ ★ ★ ★ ★ ★ ★ ★ ★

MODIFIERINGSSATS FÖR D2-KITET ENLIGT RT 6-7,1978 NU: 250,-

BUS-IC FÖR KOMPLETTERING AV D2-KORTET NU: 68,-

KCS-INTERFACE CA-6803 ENLIGT RT 9, 1978 NU: 129,-

TANGENTBORD KEYTRONIC L1648 MED LADA KOMPLETT. NU: 995,-

VIDEOMINNE CA-6806 ENLIGT RT 10, 1978, 40 TECKEN PER RAD, 24 RADER, 128 ASCII-TECKEN, 63 GRAFISKA SYMBOLER, 7 FÄRGER, BLINKNING. NU: 1.100,-

EPROM-KORT CA-6808 MED 18 SOCKLAR FÖR 2708 NU: 499,-

EPROM 2708, PROGRAMMERADE, SATSER OM 7 ST NU: 599,-

KRETSKORT CA-6805 MED KOMPONENTER. AVSETT FÖR BELL-SIGNALERING MED HÖGTALARE (INGÅR). NU: 39,-

EPROMPROGRAMMERARE CA-6807 FÖR 2708,2716 & 2532 NU: 995,-

▪▪ OVANSTÅENDE SPECIALPRISER GÄLLER T.O.M. 79-06-30 ▪▪

▪▪▪