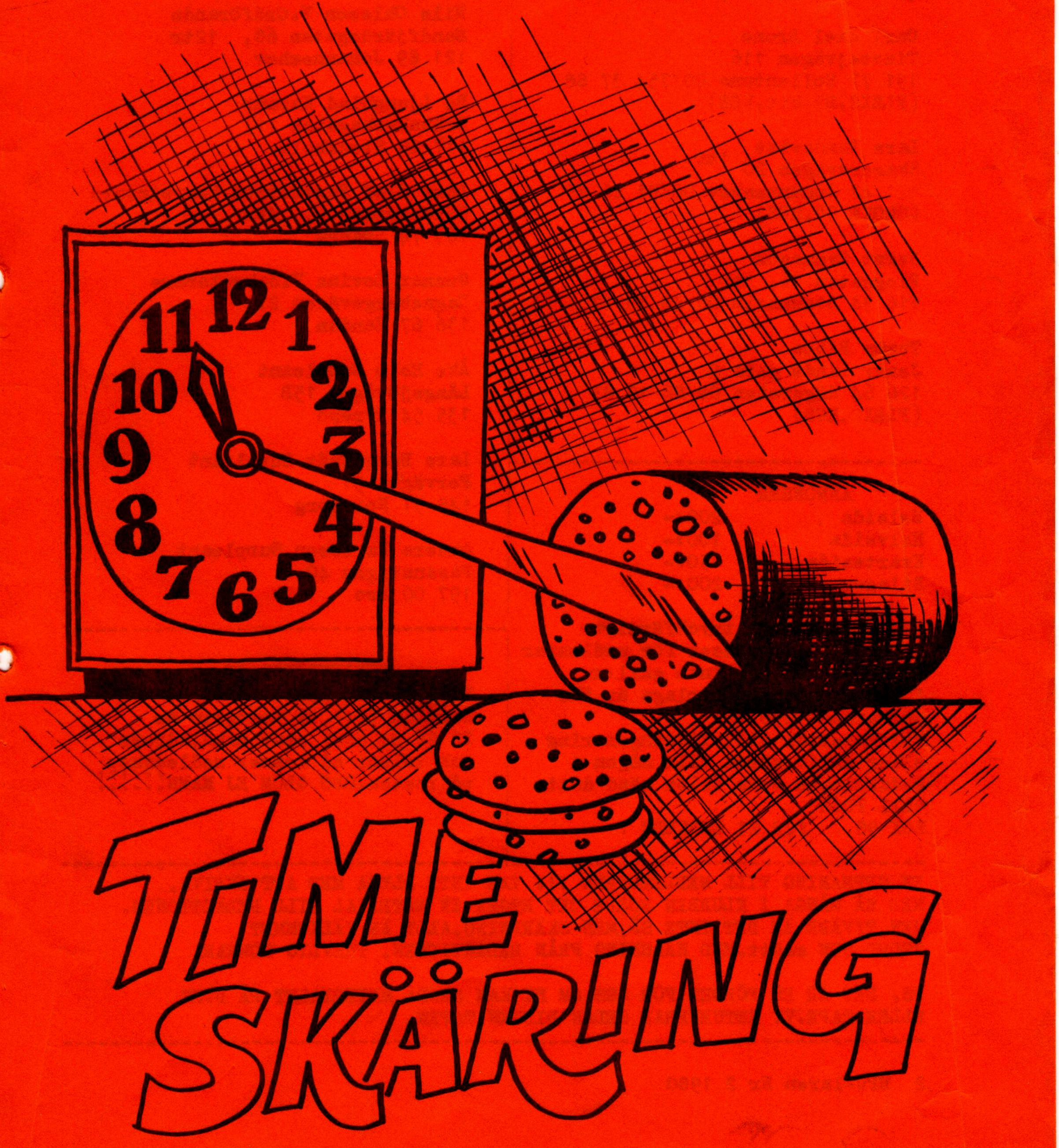
MPU-laren nr 2 1980

Tidskrift för Privatdataklubben PD68-Årgång 3



REDAKTIONEN

MPU-larenredaktionen består av följande medlemmar:

Roger Holmstrand
Rapsodivägen 168
142 00 Trångsund 08/771 38 52
(FLEX2,FLEX9,Papperstape,KCS)

Carl Axel Bruno
Timotejvägen 116
191 77 Sollentuna 08/754 37 88
(FLEX2,D2-kit,KCS)

Lars Landqvist
Skagersvägen 20 1tr
121 42 Johanneshov 08/91 26 25
(SMOKE 5",KCS)

Bernt Svensson
Ploggatan 16, 3tr
116 34 Stockholm 08/40 35 17

Tommy Riboe
Jenny Linds gata 9
126 64 Hägersten 08/88 16 83
(FLEX2,KCS)

ANNONSER

Helsida	200:-
Halvsida	100:-
Kvartssida	50:-
Bilagor	200:-

Priserna gäller tryckfärdigt material.Radannonser för medlemmar är gratis om det inte gäller firma.Tidningens upplaga är för närvarande 250 ex.
För material till tidningen ring eller skriv gärna till någon i redaktionen ovan eller sänd detta till PD68
BOX 98, 122 21 ENSKEDE 1

STYRELSEN

PD68 nuvarande styrelse:

Tommy Bladh Ordförande
Norrholmsvägen 106
132 00 Saltsjö-Boo

Nils Ohlsson V.Ordförande
Sandfjärdsgatan 68, 12tr
121 69 Johanneshov

Bo Ljungblad Kassör
Brännkyrkagatan 117
117 28 Stockholm

Lars Olof Fagerström Sekreterare
Gärdebyplan 6, 4tr
163 74 Spånga

Gunnar Lovius V.Sekreterare
Ragnebergsvägen 21
136 67 Handen

Åke Holm Suppleant
Långsjövägen 15B
135 54 Tyresö

Lars Bergstedt Suppleant
Forvägen 31
145 51 Norsborg

Anders Karlsson Suppleant
Fasanstigen 40
197 00 Bro

MEDLEMSINTRÄDE

Vill DU bli medlem i PD68?
Sätt in 75:- på postgiro nr
PG 96 04 68-7 samt namn,adress och telenr så kommer medlemskap på posten.OBS GLÖM EJ NAMN!!!!!

EN UPPMANING TILL MEDLEMMARNA !!! VAR INTE BLYGA MED ATT SKRIVA.
MED SÅ MÅNGA I KLUBBEN BORDE DET REGNA IN MATERIAL TILL REDAKTIONEN.
MEN TYVÄRR ÄR DET INTE SÅ,MEN SKÄRPNING,ALLT ÄR VÄLKOMMET
STORT SOM smått.MED HÄLSNING FRÅN REDAKTIONEN, TREVLIG SOMMAR!

PS. DET ÄR EN FÖRDEL FÖR OSS OM NI KAN SÄNDA MATERIALET PÅ DISK ELLER TAPE.VI RETURNERAR SEDAN DISKEN;TAPEN.

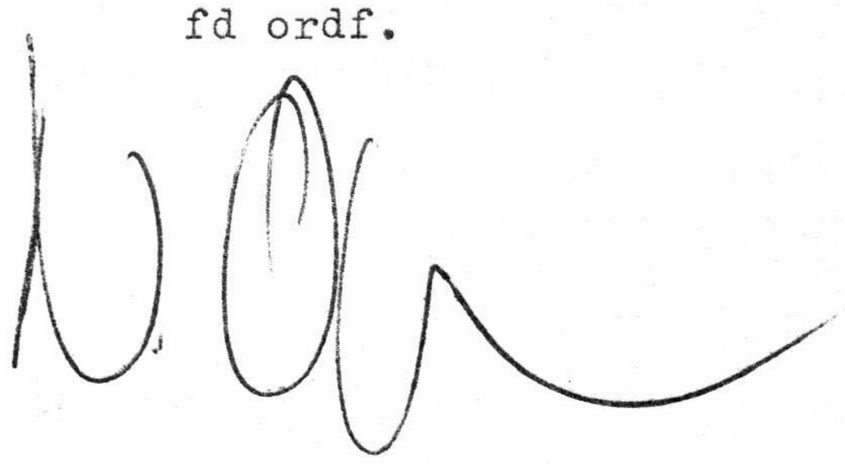
L E D A R E N

Tiden går fort, och vi har hunnit in i ett nytt decennium. Från vår lilla start har klubben vuxit till en rejäl sammanslutning av 6800-användare och mikrodator-entusiaster i allmänhet. Den gångna tiden har utnyttjats till att i demokratisk ordning styra klubben till den verksamhet som de flesta önskat.Kanske skulle man velat möta en kraftigare respons från medlemmarna i vissa frågor och eventuellt också en negativ sådan. Troligtvis har de frågor som varit på dagordningen inte varit av sådan betydelse att stormar behövt blåsa upp. En ny fråga som jag hoppas att varje medlem funderar över är det förslag som väckts om att införskaffa en klubbdator. I Sverige är det ju populärt med utredningar av alla de slag, och för att inte bryta den trenden har det på årsmötet beslutats om att tillsätta en kommitté. Deras uppgift blir att försöka förutsäga konsekvenserna av införskaffandet av en dator, samt hur densamma ska fungera och brukas. Inte minst den ekonomiska aspekten måste beaktas ur alla vinklar. Nyttjandegraden är likaså ett frågetecken.

I och med det gångna årsmötet har också en del andra förändringar inträtt. För att hålla en klubb igång fordras inte bara en målsättning, utan också ideer om hur man skall nå denna. De ideer som funnits har nu till en stor del förverkligats och det är dags för nya friska tag om båten ska glida framåt.

I samband med årsmötet valdes den nya styrelsen med suppleanter. Visserligen finns lite nytt blod med, men personligen hade jag hoppats på en kraftigare blandning. Redan vid de förberedande samtalen inför årsmötet lät jag meddela att jag inte önskade sitta kvar ytterligare en period som ordförande. Detta av flera skäl. Ett av de viktigaste är det ovan nämnda, men jag känner också att jag p g a ändrade arbetsförhållanden i mitt dagliga liv inte skulle ha kunnat få ett ordentligt tag om åran. Nu kommer jag visserligen ändå att finnas med i styrelsen, men klubban är överlämnad till fd vice ordföranden Tommy Bladh. I och med detta kommer jag inte heller att kludda på den här sidan och sänder över pennan till Tommy med en önskan att han får lika mycket skrivkramp som jag själv fått.
Tack för en trevlig tid och glöm inte att NI är klubben.

Tecknar Nils Ohlsson
fd ordf.



KUNDANPASSADE MIKROPROCESSOR SYSTEM FÖR STYRNING,ÖVERVAKNING ETC!

Ange önskad CPU och EPROM, samt antal EPROM, ROM, PIA och ACIA med önskade adresser. Ytterligare TTL-kretsar, OPTO-kopplare och reläer kan relativt prisbilligt integreras. Kortet levereras färdigt med flatkabel för anslutning, fullt bestyckat eller utan IC-kretsar. KYBERMATIC HB 0755-57854
Rabatt lämnas till PD68-medlemmar.

RADANNONSER
==========

KÖPES

Tangentbord,ASCI
Ted Johansson 013-151859

SÄLJES:

Minneskretsar typ 4060 (4k. dyn) 22-pinnars 35:-/st.
Minneskretsar lika 2114 (4k. stat.)
18-pinnars DIL 69:-/st.
Kontakta L O Fagerström
Tel h. 08-7616190
a. 08-7878191

Monterad och fullt fungerande MEK 6800 D2-kit inkl. all litteratur + strömförsörjning 1500:-
Erik Mårtensson
Drottninggatan 45 A
961 45 BODEN
Tel. 0921-11644

2708 EPROM fabr. MOTOROLA
pris: 59:50/st.
Bernt 08-448864

EPROM-kort typ CÅ-Elektronik inkl. EPROM 2708
Tommy R.
08-881683

TSC texteditor 100kr TSC textprocessor 200kr TSC assembler 150kr på 8" eller 5" disk. Compilerande Basic. 8" med assembler editor runtime för 6800 av fabrikat inclusive avancerat chackspel 50kr. SOFTWARE DYNAMIC med orginal manual allt för 1250kr.+Flex med extra utilitys 1702A 5kr st Eprom 156*8 2401 2k skiftregister 5kr st. Optokopplare Fcd 810 1kr Högnivålogik i mängder. 2102 A 450ns,A4,A6,N,minnen Tecken generatorer, 4096 dynamiska minnen.
Maskintid för 6809 finnes för uthyrning.

Erling Andeberg tel 08/626861.

6800 operativsystem Flex 2.0 på 5" SWTPC disk. pris 200:- 6800 TSC Basic på 5" SWTPC disk pris 300:- SWTPC CT-82 Editor för terminal CT-82 på SWTPC 5" disk. pris 100:- SWTPC PR 80 printer.80 kolum.120 tecken per sekund.Svenska tecken små och stora.Parallellport pris 5000:- Manualer medföljer till samtliga artiklar. Magnus Tauson tel 08/7589251

CT-64 TERMINAL.
Behöver Du en terminal och dessutom hör till dem, som aldrig ger upp hoppet att kunna lokalisera en felaktig IC-krets? I så fall kan Du förmånligt få överta min SWTPC CT-64 terminal, i befintligt skick. Har med oscilloskop konstaterat att det mesta fungerar, men pga bristande tid inte kunnat lokalisera den felande kretsen. 32 eller 64 tecken/rad,stora/små bokst., 2 pages, kontrolltecken osv. komplett med serieinterface.
Säljes Även:SWTPC kassetinterface AC-30 för två bandspelare.Aut. fjärrkontroll. Gunnar Lovius Tel 08/7771560

16k statiskt RAM-kort,samma som Åke Holm presenterat i R&T. Morgan Nöjdberg Villavägen 16. 830 44 NÄLDEN Tel 0640/205 45

```
          NAM       FLEXMOD
          OPT       PAG,NOG
*
START     EQU       $AD00     TRANSFER ADRESS   BUGGEN
**********************************                ======
          ORG       $BEA3
*
INVEC     FDB       $E1AC
OUTVEC    FDB       $E1D1
ACIA      FDB       $8008
TIMER     FDB       $8030
IRQ       FDB       $A000
SWI       FDB       $A00F
MON       FDB       $E0D0
PCV       FDB       $A03E
*
*
 ORG $AA6C
 FDB $AC10 YEAR
*
 ORG $AA73
 FDB $AC0E
*
 ORG $AA7A
 FDB $AC0F DAG
*
 ORG $AAD2
*
          FCC       'DATE (ÅÅ,MM,DD)? '
          FCB       4
*
 ORG $AC5F
 FCC 'What?'
 FCB 4
 FCC /Can't transfer/
 FCB 4
 FCC 'Not found'
 FCB 4
 FCC 'Disk error #'
 FCB 4
 FCC 'Drives not ready'
**********************************
          ORG       $B02C
*
          FCB       $5E       STORA Å,Ä,Ö ÄR OCKSÅ TECKEN SAMT ^
*
          ORG       $B034
*
          FCB       $7E       ÄVEN SMÅ å,ä,ö OCH ~
**********************************
          END       START
```

Från Anders Karlsson, Fasanstigen 40, 197 00 BRO har vi fått följande tips: Procedur vid ändring av inmatningsformat för datum i FLEX 2 resp FLEX 9till eropeisk standard.
FLEX 2 ------ AA6D (0E) 10 AA74 (0F) 0E AA7B (10) 0F samt textsträng från AAD2
FLEX 9 ------ CA5F (0E) 10 CA66 (0F) 0E CA6D (10) 0F samt sträng från CAD6
Ändra direkt på disketten med REPAIR.cmd föreslår Anders.
I detta sammanhang vill jag komplettera med en metod som ger automatisk dokumentation över alla ändringar, dessa sammanställs i samma assemblerfil som definierar IO-portar mm.

Tommy R

Jag har slagits av tanken
att någon undrar vad programmet
'BASFORM' som var listat i förra
MPU-laren gör för nytta.
Här kommer därför ett exempel
på bästa 'MÅLA-OM-DITT-KÖK'-vis:

FÖRE

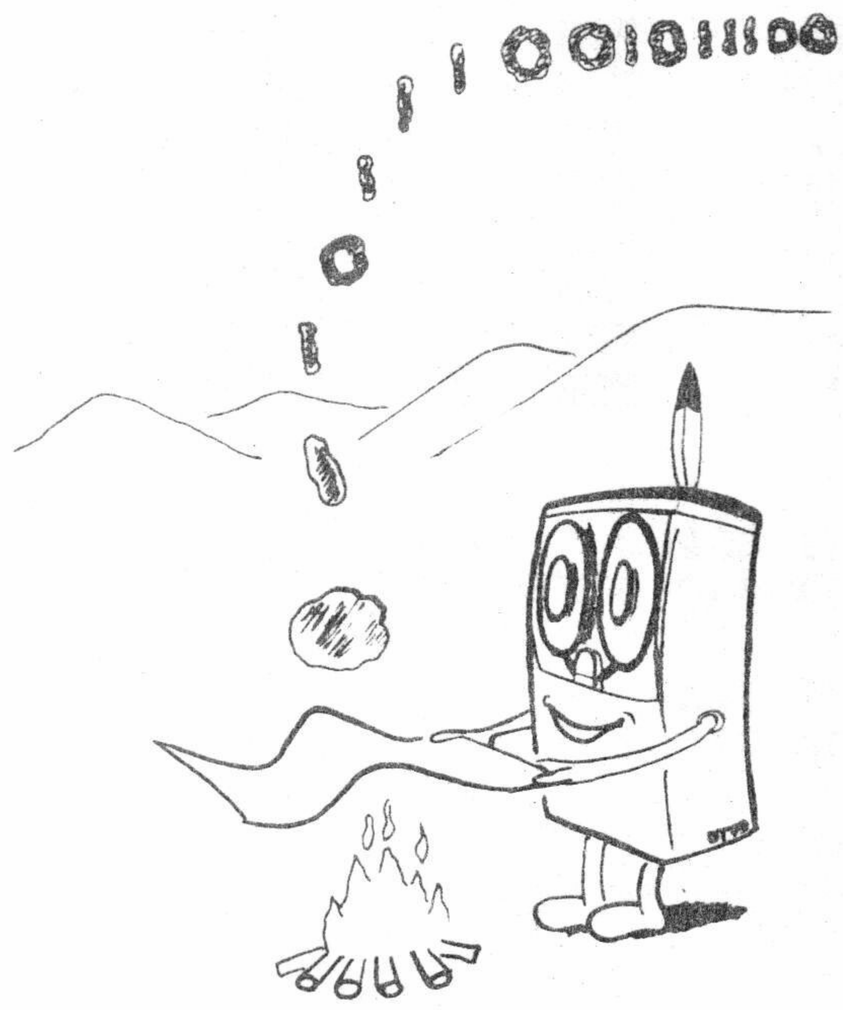
```
10 REM DEMO-SNUTT
20 A=B
30 B=C
40 C=A
50 FOR D=E TO F
60 FOR I=J TO K
70 PRINT "Jag har hakat upp mig"
80 INPUT X
90 NEXT I
100 NEXT D
110 END
```

EFTER

```
10 REM DEMO-SNUTT
20 A=B
30 B=C
40 C=A
50    FOR D=E TO F
60       FOR I=J TO K
70          PRINT "Jag har hakat upp mig"
80          INPUT X
90       NEXT I
100   NEXT D
110 END
```

Det senare blir ju mycket tydligare,eller hur?

PS. Kritik mot Demo-programmet mottages ej!
Hälsningar
Tommy R



På omstående sida har Tommy Bladh gjort en översättning och en beskrivning av TSCs Textprocessor.

Varje dag produceras det millioner ord av text. Det finns ord i dagstidningar, tidskrifter, flygblad, brev, dokument, kataloger och manualer som alla behöver organiseras före publiceringen. Det skulle förvisso vara en aldrig avslutande företagsamhet om all textformattering måste ske manuellt. Det skulle helt enkelt inte vara möjligt. Tack vare datorer och program kallade "textprocessorer" blir formattering av text en ganska trivial historia. Textprocessorn medger bekväm och exakt sid-layot och hantering. Den slutliga kopian blir extremt nätt och läsbar. En önskvärd egenskap hos varje tryckt sak.

Vad kan då göras med textprocessorer? Den enklaste funktionen åstadkommer exakt passform av sidan. Med andra ord, om texten skall ha 20 mm marginaler med sidans nummer centrerat vid botten av varje sida, och kanske en speciell rubrik i början på varje sida, kommer processorn automatiskt att ge detta. Förutsatt att den givits några lämpliga kommandon. Rad centrering är en annan finess som tillhandahålls. Ett flertal typer är tillgängliga, vilka innehåller vänster marginal justering (vänster kant rak, höger ojämn), höger marginal justering enbart (vänster ojämn, höger rak), vänster och höger (båda marginalerna är raka), och centrerad justering (båda kanterna ojämna men raderna är centrerade). En mångsidig textprocessor ger också möjlighet till speciella finesser såsom fotnots kontroll. TSC Text Processing System tillhandahåller alla de finesser som nämnts ovan.

För att ge lite insikt i hur man använder en textprocessor, kommer ett flertal exempel att visas där reportoaren i TSC:s Text Processor används. Kommandona som används av textprocessorer varierar från system till system men många används på likartat sätt. TSC använder en blandad metod med kommandon och text. För att ge processorn ett kommando är det nödvändigt att börja det i kolumn ett på en ny rad. Kommandot inleds med kontrolltecknet, en punkt (.). Detta är den metod som används av de flesta storskaliga systemen. NROFF inkluderat. Detta är ett program som skrivits vid Bell Labs. Det körs under de flesta operativsystemen, inklusive UNIX Time Sharing System, som är det system efter vilket TSC Text Processor är modellerat.

MILJÖ

Innan några specifika exempel visas, skall en beskrivning av "miljö" göras. Miljö är ett begrepp som beskriver de grundläggande sid och formatterings egenskaper som kommer att vara aktiva för såvitt inget annat specificeras. Den initiala, även kallade default miljön är mycket viktig. TSC:s Text Processor kommer att utföra följande helt utan någon kommando information: justering av vänster och höger marginal med en radlängd på 65 tecken. Sidans längd blir 66 rader. Andra initiala miljö egenskaper gör det möjligt att sända blanka rader till output, att en rad kan starta med en blank eller flera. Detta skapar en rad där dessa blanka behandlas som icke utfyllbara tecken av processorn varför de kommer att uppträda som blanka på utskriften och ej identifieras av processorn. (Detta betyder att dessa blanka ej kommer att "spridas ut" av justerings rutinerna.) Med miljön uppsatt enligt ovan är det fullt möjligt att ta vilken text fil som helst, utan inbäddade kommandon, och likväl få en känsligt formatterat utskrift. Detta är en viktig egenskap som många text processor konstruktörer förbiser. Miljön kan givetvis ändras och modifieras vid vilken tidpunkt

som helst med speciella kommandon och ge den önskvärda personliga utformningen av jobbet.

KOMMANDON

Låt oss titta på några speciella kommandon hos TSC processorn. Ett av de enklaste kommandona är det för centrering, .CE n, där n står för det antal rader som skall centreras inom den gällande radstorleken. För att använda detta kommando är det liksom för de övriga endast nödvändigt att placera det omedelbart före de rader som det skall styra. Exempelvis:

```
.CE 2
Att konstruera en Text
En introduktion
```

kommer att lista två rader prydligt centrerade på sidan. Man kan lätt inse att detta är mycket enklare än att manuellt försöka beräkna det rätta antalet mellanslag.

Den initiala miljön beskrevs nyss. Alla parametrarna kan enkelt ändras genom de kommandon som påverkar dem. Ett av kommamdona är .LN n, som sätter aktuell rad längd. För att erhålla en radlängd av 33, som i denna artikel, behövs bara kommandot:

```
.LN 33
```

Radlängden är nu 33 kolumner. En annan parameter som är lätt att sätta är sidans längd. Den definieras med kommandot .PL n, där n är det önskade antalet rader per sida. Några andra kommandon som ändrar miljö parametrarna är .FI och .NF vilka slår till resp ifrån utfyllnings mod. Utfyllning betyder att så många ord som rymms på aktuell rad klämms in på utmatad rad. Detta ger en rak vänster marginal och en något ojämn höger kant. Ingen utfyllnad helt enkelt kopierar inmatad rad till utmatad rad. Det skall observaras att "fill" måste vara påslagen för att marginal utjämning skall ske. Justerings egenskapen kan slås av genom att använda .NJ för "no justification" eller typ av justering kan slås till genom .JU x. "X" är val tecken för önskad typ av utfyllning. Det kan vara R för höger marginal (Right hand), C för centrerad, eller N för normal (vänster + höger). Anges inget tecken återgår processorn i tidigare mod. Vänster justering fås genom att slå till "fill" och justering från.

STORA/SMÅ BOKSTÄVER

En annam miljö parameter är styrning av stora och små bokstäver. Detta är en finess som TSC är ensamma om. Den gör det möjligt att få text med små bokstäver trots att man bara har ett tangentbord med upper-case, dvs stora bokstäver. Kommandona .CP och .NC slår till resp från denna mod. Om dennamod är till konverteras alla tecken till små såvitt de inte föregås av tecknet . Ett skall ses som ett tryck på skrivmaskinens skift tangent. En annan variant är av typen "skifta och lås". Genom att trycka ^ (pil upp) översätts alla efterföljande tecken till stora tills nästa ^ påträffas.

Det är ofta önskvärt att kunna få dubbel radframmatning. Exempelvis vid manuskript e.dyl. TSC processorn ger detta genom kommandot .MS n, där n är antalet blankrader. Default värde är 2 ifall inget värde anges. Vanlig radmatning kan återfås med kommandot .MS 1 eller .SS för "single spacing".

MARGINALER OCH INDENTERiNG

En annan grupp av kommandon har att göra med vänstermarginal och indentering. Med indentering menas inbrytning på raden. Vänster marginalen är normalt satt till 0 eftersom utmatnings mediet (printern) normalt håller med egen vänster marginal

(pappers positionering). En del applikationer kräver en större marginal. Därvid kan .LM n användas för att definiera om den till n extra blanka bred. Indentering likvärdigt med vänstermarginal kontroll med ett undantag, .LM n bevarar radlängden och flyttar helt enkelt raden n blanka högerut. .IN n å andra sidan reducerar radlängden med n kolumner för att bevara högermarginalen. Genom att sätta tillbaka indenteringen till 0 återställes den fulla radlängden igen. En annan form av indentering kan fås med "single line indent" kommandot .SI n. Detta är identiskt med indent utom i att automatisk återgång till 0 sker då raden utmatats. Man skall observara att kommandona för vänster marginal, indent och single indent är additiva. Om sålunda följande kommandon är sammanställda:

```
.LM 10
.IN 8
.SI 5
```

blir den resulterande raden föregången av 23 blanka, efterföljande kommer att föregås av 18 blanka förutsättande att inget annat .SI kommando ges.

FÖRSIKTIGHET

En liten notis för påkallande av uppmärksamhet angående vissa av processorns kommandon kan vara på sin plats. De flesta kommandon utför endast sin speciella uppgift men andra ger också en "line break". Detta betyder att den rad som håller på att sammanställas i rad bufferten tvingas till utmatning. Normalt behålls en rad i bufferten tills den specificerade rad längden har uppnåtts, vid vilken tidpunkt justering kan ske, beroende på tillslagen mod (även förutsatt att "fill" är tillslagen). Brytningen medför att den partiella raden utmatas utan att utfyllas, men med tillämplig justering. Detta är användbart för att starta nya stycken eller nya textblock. Några av de kommandon som ger en brytning är: Ö .CE, .FI, .NF, .IN, och .SI. I bland är det önskvärt att dessa kommandon inte ger brytning. Det kan man fixa genom att använda "no break" kontroll tecknet (:). Hittills har alla kommandon föregåtts av det normala kontrolltecknet, en punkt. För att erhålla en indentering av 10 utan att orsaka en brytning skall följande användas:

```
:in 10
```

Kolonet kan användas till alla kommandon, vare sig detta ger brytning eller ej.

BLANKA RADER

Det är ofta nödvändig att producera en sektion av en eller flera blanka rader. Space kommandot, .SP n, kan nyttjas för att utmata n antal blanka rader. Space kommandot ger också brytning. Om n inte ges kommer processorn att mata ut en blank rad. Ett krav kan vara att samtliga blanka rader skall rymmas på en och samma sida. Exempelvis för att ge plats för en bild eller ritning. TSC Text Procesor tillåter detta genom "spara utrymme" kommandot .SV n, där n är antalet behövliga tomma rader. Om det inte finns n rader kvar på sidan, matas inget utrymmet ut utan utskriften fortsätter och n sparas för senare användning. När nästa sida nås, kan "mata ut sparade utrymme" kommandot .OS användas. Då produceras det i sparade antal behövliga rader. En bekväm metod att ge .OS kommer att visas senare. Ett likartat kommando är "behöver rader" kommandot .NL n, där n är rad antalet. Detta kommando säger att det måste finnas n rader kvar på sidan och om så inte är fallet, byt sida. Detta är bekvämt när man vill hålla speciella text block samman (ej utspillda över sid gränser), eller för att inte starta ett nytt stycke längst ned på en sida om bara ett par rader återstår.

SIDO RUBRIKER

De kommandon som nu beskrivits kommer att ge mycket trevlig sid formattering. Om dessa var de enda tillgängliga skulle mycken tid och möda vara sparad. TSC Text Processor offererar emellertid många flera kommandon och möjligheter. En finess som ofta behövs i dokument och manualer är sido titlar. Det finns många sätt att åstadkomma rubriker men TSC processorn använder ett litet kommando som har utseendet:

```
.TL 'fält1'fält2'fält3'
```

där fält1 är vänster justerat, fält2 är centrerat, och fält3 är höger justerat. Vilken som helst eller alla av fälten kan finnas. En annan finess är att kunna skriva den aktuella sidans nummer i texten. Varhelst procent tecknet(%) förekommer kommer det att ersättas med sidans nummer. Ett par exempel kommer att klargöra användningen av titel kommandot.

```
.TL 'Huvud rubrik'
.TL ''Centrerad rubrik'Datum'
.TL ''-%-''
```

Den första raden kommer att vänster justera "Huvud rubrik" på sidan. Exempel nummer två centrerar "Centrerad rubrik" och höge justerar "Datum". Det sista exemplet visar hur sidans nummer skrivs centrerat mellan två bindestreck.

Nu skulle det smaka fågel om det fanns ett sätt att få rubriken (och kanske en del andra kommandon) att exekvera i toppen och/eller i botten på varje utmatad sida. TSC Text Processor erbjuder två avancerade möjligheter för att utföra denna sak: makros och traps. En makro är ett set av kommandon som grupperats samman och givna ett namn. När namnet senare efereras, utförs hela gruppen av kommandon. I korthet så resulterar detta i förmågan att skriva speciella program med användande av kommando uppsättningen hos processorn för att utföra special uppgifter såsom huvuden, stycken, special rubriker etc. En Trap (fälla) gör att man kan specificera en speciell rad på utskrifts sidan där en specifik makro skall utföras. För att lösa rubrik problemet enligt ovan, är det praktiskt att definiera två makron. En header makro och en fot makro. Ändamålet med headern är att få toppen på varje sida att se lika dan ut. Fot makron arbetar i botten på varje sida. Antag att det krävs att varje sida skall ha tre blanka rader följt av en centrerad rubrik och att botten på varje sida skall ha ett inom binde streck centrerat sidnummer. De följande makrona och trap placeringarna kommer att tillfredställa kravet.

```
.DM HD
:SP 3
.TL ''Sido Rubrik''
:SP 3
..

.DM FT
:SP 3
.TL'''-%-''
:PG
..

.AT 1 HD
.AT -7 FT
```

DM kommandot definierar en makro och den första listade i exemplet definierar en huvud makro med namnet HD. Huvud makcron kommer att ge tre blanka rader (utan att ge brytning, då : är givet), skriva en centrerad rubrik och slutligen skriva ytterligare tre blankrader utan brytning. Den sista raden i makro definitonen är ".." och är ett kommando för avslutning av makro definitionen. Den andra makro definitionen är FT och används för sidans fot. När den utförs kommer tre blankrader att matas ut, ett centrerar sidnummer att skrivas samt att sidbyte sker. Kommandot .AT används för att sätta trap läge..AT HD får huvud makron att utföras på rad 1 för varje ny sida. .AT -7 FT får fot makron

att utföras 7 rader från slutet på varje ny sida. Möjligheten att sätta trap lägen och definiera makron gör rubriker och fötter extremt enkla och effektiva.

ÄNDRINGSFÖRMÅGA

En av de viktigaste aspekterna med användandet av en text processor är möjligheten att göra några få mindre ändringar och vidklyftigt kunna förändra utseendet på den sluliga kopian. Tag som exempel att man i sista sekunden beslutar att det skulle se snyggare ut om varje sida inleds med fyra blanka rader i stället för tre. Om dokumentet prepparerades för hand skulle detta kräva en omskrivning av hela arbetet. Använder man däremot en liten text processor är allt man behöver göra en förändring av rad räknaren före varje rubrik. Med TSC Text Processor och dess förmåga att definiera makros, resulterar detta att endast en enda rad i hela textfilen behövs ändras. Rad nr två i huvud makron som fn. är ":SP 3" ändras till ":SP 4". En bagatell ändring och det önskade resultatet är uppnått! Man bör därför hålla i minnet att då man prepparerar dokument med en processor som tillhanda håller makro funktioner, skall de ofta använda kommando strängarna definieras i en makro för att enkelt medge små globala förändringar då så önskas.

VILLKORLIGA KOMMANDON

Det finns ännu mera avancerade egenskaper i TSC Text Processing System. En av dessa är förmågan till villkorlig exekvering av kommandon. Det finns fyra former för detta kommando:

```
.IF O .XX
.IF E .XX
.IF N .XX
.IF !n .XX
```

där O och E står för ODD (udda) och EVEN (jämn), N står för ett tal, ett tal register (förklaras strax) eller ett uttryck bestående av tal och talregister. Utropstecknet är negations operatorn (NOT) och .XX är ett kommando eller makro namn. Kommandot fungerar så här: OM (IF) villkoret är sant (sidan är udda eller jämn eller talet eller uttrycket är större än noll) utförs kommandot XX. Annars inte. Genom att föregå uttrycket med ! utförs XX endast om villkoret inte är sant (mindre än eller lika med noll). Den följande speciella huvud makro definitionen illustrerar detta kommando.

```
.DM HD
:SP 3
.IF O .TL '''Rubrik'
.IF E .TL 'Rubrik'''
:SP 2
..

.DM HD
:SP 3
.IF %-1 .TL ''Rubrik''
:SP 2
..
```

Den första huvud definitionen ger en höger justerad rubrik på udda numererade sidor och en vänster justerad på jämna sidor. Den andra definitionen kommer att skriva en centrerad rubrik på varje sida utom sida nummer ett, eftersom värdet på uttrycket kommer att vara noll när sido nummret är ett (kom ihåg att % representerar aktuellt sidnummer).

TALREGISTER

En annan egenskap i TSC processorn är möjligheten att använda tal register. Två typer existerar. En som tillåter att man läser och kan påverka vissa system parametrar. Exempelvis datum, sid nummer, aktuellt inbrytnings värde, vänster marginal, aktuellt kolumnvärde, rad på sidan och rad längd. Den andra typen är användar definierbar och kan behandlas

precis som man använder variabler i program. Tal register är de enkla bokstäverna A-Z och det redan introducerade procent tecknet (%). Flera andra funktioner är associerade med talregistren i TSC processorn. Dessa inkluderar auto inkrement (automatisk upp stegning), tilldelning av värden till registren, låta dem ingå i uttryck samt möjligheten att skriva ut vilket register som helst i stilarna: Arabisk, stora Romerska eller små Romerska siffror.

En del processorer, TSC's inkluderad, medger kommunikation mellan processorn och operatören under själva text behandlingen. Tre av dessa kommandon har följande format:

```
.ST
.TM valfri textsträng
.GI valfri textsträng
```

Det första kommandot kommer att avbryta processeringen och skriva STOP på operatörens terminal. Detta kan vara önskvärt ifall speciell hantering av papperet eller annan handpåläggning krävs. När processorn har stoppats kan den åter startas genom att man ger något tecken på terminalen. Dock ej S eftersom detta avbryter processeringen. Det andra kommandot sänder en "valfri textsträng" till terminalen. Det kan användas före STOP kommandot för att ge en instruktion till operatören. Det sista kommandot (Get Input) tar in en text sträng från terminalen och sätter in den i utskriften. Ett exempel när detta är mycket värdefullt är vid preparerandet av standard brev. Processorn kan begära in namn och adress från operatören via tangentbordet och automatiskt sätta in detta i texten.

OMSTYRNING AV TEXT

Ytterligare ett kommando skall slutligen beskrivas i denna introduktion, "omstyra text" kommandot. I bland är det praktiskt att spara undan påträffad text för senare bruk. Ett exempel är då man försöker skriva fotnötter. Det skulle vara fint om man omedelbart efter själva referensen även kunde skriva själva fotnotens text, men spara den för insättning i slutet av sidan. Den mekanism som medger det här sättet att arbeta kallas "omstyrning" och är bara tillgänglig på de mera komplexa text procesrna, såsom TSC's.
Två typer av omstyrning existerar vanligtvis:

```
.DI XX
.DA XX
```

där .DI instruerar processorn att omstyra efterföljande text till en lagrings area med namnet XX. Nästa kommando .DA omstyr och lägger till på slutet i text arean kallad XX. Under omstyrning sker all normal text behandling precis som vanligt, men i stället för att mata ut resultatet skrivs det in i en speciell area internt i processorn. Omstyrningen fortsätter tills ett omstyr kommando utan namn påträffas. För att återfå den omstyrda texten räcker det att anropa den med dess namn likt ett makro anrop.

MAKRO UPPSÄTTNING

Såsom en avancerad övning och demonstration av omstyrnings processen (likväl som av de många andra processor kommandona) kommer ett fullständigt set av makron för hantering av fotnoter att beskrivas. Läsaren bör ha i minnet att följande exempel är mycket komplexa. Flera läsningar kommer förmodligen att krävas för att få fullständig förståelse för funktionen.

```
.NR B 7
.DM HD
:SP 2
.IF %-1 .TL 'FOTNOT TEST'
:SP 2
.AU 1
```

```
.NR X 0
.NR W 0-#B
.IF #V .TR
.NS
..

.DM FO
.NR V 0
.IF #X .FT
.CH FO-#B
:PG
..

.DM BF
.DA TX
.EV 1
.IF !#+X-1 .SA
.DM EF
.BR
.EV 0
.DI
.NR W-#V
.CH FO #W
.IF #N-#P-#W .CH FO #N+1
..

.DM SA
---------------------
.BR
..

.DM TR
.BF
.NF
.FE
.FI
.EF
..

.DM FN
.DI FE
..

.DM FT
.EV 1
.NF
.TX
.RM TX
.DI
.FI
.EV 0
..

.AT 1 HD
.AT -#B FO
.AT -4 NM
.CH FO 70
.AT -#B FN
.CH FO-#B
.EV 1
.AU 1
.LN 55
.EV
```

Detta exempel är ganska likt det som givits i "NROFF Users' Manual" skriven av J. Ossanna, Bell Laboratories. För att använda dessa makron räcker det att sätta in deras definitioner i början av textfilen samt att omedelbart efter att en fotnot referns givits, anropa makro BF. Efter anropet skrivs texten i fotnoten och den avslutas med ett anrop av EF.

En beskrivning av makrona följer nu. Den första raden definierar tal register B och sätter det till 7. Detta register används för att specificera storleken (i rader) för botten marginalen. En huvud makro definition följer (HD) och medger ett flertal funktioner. Efter två tomma rader matas rubriken ut för såvitt det inte är sida ett (IF kommandot). Ytterligare två rader produceras och auto-increment värdet sätts till ett. Register X nollställs och används senare för att räkna antal fotnoter på aktuell sida. Därefter sätts W till positionen för botten marginalens trap. Den kommer att justeras om nödvändigt då fotnötter adderas. Kommandot IF #V kollar om det finns någon återstående fotnot text från föregående sida. Om så är fallet återprocesseras denna (register V innehåller rad antal av sista omstyrningen). Slutligen slås "ej blanka" moden till och undertrycker ev blanka som i onödan kanske skulle utskrivits vid sidans topp.

Fot makron FO nollställer omstyrnings räknaren V och kollar värdet i X. Om X ej är noll, vilket betyder att det fanns fotnoter på sidan, infogas makro FT. Foten återställs därefter till sin ursprungliga position med hjälp av Change (ändra) kommandot definierat av B. Det sista kommandot utför ett

sidkast. Makro NM skriver ett centrerat sid nummer i botten på varje sida.

Starta-fotnot makron BF inleds med en adderande omstyrning till arean med namnet TX. Miljön växlas, och om det är första fotnoten på sidan får makro SA skriva ut en rad understrykningar. Omstyrning av fotnot texten pågår tills makro EF anropas. Vid denna tidpunkt exekveras en "break" och den ursprungliga miljön återställs. Omstyrningen stoppas med DI kommandot. Register W uppdateras med antal omstyrda rader och trapen för fot raden är uppdaterad för att kompensera de adderade fotnot raderna. Slutligen tar man hänsyn till om fallet då de antal omstyrda raderna var så många att de flyttar fot trapen upp förbi aktiv rad. Då omsätts trap värdet till nästa rad. TR är ansvarig för omflyttning av de fotnot rader som inte rymms på sidan. Detta händer mycket sällan men kan inträffa då en fotnot är mycket stor och refereras nära slutet på sidan.

Makro FT används för att läsa tillbaka omstyrd text. Den växlar miljöer, sätter "no fill" mod och anropar TX vilket är den riktiga fotnot texten. TX tas sedan bort från makr listan, fill mod återställs, och miljöerna växlas. Den sista gruppen av rader i exemplet använs för att definiera trap lägena för de olika makrona. Huvudet sätts till rad ett, och NM sätts att exekveras fyra rader från sidans slut. Trapen för foten planteras vid -#B, därefter flyttad förbi sidans slut medan FN också placerats vid -#B. FO är därefter tillbaka flyttad till ursprungs stället. Detta medför att FO och FN placerats på samma rad, men trap FN kan bara inträffa om fot trapen flyttats upp genom att en fotnot påträffats. De sista raderna växlar till miljö ett och initierar denna för en radlängd av 55 och auto stegning med ett.

Miljöväxling som nämnts ovan är en egenskap som många av de större text processorerna (inklusive TSC's) har. Det tillåter att man växlar alla större viktiga parametrar på en gång.

DISK VERSIONEN

Ett avtal tilläggs kommandon finns för disk versionen an text processorn. Alla dessa kommandon är knutna till begreppet "data fil". Data filen består av ett antal "block". Varje block är uppbyggt av "items". Ett item (sv. ungefär "artikel") kan bestå av valfr text eller processor kommandon föjt av ett "end of item" tecken. Detta är initialt ett ">" men kan omdefinieras med .IC kommandot. Block slut anges med ett tomt item. Det anges med ">>".

Det finns processor kommandon för insättning av items i texten (.RI), hoppa över items (.NI), flytta till nytt block (.NB) samt för att öppna och stänga datafiler.

Ett exempel på block kan vara personuppgifter för en individ. Ett item kan då vara namnet,ett annat adressen, postadress mm. I en textmängd som formar ett standard brev kan man personligen adressera folk genom att låta text processorn läsa en data fil med person uppgifter och insätta dem i brevet.

```
.IC c
.OF NAME
.CF
.RI
.NI n
.NB n
```

Med .IC c kan man definiera om end of item tecknet. Öppna en data fil kan man göra med .OF NAME.NAME kan anges på kommado raden på vanligt FLEX sätt. Om man ej gör detta kommer text processorn att fråga efter fil namnet och man får slå in det

från terminalen. En öppnad fil kan stängas med .CF. Därefter kan en annan fil öppnas om så önskas. Läsning av items göres med .RI. Antal tecken läst är också tillgängligt i ett nummer register. Normalt kanske man läser items sekvensiellt,dvs en efter en. Med hjälp av .NI n kan man hoppa över "n" antal items. Att hoppa till nästa data block måste man göra med .NB n. Med .NI och .RI kan man ej förflytta sig till nästa block.

Disk version har ytterligare ett kommando .UL. Med detta kan man lägga ut "under line" tecken under efterföljande rad. Används för att stryck under exempelvis rubriker. Eftersom processorn använder back-space tecken för att backa får man se till att ens printer klarar av detta.

När man använder flex skivor kan alla definitioner av kommandon och makros läggas på en fil med namnet "MACRO.TXT". När textprocessorn startas kommer den att söka efter denna fil och läsa in innehållet. Detta betyder t.ex. att MPU-lar redaktionen kan ha en egen uppsättning kommandon anpassade efter den printer som man har medan jag själv kan ha en enklare variant, anpassad till CRT terminalen. Vi slipper således gräva ned detta i de individuella text filerna.

Vid FLEX anropet av processorn kan de textfiler som skall behandlas anges på kommandoraden separerade av komman. Anger man "*" frågar processorn efter filnammet när det är dags att ta den filen. Då kan man exempelvis byta flexskiva e.dyl.

Vissa printrar kan kommenderas att skriva med dubbel höjd på tecknen. Ett kommando för detta finns i TSC's text processor.

SAMMANFATTNING

Det som nyss beskrivits är endast tänkt att vara en introduktion till text processering. Många kommandon och finesser har hoppats över. De som inkluderats är de mest använda och de som ger mest kontroll möjligheter och flexibilitet. Genom de fantastiska kombinations möjligheterna som finns i makro tekniken är TSC's text processor mycket mer än bara de enskilda kommandona.

Hela den här artikeln är skriven med hjälp av TSC Text Processor. De planer vi har är att "sätta" hela MPU-laren med dess hjälp. Jag måste säga att det var oerhört mycket enklare att utforma innehållet i artikeln med processorn i åtanke. Texten är inlagd med TSC's Text Editor under FLEX 2.0. Jag har aldrig som tidigare behövt bekymra mig om orden skall rymmas på spalten. Har jag råkat stava fel har det varit enkelt att direkt korrigera misstaget. Korrektur rättning är ingen match eftersom editorn är synnerligen kraftfull.

Man får dock tänka på några småsaker, åtminstone om spaltbredden är så pass liten som våra 33 tecken. Långa ord bör i görligaste mån undvikas tillsammans. Likaså bör ett mellanslag ske efter komma tecken. Allt detta för att ge processorn en rimlig chans att sprida texten snyggt.

TSC Text Processor System finns i både kassett och disk version. Den säljs av bla. Swedish Electronic. Priset är c:a 280 Skr för kassett versionen och 470 Skr för disk versionen. Detta gällde FLEX2. I 8" version kostar den 540 Skr. Priserna avser också 6800.6809 versionerna var ännu ej prissatta. Notera att det ej kommer att finnas en kassett version för 6809.

TBl

FLEX FÖR EXORCISERN

Nu kan man även köra FLEX på Motorolas EXORciser system med EXORdisk II eller III. TSC har annonserat (jan 80) att man har en version av FLEX för detta ändamål. Ingen modifiering av hårdvaran behövs göras. Möjligtvis med undantag av minnes positionering. Man bootar som vanligt från en disk. Men FLEX i stället för MDOS. Priset för EXORciser FLEX inklusive disk editor, assembler och en full uppsättning manualer är $150. Övrig program support är debug paket, text processor, två snabba BASIC tolkar, sort/merge mm. En 6809 version är under fullbordande.

UNIX OCH FLEX - NY DOS

TSC håller på och avslutar framtagandet av ett helt nytt operativ system kallat UniFLEX. Ideerna har man fått främst från två andra operattiv system, FLEX och UNIX. FLEX är TSC själva föräldrar till medan UNIX kommer från Bell Labs. UniFLEX bibehåller flexibiliteten och användar enkelheten som kännetecknar FLEX samt inkorporerar några av de allmänt accepterade strukturerna hos UNIX.

UniFLEX är ett sant multi tasking, multi user operativ system. Det supportar ett hierarkiskt filsystem vilket tillåter filstorlekar upp till en miljard bytes och disk kapacitet på över åtta miljarder bytes. Alla filer är fullt skyddade. En användare kan läsa, skriva eller exekvera skyddade filer på individuell basis. All system I/O är oberoende av enheter eftersom I/O enheter och filer hanteras på ett likartat sätt. Varje kombination av avbrotts drivna enheter kan anslutas till systemet. En konfiguration kan ha en nittio megabyte disk, en trettio megabyte disk, två åtta tums floppy, en parallell driven printer samt sex användar terminaler.

Ett exekverande jobb (task) kan initiera ett annat jobb på asynkront sätt. Kommunikation mellan jobb supportas också. Jobb swapping kan användas på de system som innehåller en lämplig swapping enhet. Swapping betyder att jobb kan flyttas från primärminnet till yttre minne och vice versa. Detta kan inträffa när ett jobb av högre prioritet behöver plats i primärminnet eller då jobbet väntar på yttre händelse. Exempelvis I/O från disk eller terminal. För realtids applikationer finns möjligheten att låsa ett jobb i primär minnet.

En användare får sin kontakt med systemet huvudsakligen via kommando språket. Med detta matar man in direktiv till systemet. Det supportar en vid variation av egenskaper inklusive parameter överföring, kommando variabler, kontroll flödes primitiver samt olika typer av namn jämförelser. Input till kommando språket kan komma från terminalen eller från vilken fil som helst (inklusive andra enheter). Detta tillåter att väldigt komplexa kommando filer skapas och körs enligt önskan.

UniFLEX har designats med inriktning mot de större mikrodator systemen. TSC anser att små system förstjänar små kompakta operativ system (som FLEX), medan större system fordrar mycket större sofistikation. Man säger dock att FLEX kommer att sopporteras till 100% efersom den lilla mikrodatorn kommer att dominera som system under avsevärd tid. UniFLEX är för den kommande generationen av mikrodatorer. System med massiv bestyckning av primär och sekundär minne. Det kommer att bli det operativsystem som TSC inte bara kommer att tillhandahålla för 6809 system, utan också alla 68000 baserade maskiner.

Maskinvaru krav

UniFLEX kommer att kräva ett system baserat antingen på 6809 eller 68000. Inget försök till 6800 version kommer att göras. Centralenheten måste ha utökad adressering (minimum 19 adress ledningar) och någon form av minnes segmenterings hantering. Funktionen måste kunna avdela segment ej större än 4k över hela cpu området. Operativsystemet kommer att ha multiple mapping register funktioner likväl som hårdvaru skydd av tilldelat minne. Denna funktion är dock ej nödvändig. Minumum 96k RAM kommer att erfordras.

All system I/O enheter måste vara avbrotts drivna. Terminaler kan köras antingen från en ACIA type kontroller eller genom en I/O buffer processor. Åtminstone måste en 8" single density flex skive enhet finnas. Drivenheten måste vara DMA (direkt minnes access) kontrollerad och avbrotts producerande. Denna drivenhet kommer att användas för program transport och system boot. Varje annan typ av disk controller kan adderas till systemet, förutsatt att den är av DMA typ.

System support

Grundversionen av UniFLEX kommer att inkludera operativ systemet, ett flertal system utilities (hjälp program), text editor, makro assembler och program för system generering. Program för underhåll av systemet kommer att finnas tillgängligt som option (extra tillbehör). Varje kopia som säljs av UniFLEX kommer att öronmärkas genom att innehålla en enda cpu licens samt ett unikt serie nummer. Varje yttrligare mjukvara som inköps kommer att kräva ett matchande serie nummer för att fungera riktigt. Det här är ett nytt och intressant initiativ för att få bukt med piratkopieringen. Återstår att se hur heltäckande det kommer att bli. De som köper sytem underhåll kommer att få uppdateringar på 8" flex skiva. Detta kommer att vara fixade 'buggar' och förbättringar i systemet. De som ej köper underhåll kommer endast att informeras om fel under en period av 30 dagar.

För närvarande befinnersig en hel del support mjukvara i utvecklings skedet. I detta ingår en C kompilator, Pascal, Text processor system, debug paket och en Basic compilator som är kompatibel med TSC Extended Basic. Många andra program funderar man på. Bl.a. Fortran, RPG/II och ett data bas hanterings system.

TSC säger som slutkläm i sitt meddelande att man inte vill påstå att UniFLEX är DET SLUTLIGA operativ systemet. Men man tycker att UniFLEX representerar 'state-of-the-art' bland operativ system. Det innehåller många finesser från 'stora system' utan att man förlorar vänligheten hos det lilla systemet.

NY SWTPC ASSEMBLER

SWTPC annonserar att deras 6809 Resident Optimizing Assembler finns att allmänt tillgå. De säger att den tidigare har använts flitigt vid program utveckling inom det egna företaget och är en stark förbättring över vad som tidigare funnits att tillgå. Assemblern skapades med tanke på att understöda modulär block strukturerad programmering. Den skall innehålla en massa finesser som gör det lättare att snabbt skapa felfria program med ett minimum av möda (sug på den karamellen !). De viktiga egenskaperna inkluderar valfri längd på symboler, procedurer och data strukturer, lokala och allmänna register, multipla program räknare, automatisk optimering av hopp instruktioner, nästlade inkluderingar av biblioteks filer, kraftfull fel

kontroll och mycket mera.

Assemblern går att köra på vilken SWTPC 6809 disk baserat dator system som helst. Minst 24k minne ska man ha. Program disketten innehåller den residenta 6809 assemblern komplett med en overlay för symbol tabell formatteraren samt utility program för konvertering mellan binär filer och pappers tape. ASM09 disken finns som både 5" och 8" FLEX diskett.

Överst Ser vi råtext med inbäddade styrkommandon. Inlagt från tangentbord med endast stora bokstäver.

Under Ser vi samma information men efter behandling med textprocessorn.

TB1

TEST OF SEVERAL
PROCESSOR COMMANDS

This example shows how commands and text can be intermixed for later processing by a text processor. The example started by centering two lines followed by a single indent to signify the start of a paragraph. The capitalization mode is on and the upper case shift characters are being used.

The adjust mode was just changed to centering as well as a line length of 45. The left margin was set to 10 to give a nicely centered narrow line. Special effects like these are easily accomplished.

The parameters were just switched back so the line appearance will be restored. This is a short example but should show how the commands can be integrated with the text.

```
.CP
.SP 2
.CE 2
↑TEST OF SEVERAL↑
↑PROCESSOR COMMANDS↑
.SP
.SI 5
@THIS EXAMPLE SHOWS HOW COMMANDS AND TEXT CAN BE INTERMIXED
FOR LATER PROCESSING BY A TEXT PROCESSOR.
@THE EXAMPLE STARTED BY CENTERING TWO LINES FOLLOWED
BY A SINGLE INDENT TO SIGNIFY THE START OF A PARAGRAPH.
@THE CAPITALIZATION MODE IS ON AND THE UPPER CASE SHIFT
CHARACTERS ARE BEING USED.
.SP
.LM 10
.LN 45
.JU C
@THE ADJUST MODE WAS JUST CHANGED TO CENTERING
AS WELL AS A LINE LENGTH OF 45.
@THE LEFT MARGIN WAS SET TO 10 TO GIVE A NICELY
CENTERED NARROW LINE.
@SPECIAL EFFECTS LIKE THESE ARE EASILY ACCOMPLISHED.
.SP
.LM 0
.LN 65
.JU N
@THE PARAMETERS WERE JUST SWITCHED BACK SO THE
LINE APPEARANCE WILL BE RESTORED.
@THIS IS A SHORT EXAMPLE BUT SHOULD SHOW HOW THE
COMMANDS CAN BE INTEGRATED WITH THE TEXT.
```

INTERFACE FÖR IBM SELECTRIC

Förr eller senare kommer man att vilja ha en utskrifts möjlighet i sin datoranläggning. Därvid hamnar man ofelbart i en besvärlig sits. För då måste man välja, vilket som bekant alltid är knepigt. Man måste (med plånboken i bakhuvudet), bestämma sig för om man vill skriva snabbt eller skönt. Nu tänker vi dessutom på något som kan ta minst A4. Planerar man att skriva verklig skönskrift, brev, rapporter e.dyl, kan man nog fundera ett par gånger på en begagnad IBM Selectric. Det finns en typ som brukar kallas BCD maskin och har gått som konsol skrivare. Den har den välkända kulan med alla tecken på, tangentbord samt inbyggt reläpaket i skrovet. Den kan också utrustas med pigg vals för kedje format på papperet. En nackdel med den här typen av skrivare är dess snabbhet. Dvs brist på sådan. Max 15 tecken i sekunden kan du räkna med. Sedan tar vagnreturen en avsevärd tid. Slamrar gör den också. Det fina är dock att man kan få skrivmaskins kvalite på det utmatade. Perfekt om man använder en text processor som skriv hjälp. Priset brukar vara runt 1000 kr för en begagnad BCD maskin.

Den här typen av maskin finns i två sorter. BCD och Correspondence maskin. Skillnaden är främst om maskinen även skall användas för input. BCD maskinen har då tecknen lagda på ett visst sätt samt den interna wirningen utförd så att en 6-bit kod direkt kommer ut på AMP kontakten där bak. Den kvicktänkte inser då genast att med 6 bitar klarar man inte alla tecken. Mycket riktigt. På kulan finns i stort sett bara stora bokstäver (oskiftat), siffrorna samt de vanligaste special tecknen (+- etc). Denna begränsning hänför sig till maskinens tidigare uppgit. Correspondence maskinen har ingen sådan intern 'avkodning'. All avkännings kontakter går ut p baksidan. Om man bara skall h maskinen till utskrift spelar de ingen roll. I båda maskinern har man samma kontroll öve kulans positionering. Själ skriver jag ibland med e skrivmaskins kula i BCD maskinen Så länge datorn sköte utmatningen spelar detta inge roll. Tangentbordet kan dock int användas eftersom tecknen int ligger på samma plats på kulorn (minns det jag sa om stor bokstäver i oskiftat läge). Äve detta kan lösas. Genom at flytta runt de armar som sty kodningen rent mekansiskt kan ma få allt att stämma. Fråga Gunna Lovius, han har gjort det.

Jag har nöjt mig med att låt IBM:en vara skrivare. Hårdvara har därvid blivit mycket enkel Uppgiften för den är endast at kunna manövrera 14 reläer sam att medge test på READY från IBM Vidare vill jag kunna välj mellan två kulor. En test til behövs således. Allt det här löstes enklast med en MC6821 PI samt två relädrivar kapslar frå RIFA. Dessa kan i och försi ersättas med darlington kopplad transistorer. Men har ma motvilja mot koppla och löda s har man. Mot databussen ligge PIA:n. Varje sida driver sj ingångar på relädrivaren Eftersom de sju reläerna som sty kulan och de sju som kontrollera LF, CR BS o.dyl ej dra samtidigt, skulle man kunn multiplexa och spara PIA trådar Men som sagt, varför löda onödan. Bit sju på respektiv sida programmeras som ingångar Den ena går till en switch oc anger vilken av de fn. tv kulorna som sitter i. Den andr är ett OCH villkor från tes kontakter inne i skrivmaskinen In på PIA kommer ett READY/BUS villkor. Med detta sty programmet maskinen med optima snabbhet. Hårdvaran, dvs PIA relädrivarna, en transistor, diverse kondingar och motstån samt en lysdiod, har monterats p

ett labb kort från CÅ-Elektronik som heter CÅ-8091. Det är ett litet periferi kort med en PIA plats samt utrymme för montering av valfria IC kapslar. En massa luft blir över på kortet varför jag förmodligen kommer att sätta dit en egen processor (6802) och låta den få ett standard V24 (RS-132) snitt. Då kan man ansluta IBM:en till vilken serie port som helst. Just nu drivs magneterna från ett separat labb aggregat.

Jag tror nog att program listan skall vara tillräckligt självförklarande för att förstå funktionen. Men OK..vi tar det lite löst här ändå. När ett tecken kommer till rutinen OUTCHA sker följande:

Man kollar först om det är line-feed. Om föregående tecken var CR ignorerar man detta LF. Detta beror på att IBM ger en automatisk index (LF) vid vagnretur. Annars går man till nästa steg. Där jämförs vissa kontrolltecken som har egna specifika magneter mot en tabell. Hittar man tecknet i CONCHA ligger reläkoden som 1,X. Jag har lagt in automatisk nedgång till oskiftat läge vid tecknen CR och LF. Det är nämligen ganska tråkigt att få maskinen låst i skiftat läge om man vill skriva för hand. Kom i håg att shift är bistabilt styrt av två magneter. Nåväl, var det inte heller ett kontroll tecken av ovanstående typ, kollar man att det ej ligger under hex 20 (SP). I så fall behöver vi inte göra något. Nästa steg blir annars att addera ASCII tecknet minskat med hex 20, till en pekare i en tabell. Denna tabell är organiserad i ASCII ordning. I varje position finns en magnet kod för att skriva motsvarande tecken. Det är sju magneter som kan behöva manövreras för att vicka (tilt) och snurra (rotate) kulan till rätt position. Om tecknet måste skiftas är högsta biten satt i tabell ordet. Genom att koderna ligger i en tabell kan man byta kula genom att peka ut en annan tabell. I programmet ligger fn två kulor. Då vissa tecken saknas på kulan kan man låta programmet utföra vissa jippon. Ett saknat semikolon skrivs exempelvis som punkt, back-space och kolon. Då BCD kulan bara har små fyrkanter i stället för skiftade bokstäver, ligger samma tecken i tabellen både oskiftat som skiftat.

Innam man drar några magneter kontrolleras att maskinen är READY. En etta på PA7 ger ledig status. För att plocka bort kontakt studsar, måste programmet testa en READY 255 ggr i rad innan vi är nöjda. Därefter dras magneterna. Efter detta får rutinen BUSY se till att villkoret READY försvinner. Annars skulle datorn hinna mata ut nästa tecken innan ledig kontakten hunnit gå ifrån. När READY försvunnit behöver inte magneterna dras längre. PIA utgången nollställs. I BUSY rutinen finns ytterligare en tidsövervakning. Om villkoret ej kommer skulle magneterna ligga aktiverade i evig tid. För att undvika små rökpuffar sker en ny initiering (nollning) av PIA efter extra lång tid (X-reg loop).

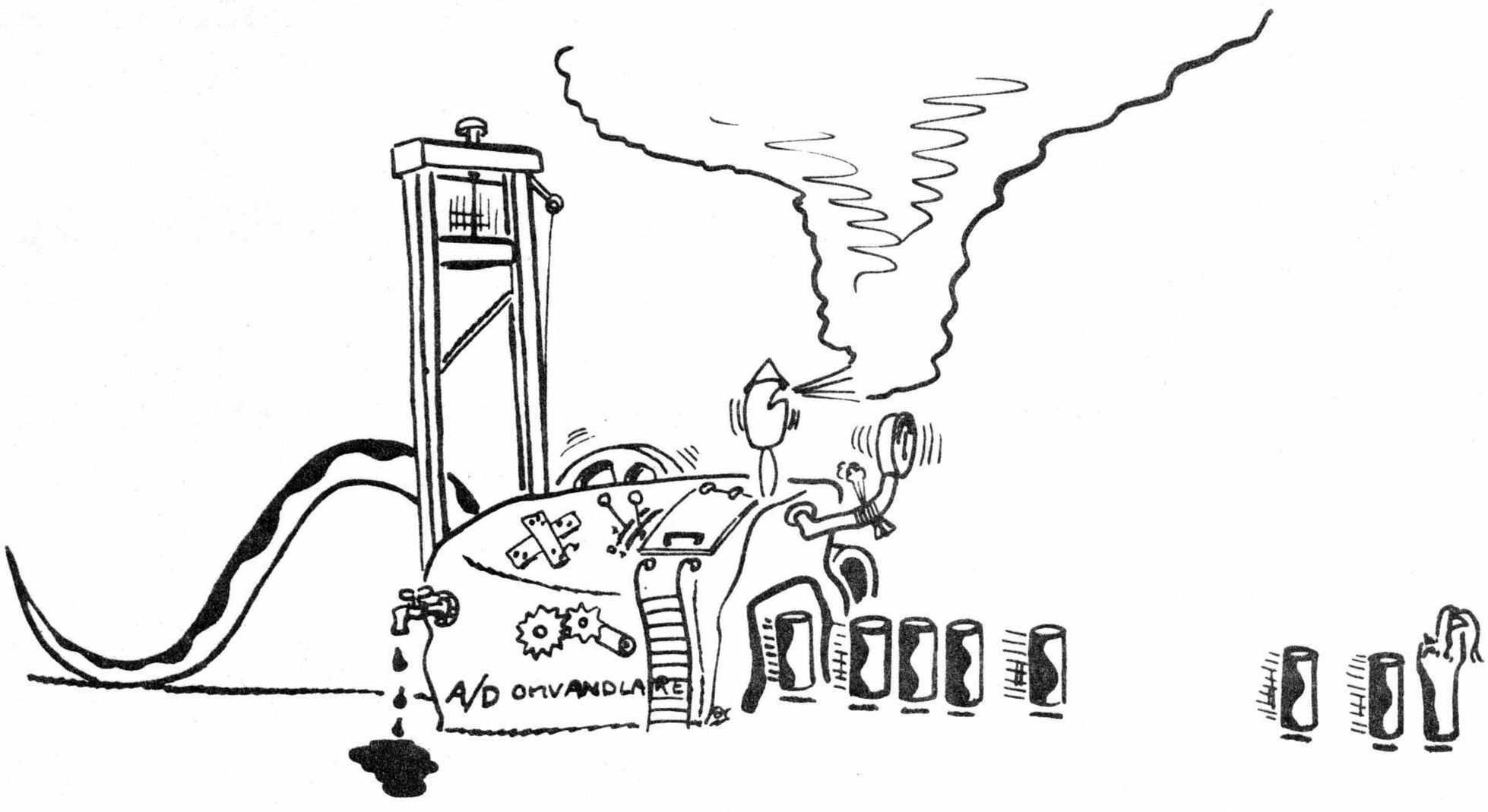
I slutet på programmet ligger två rutiner för uttestning av nya kulor. Den första mottar ett valfritt ASCII tecken från tangent bordet och skall skriva ut det på normalt sätt. Den andra anropas med hex code och ställer direkt ut kulans reläer. Enkelt att skicka in koderna hex 01-7F i dels oskiftat dels skiftat läge och se vad som skrivs. Allra sist ligger de vektorer som FLEX2 behöver för att kunna använda detta program som en PRINT.SYS rutin.

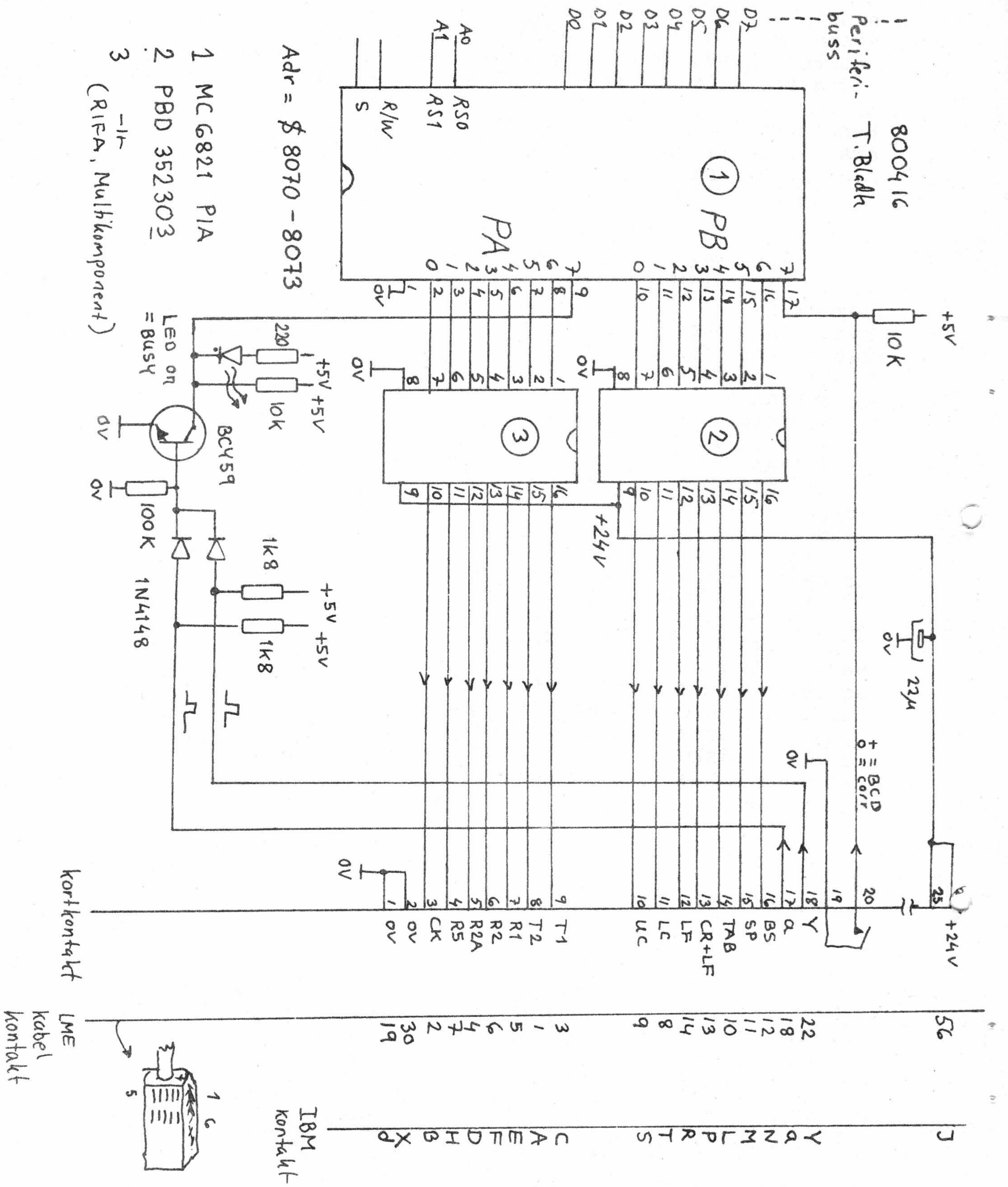
Observera att inga större kod optimeringar har försökts göras. För FLEX2 laddar denna PRINT.SYS i DOS arean för FLEX9 ($C100). Vid körning av FLEX9 laddar den i stället i FLEX2's DOS area ($A100). Alla register förblir

orörda. En liten tråkig grej återstår dock. Vid RESET av datorn återställs även PIA:n. Detta gör alla periferi ledningar till ingångar (ettor). Resultatet blir att alla reläer drar samtidigt. Den bästa medicinen mot detta är att initiera PIA'n i monitorn (TBUG etc).

I ritningen över interfacet finns IBM beteckningarna för de magneter och kontakter som nyttjats. Även den långa LME kabel som följde med maskinen finns dokumenterad med motsvarande AMP kontakt beteckning. Denna maskin hade gemensam plus i sin wirning. Ett relä dras därid genom jordning. Vissa maskiner, förmodligen alla correspondence maskiner har genmensam jord på reläerna. Då får man skicka in ström genom pnp transistorer. Kolla upp detta !!! Min BCD maskin går med +24V. En correspondence maskin har förmodligen +48 volt. Magneterna verkar dra c:a 125 mA/st. Ett aggregat som kan ge 1A med några buffliga kondensatorer mot magnet sidan borde räcka. Alla magneter drar ju gudskelov inte samtidigt. Dessutom är de inte dragna 100% i tid heller.

TBl




800416
T. Bladh
Periferi-buss
Adr = $ 8070 - 8073
1 MC 6821 PIA
2 PBD 352303
3 -"- (RIFA, Multikomponent)
PA
PB
LED on = BUSY
BCY59
1N4148
+5V
+24V
10K
100K
1k8
220
22µ
+ = BCD
o = corr
kortkontakt
LME kabel kontakt
IBM kontakt

MOTOROLA EXORset 30



Det nya utvecklingssystemet för M6809 – marknadens kraftfullaste 8-bitars MPU.

EXORset är ett kompakt utvecklingssystem, helt kompatibel med EXORciser, EXORterm och Micromodules – Motorola's kortsystem. Den kan tillsammans med Micromodules användas som centralenhet i t ex styrsystem för verkstadsindustrin. Den kraftfulla basicen, dubbla "floppyn" och det låga priset (ca 30.000:-) är ytterligare argument för EXORset 30.

Begär ytterligare information från Distributören!

MOTOROLA Semiconductor

DISTRIBUTÖREN

08/13 21 60

INTERELKO AB
Box 32
122 21 ENSKEDE

```
                    *
                    * PRINT.SYS FOR IBM SELECTRIC BCD TYPEWRITER
                    * WITH BCD/CORRESPONDENCE BALLS
                    * REV C1 1980-04-03 T BLADH
                    *
                    * HARDWARE:
                    * ---------
                    * PIA AT $8070 WITH :
                    *   PA0 CK  MAGNET
                    *   PA1 T2  MAGNET
                    *   PA2 T1  MAGNET
                    *   PA3 R5  MAGNET
                    *   PA4 R2A MAGNET
                    *   PA5 R2  MAGNET
                    *   PA6 R1  MAGNET
                    *   PA7 READY TEST
                    *
                    *   PB0 UC  UPPER CASE MAGNET
                    *   PB1 LC  LOWER CASE MAGNET
                    *   PB2 LF  (INDEX)    MAGNET
                    *   PB3 CR  (CR+LF)    MAGNET
                    *   PB4 TAB TABULATOR  MAGNET
                    *   PB5 SP  SPACE      MAGNET
                    *   PB6 BS  BACK-SPACE MAGNET
                    *   PB7 TEST FOR BCD/CORR BALL
                    *                    1    0  LEVELS
                    *
                    * DECLARATIONS
                    * ------------
                    *
8070                PIA     EQU     $8070       I/O PORT
8070                IBMDAT  EQU     $8070       PRINT CHAR PORT
8072                IBMCON  EQU     $8072       CONTROL CHAR PORT
0002                LOWCAS  EQU     $02         LOWER CASE MAGNET
0001                UPPCAS  EQU     $01         UPPER CASE MAGNET
000D                CR      EQU     $0D         CR CODE
000A                LF      EQU     $0A         LINE FEED CODE
0008                BS      EQU     $08         BACK-SPACE CODE
                    *
                    * START OF PROCESSING
                    * -------------------
                    *
C100                        ORG     $C100
                    *
                    * SUB ( INIT )
                    *
                    * ARRANGE THE MC6821 PIAA AS TWO OUTPUT
                    * 7 BIT RELAY DRIVERS ( BIT 0-6 )
                    * AND TWO TEST INPUT BITS ( BIT 7 )
                    * INITIATE IBM IN LOWER CASE
                    *
C100                INIT    EQU     *
C100 CE 80 70               LDX     #PIA
C103 6F 01                  CLR     1,X         SET TO DATA DIR REG
C105 6F 03                  CLR     3,X
C107 86 7F                  LDA A   #$7F        7 OUTPUTS
C109 A7 00                  STA A   0,X
C10B A7 02                  STA A   2,X
                    *
```

```
                                LDA A  #4          CHANGE TO DATA
                                STA A  1,X
                                STA A  3,X
                        *
                                LDA A  #LOWCAS     START IN LOWER CASE
                                STA A  CASE        REMEMBER PRESENT CASE
                                STA A  LASTCH      LAST CHAR NOT CR
                                STA A  2,X         PULL MAGNET
                                JSR    RE10        USE AS DELAY ONLY
                                CLR    PIA+2       RELEASE MAGNET
                                RTS
                        *
                        * SUB ( OUTCHA )
                        *
                        * ROUTINE FOR PRINTING ONE ASCII CHARACTER
                        * ACCEPTS CODES ( 00 - FF )
                        *
C124                    OUTCHA  EQU    *
C124 84 7F                      AND A  #$7F        BIT 7 IS NO-NO
C126 FF C1 FC                   STX    XSAVE       PUSH REGISTERS
C129 36                         PSH A
C12A 37                         PSH B
                        *
C12B 81 0A                      CMP A  #LF         TRAP CR+LF
C12D 26 0F                      BNE    OUTC05
                        *
                        * IBM PERFORMES AN INDEX (LF) WHEN
                        * CR MAGNET IS PULLD. IGNORE THE LF
                        * IF PRECEEDED BY A CR.
                        *
C12F C6 0D                      LDA B  #CR
C131 F1 C1 FF                   CMP B  LASTCH
C134 26 08                      BNE    OUTC05      IF NO PRINT LF
C136 B7 C1 FF                   STA A  LASTCH      UPPDATE
C139 BD C1 DC                   JSR    READY       WAIT UNTIL CR IS DONE
C13C 20 4A                      BRA    EXIT
                        *
                        * CHECK IF CONTROL CHARACTER
                        *
C13E B7 C1 FF           OUTC05  STA A  LASTCH      REMEMBER
C141 CE C2 00                   LDX    #CONCHA     CONTROL CHARACTER TABL
C144 A1 00              OUTC10  CMP A  X           MATCH ?
C146 27 6B                      BEQ    FOUND
C148 08                         INX
C149 08                         INX
C14A 8C C2 0A                   CPX    #CONEND     CHECKED ALL
C14D 26 F5                      BNE    OUTC10      TRY NEXT
                        *
                        * TRAP REST OF UN NON-PRINTABLE CHARS
                        *
C14F 81 20                      CMP A  #$20        (00-20)
C151 25 35                      BCS    EXIT
                        *
                        * TRY ASCII TABLE(S)
                        * TEST WHAT BALL WE WOULD LIKE
                        *
C153 7D 80 72                   TST    IBMCON
C156 2B 05                      BMI    BCDTAB      1=BCD, 0=CORR
C158 CE C2 49                   LDX    #CORCHA-$21 PRESTIGE ELITE 72 BA
C15B 20 03                      BRA    OUTC15
```

```
120                     *
121  C15D CE C1 E9  BCDTAB  LDX     #BCDCHA-$21 BCD 6400 BALL
122  C160 4C        OUTC15  INC A
123  C161 08        OUTC20  INX                 X = X + A
124  C162 4A                DEC A
125  C163 26 FC             BNE     OUTC20
126  C165 A6 00             LDA A   X           APPROPRIATE MAGNET CODE IS FETC
127  C167 27 1F             BEQ     EXIT        IF ZERO - DONT BOTHER
128                     *
129                     * CHECK IF CASE MUST BE SHIFTED
130                     *
131  C169 2B 0C             BMI     OUTC30      BIT 7 =1 IF UPPER CASE
132  C16B C6 02             LDA B   #LOWCASE    ARE WE IN LOWER CASE
133  C16D F1 C1 FE          CMP B   CASE
134  C170 27 14             BEQ     OUT
135  C172 F7 C1 FE          STA B   CASE        NEW CASE
136  C175 20 0A             BRA     OUTC40
137                     *
138  C177 C6 01     OUTC30  LDA B   #UPPCAS     ARE WE IN UPPER CASE
139  C179 F1 C1 FE          CMP B   CASE
140  C17C 27 08             BEQ     OUT
141  C17E F7 C1 FE          STA B   CASE        NEW CASE
142                     *
143  C181 36        OUTC40  PSH A
144  C182 17                TBA                 CONTROL CODE
145  C183 8D 4C             BSR     OUTCNT
146  C185 32                PUL A
147  C186 8D 3E     OUT     BSR     OUTASC      PRINT CHAR
148  C188 FE C1 FC  EXIT    LDX     XSAVE       PULL REGISTERS
149  C18B 33                PUL B
150  C18C 32                PUL A
151                     *
152                     * FIX SOME MISSING CHARACTERS BY PRINTING
153                     * TWO ON TOP OF EACH OTHER.
154                     * SEMICOLON = . BS :
155                     * EXCLAM    = ' BS . (ELITE 72)
156                     *
157                     * ++ WARNING ++ RECURSIVE CALLS
158                     *
159  C18D 81 3B             CMP A   #';
160  C18F 26 0C             BNE     RETO01
161  C191 36               PSH A
162  C192 86 08             LDA A   #BS
163  C194 BD C1 24          JSR     OUTCHA
164  C197 86 3A             LDA A   #':
165  C199 BD C1 24          JSR     OUTCHA
166  C19C 32                PUL A
167                     *
168  C19D 7D 80 72  RETO01  TST     IBMCON      WHICH BALL
169  C1A0 2B 10             BMI     RETURN
170  C1A2 81 21             CMP A   #'!         EXCLAMATION CHAR
171  C1A4 26 0C             BNE     RETURN
172  C1A6 36                PSH A
173  C1A7 86 08             LDA A   #BS
174  C1A9 BD C1 24          JSR     OUTCHA
175  C1AC 86 27             LDA A   #''
176  C1AE BD C1 24          JSR     OUTCHA
177  C1B1 32                PUL A
178  C1B2 39        RETURN  RTS
179                     *
```

```
180  C1B3                FOUND   EQU     *
181  C1B3 A6 01                  LDA A   1,X         CONTROL CODE
182  C1B5 85 02                  BIT A   #LOWCASE    FORCED LOWER CASE
183  C1B7 27 05                  BEQ     FOUN10
184  C1B9 C6 02                  LDA B   #LOWCASE
185  C1BB F7 C1 FE               STA B   CASE
186  C1BE 8D 11          FOUN10  BSR     OUTCNT
187  C1C0 20 C6                  BRA     EXIT
188                      *
189                      * SUB ( TEST )
190                      *
191                      *FOR FLEX OUTPUT SPOOLING
192                      *
193  C1C2                TEST    EQU     *
194  C1C2 7D 80 70               TST     IBMDAT
195  C1C5 39                     RTS
196                      *
197                      * SUB ( OUTASC )
198                      *
199                      * PRINTS ONE ASCII CHARACTER
200                      *
201  C1C6                OUTASC  EQU     *
202  C1C6 8D 14                  BSR     READY
203  C1C8 B7 80 70               STA A   IBMDAT      PULL MAGNETS
204  C1CB 8D 1A                  BSR     BUSY        WAIT FOR BUSY REPLY
205  C1CD 7F 80 70               CLR     IBMDAT      RELEASE MAGNETS
206  C1D0 39                     RTS
207                      *
208                      * SUB ( OUTCNT )
209                      *
210                      * ACTIVATES CONTROL MAGNETS
211                      *
212  C1D1                OUTCNT  EQU     *
213  C1D1 8D 09                  BSR     READY
214  C1D3 B7 80 72               STA A   IBMCON      PULL MAGNETS
215  C1D6 8D 0F                  BSR     BUSY
216  C1D8 7F 80 72               CLR     IBMCON      RELEASE MAGNETS
217  C1DB 39                     RTS
218                      *
219                      * SUB ( READY )
220                      *
221                      * CHECKS FOR READY STATUS FROM IBM
222                      * IF PA7 = 1 THEN NOT BUSY
223                      *
224  C1DC                READY   EQU     *
225  C1DC C6 FF                  LDA B   #$FF        DEBOUNCE SUPERVISION
226  C1DE 7D 80 70       RE10    TST     IBMDAT
227  C1E1 2A F9                  BPL     READY       0=BUSY
228  C1E3 5A                     DEC B               TEST READY 255 TIMES
229  C1E4 26 F8                  BNE     RE10
230  C1E6 39                     RTS
231                      *
232                      * SUB ( BUSY )
233                      *
234                      * WAITS UNTIL IBM GETS BUSY
235                      *
236  C1E7                BUSY    EQU     *
237  C1E7 CE 3F FF               LDX     #$3FFF      SAFETY TIME SUPERVISIO
238  C1EA 09             BU05    DEX
239  C1EB 27 0B                  BEQ     BURES       IF 0 THEN WE ARE IN TR
```

```
240  C1ED C6 FF              LDA B  #$FF      DEBOUNCE DELAY
241  C1EF 7D 80 70  BU10     TST    IBMDAT    0=BUSY
242  C1F2 2B F6             BMI    BU05      1=NOT BUSY YET
243  C1F4 5A                DEC B
244  C1F5 26 F8             BNE    BU10
245  C1F7 39                RTS
246                    *
247                    * FAILURE - RELAY LATCH UP.
248                    * DO NOT RISK FIRE IN MAGNETS.
249                    * RESET ALL MAGNETS.
250                    *
251  C1F8 BD C1 00  BURES   JSR    INIT      INITIATE
252  C1FB 39                RTS
253                    *
254                    * VARIABLES
255                    * ---------
256                    *
257  C1FC              XSAVE   RMB    2         SAVE X-REG HERE
258  C1FE              CASE    RMB    1         PRESENT U/L CASE
259  C1FF              LASTCH  RMB    1         LAST CHAR PRINTED
260                    *
261                    * INITIAL DATA
262                    * ------------
263                    *
264                    * TABLE FOR CONTROL CODES.
265                    * GO TO LOWER CASE WHEN PRINTING
266                    * CR OR LF (LEAVE THE IBM MANUALLY USEABLE).
267                    *
268  C200 20           CONCHA  FCB    $20       SP
269  C201 20                   FCB    $20       MAGNET CODE
270                    *
271  C202 08                   FCB    $08       BS
272  C203 40                   FCB    $40
273                    *
274  C204 09                   FCB    $09       TAB
275  C205 10                   FCB    $10
276                    *
277  C206 0D                   FCB    CR        CR
278  C207 0A                   FCB    $0A       CR + LOWER CASE
279                    *
280  C208 0A                   FCB    LF        INDEX (LF)
281  C209 06                   FCB    $06       LF + LOWER CASE
282                    *
283  C20A              CONEND  EQU    *
284                    *
285                    * TILT AND ROTATE CODES
286                    * ----------------------
287                    *
288                    * CODE IS EQUAL TO PIA OUTPUT PORT
289                    * BIT 7 IS 1 IF CHAR IS IN UPPER CASE
290                    *
291                    * TABLE FOR BCD 6400 BALL
292                    *
293  C20A              BCDCHA  EQU    *
294                    *  ASCII $20 - 2F
295  C20A 00                   FCB    $00,$01,$82,$FE,$BE,$DF,$9F,$82
296  C212 CF                   FCB    $CF,$C6,$06,$B6,$02,$0C,$01,$7A
297                    *  ASCII $30 - 3F
298  C21A 0E                   FCB    $0E,$7F,$37,$3E,$57,$5E,$16,$1F
299  C222 4F                   FCB    $4F,$46,$81,$02,$F0,$D7,$8E,$99
```

```
300                         *  ASCII $40 - 4F
301   C22A CF                     FCB    $CF,$79,$31,$38,$51,$58,$10,$19
302   C232 49                     FCB    $49,$40,$7C,$34,$3D,$54,$5D,$15
303                         *  ASCII $50 - 5F
304   C23A 1C                     FCB    $1C,$4C,$45,$32,$3B,$52,$5B,$13
305   C242 1A                     FCB    $1A,$4A,$43,$0B,$08,$04,$00,$75
306                         *
307                         * BCD 6400 BALL HAS ONLY CAPITALS
308                         * SET UPPER CASE CHARS EQU TO LOWER CASE CHAR
309                         *
310                         *  ASCII $60 - 6F
311   C24A 00                     FCB    $00,$79,$31,$38,$51,$58,$10,$19
312   C252 49                     FCB    $49,$40,$7C,$34,$3D,$54,$5D,$15
313                         *  ASCII $70 - 7F
314   C25A 1C                     FCB    $1C,$4C,$45,$32,$3B,$52,$5B,$13
315   C262 1A                     FCB    $1A,$4A,$43,$0B,$08,$04,$00,$75
316                         *
317                         * CODE FOR PRESIGE ELITE 72 BALL
318                         *
319   C26A                  CORCHA  EQU    *
320                         *  ASCII $20 - 2F
321   C26A 00                     FCB    $00,$34,$98,$F4,$0D,$E6,$EE,$FE
322   C272 9E                     FCB    $9E,$86,$FA,$B6,$18,$01,$34,$C6
323                         *  ASCII #30 - 3F
324   C27A 46                     FCB    $46,$4A,$37,$3E,$4F,$57,$16,$5E
325   C282 1F                     FCB    $1F,$07,$B4,$34,$58,$CE,$D8,$96
326                         *  ASCII $40 - 4F
327   C28A 8E                     FCB    $8E,$9C,$82,$9A,$DB,$D2,$B8,$F9
328   C292 C3                     FCB    $C3,$95,$F0,$93,$CA,$FC,$B2,$CC
329                         *  ASCII $50 - 5F
330   C29A D1                     FCB    $D1,$90,$DD,$C5,$F3,$BB,$BD,$84
331   C2A2 FA                     FCB    $FA,$C0,$F6,$D4,$B0,$C8,$7F,$81
332                         *  SMALL LETTERS
333                         *  ASCII $60 - 6F
334   C2AA F4                     FCB    $F4,$1C,$02,$1A,$5B,$52,$38,$79
335   C2B2 43                     FCB    $43,$15,$70,$13,$4A,$7C,$32,$4C
336                         *  ASCII $70 - 7F
337   C2BA 51                     FCB    $51,$10,$5D,$45,$73,$3B,$3D,$04
338   C2C2 7A                     FCB    $7A,$40,$76,$54,$30,$48,$7F,$00
339                         *
340                         * ( TEST ROUTINES )
341                         *
342                         * SEND A CHARACTER FROM THE KEYBOARD TO IBM
343                         *
344   C400                        ORG    $C400
345   C400 BD C1 00               JSR    INIT
346   C403 BD E1 AC       LOOP    JSR    $E1AC       ( INEEE )
347   C406 BD C1 24               JSR    OUTCHA
348   C409 20 F8                  BRA    LOOP
349                         *
350                         * SEND A HEX CODE AND SEE WHAT CHARACTER YOU GE
351                         *
352   C40B BD C1 00               JSR    INIT
353   C40E BD E0 55       LOOP2   JSR    $E055       ( BYTE )
354   C411 BD C1 C6               JSR    OUTASC      PULL MAGNETS
355   C414 20 F8                  BRA    LOOP2
356                         *
357                         * FLEX PRINT.SYS VEKTORS
358                         *
359   ACC0                        ORG    $ACC0
```

```
  360   ACC0 7E C1 00          JMP     INIT
  361                    *
  362   ACD8                   ORG     $ACD8
  363   ACD8 7E C1 C2          JMP     TEST
  364                    *
  365   ACE4                   ORG     $ACE4
  366   ACE4 7E C1 24          JMP     OUTCHA
  367                    *
  368                           END

NO ERROR(S) DETECTED

   SYMBOL TABLE:

BCDCHA  C20A    BCDTAB  C15D    BS      0008    BU05    C1EA    BU10    C1EF
BURES   C1F8    BUSY    C1E7    CASE    C1FE    CONCHA  C200    CONEND  C20A
CORCHA  C26A    CR      000D    EXIT    C188    FOUN10  C1BE    FOUND   C1B3
IBMCON  8072    IBMDAT  8070    INIT    C100    LASTCH  C1FF    LF      000A
LOOP    C403    LOOP2   C40E    LOWCAS  0002    OUT     C186    OUTASC  C1C6
OUTC05  C13E    OUTC10  C144    OUTC15  C160    OUTC20  C161    OUTC30  C177
OUTC40  C181    OUTCHA  C124    OUTCNT  C1D1    PIA     8070    RE10    C1DE
READY   C1DC    RETO01  C19D    RETURN  C1B2    TEST    C1C2    UPPCAS  0001
XSAVE   C1FC
```

NYA MEDLEMMAR

TILLKOMMNA I REGISTRET EFTER MPU-LAREN NR 1/80

ANDERSSON, LEIF	REPSLAGAREGATAN 3C 2TR	732 00 ARBOGA
BLADH, BERTIL	PROFESSORSGATAN 5A	214 53 MALMÖ
DIDACT LÄRMOMED PROD	BOX 20	581 02 LINKÖPING
ENGSTRÖM, GÖRAN	GLASMÄSTAREGATAN 15	412 62 GÖTEBORG
ERIKSSON, P H	KUNGSGÅRDSVÄGEN 39	791 41 FALUN
ERLANDSON, MARTIN	OPALVÄGEN 11	442 60 KODE
HERMANSSON, URBAN	ÖSTERVÄGEN 34	191 49 SOLLENTUNA
HÖRNQVIST, CLAES-GÖRA	SVETSARVÄGEN15	175 73 JÄRFÄLLA
JANZEN, TORD	SPRÅKGRÄND	902 41 UMEÅ
JOHANSSON, SUNE	SVEDJEMARKSGATAN 42	281 00 HÄSSLEHOLM
KASK, AVO	HÖDERVÄGEN 8	184 02 ÖSTERSKÄR
MAD, C/O L SÖDERLUND	SÖDRA GULSPARVSGATAN 42	214 61 MALMÖ
NORDSTRÖM, PER	STORA FRIDHEM	614 00 SÖDERKÖPING
OLSEN, LENNART	RÖNNBÄRSVÄGEN 53	196 31 KUNGSÄNGEN
PETTERSON, GUNNAR	SÖNDAGSVÄGEN 112	123 60 FARSTA
SANDBERG, ULF	FORNMINNESVÄGEN 35	186 00 VALLENTUNA
WIBERG, TORULF	ÖVREGATAN 5B	261 61 LANDSKRONA

FACIT 4553 MATRISSKRIVARE OCH ENKORTSDATOR GER EN ANVÄNDBAR PRINTER

I bland så faller det smulor från den rike mannens bord. Jag råkade stå ivägen när det ramlade ner en FACIT 4553 matrisskrivare. Vad gör man då med en sådan? Vid första anblicken befinner sig 6800'an och printerns papper på långt avstånd från varandra.

4553'an består i princip av en mekanisk enhet. Vidare har den elektronik som behövs för att trycka kolumnerna i en 5x7 matris samt för att klara vagnretur och radframmatning. Dessa är för övrigt sammankopplade. Kommunikationen med yttervärlden sker över ett så kallat SP1-snitt. CR/LF är inte avkodade utan signal ges över en särskild ledning i snittet. Någon teckengenerator ingår ej. Man kan alltså generera tecken efter egna id Öer. Det går dock inte att gå utanför 5x7 matrisen.

Hur skall nu datorns ASCII-tecken kunna övertalas att upstå i läsbar form på papperet? Ja, den enda rationella lösningen är väl en enkortsdator.

Vad behövs då? Jo, en CPU (6800), en PIA (6821), en ACIA (6850) och ett EPROM (1 kbyte). Dessutom en avkodare74LS138) och en invertrare (74LS14).Någon buffring av data eller adressbussen behövs inte. Byt dock inte 74LS-- mot 74--.

PIA'n driver printern direkt utan buffring. 6821 är kanske att föredra framför 6820. Dess utgångar tål ju lite mer.

Schemat frmgår av figur.1. Jag har inte ritat in alla ledningar. Det som inte är självförklarande har förtydligats. Rent allmänt så skall ju t.ex. alla pinnar märkta D1 kopplas ihop o.s.v.

Runt mitt arbetsbord klarar jag mig med överföring via TTL-nivåer. Klockan till ACIA'n tas från datorn. Den visade ocillatorn fungerar utmärkt i brist på 4 MHz kristall!

ACIA'n är intruptdriven (NMI). RTS kopplas till CTS på datorns ACIA. Ett par omkpplare används för val av tecken uppsättning och automatisk sidformatering. Fig.2 ger en bild av förbindelserna mellan enkortsdatorn, printern och yttervärlden.Så långt hårdvaran.

Då jag inte vet hur red. Tänker klippa sönder källtexten så kommer jag att referera till radnummren i fortsättningen.

Rad 26-37 är ett sammandrag av signalerna mellan PIA'n och printern. Det enda RAM som behövs är det som finns i 6802'an. Det är fördelat på 80 bytes till teckenbuffern, 15 bytes för diverse pekare och flaggor samt resten till stacken. På rad 47 står "NO IRQ". Skall vara "IRQ".

Rad 68-114 initierar portarna, rensar buffern,sätter upp diverse pekare och flaggor samt väntar några sekunder medan printern skall komma igång. En rivmarkering görs på papperet om moden är AUTO. I annat fall bara en LF(CR).

Rad 116-128 är huvudlopen. COLFLG talar om dels om ett tecken håller på att tryckas och dels vilken kolumn som tryvks. COLFLG=0 innebär att tecknet är klart. I så fall testas WFLG. Här innebär 0 att utskriften kan börja. BUF är en räknare som anger hur många tecken som för tillfället finns i buffern.

Rad 131-165 hanterar ACIA'n. Eftersom NMI används så är rutinen "fristående" och som jag strax skall förklara så är NMI ett måste. Om det inkomna tecknet är CR ställs WFLG=0 och utskriften börjar. Detsamma händer om buffern blir full. Då sätts också RTS=1. I annat fall

skulle buffern svämma över! Tillhör tecknet de lagliga -0A, 0C, 1C, 20-7F så deponeras det bara i buffern och pekaren uppdateras. OBS att buffern är cirkulär.

Nu ett litet hopp till rad 221. Detta för att få det hela i någon slags logisk ordning. EVAL hämtar nästa tecken i buffern och utvärder det. 0A ger givetvis line feed och 0C matar ut resten av sidan. 1C ger några LF samt en rivmarkering. CR behöver ej kodas av eftersom LF ger en automatisk vagnretur. Dessutom blir rutinen för sidformatteringen enklare.

Är tecknet inget av det ovannämda så skall det tryckas. Först testas om det är ett svenskt tecken och en omkodning sker i så fall. Därefter räknar EVAL98 fram en pekare till teckenmatrisen. Innan EVAL lämnas uppdateras pekaren. Om buffern har minskat tillräcklig återställs RTS till 0 så att datorn kan mata ut fler tecken.

Rad 168-181 är den egentliga print rutinen. Den tar fram en kolumn från tecken matrisen och lägger den på printerns ingång. Bit 7 skall alltid vara 0. I viloläge är READY=1. STROBE kan nu sättas =1 och printern svarar då med att nollställa READY. STROBE nollstäls. Programmet väntar nu på att READY skall gå till "1" innan nästa kolumn hämtas från matrisen. När sista kolumn tryckts skall bit 7 sättas =1. Det finns kanske någon genväg men SP1 snittet trivs bäst om man använder det som FACIT tänkt sig!!

Eftersom bit 7 fungerar som teckensynk kommer resultatet att likna kinesiska om den ställs om på fel ställe. Printern kan skriva tecken för tecken men arbetar normalt synkront. Datorn hinner ju plocka fram ny information i tillräcklig fart.

Rad 200-210 hanterar vagnreturen. Signal ges via CA och en normal SP1 cyckel krävs dessutom.

Rad 302-332 sköter den automatiska sidformateringen. En räknare -LINO- håller reda på radnummret och löser ut 3 LF, rivmarkering och ytterligare 3 LF. På så sätt erhålls automatiska sidor i A4 format.

Rad 335-339 hanterar FORM FEED. Rutinen tittar på LINO och matar papper tills LINO är =0,d.v.s resten av sidan matas ut.

Så följer teckenmatrisen (PUH!!). Hur den är uppbygd framgår av fig.3. Den börjar på HEX 20 och fortsätter sedan till HEX 7F. De svenska tecknen ligger sist.

Något om de problem som uppstod. Overrun (d.v.s att ett tecken skriver över det närmast föregående) i ACIA'n var det första problemmet. I den första varianten av programmet använde jag inte NMI utan gick ut och testade om några tecken kommit. Resultatet var att en massa kod gick åt att tömma ACIA'n. En del rutiner innebär ganska mycket snurrande. NMI löste den detaljen.

Nu svämmade buffern över i stället. Stor förvåning!! Särskilt som jag sätter RTS för att "bromsa" datorns vilda framfart. Efter en del mätande visade det sig att ACIA'n inte alls slutar sända bara för att CTS går hög. Den tar inte emot flera tecken, men det som kommit in sänds!! Man får räkna med 2-3 tecken. Ett i RXACIA'n, ett på väg över och kanske ett som väntar på att sändas. Det blev att bromsa lite tidigare. MOTOROLA säger inte något om detta i sina böcker. Men det är kanske en självklarhet!!

Ett annat problem (för mig) var att hinna räkna fram pekaren till marisen tillräckligt snabbt. Men det löstes genom goda råd från mer programeringskunniga vänner.

Till sist. Det här är ingen exakt byggbeskrivning. Men kanske någon kan hitta något användbart och då är ju alltid en del vunnet. Programmet finns i källtext och vill någon ha det antingen i den formen eller i assemblerad form så går det bra. Men ring mig först eller skriv en rad,Jag kan också bränna 2708.

PS:Den här texten är också framställd med hjälp av en dator. Därav de snygga marginalerna. Innehållet och stavfelen kan man inte gör något åt ens med 6800.

STIG BOBERG

Listnining på sid 35

PRIVATDATAKLUBBEN PD68 SALDO 1979

Intäckter		Utgifter	
Medl.avg.	13.300:-	MPU-laren	4.962:85
Annonser	1.600:-	Tryck+porto	5.041:35
Kassa in	5.794:-	Reseers.	100:-
		Kassetter	635:15
		Boxhyra	30:-
		Återbet.	75:-
		Div.	144:90
		Kassa ut	9.704:75
	20.694:-		20.694:-

UtgÅende balans 79-12-31

Tillg. p.giro	9.704:75	Skulder	0:-

ÅRSMÖTE 27 MARS 1980

Efter att mötet öppnats av ordförande Nils Ohlsson valdes Arne Tillbom till mötesordförande och Bo Ljungblad till mötessekr. Frågan om kallelse behörigen skett, blev med ja besvarad, även om inte alla uppfattat kallelsen högst upp på infobladet för någon månad sedan. Då det inte inkommit några förhandsanmälda frågor lyssnade vi till styrelsens verksamhetsberättelse som också skickades ut i samband med mötet. Kassören Åke Holm berättade om klubbens ekonomiska ställning som befanns vara god. Då en av de valda revisorna, Roger Holmstrand varit oanträffbar lång tid innan årsmötet och revisorsuppleant ej var vald har revision skett av endast en revisor, Per-Erik Svennberg som fann allt vara i god ordning. Då klubben har lättkontrollerade inkomster (medlemsavgifter + annonser) och utgifter (tryck + portokostnader) godkände mötet revision med bara en revisor. Ansvarsfrihet beviljades den avgående styrelsen. Frågan hur vinsten eventuellt ska användas sköts upp till punkt 13 på dagordningen. Vidare bestämdes att klubbens årsavgift på 75:- kvarstår ytterligare ett år. Då både Nils Ohlsson och Åke Holm har mer än fullt i det dagliga arbetet valdes till ny ordförande Tommy Bladh och till ny Kassör Bo Ljungblad. Till styrelsen i övrigt valdes Anders Karlsson, Gunnar Lovius, Åke Holm, Lars Bergstedt, Lars-Olof Fagerström och Nils Ohlsson. Till revisorer valdes Werner Gustavsson och Lennart Uusitalo. Revisorsuppl. Roger Holmstrand. Till valberedning valdes Arne Tillbom och Sven-Erik Ringström med Sven-Erik som sammankallande. Under det kommande året ska MPU-laren komma ut som vanligt dvs. så fort mtr. räcker till. Redaktionskommiten kvarstår. Den gamla styrelsen lämnade förslag om en klubbdator, och mötet tillsatte en kommite för att undersöka möjligheterna för en dylik.

Stora diskusitioner med frågor:

Vad får den kosta
Ett eller två modem OSV.

Kommiten Skickar enkät till medlemmarna och undersöker grundligt. Kommiten består av: Tommy Ribboe (sammankallande) Robert Lundin, Lennart Uusitalo, Nils Ohlsson och Jason King. Inga motioner inkommna och mötet förklarades avslutat.

```
                        NAM     PRTDRV 2.0
                        *
                        *
                        *80-01-16
                        *STIG BOBERG
                        *HJORTRONVAGEN 26
                        *59054 STUREFORS
                        *
                        *013/118311
                        *
                        *INTENDED AS DRIVER FOR FACIT
                        *PRINTER WITHOUT CHARACTER ROM
                        *
                        *AUTOMATIC PAGE FORMATTING WITH CUTTING LINE
                        *
                        *NORMAL ASCII OR SWEDISH PRINT
                        *
                        *ACIA IRQ CONNECTED TO NMI
                        *
                        *
                        *MAY BE COPIED FOR PERSONAL USE ONLY
                        *
                        *

                        *ACIA AT $B008
                        *PIA AT $B004
                        *RAM AT $00-7F - 6802 INTERNAL
                        *PROM AT $FC00 - 1 KBYTE CONTAINS PROGRAM
                        *AS WELL AS CHARACTER MATRIX
                        *
                        *PIA PA0-7 >>>>>> FACIT D1-8
                        *PIA PB7   <<<<<< FACIT AC (READY)
                        *PIA PB6   <<<<<< SWITCH INT/SWEDISH
                        *PIA PB5   <<<<<< SWITCH NORMAL/AUTO PAGE
                        *PIA PB1   >>>>>> FACIT CA (CR//LF)
                        *PIA PB0   >>>>>> FACIT SC (STROBE)

                        *POSITION OF PORTS
B008                    ACIAC   EQU     $B008
B009                    ACIAD   EQU     $B009
B004                    PIAAD   EQU     $B004
B005                    PIAAC   EQU     $B005
B006                    PIABD   EQU     $B006
B007                    PIABC   EQU     $B007
                        *CONSTANTS FOR ACIA - *16, 1 STOPBIT

                        *NO IRQ, RTS 0 OR 1
00D5                    RTS0    EQU     $D5
0095                    RTS1    EQU     $95


                        *BUFFER ETC
                                ORG     0
0000                    BUFBEG  RMB     79      START OF BUFFER
0000                    BUFEND  RMB     1       END OF D:0!
004F                    COLFLG  RMB     1       PRINT FLG - COL #
0050                    COLAD   RMB     2       POINTER TO TABLE
0051                    BUF     RMB     1       # OF CHR IN BUF
0053
```

```
57   0054                STORE   RMB    2          WHERE NEXT CHR GOES
58   0056                PRINT   RMB    2          NEXT PRINTED
59   0058                TEMP    RMB    2          TEMPORARY
60   005A                TEMP1   RMB    1
61   005B                CHRFLG  RMB    1          1=SWEDISH
62   005C                PGFLG   RMB    1          1=AUTO PAGE
63   005D                LINO    RMB    1          LINE # AUTO MODE
64   005E                WFLG    RMB    1          1=HOLD PRINT
65                       *
67   FC00                        ORG    $FC00
68   FC00 8E 00 7F       START   LDS    #$7F
69                       *WAIT FOR PRINTER TO STABILIZE
70   FC03 C6 04                  LDA B  #4         DELAY ABOUT
71   FC05 CE 00 00       DLY     LDX    #0         FOUR SECONDS
72   FC08 09             DLY1    DEX
73   FC09 26 FD                  BNE    DLY1
74   FC0B 5A                     DEC B
75   FC0C 26 F7                  BNE    DLY
76                       *SET UP ACIA
77   FC0E 86 03                  LDA A  #3
78   FC10 B7 B0 08               STA A  ACIAC      MASTER CLEAR
79   FC13 86 D5                  LDA A  #RTS0
80   FC15 B7 B0 08               STA A  ACIAC
81                       * AND PIA
82   FC18 CE FF 04               LDX    #$FF04
83   FC1B FF B0 04               STX    PIAAD
84   FC1E CE 0F 04               LDX    #$0F04
85   FC21 FF B0 06               STX    PIABD
86   FC24 86 80                  LDA A  #$80
87   FC26 B7 B0 04               STA A  PIAAD      SET SYNC BIT = 1
88   FC29 7F B0 06               CLR    PIABD      SET SC AND CA = 0
89                       *CLEAR BUFFER
90   FC2C CE 00 00               LDX    #BUFBEG    SET X
91   FC2F 4F                     CLR A
92   FC30 A7 00          STARTO  STA A  0,X
93   FC32 08                     INX
94   FC33 8C 00 5F               CPX    #WFLG+1
95   FC36 26 F8                  BNE    STARTO
96   FC38 7C 00 5E               INC    WFLG       SET PRINT FLAG
97                       *SET CHAR AND PAGE FLAGS
98   FC3B C6 01                  LDA B  #1         SET B
99   FC3D B6 B0 06               LDA A  PIABD      READ SWITCHES
100  FC40 48                     ASL A
101  FC41 85 80                  BIT A  #$80       CHAR SET
102  FC43 2A 04                  BPL    START2     NORMAL
103  FC45 D7 5B                  STA B  CHRFLG     SWEDISH
104  FC47 20 03                  BRA    START3
105  FC49 7F 00 5B       START2  CLR    CHRFLG
106  FC4C 48             START3  ASL A
107  FC4D 85 80                  BIT A  #$80       AUTO?
108  FC4F 2A 07                  BPL    START4     NORMAL
109  FC51 D7 5C                  STA B  PGFLG      AUTO PAGE
110  FC53 BD FD AE               JSR    PAGE0      FORM CUTTING LINE
111  FC56 20 06                  BRA    SCAN
112  FC58 7F 00 5C       START4  CLR    PGFLG
113  FC5B BD FD 00               JSR    LFCR       NEW LINE
115                      *SCAN ROUTINE
116                      *SEE IF CHAR IS PRINTED
117  FC5E 7D 00 50       SCAN    TST    COLFLG     PRINTING?
118  FC61 27 04                  BEQ    SCAN2      NO
```

```
119  FC63 8D 69          BSR    PRINT2     PRINT NEXT COLUMN
120  FC65 20 F7          BRA    SCAN       LOOK FOR INPUT
121                   *CHARACTER IN BUFFER???
122  FC67 7D 00 5E  SCAN2  TST    WFLG       IS FLAG SET?
123  FC6A 26 08          BNE    SCAN3      IF SO DONT PRINT YET
124  FC6C 7D 00 53       TST    BUF        EMPTY?
125  FC6F 27 03          BEQ    SCAN3      YES, LOOP AGN
126  FC71 BD FD 1B       JSR    EVAL       GET NEXT CHAR FROM BUF
127  FC74 20 E8     SCAN3  BRA    SCAN
128                   *
129                   *GET INPUT STORE IT IN BUFFER
130  FC76 B6 B0 09  GETCH  LDA A  ACIAD      GET CHARACTER
131  FC79 84 7F          AND A  #$7F        MASK BIT 7
132  FC7B 81 0D          CMP A  #$0D        CR?
133  FC7D 26 05          BNE    GETCO
134  FC7F 7F 00 5E       CLR    WFLG        BEGIN PRINT
135  FC82 20 3A          BRA    GETCH3
136  FC84 81 0A     GETCO  CMP A  #$0A        LINE FEED ?
137  FC86 27 13          BEQ    GETCHO
138  FC88 81 1F          CMP A  #$1F        PRINTABLE ?
139  FC8A 22 0F          BHI    GETCHO
140  FC8C 7D 00 5C       TST    PGFLG
141  FC8F 27 2D          BEQ    GETCH3      NOT AUTO MODE
142  FC91 81 0C          CMP A  #$0C        FORM FEED
143  FC93 27 06          BEQ    GETCHO
144  FC95 81 1C          CMP A  #$1C        PAGE HEADER
145  FC97 27 02          BEQ    GETCHO
146  FC99 20 23          BRA    GETCH3
147  FC9B DE 54     GETCHO LDX    STORE
148  FC9D A7 00          STA A  0,X         STORE IT
149  FC9F 7C 00 53       INC    BUF         BUFFER SIZE!
150  FCA2 96 53          LDA A  BUF
151  FCA4 81 49          CMP A  #BUFEND-BUFBEG-6 FULL?
152  FCA6 23 08          BLS    GETCH1
153  FCA8 7F 00 5E       CLR    WFLG        BUFFER FULL - PRINT!!
154  FCAB 86 95          LDA A  #RTS1
155  FCAD B7 B0 08       STA A  ACIAC       NO MORE
156  FCB0 8C 00 4F  GETCH1 CPX  #BUFEND
157  FCB3 27 04          BEQ    GETCH2      RESET POINTER
158  FCB5 08             INX
159  FCB6 DF 54          STX    STORE
160  FCB8 3B             RTI
161  FCB9 CE 00 00  GETCH2 LDX  #BUFBEG     RESET!
162  FCBC DF 54          STX    STORE       NEW STORE ADDRESS
163  FCBE 3B        GETCH3 RTI
164                   *
166                   *PRINT ROUTINE
167  FCBF 96 50     PRINTO  LDA A  COLFLG
168  FCC1 81 01             CMP A  #1        CHR READY,SYNC!
169  FCC3 26 08             BNE    PRINT1
170  FCC5 86 80             LDA A  #$80
171  FCC7 B7 B0 04          STA A  PIAAD     SET SYNC BIT
172  FCCA 7F 00 50          CLR    COLFLG
173  FCCD 39        PRINT1  RTS
174  FCCE DE 51     PRINT2  LDX    COLAD     POINT TO CHR ROW
175  FCD0 A6 00             LDA A  0,X       GET COL
176  FCD2 7A 00 50          DEC    COLFLG
177  FCD5 8D 04             BSR    SP1       PRINT IT
178  FCD7 08                INX              SET TO NEXT COL
179  FCD8 DF 51             STX    COLAD
```

```
180  FCDA 39                    RTS
181                        *SP1 HANDLER (FACIT)
182  FCDB B7 B0 04         SP1      STA A  PIAAD      STORE COLUMN
183  FCDE B6 B0 06         SP10     LDA A  PIABD
184  FCE1 8A 01                     ORA A  #1         SET STROBE
185  FCE3 B7 B0 06                  STA A  PIABD
186  FCE6 7D B0 06         SP11     TST    PIABD      AC=0 (PR NOT READY)
187  FCE9 2B FB                     BMI    SP11       NO, WAIT
188  FCEB B6 B0 06                  LDA A  PIABD
189  FCEE 84 FE                     AND A  #$FE       RESET BIT 0
190  FCF0 B7 B0 06                  STA A  PIABD      REMOVE STROBE
191  FCF3 7D 00 50                  TST    COLFLG
192  FCF6 27 02                     BEQ    SP12
193  FCF8 8D C5                     BSR    PRINTO
194  FCFA 7D B0 06         SP12     TST    PIABD      AC=1?
195  FCFD 2A FB                     BPL    SP12       NO!
196  FCFF 39                        RTS
197                        *
198                        *LF/CR ROUTINE. CA STROBED
199  FD00 B6 B0 06         LFCR     LDA A  PIABD
200  FD03 8A 02                     ORA A  #2         D1=1 (FACIT LF/CR)
201  FD05 B7 B0 06                  STA A  PIABD
202  FD08 8D D4                     BSR    SP10       STROBE PRINTER
203  FD0A B6 B0 06                  LDA A  PIABD
204  FD0D 84 FD                     AND A  #$FD       RESET PB1 (CA)
205  FD0F B7 B0 06                  STA A  PIABD
206  FD12 7D 00 5C                  TST    PGFLG      1=AUTO PAGE
207  FD15 27 03                     BEQ    LFCR1
208  FD17 BD FD A4                  JSR    PAGE       PAGE ROUTINE
209  FD1A 39               LFCR1    RTS
210                        *
211                        *
213                        *GET NEXT CHAR FROM BUFFER AND
214                        *EVALUATE IT
215                        *CR AND LF IS COUPLED IN THE PRINTER
216                        *CR IS THEREFORE SKIPPED AS IT IS
217                        *AUTOMATICALLY MADE ON THE FIRST
218                        *LF RECEIVED
219                        *
220  FD1B DE 56            EVAL     LDX    PRINT
221  FD1D 7A 00 53                  DEC    BUF        ONE CHAR GONE
222  FD20 26 03                     BNE    EVALO
223  FD22 7C 00 5E                  INC    WFLG       EMPTY, HOLD PRINT
224  FD25 A6 00            EVALO    LDA A  0,X        GET CHARACTER
225  FD27 81 0A                     CMP A  #$0A       LF?
226  FD29 27 10                     BEQ    EVAL2
227  FD2B 81 0C                     CMP A  #$0C       FORM FEED
228  FD2D 27 10                     BEQ    EVAL4
229  FD2F 81 1C                     CMP A  #$1C       "HOME"
230  FD31 27 11                     BEQ    EVAL5
231  FD33 8D 25                     BSR    EVAL90     FIND TABLE POS
232  FD35 C6 06                     LDA B  #6         5 COL+SYNC
233  FD37 D7 50                     STA B  COLFLG     SET COLFLAG
234  FD39 20 0C            EVALO1   BRA    EVAL8
235                        *
236  FD3B 8D 43            EVAL2    BSR    LFCR
237  FD3D 20 08                     BRA    EVAL8
238                        *
239  FD3F BD FD E1         EVAL4    JSR    FORMF
240  FD42 20 03                     BRA    EVAL8
```

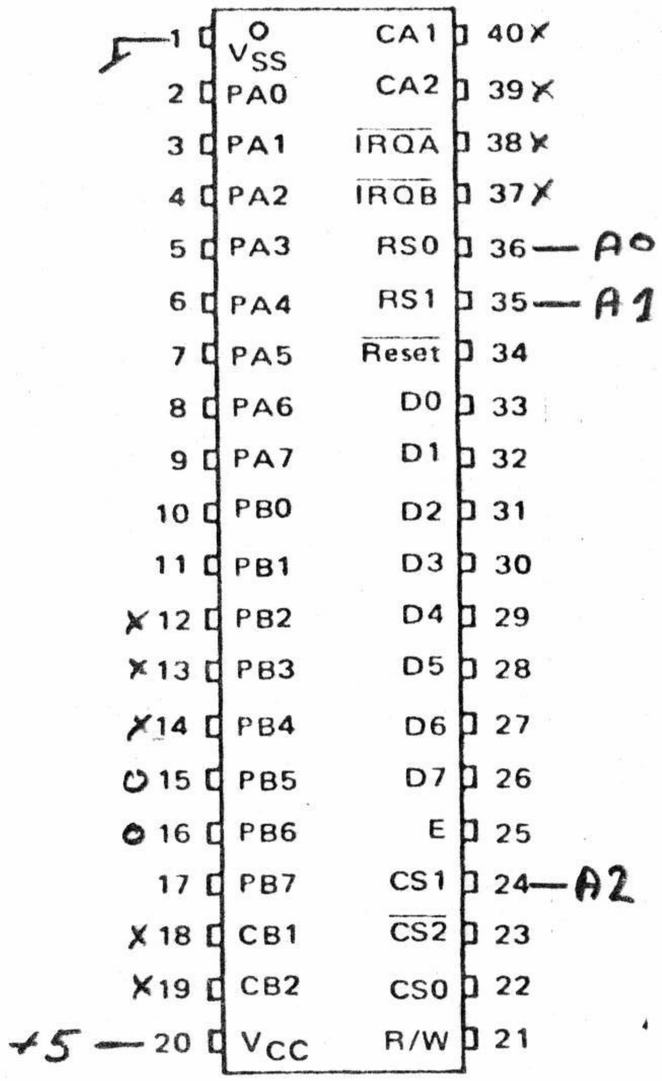
```
241                          *
242  FD44 BD FD AE  EVAL5    JSR     PAGEO
243                          *
244                          *ADJUST PRINT POINTER
245                          *BUFFER IS CIRCULAR
246                          *
247  FD47 DE 56     EVAL8    LDX     PRINT
248  FD49 8C 00 4F           CPX     #BUFEND
249  FD4C 27 05              BEQ     EVAL81
250  FD4E 08                 INX
251  FD4F DF 56              STX     PRINT
252  FD51 20 45              BRA     EVAL99
253  FD53 CE 00 00  EVAL81   LDX     #BUFBEG
254  FD56 DF 56              STX     PRINT       RESET POINTER
255  FD58 20 3E              BRA     EVAL99
257                      *SEE IF SWEDISH
258                      *CHANGE CODE IF NECESSARY
259                      *
260  FD5A 7D 00 5B  EVAL90   TST     CHRFLG      1=SWEDISH
261  FD5D 27 1F              BEQ     EVAL98      NO!!!
262  FD5F CE FD 71           LDX     #TAB3
263  FD62 A1 00     EVAL91   CMP A   0,X
264  FD64 27 07              BEQ     EVAL93      FOUND!
265  FD66 6D 01              TST     1,X
266  FD68 27 14              BEQ     EVAL98      END OF TABLE
267  FD6A 08                 INX
268  FD6B 20 F5              BRA     EVAL91      TRY AGN
269  FD6D A6 07     EVAL93   LDA A   7,X         TRANSLATE
270  FD6F 20 0D              BRA     EVAL98
271                          *
272  FD71 5B        TAB3     FCB     $5B,$5C,$5D,$7B,$7C,$7D,0
273  FD78 80                 FCB     $80,$81,$82,$83,$84,$85
274                          *
275                          *FIND POS IN TABLE
276                          *ACCA=ASCII
277  FD7E CE FD EC  EVAL98   LDX     #TABBEG
278  FD81 DF 58              STX     TEMP
279  FD83 5F                 CLR B
280  FD84 80 20              SUB A   #$20        MOVE 32 DOWN
281  FD86 97 5A              STA A   TEMP1       SAVE
282  FD88 48                 ASL A
283  FD89 59                 ROL B               X2
284  FD8A 48                 ASL A
285  FD8B 59                 ROL B               X4
286  FD8C 9B 5A              ADD A   TEMP1
287  FD8E C9 00              ADC B   #0
288  FD90 9B 59              ADD A   TEMP+1
289  FD92 D9 58              ADC B   TEMP
290  FD94 D7 51              STA B   COLAD       (ASCII-32)*5
291  FD96 97 52              STA A   COLAD+1     +TABBEG
292  FD98 96 53     EVAL99   LDA A   BUF         GET BUFFER SIZE
293  FD9A 81 28              CMP A   #40         > 40 ?
294  FD9C 22 05              BHI     EVA100      NO MORE JUST NOW
295  FD9E 86 D5              LDA A   #RTSO
296  FDA0 B7 B0 08           STA A   ACIAC       FILL BUFFER
297  FDA3 39        EVA100   RTS
298                          *
300                          *AUTO PAGE MODE
301  FDA4 86 40     PAGE     LDA A   #64         MAX PRINTED
302  FDA6 91 5D              CMP A   LINO
```

```
303   FDA8 27 04          BEQ     PAGEO       NEXT PAGE
304   FDAA 7C 00 5D       INC     LINO
305   FDAD 39             RTS
306                  *
307   FDAE 8D 02     PAGEO BSR     PAGE1
308   FDB0 20 0C          BRA     PAGE19
309                  *
310   FDB2 7F 00 5C  PAGE1 CLR     PGFLG       TEMPORARY!
311   FDB5 C6 03     PAGE10 LDA B  #3          3*LF
312   FDB7 BD FD 00  PAGE11 JSR     LFCR
313   FDBA 5A              DEC B
314   FDBB 26 FA           BNE     PAGE11
315   FDBD 39              RTS
316                  *PRINT CUTTING LINE
317   FDBE 86 4F     PAGE19 LDA A  #79
318   FDC0 97 58           STA A   TEMP
319   FDC2 CE FE 2D  PAGE20 LDX     #CUT
320   FDC5 DF 51           STX     COLAD
321   FDC7 86 06           LDA A   #6
322   FDC9 97 50           STA A   COLFLG      SET FLAG
323   FDCB BD FC CE  PAGE21 JSR     PRINT2
324   FDCE 7D 00 50        TST     COLFLG      DONE?
325   FDD1 26 F8           BNE     PAGE21
326   FDD3 7A 00 58        DEC     TEMP        NEXT CHR
327   FDD6 26 EA           BNE     PAGE20      NEXT CHR
328   FDD8 8D DB           BSR     PAGE10
329   FDDA 7C 00 5C        INC     PGFLG       RESET FLAG
330   FDDD 7F 00 5D        CLR     LINO
331   FDE0 39              RTS
332                  *
333                  *FORMFEED
334   FDE1 7D 00 5D  FORMF TST     LINO
335   FDE4 27 05           BEQ     FORM2       DONE!
336   FDE6 BD FD 00        JSR     LFCR        DO LINE FEED
337   FDE9 20 F6           BRA     FORMF
338   FDEB 39        FORM2 RTS
340                  *FOLLOWS NORMAL ASCII CHARACTER SET
341                  *6 SWEDISH LETTER PLACED AT THE END
342                  *OF THE TABLE
343   FDEC           TABBEG EQU    *
344   FDEC 00              FCB     $00,$00,$00,$00,$00
345   FDF1 00              FCB     $00,$00,$5F,$00,$00
346   FDF6 00              FCB     $00,$07,$00,$07,$00
347   FDFB 14              FCB     $14,$7F,$14,$7F,$14
348   FE00 24              FCB     $24,$2A,$7F,$2A,$12
349   FE05 23              FCB     $23,$13,$08,$64,$62
350   FE0A 36              FCB     $36,$49,$56,$20,$50
351   FE0F 00              FCB     $00,$04,$02,$01,$00
352   FE14 1C              FCB     $1C,$22,$41,$00,$00
353   FE19 00              FCB     $00,$00,$41,$22,$1C
354   FE1E 22              FCB     $22,$14,$7F,$14,$22
355   FE23 08              FCB     $08,$08,$3E,$08,$08
356   FE28 00              FCB     $00,$40,$30,$00,$00
357   FE2D 08        CUT   FCB     $08,$08,$08,$08,$08
358   FE32 00              FCB     $00,$60,$60,$00,$00
359   FE37 20              FCB     $20,$10,$08,$04,$02

361   FE3C 3E              FCB     $3E,$51,$49,$45,$3E
362   FE41 00              FCB     $00,$42,$7F,$40,$00
363   FE46 62              FCB     $62,$51,$49,$49,$46
```

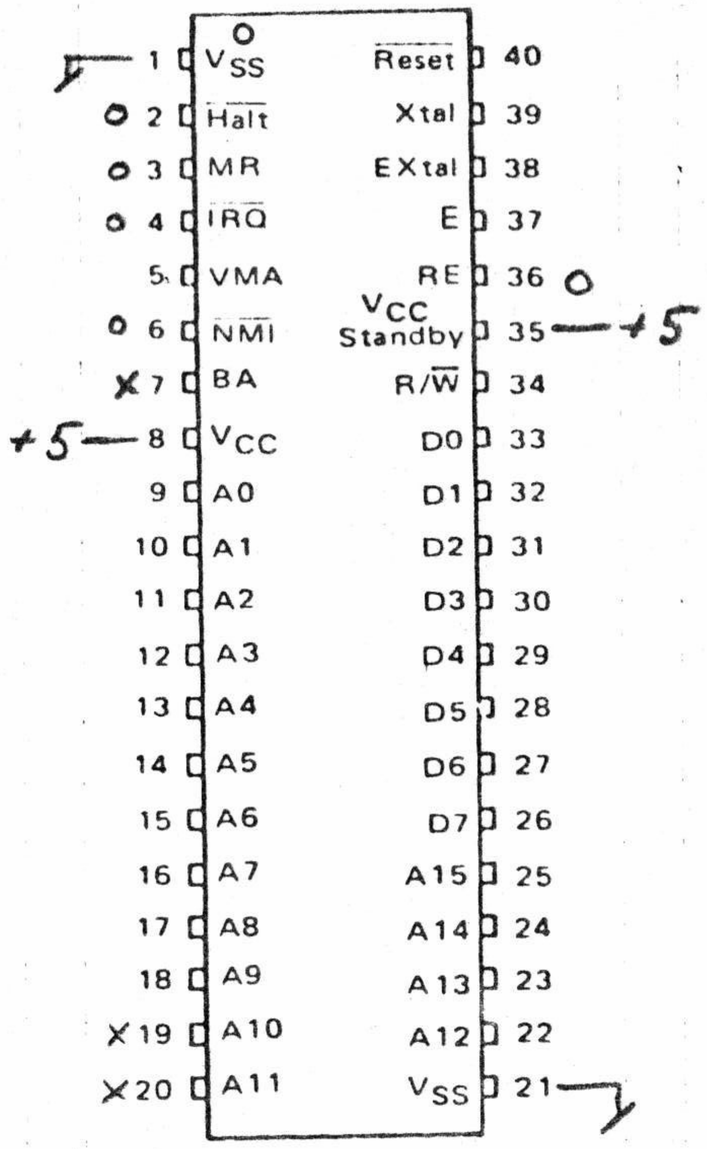
```
364   FE4B 21     FCB     $21,$41,$49,$4D,$33
365   FE50 18     FCB     $18,$14,$12,$7F,$10
366   FE55 27     FCB     $27,$45,$45,$45,$39
367   FE5A 3C     FCB     $3C,$4A,$49,$49,$31
368   FE5F 01     FCB     $01,$71,$09,$05,$03
369   FE64 36     FCB     $36,$49,$49,$49,$36
370   FE69 46     FCB     $46,$49,$49,$29,$1E
371   FE6E 00     FCB     $00,$66,$66,$00,$00
372   FE73 00     FCB     $00,$40,$34,$00,$00
373   FE78 08     FCB     $08,$14,$22,$41,$00
374   FE7D 14     FCB     $14,$14,$14,$14,$14
375   FE82 00     FCB     $00,$41,$22,$14,$08
376   FE87 02     FCB     $02,$01,$59,$05,$02

378   FE8C 3E     FCB     $3E,$41,$5D,$59,$4E
379   FE91 7E     FCB     $7E,$09,$09,$09,$7E
380   FE96 7F     FCB     $7F,$49,$49,$49,$36
381   FE9B 3E     FCB     $3E,$41,$41,$41,$22
382   FEA0 7F     FCB     $7F,$41,$41,$41,$3E
383   FEA5 7F     FCB     $7F,$49,$49,$49,$41
384   FEAA 7F     FCB     $7F,$09,$09,$09,$01
385   FEAF 3E     FCB     $3E,$41,$41,$59,$79
386   FEB4 7F     FCB     $7F,$08,$08,$08,$7F
387   FEB9 00     FCB     $00,$41,$7F,$41,$00
388   FEBE 20     FCB     $20,$40,$40,$40,$3F
389   FEC3 7F     FCB     $7F,$08,$14,$22,$41
390   FEC8 7F     FCB     $7F,$40,$40,$40,$40
391   FECD 7F     FCB     $7F,$02,$0C,$02,$7F
392   FED2 7F     FCB     $7F,$04,$08,$10,$7F
393   FED7 3E     FCB     $3E,$41,$41,$41,$3E
395   FEDC 7F     FCB     $7F,$09,$09,$09,$06
396   FEE1 3E     FCB     $3E,$41,$51,$21,$5E
397   FEE6 7F     FCB     $7F,$09,$19,$29,$46
398   FEEB 26     FCB     $26,$49,$49,$49,$32
399   FEF0 01     FCB     $01,$01,$7F,$01,$01
400   FEF5 3F     FCB     $3F,$40,$40,$40,$3F
401   FEFA 1F     FCB     $1F,$20,$40,$20,$1F
402   FEFF 7F     FCB     $7F,$20,$18,$20,$7F
403   FF04 63     FCB     $63,$14,$08,$14,$63
404   FF09 03     FCB     $03,$04,$78,$04,$03
405   FF0E 61     FCB     $61,$51,$49,$45,$43
406   FF13 00     FCB     $00,$7F,$41,$41,$41
407   FF18 02     FCB     $02,$04,$08,$10,$20
408   FF1D 41     FCB     $41,$41,$41,$7F,$00
409   FF22 04     FCB     $04,$02,$7F,$02,$04
410   FF27 40     FCB     $40,$40,$40,$40,$40

412   FF2C 00     FCB     $00,$00,$03,$04,$00
413   FF31 20     FCB     $20,$54,$54,$54,$78
414   FF36 7F     FCB     $7F,$44,$44,$44,$38
415   FF3B 38     FCB     $38,$44,$44,$44,$08
416   FF40 38     FCB     $38,$44,$44,$44,$3F
417   FF45 38     FCB     $38,$54,$54,$54,$18
418   FF4A 04     FCB     $04,$7E,$05,$01,$02
419   FF4F 48     FCB     $48,$54,$54,$54,$7E
420   FF54 7F     FCB     $7F,$04,$04,$04,$78
421   FF59 00     FCB     $00,$44,$7D,$40,$00
422   FF5E 20     FCB     $20,$40,$40,$40,$3D
423   FF63 7F     FCB     $7F,$10,$08,$14,$60
424   FF68 00     FCB     $00,$41,$7F,$40,$00
```

```
425  FF6D 7C                FCB     $7C,$04,$18,$04,$7C
426  FF72 7C                FCB     $7C,$04,$04,$04,$78
427  FF77 38                FCB     $38,$44,$44,$44,$38

429  FF7C 7C                FCB     $7C,$14,$14,$14,$08
430  FF81 08                FCB     $08,$14,$14,$14,$7C
431  FF86 7C                FCB     $7C,$08,$04,$04,$08
432  FF8B 08                FCB     $08,$54,$54,$54,$20
433  FF90 04                FCB     $04,$3F,$44,$40,$20
434  FF95 3C                FCB     $3C,$40,$40,$20,$7C
435  FF9A 1C                FCB     $1C,$20,$40,$20,$1C
436  FF9F 3C                FCB     $3C,$40,$30,$40,$3C
437  FFA4 44                FCB     $44,$28,$10,$28,$44
438  FFA9 0C                FCB     $0C,$50,$50,$50,$3C
439  FFAE 44                FCB     $44,$64,$54,$4C,$44
440  FFB3 00                FCB     $00,$08,$36,$41,$41
441  FFB8 00                FCB     $00,$00,$33,$00,$00
442  FFBD 41                FCB     $41,$41,$36,$08,$00
443  FFC2 02                FCB     $02,$01,$02,$04,$02
444  FFC7 7F                FCB     $7F,$7F,$7F,$7F,$7F
446                     *SWEDISH LETTERS
447                     *
448  FFCC 7D                FCB     $7D,$0A,$0A,$0A,$7D
449  FFD1 3D                FCB     $3D,$42,$42,$42,$3D
450  FFD6 7C                FCB     $7C,$0A,$0B,$0A,$7C
451  FFDB 20                FCB     $20,$55,$54,$55,$78
452  FFE0 38                FCB     $38,$45,$44,$45,$38
453  FFE5 20                FCB     $20,$54,$55,$54,$78


455  FFFC                   ORG     $FFFC
456  FFFC FC 76             FDB     GETCH       NMI VECTOR
457  FFFE FC 00             FDB     START       START VECTOR
458                         END

NO ERROR(S) DETECTED

    SYMBOL TABLE:

ACIAC   B008    ACIAD   B009    BUF     0053    BUFBEG  0000    BUFEND  004F
CHRFLG  005B    COLAD   0051    COLFLG  0050    CUT     FE2D    DLY     FC05
DLY1    FC08    EVA100  FDA3    EVAL    FD1B    EVALO   FD25    EVALO1  FD39
EVAL2   FD3B    EVAL4   FD3F    EVAL5   FD44    EVAL8   FD47    EVAL81  FD53
EVAL90  FD5A    EVAL91  FD62    EVAL93  FD6D    EVAL98  FD7E    EVAL99  FD98
FORM2   FDEB    FORMF   FDE1    GETCO   FC84    GETCH   FC76    GETCHO  FC9B
GETCH1  FCBO    GETCH2  FCB9    GETCH3  FCBE    LFCR    FDOO    LFCR1   FD1A
LINO    005D    PAGE    FDA4    PAGEO   FDAE    PAGE1   FDB2    PAGE10  FDB5
PAGE11  FDB7    PAGE19  FDBE    PAGE20  FDC2    PAGE21  FDCB    PGFLG   005C
PIAAC   B005    PIAAD   B004    PIABC   B007    PIABD   B006    PRINT   0056
PRINTO  FCBF    PRINT1  FCCD    PRINT2  FCCE    RTSO    00D5    RTS1    0095
SCAN    FC5E    SCAN2   FC67    SCAN3   FC74    SP1     FCDB    SP10    FCDE
SP11    FCE6    SP12    FCFA    START   FCOO    STARTO  FC30    START2  FC49
START3  FC4C    START4  FC58    STORE   0054    TAB3    FD71    TABBEG  FDEC
TEMP    0058    TEMP1   005A    WFLG    005E
```

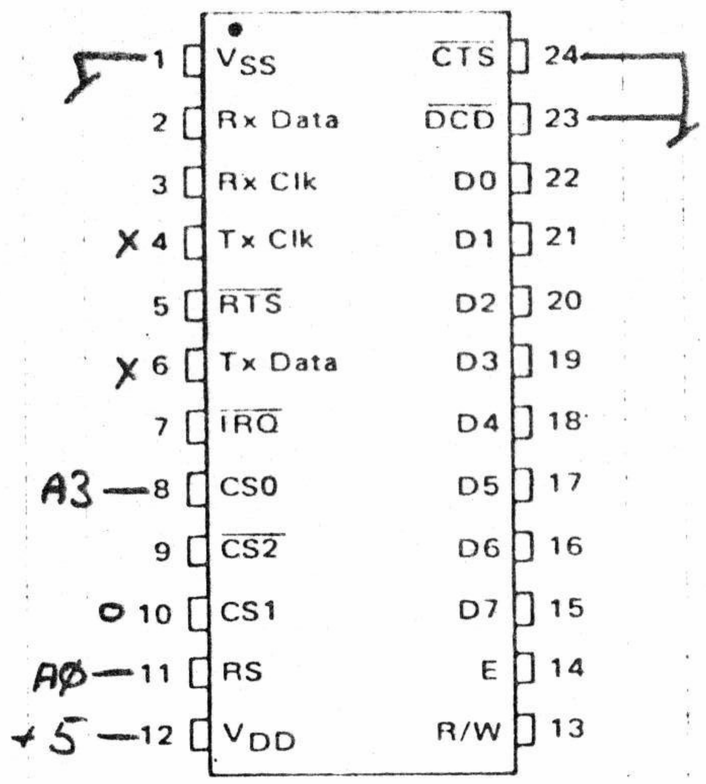
6821



6802



6850



2708

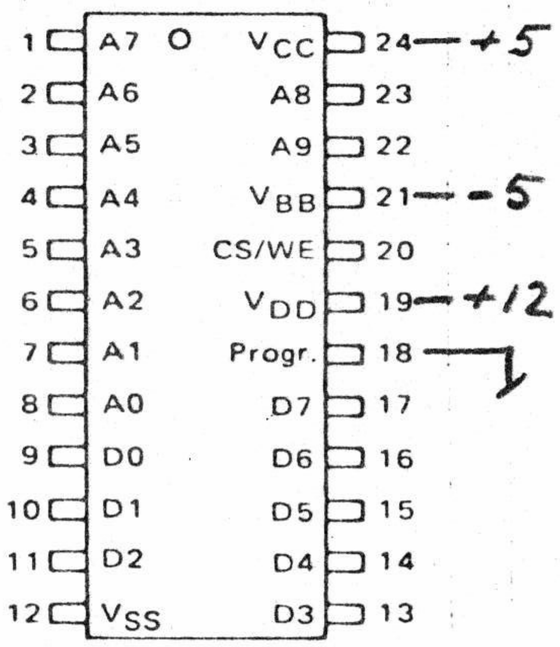


Fig 1

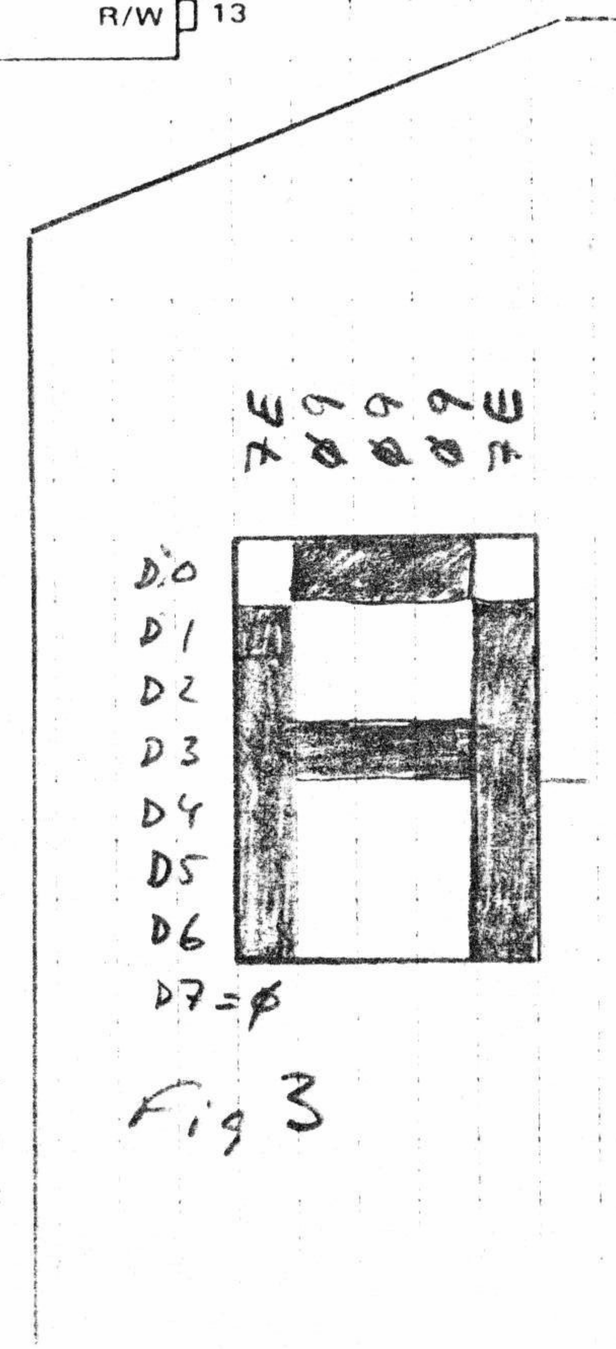
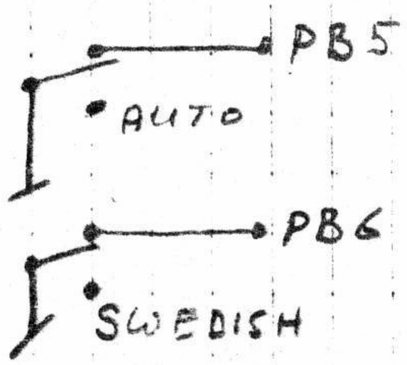


Fig 3

PIA		FACIT
PA0	——	D1
1	——	D2
2	——	D3
3	——	D4
4	——	D5
5	——	D6
6	——	D7
7	——	D8
PB 0	——	SC
1	——	CA
7	——	AC

Fig 2



* FLEX 2.0 FIX *

Skulle inte ovanstående rubrik kunna bli en återkommande serie i MPU-laren.... Många av oss har säkert kommit på en hel del trevlig saker att förbättra Flex 2.0 med. Även nya utilities är vi alla intreserade av också.

Om man har en snabb terminal vill man ofta använda pause- och dp funktionen i TTYSET. Använder man sedan P:CMD utility så ställs PS=OFF. För att slippa detta kan man editera om P.CMD och då ta bort PAUSE EQU $AC09 och vidare ta bort CLR PAUSE och sedan assemblera om P.CMD. Man kan ju också kalla den nya filen för P2.CMD så är det bara att välja.

Vidare vill man ju ibland ändra PS-funktionen och istället för att då skriva TTYSET,PS=Y eller PS=N så kan man använda de listade Y.CMD och N.CMD programmen.

XOUT är ett mycket bekvämt kommando och finns ett enkelt sätt att ändra detta kommando till att sudda vilken fil som helst. Ändring av XOUT till t.ex XTXT.CMD

```
 +++GET,XOUT.CMD
 +++MON
 $M A15B 4F Ändra till 54
 $ A161 55 ändra till 58
  $A167 54 ändra till 54
 $G
 +++SAVE.LOW,XTXT.CMD,A100,A1FA,A
100
```

ÅKE STENÄNG

Här kommer ett genmäle från Norge.Som ni ser så är det inte så lätt att slå norrmännen på fingrarna.Och de kan ge igen lika bra som de tyvär ganska självgoda svenskarna.

Reidar Strand
c/o Ola Tandberg Elektro A/S
Skedsmogt. 25
Oslo 6 - Norge

Oslo, den 14. februar 1980

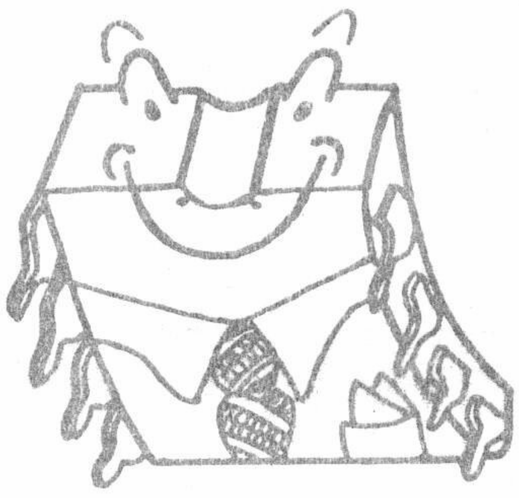
Privatdataklubben PD68
Box 98

12221 ENSKEDE 1
S v e r i g e

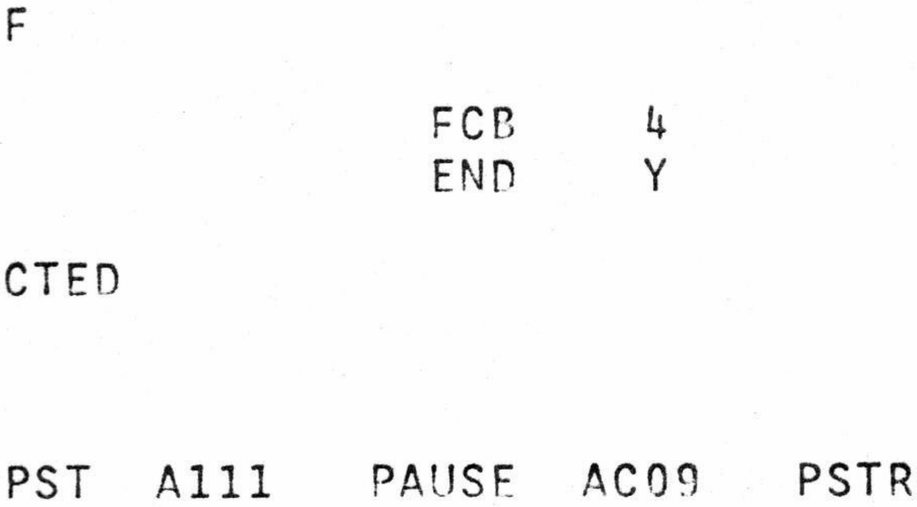
68001-N. INFO FRA NORGE.

Etter H. Degermans notis i MPU-Laren nr. 4, om den norske 1 pins-versjonen av 68000, har vi fått en del henvendelser fra svensker som mener seg kvalifisert til å ta den i bruk.
Deres største bekymring har imidlertid vært:
På hvilken måte skal den monteres?
Svensker, det er ikke noe problem!
Den virker like bra begge veier.

Reidar Strand
R. Strand



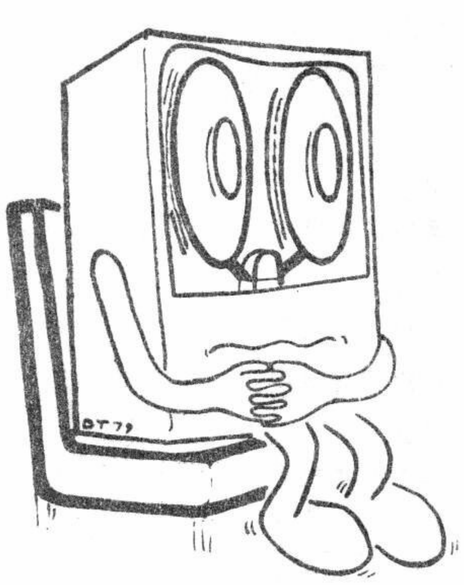
```
 1                          NAM     Y.CMD
 2                  *
 3                  **********************
 4                  * PAUSE=YES (ON) *
 5                  **********************
 6                  *
 7  A840            FCB     EQU     $A840
 8  AC09            PAUSE   EQU     $AC09
 9  00FF            VALUE   EQU     $FF
10  AD1E            PSTRNG  EQU     $AD1E
11  AD03            WARMS   EQU     $AD03
12                  *
13  A100                    ORG     $A100
14  A100 20 01      Y       BRA     YP
15  A102 01         VN      FCB     1
16  A103 86 FF      YP      LDA A   #VALUE
17  A105 B7 AC 09           STA A   PAUSE
18  A108 CE A1 11           LDX     #NOPST
19  A10B BD AD 1E           JSR     PSTRNG
20  A10E 7E AD 03           JMP     WARMS
21  A111 54         NOPST   FCC     /TTYSET,PS=ON/
    A112 54 59
    A114 53 45
    A116 54 2C
    A118 50 53
    A11A 3D 4F
    A11C 4E
    A11D 04                 FCB     4
    23                      END     Y

NO ERROR(S) DETECTED

   SYMBOL TABLE:

FCB     A840   NOPST   A111   PAUSE   AC09   PSTRNG AD1E   VALUE  00FF
VN      A102   WARMS   AD03   Y       A100   YP     A103
```

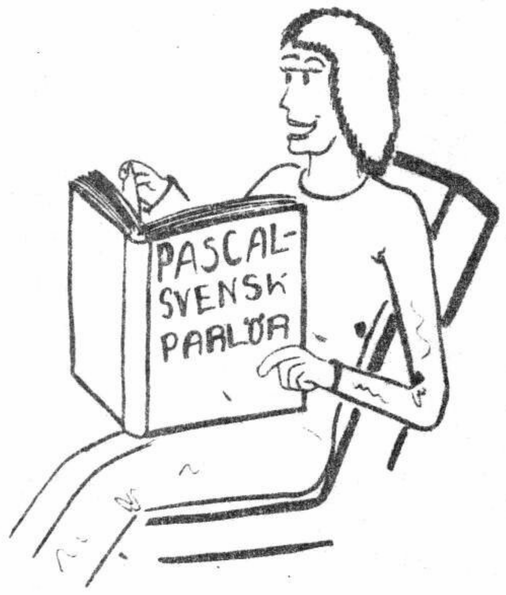


```
A15F 22          NOPST   FCC     /"PRINT.SYS" NOT FOUND/
A160 50 52
A162 49 4E
A164 54 2E
A166 53 59
A168 53 22
A16A 20 4E
A16C 4F 54
A16E 20 46
A170 4F 55
A172 4E 44
A174 04                  FCB     4
A175 22          ERSTR   FCC     /"P" MUST BE FOLLOWED BY A COMMAND/
A176 50 22
A178 20 4D
A17A 55 53
A17C 54 20
A17E 42 45
A180 20 46
A182 4F 4C
A184 4C 4F
A186 57 45
A188 44 20
A18A 42 59
A18C 20 41
A18E 20 43
A190 4F 4D
A192 4D 41
A194 4E 44
A196 04                  FCB     4
A843                     ORG     $A843
A843 FF                  FCB     $FF
A844 50                  FCC     /PRINT/
A845 52 49
A847 4E 54
A849 00                  FCB     0,0,0
A84A 00
A84B 53                  FCC     /SYS/
A84C 59 53
                         END     P

NO ERROR(S) DETECTED

   SYMBOL TABLE:

EOL     AC02    ERSTR   A175    FCB     A840    FMS     B406    FMSCLS  B403
INDEX   0010    LOAD    AD30    LSTTRM  AC11    NFER    0004    NOPST   A15F
OUTCH   AD0F    P       A100    P1      A103    P12     A111    P15     A137
P2      A143    P25     A14C    P3      A151    P4      A154    P8      A15A
PINIT   ACC0    POUT    ACE4    PR1     ACFC    PSTRNG  AD1E    RENTER  AD06
RPTERR  AD3F    VN      A102    WARMS   AD03
```

```
                                NAM     P2.CMD
                                ORG     $A100
A100
0010                    INDEX   EQU     $0010
A840                    FCB     EQU     $A840
AD30                    LOAD    EQU     $AD30
B406                    FMS     EQU     $B406
B403                    FMSCLS  EQU     $B403
AD06                    RENTER  EQU     $AD06
0004                    NFER    EQU     $4
AD1E                    PSTRNG  EQU     $AD1E
AD3F                    RPTERR  EQU     $AD3F
AD03                    WARMS   EQU     $AD03
AC11                    LSTTRM  EQU     $AC11
AC02                    EOL     EQU     $AC02
ACC0                    PINIT   EQU     $ACC0
ACE4                    POUT    EQU     $ACE4
AD0F                    OUTCH   EQU     $AD0F
ACFC                    PR1     EQU     $ACFC
A100 20 01              P       BRA     P1
A102 03                 VN      FCB     3
A103 B6 AC FC   P1              LDA A   PR1
A106 27 09                      BEQ     P12
A108 CE A8 40                   LDX     #FCB
A10B C6 1B                      LDA B   #27
A10D E7 01                      STA B   1,X
A10F 20 40                      BRA     P3
A111 B6 AC 11   P12             LDA A   LSTTRM
A114 81 0D                      CMP A   #$D
A116 27 42                      BEQ     P8
A118 B1 AC 02                   CMP A   EOL
A11B 27 3D                      BEQ     P8
A11D B6 AC E4                   LDA A   POUT
A120 81 39                      CMP A   #$39
A122 26 13                      BNE     P15
A124 CE A8 40                   LDX     #FCB
A127 86 01                      LDA A   #1
A129 A7 00                      STA A   0,X
A12B BD B4 06                   JSR     FMS
A12E 26 13                      BNE     P2
A130 86 FF                      LDA A   #$FF
A132 A7 3B                      STA A   $3B,X
A134 BD AD 30                   JSR     LOAD
A137 BD AC C0   P15             JSR     PINIT
A13A CE AC E4                   LDX     #POUT
A13D FF AD 10                   STX     OUTCH+1
A140 7E AD 06                   JMP     RENTER
A143 A6 01      P2              LDA A   1,X
A145 81 04                      CMP A   #NFER
A147 26 08                      BNE     P3
A149 CE A1 5F                   LDX     #NOPST
A14C BD AD 1E   P25             JSR     PSTRNG
A14F 20 03                      BRA     P4
A151 BD AD 3F   P3              JSR     RPTERR
A154 BD B4 03   P4              JSR     FMSCLS
A157 7E AD 03                   JMP     WARMS
A15A CE A1 75   P8              LDX     #ERSTR
A15D 20 ED                      BRA     P25
```

```
1                                NAM     N.CMD
2                       *
3                       **********************
4                       * PAUSE=NO (OFF) *
5                       **********************
6                       *
7   A840                FCB     EQU     $A840
8   AC09                PAUSE   EQU     $AC09
9   0000                VALUE   EQU     $00
10  AD1E                PSTRNG  EQU     $AD1E
11  AD03                WARMS   EQU     $AD03
12                      *
13  A100                        ORG     $A100
14  A100 20 01          N       BRA     NP
15  A102 02             VN      FCB     2
16  A103 86 00          NP      LDA A   #VALUE
17  A105 B7 AC 09               STA A   PAUSE
18  A108 CE A1 11               LDX     #NOPST
19  A10B BD AD 1E               JSR     PSTRNG
20  A10E 7E AD 03               JMP     WARMS
21  A111 54             NOPST   FCC     /TTYSET,PS=OFF/
    A112 54 59
    A114 53 45
    A116 54 2C
    A118 50 53
    A11A 3D 4F
    A11C 46 46
    22  A11E 04                 FCB     4
    23                          END     N

NO ERROR(S) DETECTED

   SYMBOL TABLE:

FCB     A840    N       A100    NOPST   A111    NP      A103    PAUSE   AC09
PSTRNG  AD1E    VALUE   0000    VN      A102    WARMS   AD03
```





```
0005 REM COPYRIGHT ULF BOSTRÖM
0010 REM A=TABELL ÖVER ALLA SPELARES KORT
0020 REM M=VÄXEL SOM ANGER OM PASS ELLER SPEL SPELAS
0030 REM N ANGER VALÖR PÅ KORT
0040 REM O INNEHÅLLER NUMMER PÅ SPELAREN VARS KORT ÄR HÖGST
0050 REM P ANGE FÄRG PÅ RT OM EJ RÄT FÄRG FINNS
0060 REM S ANGER FÄRG PÅ SPELAT KORT
0070 REM W=VÄXEL SOM ANGER OM MEDSPELAREN HAR DRAGIT
0080 PRINT CHR$(5)
0085 PRINT
0090 PRINT "DETTA ÄR 4-MANS WHIST, KAN DU SPELA DET";
0100 INPUT Q$
0110 IF Q$="JA" THEN 210
0120 IF Q$="NEJ" THEN 160
0130 PRINT
0140 PRINT "SVARA BARA JA ELLER NEJ, TACK"
0150 GOTO 100
0160 PRINT
0170 PRINT "DET VAR ILLA - DÅ MÅSTE DU LÄRA DIG"
0180 PRINT
0190 PRINT "LYCKA TILL"
0200 GOTO 2280
0210 DIM A(4,13),B$(4),C$(13),E(6),K$(2),L$(4),V(2,4),W(4)
0212 PRINT
0213 PRINT "DRÖJ NÅGON MINUT....."
0214 PRINT "JAG DELAR SAKTA....."
0220 FOR I=1TO 4
0230 READ B$(I)
0240 DATA "HJÄRTER","RUTER","KLÖVER","SPADER"
0250 NEXT I
0260 FOR I=1TO4
0270 W(I)=0
0280 NEXT I
0290 FOR I=1TO13
0300 READ C$(I)
0310 DATA "TVÅ","TRE","FYRA","FEM","SEX","SJU"
0320 DATA "ÅTTA","NIO","TIO","KNEKT","DAM","KUNG","ESS"
0330 NEXT I
0340 FOR I=1TO2
0350 READ K$(I)
0360 DATA "PASS","SPEL"
0370 NEXT I
0380 FOR I=1 TO 4
0390 READ L$(I)
0400 DATA "DU","SPELARE 1","JAG DATORN","SPELARE 2"
0410 NEXT I
0420 FOR I=1TO4
0430 FOR J=1TO13
0440 A(I,J)=0
0450 NEXT J
0460 NEXT I
0470 FOR I=1TO4
0480 E(I)=1
0490 NEXT I
0500 FOR X=1TO13
0510 FOR Y=1TO3
0520 GOSUB 2290
0530 A(I,J)=Y
0540 NEXT Y
0550 NEXT X
```

```
0552 FOR I=1 TO 4
0554 FOR J=1 TO 13
0556 IF A(I,J)=0 THEN A(I,J)=4
0558 NEXT J
0559 NEXT I
0560 PRINT
0570 PRINT
0580 PRINT "OK, DÅ BÖRJAR VI. KORTEN ÄR UTDELADE."
0590 F=0
0600 GOSUB 2340
0610 FOR F=2TO4
0620 G=0
0630 GOSUB 2510
0640 IF G>4 THEN E(F)=2
0650 NEXT F
0660 PRINT
0670 PRINT "VAD VÄLJER DU, PASS ELLER SPEL";
0680 INPUT H$
0690 IF H$=K$(1) THEN 720
0700 IF H$<>K$(2) THEN 670
0710 E(1)=2
0720 REM DAGS ATT PRINTA SPELPLAN
0730 PRINT "    "
0740 FOR I=1 TO 5
0750 PRINT
0760 NEXT I
0770 PRINT "        ";L$(3);" VALDE ";K$(E(3))
0780 FOR I=1TO5
0790 PRINT
0800 NEXT I
0810 PRINT L$(2);"-";K$(E(2));"    ";L$(4);"-";K$(E(4))
0820 FOR I=1TO5
0830 PRINT
0840 NEXT I
0850 PRINT "        ";L$(1);" VALDE ";K$(E(1))
0860 FOR I=1TO4
0870 PRINT
0880 NEXT I
0890 M=0
0900 FOR I=1TO4
0910 IF E(I)=1 THEN 940
0920 M=I
0930 GOTO 960
0940 NEXT I
0950 IF M=0 THEN 1000
0960 PRINT L$(I);" VALDE ";K$(2)
0970 I=I-1
0980 IF I=0 THEN I=4
0990 GOTO 1020
1000 PRINT "ALLA VALDE ";K$(1)
1010 I=1
1020 PRINT "DET BETYDER ATT ";L$(I);" BÖRJAR"
1030 PRINT
1040 PRINT "ÄR DU KLAR";
1050 INPUT Q$
1060 IF Q$<>"JA" THEN 1040
1070 T=T+1
1080 PRINT
1090 ON I GOTO 1100,1270,1390,1540
1100 F=0
```

```
1110 GOSUB 2340
1120 GOSUB 3320
1130 IF W(1)=1 THEN 1170
1140 IF M=0 THEN V(1,1)=I
1150 IF M>0 THEN V(2,1)=I
1160 W(1)=1
1170 S=X
1180 O=1
1190 I=1
1200 GOSUB 3600
1210 Y=J
1220 FOR I=2TO4
1230 GOSUB 4240
1240 GOSUB 3600
1250 NEXT I
1260 GOTO 1680
1270 O=2
1280 GOSUB 3800
1290 FOR I=2TO4
1300 GOSUB 4240
1310 GOSUB 3600
1320 NEXT I
1330 GOSUB 4150
1340 GOSUB 4090
1350 GOSUB 2340
1360 GOSUB 3320
1370 GOSUB 3690
1380 GOTO 1680
1390 O=3
1400 GOSUB 3800
1410 FOR I=3TO4
1420 GOSUB 4240
1430 GOSUB 3600
1440 NEXT I
1450 GOSUB 4150
1460 GOSUB 4090
1470 GOSUB 2340
1480 GOSUB 3320
1490 GOSUB 3690
1500 I=2
1510 GOSUB 4240
1520 GOSUB 3600
1530 GOTO 1680
1540 O=4
1550 GOSUB 3800
1560 I=4
1570 GOSUB 4240
1580 GOSUB 3600
1590 GOSUB 4150
1600 GOSUB 4090
1610 GOSUB 2340
1620 GOSUB 3320
1630 GOSUB 3690
1640 FOR I=2TO3
1650 GOSUB 4240
1660 GOSUB 3600
1670 NEXT I
1680 PRINT
1690 PRINT
1700 PRINT L$(O);" TOG STICKET"
```

```
1710 PRINT
1720 PRINT
1730 PRINT
1740 ON 0 GOTO 1770,1750,1770,1750
1750 E(6)=E(6)+1
1760 GOTO 1780
1770 E(5)=E(5)+1
1780 I=0
1790 IF T=13 THEN 1850
1800 PRINT
1810 PRINT "VI HAR ";E(5);"       DOM HAR ";E(6)
1815 PRINT
1820 PRINT "DAGS FÖR NÄSTA - SKRIV N NÄR DU ÄR KLAR"
1830 INPUT Q$
1835 PRINT CHR$(5)
1840 GOTO 1070
1850 PRINT
1860 PRINT
1870 PRINT
1880 PRINT "JAHA, DÅ SKALL VI SE..."
1890 PRINT
1900 PRINT L$(1);" OCH ";L$(3);" FICK ";E(5);" STICK OCH ";
1905 PRINT
1910 PRINT L$(2);" OCH ";L$(4);"  ";E(6);" STICK"
1930 PRINT
1940 T=E(6) - E(5)
1950 IF M>0 THEN 2050
1960 PRINT
1970 PRINT "ALLA VALDE ";K$(1)
1980 IF T>0 THEN 2020
1990 I=2
2000 J=4
2010 GOTO 2210
2020 I=1
2030 J=3
2040 GOTO 2210
2050 IF T>0 THEN 2160
2060 IF M=1 THEN 2080
2070 IF M<>3 THEN 2130
2080 PRINT
2090 PRINT L$(M);" VALDE ";K$(2)
2100 IF M=1 THEN 2020
2110 IF M=3 THEN 2020
2120 GOTO 1990
2130 PRINT
2140 PRINT "BRAVO!! NI TOG STRAFF PÅ DOM.."
2150 GOTO 2020
2160 IF M=2 THEN 2080
2170 IF M=4 THEN 2080
2180 PRINT
2190 PRINT "AJ AJ, DET GICK INGET VIDARE DET DÄR, BARA ";E(5);" STICK
2200 GOTO 1990
2210 PRINT
2220 PRINT "OCH DET BETYDER ATT ";L$(I);" OCH ";L$(J);" HAR VUNNIT"
2230 PRINT
2240 PRINT "VILL DU PRÖVA LYCKAN IGEN";
2250 INPUT Q$
2260 IF Q$="JA"THEN 420
2270 IF Q$<>"NEJ" THEN 2240
2280 END
```

ECBUG.

Här presenteras programlistan till CÅ:s lilla terminalkort för nybörjare vilket beskrevs i Radio & Television nr 5, 1980. Denna lista gäller monitorprogrammet för 6800-versionen, kallat ECBUG. Versionen för 6809 kallas FOBUG och den listan kommer i ett senare nummer av MPU-laren.

RH

```
1                NAM    ECBUG
2              *
3              * COPYRIGHT CÅ-ELEKTRONIK AB
4              *          BOX 2010
5              *          S-135 02 TYRESÖ SWEDEN
6              *
7              * 1980-03-20
8              *
9              ************************************************************
10             *
11             * MONITORPROGRAM FÖR MC6808
12             * PROGRAMMET INGÅR SOM EN DEL I ETT DATORBYGGE
13             * FÖR NYBÖRJARE OCH ENKLARE TILLÄMPNINGAR.
14             * DATORN BESTÅR AV :
15             * CPU-KORT CÅ-6819 MED MC6808 OCH 1 K DATAMINNE (RAM)
16             * MODERKORT CÅ-800 MED 2 KONTAKTER (J15 OCH J16).
17             * DISPLAYKORT CÅ-8006 MED TANGENTKORT
18             * PROGRAMMINNE ECBUG I EPROM (PÅ KORT CÅ-6819).
19             *
20             ************************************************************
21             * FUNGERAR SOM MOTOROLAS D2-KIT
22             * MEN HAR STATISK DISPLAY
23             * ECBUG FINNS ÄVEN FÖR 6809 CPU OCH HETER DÅ FOBUG.
24             ************************************************************
25             *
26             * TANGENTERNAS FUNKTIONER :
27             * 0-9 SAMT A-F HEXADECIMAL TECKENINMATNING
28             * A-F ÄR ÄVEN KOMMANDOTANGENTER
29             * A ALTER MEMORY, MINNESMANIPULERING
30             * B SÄTT IN EN BRYTPUNKT
31             * C CONTINUE, FORTSÄTT PROGRAMMET FRÅN BRYTPUNKT
32             * D DELETE BRKPT, TA BORT ALLA BRYTPUNKTER
33             * E EXECUTE PROGRAM, GÅ TILL ADRESS OCH STARTA PROGRAM
34             * L LADDA MINNET FRÅN KASSETT S1-FORMAT
35             * P PUNCH MEMORY, SPELA IN DATA PÅ KASSETT S1-FORMAT
36             * R DISPLAY REGISTERS
37             *
38             *
39             * ECBUG LIGGER I EPROM 2708 PÅ ADRESS $EC00
```

```
40                *
41                * DE SEX LYSDIODDISPLAYERNA ADRESSERAS SOM RAM-CELLER.
42                * BIT 0-3 GER ETT HEXADECIMALT TAL MELLAN 0 OCH F
43                * BIT 4-6 ANVÄNDS EJ
44                * BIT 7 HÖG TÄNDER DECIMALPUNKTEN
45                *
46                * DATA FRÅN TANGENTBORDET ERHÅLLS SOM
47                * ETT TAL MELLAN $00 OCH $13
48                *
49  80E0    DS1       EQU     $80E0     MSB ADRESSINDIKATOR (VÄNSTRA DISPLAYEN)
50  80E1    DS2       EQU     $80E1
51  80E2    DS3       EQU     $80E2
52  80E3    DS4       EQU     $80E3     LSB ADRESS
53  80E4    DS5       EQU     $80E4     MSB DATA
54  80E5    DS6       EQU     $80E5     LSB DATAINDIKATOR
55                *
56  80E6    KEY       EQU     $80E6     DATA FRÅN TANGENTBORD
57  80E8    INFLG     EQU     $80E8     BIT 7 ÄR NOLL OM TANGENT ÄR TRYCKT
58  80EE    BELL      EQU     $80EE     ACCESS GER PULS TILL BELLKORT CÅ-6805
59  008C    SKIP2     EQU     $8C
60  8020    TAPE      EQU     $8020     ACIA FÖR KCS-MODEM
61  8021    TAPED     EQU     $8021
62                *
63                * RAMCELLER FÖR DATALAGRING
64                *
65  A000              ORG     $A000
66  A000    RAM       EQU     *
67                *
68  A000    IRQV      RMB     2         IRQ-VEKTOR
69  A002    BEGA      RMB     2         STARTADRESS VID PUNCH MM
70  A004    ENDA      RMB     2         SLUTADRESS VID PUNCH MM
71  A006    NMIV      RMB     2         NMI VEKTOR
72  A008    SPEK      RMB     2         ANVÄNDARENS STACKPEKARE
73  A00A    CKSM      RMB     1         CHECKSUMMA
74  A00B    BYTECT    RMB     1         BYTE RÄKNARE
75  A00C    XHI       RMB     1         TEMP LAGRING AV X-REGISTER MSB-HALVA
76  A00D    XLO       RMB     1         TEMP LAGRING AV X-REGISTER LSB-HALVA
77  A00E    SWIVEK    RMB     2         EGEN SWI-VEKTOR
78  A010    VFLAG     RMB     1         ANTAL INSATTA BRYTPUNKTER
79  A011    BPTAB     RMB     15        TABELL MED MAX 5 BRYTPUNKTER
80  A020    RDS1      RMB     1         RAMCELL FÖR DS1
81  A021    RDS2      RMB     1
82  A022    RDS3      RMB     1
83  A023    RDS4      RMB     1
84  A024    RDS5      RMB     1
85  A025    RDS6      RMB     1
86  A026    MCONT     RMB     1
87  A027    ADRESS    RMB     2         PEKARE FÖR AKTUELL DISPLAY
88  A029    DSPEK     RMB     2         CELL FÖR BLINKANDE DISPLAY
89  A02B    BLRAM     RMB     2
90  A02D    BLFLG     RMB     1         FLAGGA FÖR BLINKNING
91  A02E    DPFLG     RMB     1         1=TÄND DEC.PUNKT, 0=SLÄCKT
92  A02F    BLTID     RMB     2         CELL FÖR BLINKHASTIGHET
93  A031    BUFF      RMB     10
94  A07F    STACK     EQU     RAM+$7F
95                *
96  EC00              ORG     $EC00
97  EC00    PGM       EQU     *
98                *
99                * SERVICERUTIN FÖR IRQ
```

```
100                 *
101   EC00 FE A0 00 IRQSER  LDX     IRQV       ADRESS FÖR IRQ-RUTIN
102   EC03 6E 00    JMP     0,X        HOPPA DIT OCH TAG HAND OM AVBR
103                 *
104                 * SERVICERUTIN FÖR NMI
105                 *
106   EC05 FE A0 06 NMISER  LDX     NMIV       ADRESS FÖR NMI-RUTIN
107   EC08 6E 00    JMP     0,X        HOPPA DIT OCH TAG HAND OM AVBR
108                 *
109                 * SERVICERUTIN FÖR SWI
110                 *
111   EC0A FE A0 0E SWISER  LDX     SWIVEK     ADRESS FÖR SWI-RUTIN
112   EC0D 6E 00    JMP     0,X        HOPPA DIT OCH TAG HAND OM AVBR
113                 *
114   EC0F CE 80 E0 CLEARD  LDX     #DS1       RUTIN FÖR ATT NOLLA IN
115   EC12 C6 06    CLEAR   LDA B   #6
116   EC14 6F 00    CL300   CLR     0,X
117   EC16 08               INX
118   EC17 5A               DEC B                 6 POSITIONER
119   EC18 26 FA            BNE     CL300
120   EC1A 39               RTS
121                 *
122   EC1B CE ED DD START   LDX     #RETINT
123   EC1E FF A0 00 STX     IRQV
124   EC21 CE ED DE LDX     #SFE       ADRESS FÖR SWI
125   EC24 FF A0 06 STX     NMIV
126   EC27 FF A0 0E STX     SWIVEK
127   EC2A 8E A0 77 LDS     #STACK-8   INITIERA STACKPEKAREN
128   EC2D BF A0 08 STS     SPEK
129   EC30 86 03    LDA A   #3         NOLLA ACIA FÖR KCS
130   EC32 B7 80 20 STA A   $8020
131   EC35 7F A0 2E CLR     DPFLG
132   EC38 7F A0 10 KILL    CLR     VFLAG
133   EC3B BE A0 08 CONTRL  LDS     SPEK       FRISKA UPP STACKEN
134   EC3E 8D CF    BSR     CLEARD
135   EC40 CE A0 20 ST200   LDX     #RDS1      NOLLSTÄLL ÄVEN RAMCELL
136   EC43 8D CD    BSR     CLEAR
137   EC45 8D 29    BSR     LBLINK     LÅNGSAMT BLINK
138   EC47 BD EC E0 JSR     DEL200     LÄGG IN BLTID I BLRAM
139                 *
140                 * ÅTERSTÄLL DISPLAYPEKAREN PÅ FÖRSTA DISPLAYEN OCH TÄND
141                 * BLINKANDE DECIMALPUNKT
142                 *
143   EC4A CE A0 20 LDX     #RDS1      ADRESS TILL FÖRSTA DISPLAYEN
144   EC4D FF A0 29 STX     DSPEK      ADRESSEN TILL FÖRSTA DISPLAYEN
145   EC50 8D 2A    LOOP10  BSR     INCH       HÄMTA TECKEN
146   EC52 8D 1F    BSR     SBLINK
147   EC54 CE EF DF LDX     #INTAB     KOLLA MED KOMMANDOTABELLEN
148   EC57 A1 00    LOOP20  CMP A   0,X        OM TECKNET FINNS
149   EC59 27 0F    BEQ     MATCH      I SÅ FALL HOPPA
150   EC5B 8C EF F7 CPX     #TABEND    SLUT PÅ TABELLEN ?
151   EC5E 26 05    BNE     LOOP30     FEL TECKEN INMATAT
152   EC60 7D 80 EE TST     BELL       GE PIIP
153   EC63 20 EB    BRA     LOOP10
154                 *
155   EC65 08       LOOP30  INX
156   EC66 08               INX                   GÅ TILL NÄSTA TECKEN I TABELLE
157   EC67 08               INX
158   EC68 20 ED    BRA     LOOP20     OCH KOLLA VIDARE
159                 *
```

```
160  EC6A EE 01     MATCH   LDX    1,X       HÄMTA ADRESS TILL SUBRUTINEN
161  EC6C AD 00     JSR    0,X        OCH UTFÖR KOMMANDOT
162  EC6E 20 CB     BRA    CONTRL     GÅ SEDAN TILLBAKA
163                 *
164                 * LÅNGSAM BLINKNING
165                 *
166  EC70 C6 22     LBLINK  LDA B  #$22
167  EC72 8C        FCB    SKIP2
168                 *
169                 * SNABB BLINKNING
170                 *
171  EC73 C6 OF     SBLINK  LDA B  #$OF
172  EC75 F7 AO 2F  STA B  BLTID
173  EC78 F7 AO 2D  STA B  BLFLG
174  EC7B 39        RTS
175                 *
176                 * SUBRUTIN FÖR ATT HÄMTA IN TECKEN FRÅN TANGENTBORDET
177                 * DECIMALPUNKTEN PÅ DISPLAYEN VARS ADRESS FINNS I DSPEK
178                 * BLINKAR MED EN HASTIGHET SOM BEROR AV VÄRDET I BLTID.
179                 *
180  EC7C 37        INCH    PSH B     SPARA B-ACK
181  EC7D FF AO 31  STX    BUFF       SPARA INDEXREGISTRET
182  EC80 8D 55     IN100   BSR    DELAY     BLINKRUTINEN
183  EC82 7D AO 2D  TST    BLFLG      SKALL PUNKTEN SLÄCKAS ELLER TÄNDAS ?
184  EC85 27 02     BEQ    IN250      INGEN ÄNDRING=O
185  EC87 8D 2F     BSR    STODSP     LÄGG UT PÅ DISPLAYEN
186                 *
187  EC89 B6 80 E8  IN250   LDA A  INFLG      ÄR KNAPP INTRYCKT ?
188  EC8C 2B F2     BMI    IN100      NEJ, GÅ RUNT IGEN
189  EC8E 7F AO 2E  CLR    DPFLG      SLÄCK DEC.PUNKTEN
190  EC91 8D 25     BSR    STODSP
191  EC93 B6 80 E6  LDA A  KEY        HÄMTA DATA
192  EC96 84 1F     AND A  #$1F       MASKA DE TRE HÖGSTA BITARNA
193  EC98 FE AO 31  LDX    BUFF
194  EC9B 33        PUL B
195  EC9C 39        RTS
196                 *
197                 * SUBRUTIN FÖR ATT FLYTTA A+B ACKUM. TILL X-REG.
198                 *
199  EC9D B7 AO OC  TFRDX   STA A  XHI
200  ECAO F7 AO OD  STA B  XLO
201  ECA3 FE AO OC  LDX    XHI
202  ECA6 39        RTS
203                 *
204                 * SUBRUTIN FÖR ATT FLYTTA X TILL A+B
205                 *
206  ECA7 FF AO OC  TFRXD   STX    XHI
207  ECAA B6 AO OC  LDA A  XHI
208  ECAD F6 AO OD  LDA B  XLO
209  ECBO 39        RTS
210                 *
211                 * RUTIN FÖR ATT LÄGGA A-ACK I RAM OCH DISPLAY
212                 *
213  ECB1 FE AO 29  STODS   LDX    DSPEK     HÄMTA ADRESSEN
214  ECB4 A7 00     STA A  0,X        LÄGG I RAM
215  ECB6 20 OD     BRA    DPOFF      OCH PÅ DISPLAY
216                 *
217  ECB8 FE AO 29  STODSP  LDX    DSPEK     HÄMTA ADRESSEN
218  ECBB A6 00     LDA A  0,X        OCH TECKNET
219  ECBD 7D AO 2E  TST    DPFLG
```

```
220  ECC0 27 03      BEQ    DPOFF
221  ECC2 8B 80      DPON    ADD A  #$80      LÄGG TILL DECIMALPUNKTE
222  ECC4 8C         FCB    SKIP2      HOPPA ÖVER NÄSTA INSTRUKTION
223                  *
224  ECC5 84 0F      DPOFF   AND A  #$0F      INGEN DEC.PT.
225  ECC7 36         IN200   PSH A     SPARA A
226  ECC8 8D DD      BSR    TFRXD      FLYTTA X TILL A &B
227  ECCA 80 20      SUB A  #$20       PEKAREN STÅR NU PÅ
228  ECCC CB C0      ADD B  #$C0
229  ECCE 8D CD      BSR    TFRDX      DISPLAYENS ADRESS
230  ECD0 32         PUL A             ÅTERTAG A
231  ECD1 A7 00      STA A  0,X        LÄGG UT PÅ DISPLAYEN
232  ECD3 7F A0 2D   CLR    BLFLG
233  ECD6 39         RTS
234                  *
235                  * SUBRUTIN FÖR BLINKFUNKTIONEN
236                  *
237  ECD7 FE A0 2B   DELAY   LDX    BLRAM     HÄMTA VÄRDET PÅ TIDSCEL
238  ECDA 09         DEX               MINSKA
239  ECDB FF A0 2B   STX    BLRAM      LÄGG TILLBAKA
240  ECDE 26 0C      BNE    DEL500     OM X EJ ÄR NOLL GÅ RTS
241  ECE0 FE A0 2F   DEL200  LDX    BLTID     NYTT VÄRDE PÅ BLINKCELL
242  ECE3 FF A0 2B   STX    BLRAM      LÄGG IN DET
243  ECE6 7C A0 2D   INC    BLFLG      VÄXLA FLAGGAN
244  ECE9 73 A0 2E   COM    DPFLG
245  ECEC 39         DEL500  RTS
246                  *
247                  * TAG IN HEXADECIMALT TECKEN
248                  *
249  ECED 8D 8D      INHEX   BSR    INCH
250  ECEF 81 0F      CMP A  #$0F       KOLLA OM HEX
251  ECF1 23 F9      BLS    DEL500
252  ECF3 7D 80 EE   TST    BELL       GE PIIP OM EJ HEXADECIMALT
253  ECF6 20 F5      BRA    INHEX      OM EJ HEX SÅ FÖRSÖK IGEN
254                  *
255                  * BYGG ADRESS I X-REG + "ADRESS"
256                  *
257  ECF8 8D 0C      BADDR   BSR    BYTE      HÖGSTA 8 BITARNA
258  ECFA B7 A0 27   STA A  ADRESS
259  ECFD 8D 07      BSR    BYTE       LÄGSTA 8 BITARNA
260  ECFF B7 A0 28   BADDR7  STA A  ADRESS+1
261  ED02 FE A0 27   LDX    ADRESS
262  ED05 39         RTS
263                  *
264                  * TAG IN TVÅ HEX. TECKEN I A-ACK OCH LAGRA I DSPEK
265                  *
266  ED06 8D E5      BYTE    BSR    INHEX     IN ETT HEXADECIMALT TAL
267  ED08 8D A7      BYTE2   BSR    STODS     LÄGG UT PÅ DISPLAY
268  ED0A 8D 2B      BSR    INCDSP     INCREMENTA DSPEK
269  ED0C 48         ASL A
270  ED0D 48         ASL A
271  ED0E 48         ASL A
272  ED0F 48         ASL A             HHHH 0000
273  ED10 16         TAB
274  ED11 37         PSH B             SPARA B-ACK
275  ED12 8D D9      BSR    INHEX      TAG IN DET ANDRA TECKNET
276  ED14 8D 9B      BSR    STODS      LÄGG UT PÅ DISPLAYEN
277  ED16 33         PUL B
278  ED17 1B         ABA
279  ED18 20 1D      BRA    INCDSP
```

```
280                *
281                * LÄGG UT PÅ ADRESSDISPLAYEN
282                *
283  ED1A CE A0 20  UTADR   LDX    #RDS1      RAM-CELL FÖR DS1
284  ED1D FF A0 29  STX     DSPEK
285  ED20 CE A0 27  LDX     #ADRESS    VÄRDE SOM SKALL UT
286  ED23 8D 03     BSR     OUT2H      MSB BYTE
287  ED25 CE A0 28  LDX     #ADRESS+1
288                *
289                * MATA UT 2 HEX TECKEN
290                *
291  ED28 A6 00     OUT2H   LDA A  0,X        HÄMTA EN BYTE
292  ED2A 36        OUT2H2  PSH A
293  ED2B 44                LSR A
294  ED2C 44                LSR A
295  ED2D 44                LSR A
296  ED2E 44                LSR A
297  ED2F 8D 80             BSR     STODS     LÄGG UT DET FÖRSTA TECKNET
298  ED31 32                PUL A
299  ED32 8D 03             BSR     INCDSP
300  ED34 BD EC B1          JSR     STODS     LÄGG UT DET ANDRA TECKNET
301  ED37 7C A0 2A  INCDSP   INC    DSPEK+1   PEKA PÅ NÄSTA
302  ED3A 39        INHEX5  RTS
303                *
304                * MINNESMANIPULERING
305                *
306  ED3B 8D BB     MEMCHG  BSR    BADDR     BEGÄR FÖRSTA ADRESS
307  ED3D 8D DB     MEM10   BSR    UTADR     LÄGG UT ADRESS PÅ DISPLAY
308  ED3F FE A0 27  LDX     ADRESS
309  ED42 BD ED B6  JSR     DATAIX    LÄGG UT DATA PÅ DISPLAYEN
310  ED45 BD EC 7C  MEM15    JSR    INCH      HÄMTA TECKEN
311  ED48 81 12             CMP A  #$12      OM RETURN GÅ RTS "R"
312  ED4A 27 EE             BEQ    INHEX5    GÅ TILL RTS
313  ED4C 81 11     MEM20    CMP A  #$11      FÖREGÅENDE ? "P"
314  ED4E 27 44             BEQ    MEM88
315  ED50 81 10             CMP A  #$10      NÄSTA ? "L"
316  ED52 27 48             BEQ    MEM94
317  ED54 81 13             CMP A  #$13      OFFSETBERÄKNING ? "S"
318  ED56 26 2C             BNE    MEM60
319  ED58 FE A0 27  LDX     ADRESS
320  ED5B FF A0 02  STX     BEGA      SPARA MINNESPOSITIONEN
321  ED5E CE A0 20  LDX     #RDS1     PEKA PÅ FÖRSTA DISPLAYEN
322  ED61 FF A0 29  STX     DSPEK
323  ED64 8D 92     BSR     BADDR     BEGÄR IN HOPPADRESSEN
324  ED66 09        DEX               KORRIGERA FÖR OFFSET
325  ED67 FF A0 04  STX     ENDA      SPARA
326  ED6A 8D AE     BSR     UTADR     LÄGG UT PÅ DISPLAYEN
327  ED6C BD EE AA  JSR     CMPADR    JÄMFÖR ADRESSERNA
328  ED6F FE A0 02  LDX     BEGA
329  ED72 FF A0 27  STX     ADRESS
330                *
331                * OM B=00 INOM RÄCKHÅLL FRAMÅT OM A<80
332                * OM B=FF INOM RÄCKHÅLL BAKÅT OM A>7F
333                *
334  ED75 4D              TST A             FRAM/BAKÅT
335  ED76 2B 04           BMI    OK         HOPPA OM BAKÅT
336  ED78 5D              TST B             =0 OM FRAMÅT
337  ED79 26 14           BNE    MEMFEL     PIIP OM FÖR STOR OFFSET
338  ED7B 5A              DEC B             B=FF NU
339  ED7C 5C              OK      INC B
```

```
340  ED7D 26 10      BNE    MEMFEL    FEL I OFFSET BAKÅT
341  ED7F 36         PSH A            SPARA OFFSETVÄRDET
342  ED80 8D 36      BSR    DATADS    SKRIV UT OFFSET PÅ DISPLAYEN
343  ED82 32         PUL A            ÅTERTAG OFFSETVÄRDET
344  ED83 8C         FCB    SKIP2     HOPPA ÖVER NÄSTA INSTRUKTION
345                  *
346  ED84 8D 82      MEM60   BSR    BYTE2      NYTT DATA
347  ED86 FE A0 27   LDX     ADRESS
348  ED89 A7 00      MEM62    STA A 0,X        LAGRA I RAM
349  ED8B A1 00      CMP A   0,X       FASTNADE DET ?
350  ED8D 27 08      BEQ     MEM92       NIX
351  ED8F 7D 80 EE   MEMFEL   TST     BELL       GE PIIP
352  ED92 20 09      BRA     MEM95
353                  *
354                  * GÅ TILL FÖREGÅENDE ADRESS I MINNET
355                  *
356  ED94 09         MEM88   DEX
357  ED95 20 06      BRA     MEM95
358                  *
359                  * STEGA TILL NÄSTA ADRESSPOSITION
360                  *
361  ED97 8D 09      MEM92   BSR     DELAYD
362  ED99 FE A0 27   MEM93   LDX     ADRESS
363  ED9C 08         MEM94   INX
364  ED9D FF A0 27   MEM95   STX     ADRESS     LAGRA NY ADRESS
365  EDA0 20 9B      BRA     MEM10
366                  *
367                  * FÖRDRÖJNINGSLOOP FÖR ATT HINNA LÄSA
368                  * AV ADRESS-DATA DISPLAYEN
369                  *
370  EDA2 FF A0 0C   DELAYD  STX     XHI
371  EDA5 37         PSH B             SPARA X OCH B
372  EDA6 C6 02      LDA B  #2
373  EDA8 CE AF FF   DELLDX  LDX     #$AFFF
374  EDAB 09         DELDEX  DEX                MINSKA X-LOOPEN
375  EDAC 26 FD              BNE     DELDEX
376  EDAE 5A                 DEC B
377  EDAF 26 F7              BNE     DELLDX
378  EDB1 33                 PUL B
379  EDB2 FE A0 0C           LDX     XHI
380  EDB5 39                 RTS
381                  *
382                  * LÄGG UT A-ACK PÅ DATADISPLAYEN
383                  *
384  EDB6 A6 00      DATAIX  LDA A  0,X         HÄMTA INDEXERAT
385  EDB8 FF A0 02   DATADS  STX    BEGA        SPARA X
386  EDBB 37                 PSH B              SPARA B
387  EDBC BD ED 2A           JSR    OUT2H2      LÄGG UT
388  EDBF 33                 PUL B
389  EDC0 CE A0 24   DEC2DS  LDX    #RDS5
390  EDC3 FF A0 29           STX    DSPEK
391  EDC6 FE A0 02          LDX    BEGA        ÅTERTAG X
392  EDC9 39                 RTS
393                  *
394                  *
395                  * GÅ TILL ADRESS OCH STARTA PROGRAM
396                  *
397  EDCA BD EC F8   GOEXEC  JSR    BADDR       HÄMTA IN EN ADRESS
398  EDCD FE A0 08           LDX    SPEK        LÄGG IN PC-VÄRDET
399  EDD0 A7 07              STA A  7,X         PÅ STACKEN
```

```
400   EDD2 B6 A0 27          LDA A  ADRESS
401   EDD5 A7 06             STA A  6,X
402                  *
403                  * CONTINUE
404                  *
405   EDD7 BD EE 90  CONTIN  JSR    BRKPIN
406   EDDA BE A0 08          LDS    SPEK       HÄMTA USER STACKPOINTER
407   EDDD 3B        RETINT  RTI               OCH GÖR EN RTI
408                  *
409                  * SWI SERVICERUTIN
410                  *
411   EDDE BF A0 08  SFE     STS    SPEK
412   EDE1 30                TSX
413   EDE2 6D 06             TST    6,X        DECREMENTERA
414   EDE4 26 02             BNE    *+4
415   EDE6 6A 05             DEC    5,X
416   EDE8 6A 06             DEC    6,X
417   EDEA 8D 7A             BSR    BRKP40     FINNS BREAKPOINTS ?
418   EDEC 27 14             BEQ    PRREG      PRINTA REGISTREN
419                *
420                * TAG BORT BREAKPOINTS I ECBUG
421                *
422   EDEE FF A0 0C  BRKP00  STX    XHI
423   EDF1 A6 02             LDA A  2,X        SKYDDA MOT MULTIDEF. BRKPT
424   EDF3 81 3F             CMP A  #$3F
425   EDF5 27 07             BEQ    BRKP10
426   EDF7 EE 00             LDX    0,X
427   EDF9 A7 00             STA A  0,X
428   EDFB FE A0 0C          LDX    XHI
429   EDFE 8D 61     BRKP10  BSR    ADD3X
430   EE00 26 EC             BNE    BRKP00
431                *
432                * PRESENTERA REGISTREN
433                *
434   EE02 BD EC 0F  PRREG   JSR    CLEARD     NOLLA INDIKATORERNA
435   EE05 8D B9             BSR    DEC2DS     PEKA PÅ DS5
436   EE07 FE A0 08          LDX    SPEK       HÄMTA STACKADRESSEN
437   EE0A 08                INX               PEKAR NU PÅ CC-REG
438   EE0B 8D A9             BSR    DATAIX
439   EE0D 8D 37             BSR    INTECK     VÄNTA IN NYTT TECKEN
440   EE0F 08                INX
441   EE10 8D A4             BSR    DATAIX     UT MED B-ACK
442   EE12 8D 32             BSR    INTECK
443   EE14 08                INX
444   EE15 8D 9F             BSR    DATAIX     UT MED A-ACK
445   EE17 8D 2D             BSR    INTECK
446   EE19 08                INX
447   EE1A FF A0 02          STX    BEGA
448   EE1D EE 00             LDX    0,X
449   EE1F FF A0 27          STX    ADRESS     VÄRDET PÅ X I STACKEN
450   EE22 BD ED 1A          JSR    UTADR      UT MED X-REG
451   EE25 FE A0 02          LDX    BEGA
452   EE28 08                INX
453   EE29 08                INX
454   EE2A FF A0 02          STX    BEGA
455   EE2D EE 00             LDX    0,X
456   EE2F FF A0 27          STX    ADRESS     VÄRDET PÅ PC
457   EE32 8D 12             BSR    INTECK
458   EE34 BD ED 1A          JSR    UTADR      UT MED PC
459   EE37 FE A0 08          LDX    SPEK       STACKVÄRDET
```

```
460  EE3A FF A0 27          STX    ADRESS
461  EE3D 8D 07             BSR    INTECK
462  EE3F BD ED 1A          JSR    UTADR      UT MED STACKPEKAREN
463  EE42 8D 02             BSR    INTECK
464  EE44 20 BC             BRA    PRREG
465                         *
466  EE46 B6 80 E8  INTECK  LDA A  INFLG      KOLLA OM KNAPP TRYCKT
467  EE49 2B FB             BMI    INTECK
468  EE4B B6 80 E6          LDA A  KEY        HÄMTA TECKNET
469  EE4E 84 1F             AND A  #$1F       MASKA D5-D7
470  EE50 81 13             CMP A  #$13       KOLLA OM S-KNAPPEN
471  EE52 27 03             BEQ    INTRTS     JO
472  EE54 7E EC 3B          JMP    CONTRL     HOPPA TILLBAKS
473                         *
474  EE57 FF A0 0C  INTRTS  STX    XHI
475  EE5A BD EC 0F          JSR    CLEARD     NOLLA DISPLAYEN FÖRST
476  EE5D FE A0 0C          LDX    XHI
477  EE60 39                RTS
478                         *
479  EE61 08        ADD3X   INX
480  EE62 08                INX
481  EE63 08                INX
482  EE64 5A                DEC B
483  EE65 39                RTS
484                         *
485  EE66 CE A0 11  BRKP40  LDX    #BPTAB     TABELL ÖVER INSATTA BREA
486  EE69 F6 A0 10          LDA B  VFLAG      ANTAL BRKP'S
487  EE6C 39        BRKRTS  RTS
488                         *
489                         * V-KOMMANDO SÄTT IN EN BRYTPUNKT
490                         *
491  EE6D BD EC F8  SETBR   JSR    BADDR      F
492  EE70 8D F4             BSR    BRKP40     FINNS LEDIG PLATS ?
493  EE72 27 0C             BEQ    BRKP60     OM LEDIG SÅ STOPPA IN DE
494  EE74 C1 05             CMP B  #5         MAX FEM BRYTPUNKTER
495  EE76 25 04             BCS    BRKP50     OK
496  EE78 7D 80 EE  BELRTS  TST    BELL       GE PIIP OM FÖR MÅNGA
497  EE7B 39                RTS
498                         *
499  EE7C 8D E3     BRKP50  BSR    ADD3X      ÖKA X MED 3
500  EE7E 26 FC             BNE    BRKP50
501                         *
502  EE80 7C A0 10  BRKP60  INC    VFLAG      ÖKA ANTALSRÄKNAREN
503  EE83 B6 A0 27          LDA A  ADRESS     ADRESS TILL MINNET
504  EE86 A7 00             STA A  0,X        IN I TABELLEN
505  EE88 B6 A0 28          LDA A  ADRESS+1
506  EE8B A7 01             STA A  1,X        LSB-DELEN
507  EE8D 7E ED A2          JMP    DELAYD     VÄNTA EN STUND
508                         *
509                         * STOPPA IN ALLA BRKPT'S I MINNET VID 'G'
510                         *
511  EE90 8D D4     BRKPIN  BSR    BRKP40
512  EE92 27 22     BRKPRT  BEQ    CMPRTS
513                         *
514  EE94 FF A0 0C  TGB     STX    XHI
515  EE97 EE 00             LDX    0,X        HÄMTA ADRESSEN
516  EE99 A6 00             LDA A  0,X        HÄMTA INSTRUKTIONEN
517  EE9B 36                PSH A
518  EE9C 86 3F             LDA A  #$3F       OPCODE FÖR SWI
519  EE9E A7 00             STA A  0,X        I MINNET
```

```
520  EEA0 FE A0 0C            LDX    XHI
521  EEA3 32                  PUL A
522  EEA4 A7 02               STA A  2,X          SPARAD OPCODE
523  EEA6 8D B9               BSR    ADD3X
524  EEA8 20 E8               BRA    BRKPRT
525                    *
526                    * JÄMFÖR START/STOPPADDRESSER
527                    *
528  EEAA B6 A0 05     CMPADR LDA A  ENDA+1
529  EEAD B0 A0 03            SUB A  BEGA+1
530  EEB0 F6 A0 04            LDA B  ENDA
531  EEB3 F2 A0 02            SBC B  BEGA
532  EEB6 39           CMPRTS RTS
533                    *
534                    * PUNCHA TILL KASSETT I S1-FORMAT
535                    * ERFORDRAR KCS-MODEM MED ACIA PÅ $8020
536                    * UNDER PUNCHNING VISAS ADRESS OCH DATA
537                    * PÅ INDIKATORERNA
538                    *
539  EEB7 86 11        PNCHOO LDA A  #$11        SÄTT UPP ACIAN FÖR
540  EEB9 B7 80 20            STA A  TAPE        TVÅ STOPPBITAR VID PUNCH
541  EEBC BD EC F8            JSR    BADDR       BEGÄR STARTADRESS
542  EEBF FF A0 02            STX    BEGA
543  EEC2 BD ED A2            JSR    DELAYD
544  EEC5 CE A0 20            LDX    #RDS1       PEKA PÅ FÖRSTA DISPLAYEN
545  EEC8 FF A0 29            STX    DSPEK
546  EECB BD EC F8            JSR    BADDR       BEGÄR STOPPADRESS
547  EECE FF A0 04            STX    ENDA
548  EED1 8D D7        PUN11  BSR    CMPADR      JÄMFÖR START/STOPP
549  EED3 26 04               BNE    PUN22
550  EED5 81 10               CMP A  #16         MINDRE ÄN 16 BYTES KVAR ?
551  EED7 25 02               BCS    PUN23       HOPPA ISÅFALL ÖVER NÄSTA INSTR.
552                    *
553                    * LÄGG UT 16 MINNESBYTES PER RECORD
554                    *
555  EED9 86 0F        PUN22  LDA A  #15
556  EEDB 8B 04        PUN23  ADD A  #4           INKLUDERA ADRESS I BYTERÄKNAREN
557  EEDD B7 A0 26            STA A  MCONT       SPARA RESULTATET
558  EEE0 80 03               SUB A  #3
559  EEE2 B7 A0 31            STA A  BUFF        DETTA RECORDS BYTEANTAL
560                    *
561                    * SÄND CR,LF,NULL OCH S1(BBAAAADDDDDDDDDDDDCC)
562                    *
563  EEE5 CE EF D3            LDX    #MTAPE1     STRÄNG MED DATA
564  EEE8 8D 35               BSR    TDATA       IVÄG MED DET
565  EEEA 5F                  CLR B              CHECKSUMMA
566                    *
567                    * SÄND BYTEANTALET (S1)BB(AAAADDDDDDDDDDDDCC)
568                    *
569  EEEB CE A0 26            LDX    #MCONT      CELL DÄR ANTALET FINNS
570  EEEE 8D 3A               BSR    PUNT2
571                    *
572                    * SÄND ADRESSEN (S1BB)AAAA(DDDDDDDDDDDDCC)
573                    *
574  EEF0 FE A0 02            LDX    BEGA        HÄMTA ADRESS FÖR RECORD
575  EEF3 FF A0 27            STX    ADRESS
576  EEF6 37                  PSH B
577  EEF7 BD ED 1A            JSR    UTADR       LÄGG UT PÅ DISPLAYEN
578  EEFA 33                  PUL B
579  EEFB CE A0 02            LDX    #BEGA       STARTADRESS FÖR VARJE RECORD
```

```
580   EEFE 8D 2A              BSR    PUNT2      FÖRSTA 2 HEX
581   EF00 8D 28              BSR    PUNT2      ANDRA 2 HEX
582                        *
583                        * SÄND DATA (S1BBAAAA)DDDDDDDDDDDD(CC)
584                        *
585   EF02 FE A0 02           LDX    BEGA       ADRESSEN TILL DATAT
586   EF05 BD ED B6  PUN32    JSR    DATAIX
587   EF08 8D 20              BSR    PUNT2
588   EF0A 7A A0 31           DEC    BUFF
589   EF0D 26 F6              BNE    PUN32
590                        *
591                        * SÄND CHECKSUMMAN (S1BBAAAADDDDDDDDDDDD)CC
592                        *
593   EF0F 53                 COM B              CHECKSUMMAN INVERTERAS
594   EF10 17                 TBA
595   EF11 8D 1C              BSR    OUT2HT
596   EF13 FF A0 02           STX    BEGA
597   EF16 09                 DEX
598   EF17 BC A0 04           CPX    ENDA       KLAR ?
599   EF1A 26 B5              BNE    PUN11
600   EF1C CE EF DC           LDX    #S9STR
601                        *
602   EF1F A6 00       TDATA  LDA A  0,X         HÄMTA TECKEN
603   EF21 81 04              CMP A  #4          SLUT ?
604   EF23 27 24              BEQ    TRTS
605   EF25 8D 16              BSR    TAPEUT
606   EF27 08                 INX
607   EF28 20 F5              BRA    TDATA
608                        *
609                        * SÄND 2 HEX TAL SAMT UPPDATERA CHECKSUMMAN
610                        *
611   EF2A EB 00       PUNT2  ADD B  0,X         UPPDATERA CHECKSUMMAN
612   EF2C A6 00              LDA A  0,X         SÄND 2 HEX TECKEN
613   EF2E 08                 INX
614   EF2F 36          OUT2HT PSH A
615   EF30 8D 18              BSR    OUTHL       OMVANDLA TILL ASCII
616   EF32 32                 PUL A
617   EF33 84 0F       OUTHR  AND A  #$0F        MASKA BORT MSB-DELEN
618   EF35 8B 30              ADD A  #$30        TILLVERKA ASCII
619   EF37 81 39              CMP A  #$39
620   EF39 23 02              BLS    TAPEUT
621   EF3B 8B 07              ADD A  #7          FÖR A-F
622                        *
623                        * UTRUTIN FÖR TAPE
624                        *
625   EF3D 36          TAPEUT PSH A
626   EF3E B6 80 20    TAPUT  LDA A  TAPE
627   EF41 47                 ASR A
628   EF42 47                 ASR A
629   EF43 24 F9              BCC    TAPUT       LEDIG ?
630   EF45 32                 PUL A
631   EF46 B7 80 21           STA A  TAPED
632   EF49 39          TRTS   RTS
633                        *
634   EF4A 44          OUTHL  LSR A
635   EF4B 44                 LSR A
636   EF4C 44                 LSR A
637   EF4D 44                 LSR A
638   EF4E 20 E3              BRA    OUTHR
639                        *
```

```
640                     * LADDA MINNET FRÅN KASSETT I S1-FORMAT
641                     * UNDER LADDNINGEN VISAS MINNESADRESSEN
642                     * PÅ ADRESSDISPLAYEN DS1-4
643                     * OCH DATA PÅ DS5-6
644                     *
645  EF50 86 15     LOAD00  LDA A  #$15      SÄTT UPP ACIAN FÖR
646  EF52 B7 80 20          STA A  TAPE      EN STOPPBIT
647  EF55 8D 45     LOAD3   BSR    TAPEIN    HÄMTA TECKEN
648  EF57 81 53             CMP A  #'S       SÖK TYP AV RECORD
649  EF59 26 FA             BNE    LOAD3
650  EF5B 8D 3F             BSR    TAPEIN
651  EF5D 81 39             CMP A  #'9       ÄR DET EOF ?
652  EF5F 27 E8             BEQ    TRTS
653  EF61 81 31             CMP A  #'1       DATA RECORD
654  EF63 26 F0             BNE    LOAD3
655  EF65 7F A0 0A          CLR    CKSM      NOLLA CHECKSUMMECELLEN
656  EF68 8D 51             BSR    TBYTE     LÄS IN ANTAL BYTES I RECORDET
657  EF6A 80 02             SUB A  #2        KOMMANDE ADRESS= 2 BYTE
658  EF6C B7 A0 0B          STA A  BYTECT
659                     *
660                     * BYGG ADRESSEN
661                     *
662  EF6F 8D 4A             BSR    TBYTE     HÄMTA MSB
663  EF71 B7 A0 27          STA A  ADRESS
664  EF74 8D 45             BSR    TBYTE     HÄMTA LSB
665  EF76 B7 A0 28          STA A  ADRESS+1
666  EF79 BD ED 1A          JSR    UTADR     LÄGG UT PÅ DISPLAYEN
667  EF7C FE A0 27          LDX    ADRESS    X HAR NU LADDADRESSEN
668                     *
669                     * LAGRA DATA
670                     *
671  EF7F 8D 3A     LOAD11  BSR    TBYTE     GÖR EN BYTE
672  EF81 7A A0 0B          DEC    BYTECT    MINSKA ANTALSRÄKNAREN
673  EF84 27 0E             BEQ    LOAD15    OM NOLL SÅ GÅ TILL BÖRJAN
674  EF86 36                PSH A
675  EF87 BD ED B8          JSR    DATADS    LÄGG UT DATA PÅ DISPLAYEN
676  EF8A 32                PUL A
677  EF8B A7 00             STA A  0,X       LAGRA I MINNET
678  EF8D A1 00             CMP A  0,X       FASTNADE DET
679  EF8F 26 08             BNE    LOAD19
680  EF91 08                INX
681  EF92 20 EB             BRA    LOAD11
682                     *
683  EF94 7C A0 0A  LOAD15  INC    CKSM
684  EF97 27 BC             BEQ    LOAD3
685  EF99 7E EE 78  LOAD19  JMP    BELRTS
686                     *
687  EF9C B6 80 20  TAPEIN  LDA A  TAPE
688  EF9F 47                ASR A
689  EFA0 24 FA             BCC    TAPEIN
690  EFA2 B6 80 21          LDA A  TAPED
691  EFA5 39                RTS
692                     *
693  EFA6 8D F4     TINHEX  BSR    TAPEIN
694  EFA8 80 30             SUB A  #$30
695  EFAA 2B 21             BMI    C1          EJ HEX
696  EFAC 81 09             CMP A  #$09
697  EFAE 2F 0A             BLE    IN1HG
698  EFB0 81 11             CMP A  #$11
699  EFB2 2B 19             BMI    C1
```

```
700   EFB4 81 16              CMP A   #$16
701   EFB6 2E 15              BGT     C1              EJ HEX
702   EFB8 80 07              SUB A   #7
703   EFBA 39         IN1HG   RTS
704                   *
705   EFBB 8D E9      TBYTE   BSR     TINHEX
706   EFBD 48                 ASL A
707   EFBE 48                 ASL A
708   EFBF 48                 ASL A
709   EFC0 48                 ASL A
710   EFC1 16                 TAB
711   EFC2 8D E2              BSR     TINHEX
712   EFC4 1B                 ABA
713   EFC5 16                 TAB
714   EFC6 FB A0 0A           ADD B   CKSM
715   EFC9 F7 A0 0A           STA B   CKSM
716   EFCC 39                 RTS
717                   *
718   EFCD 7D 80 EE   C1      TST     BELL        GE PIIP OCH ÅTERGÅ
719   EFD0 7E EC 3B           JMP     CONTRL
720                   *
721   EFD3 0D         MTAPE1  FCB     $D,$A,0,0,0,0,'S,'1,4
722                   *
723   EFDC 53         S9STR   FCB     'S,'9,4
724                   *
725                   *
726                   * TABELL FÖR KOMMANDON
727                   *
728   EFDF 0A         INTAB   FCB     $0A
729   EFE0 ED 3B              FDB     MEMCHG      MINNESMANIPULERING "M"
730   EFE2 0B                 FCB     $0B         SET BREAKPOINTS
731   EFE3 EE 6D              FDB     SETBR
732   EFE5 0C                 FCB     $0C         CONTINUE
733   EFE6 ED D7              FDB     CONTIN
734   EFE8 0D                 FCB     $0D         DELETE BREAKPOINTS
735   EFE9 EC 38              FDB     KILL
736   EFEB 0E                 FCB     $0E         EXECUTE PROGRAM
737   EFEC ED CA              FDB     GOEXEC
738   EFEE 10                 FCB     $10         LADDA I S1-FORMAT
739   EFEF EF 50              FDB     LOADOO
740   EFF1 11                 FCB     $11         PUNCHA S1-FORMAT TILL KA
741   EFF2 EE B7              FDB     PNCHOO
742   EFF4 12                 FCB     $12         REGISTERDISPLAY
743   EFF5 EE 02              FDB     PRREG
744   EFF7            TABEND  EQU     *
745                   *
746                   *
747   EFF8                    ORG     PGM+$3F8
748   EFF8 EC 00              FDB     IRQSER
749   EFFA EC 0A              FDB     SWISER
750   EFFC EC 05              FDB     NMISER
751   EFFE EC 1B              FDB     START
752                           END

NO ERROR(S) DETECTED
```

1.

ETT BASICPROGRAM FÖR SLÄKTFORSKNING

Att spåra sina egna rötter är en hobby, som alltfler ägnar sig åt. Datorn är även i detta fall ett utmärkt hjälpmedel. Ett väl genomtänkt program, som tar hand om alla data, ordnar dem i filer och tillåter redigering, är oumbärligt när högen av data växer i takt med antalet förfäder man lyckats spåra.

Detta program skrevs ursprungligen på DEC:s BASIC Plus 2 för DEC-system 2000 vid Universitetet i Chicago. BASIC Plus 2 skiljer sig avsevärt från TSC:s BASIC vad I/O-filhanteringen beträffar. Vid dimensionering av variabler och $-strängar kan man använda 'MAP', ett statement, som verkar som en sorts 'mall' vid överföring av data till I/O-buffer. Överföring mellan fil och I/O-buffer sker med 'GET' och 'PUT'. Datan kan lagras i filen sekventiellt på sedvanligt sätt eller med 'RANDOM ACCESS'. 'MAP' tillåter överföring av numeriska variabler direkt till I/O-buffer utan konvertering till $-strängar (Jfr CVT-funktionen hos TSC:s BASIC).
De flesta basicspråk saknar tyvärr 'MAP'.

Det ursprungliga programmet använder sekventiella filer, och är ett bra exempel på strukturerad basicprogrammering. En direkt omskrivning av programmet till TSC-BASIC låg nära till hands, men det visade sig snart att filhanteringen av de talrika numeriska värdena och $-strängarna kunde ske elegantare med hjälp av "Record I/O". Programmet blev dessutom betydligt kortare.

Vid "record I/O" lagras data på "record" av 252 bytes längd, dvs varje "record" motsvarar det disponibla innehållet i en disksektor. En record I/O-fil är en numrerad följd av record tillgängliga genom "random access" ungefär som datan i primärminnet. Det finns en I/O-buffer för varje sådan fil. I I/O-buffer lagrar man tillfälligt det record i filen vars data man vill manipulera. Fördelning av utrymmet för olika datasträngar inom I/O-buffret sker med hjälp av "FIELD statement". Observera att numeriska variabler måste göras om till $-strängar!

Programmet använder två I/O-buffer, vilket motsvarar två disksektorer, där varje disponibel byte utnyttjas. Detta betyder att alla data för en person får plats på två record som ligger i var sin fil. Om utrymme önskas för längre strängar, t ex för mera detaljerade källuppgifter, anmärkningar etc, kan antalet I/O-buffer ökas efter behag. Maximala antalet samtidigt öppna filer är 12. Man kan alltså manipulera max 12 I/O-buffer samtidigt.

Record-I/O är en bra metod för en effektiv och snabb filhantering. Den är vida överlägsen den vanliga sekventiella metoden, när det gäller att snabbt plocka fram större mängder av data, som inte ligger i följd i en fil. Förfarandet är dock oekonomiskt, när minneskapaciteten i varje enskild sektor inte helt kan utnyttjas. Författaren till denna artikel skrev om det ursprungliga programmet till FLEX BASIC, för att visa vad man kan göra med dess Record-I/O. Då varje sektor här nyttjas till 100 procent, blir filhanteringen synnerligen ekonomisk och man får ett program som arbetar snabbt.

Som framgår av listningen,

är programmet ganska långt. Man bör helst ha 32 K primärminne utöver 8 K för FLEX för att få plats för hantering avde växande filerna. Den som har lägre minneskapacitet, kan lägga de tre huvudsubrutinerna i skilda filer, som kan anropas med 'CHAINE-statement'. Observera att huvudsubrutin 2 har flera revideringsrutiner (rad nr 4590 - 5310), som måste ligga i samma fil, som huvudsubrutin 2.

Programmets funktion är i korthet följande: När programmet startar, frågar datorn om man vill ha en listning av alla personer i filen. Därefter sker val av program enligt nedan:

```
ANGE NUMRET PÅ DEN ÖNSKADE
PROCEDUREN

(1) FÖR IN NYA DATA
(2) LÄS AV ELLER REVIDERA DATA
(3) LISTA STAMTRÄDET
(4) AVSLUTA PROGRAMMET
```

Procedur 1 frågar först efter den aktuella personens nummer. Om numret på en i filen redan befintlig person anges, kommer datorn att betrakta personen som ny i filen och frågar efter alla data på nytt. Obs att om endast vissa data behöver ändras, sker detta enklast med procedur 2. Om numret är lika med närmaste lediga löpande nummer i filen, låter man lämpligen datorn sköta om tilldelning av nummer. Datorn begär följande information för den aktuella personen: födelsedata och plats, föräldrar, giftermål samt datum och plats för dödsfallet. Programmet har i sin nuvarande form utrymme för uppgifter om max tre giftermål. Detta bör vara tillräckligt i de allra flesta fall. Programmet har även utrymme för källinformation, för att underlätta kontroll av införda data och för framtida kompletterings. Listning 1 visar ett fingerat exempel på införande av data underpunkt 1.

Genom att anropa procedur 2 enligt tabellen ovan, listas alla data om önskad person. Listning sker efter antingen namn eller nummer. Vill man ha data listat på samtliga personer, skriver man 'ALLA'. Listning 2 visar ett exempel på en listning enligt procedur 2.

Förutom listning kan även revidering av data ske med procedur 2. När listningen för en person är klar, frågar datorn om manvill ändra någonting. Om svaret är 'ja', skrivs följande tabell ut:

```
ANGE ÖNSKAT AVSNITT
  1) NAMN, NUMMER
  2) FÖDELSE, HÄRKOMST
  3) INFO OM GIFTERMÅL
  4) INFO OM DÖDSFALL
  5) INGEN ÄNDRING
```

De 4 ändringsrutinerna är dokumentorienterade, dvs de frågar efter hur och var informationen erhållits. Detta är värdefullt för den fortsatta sökningen efter anfäder, då flera informationskällor ibland kan bidra till att man lyckas fylla en lucka i stamträdet!

Som tidigare nämnts, bör korrigeringar utföras med hjälp av procedur 2. Om man vid inskrivning av data råkar göra ett skrivfel (gäller vid procedur 1), bortser man från felet och för in resterande data. Därefter kan rättelse ske, genom att under procedur 2 begära önskad subrutin för revision enligt tabellen ovan, och förain rättelsen.

Huvudsubrutin nr 2 kan även användas för att få blanka formulär för insamling av data från landsarkiv osv. Härvid kan man lämpligen besvara alla dator-prompten med ett frågetecken. Som nummer på person väljer man en nolla. Obs att datorn kan duplicera önskat antal sådana formulär!

Genom att programmet tilldelar varje person ett unikt nummer, kan varje person identifieras, även om både för- och efternamn råkar vara identiska. Man kan också med hjälp av numren lista stamträdet för fyra generationer för en person. Listning nr 3 visar exempel på ett stamträd.

Den som är intresserad av programmet kan bespara sig några timmars möda vid tangentbordet genom att sända en 5 tums diskett till artikelförf. Helst även ett adresserat kuvert. Även den å minnets vägnar mindre resursrike kan göra sammaledes. Uppdelning av programmet i flera program för laddning med 'chain statement' kan lätt göras med hjälp av editorn.

Gunnar Lovius
Ragnebergsvägen 21
136 67 Handen.



```
10 PRINT TAB(13);"SLÄKTFORSKNING"
20 REM
30 REM WRITTEN BY STAN W.MERRIL
40 REM OF COMP RES ASSOCIATES
50 REM CHICAGO.
60 REM ÄNDRAT FÖR FLEX BASIC OCH FÖR
70 REM SVENSKA FÖRHÅLLANDEN AV
80 REM GUNNSR LOVIUS,MARS 1980
90 REM
100 REM ALLOCATE FILE AND COUNT NUMBER OF CASES ALREADY IN IT
110 REM SET COUNTERS
120 OPEN "GEN1" AS 1
130 OPEN "GEN2" AS 2
140 PRINT:PRINT:PRINT
150 ON ERROR GOTO 410
160 DIM B1$(3),B2$(3),B3$(3),B4$(3)
170 DIM C$(3),G$(3),K$(3),D2$(3)
180 REM
190 FIELD #1,8 AS B$,25ASD$,16ASE$,20ASH$,25ASQ$,8ASR$,25 AS S$,8AST$,
4ASF$,30ASD1$,16ASJ$,20ASL$,30ASD3$,16 AS C$(1)
200 REM
210FIELD#2,16ASC$(2),16ASC$(3),25ASG$(1),25ASG$(2),25ASG$(3),20ASK$(1)
,20ASK$(2),20ASK$(3),30ASD2$(1),30ASD2$(2),25ASD2$(3)
220 LET I2=0
230 LET I3=0: REM I3 COUNTS NUMBER OF CASES IN FILE
240 REM
250 PRINT "SKA JAG LISTA ALLA NAMNEN"
260 PRINT "I FILEN?"
270 INPUT W1$
280 PRINT
290 IF LEFT$(W1$,1)="J" THEN PRINT "FILEN INNEHÅLLER FÖLJANDE NAMN:"
300 PRINT
310 LET I3=I3+1
320 LET E=I3
330 GET #1,RECORD E
340 GET #2,RECORD E
350 IF LEFT$(W1$,1)="J" THEN PRINT CVT$F(B$),D$
360 REM
370 IF CVT$F(B$)>I2 THEN LET I2=CVT$F(B$)
380 IF CVT$F(R$)>I2 THEN LET I2=CVT$F(R$)
390 IF CVT$F(T$)>I2 THEN LETI2=CVT$F(T$)
400 GOTO 310
410 RESUME 430
420 REM
430 PRINT:PRINT:PRINT
440 PRINT "ANGE NUMRET PÅ DEN ÖNSKADE"
450 PRINT "PROCEDUREN:"
460 PRINT
470 PRINT TAB(10);'(1) FÖR IN NYA DATA'
480 PRINT TAB(10);'(2) LÄS AV ELLERREVIDERA DATA'
490 PRINT TAB(10);'(3) LISTA STAMTRÄDET'
500 PRINT TAB(10);'(4) AVSLUTA PROGRAMMET'
510 INPUT A
520 IF A=1 THEN GOSUB 610:REM ENTER INFORMATION
530 IF A=2 THEN GOSUB 2340:REM READ THE INFORMATION
540 IF A=3 THEN GOSUB 3420:REM LISTA STAMTRÄDET
550 IF A=4 THEN GOTO5350:REM END PROGRAM
560 GOTO 430: REM INPUT ERROR - TRY AGAIN
570 REM
580 REM PROCEDURE SUBROUTINE #1
590 REM    BIRTH CERTIFICATE INFORMATION
600 REM
```

```
610 PRINT:PRINT
620 PRINT 'ÄNDAMÅL: GENETISKA DATA TILL FILEN'
630 PRINT
640 INPUT "ANGE 'RECORDNR'=PERSONS NR I FILEN: ";Y
650 LET E=Y
660 PUT #1,RECORD E
670 PUT #2,RECORD E
680 PRINT:PRINT
690 PRINT "ANGE PERSONENS NAMN"
700 PRINT "(ANGE FLICKNAMNET I TILLÄMPL.FALL)"
710 GET #1,RECORD E
720 GET #2,RECORD E
730 INPUT A$
740 LSET D$=A$
750 PRINT:PRINT
760 PRINT "FÖR ATT UNDVIKA,"
770 PRINT "ATT ";D$
780 PRINT "FÖRVÄXLAS MED NÅGON ANNAN"
790 PRINT "SOM HAR LIKNANDE NAMN,BÖR"
800 PRINT "HAN/HON HA ETT UNIKT NR I DENNA FIL"
810 PRINT
820 PRINT "HAR NUMMER REDAN TILLDELATS?"
830 INPUT V$
840 IF LEFT$(V$,1)="J" THEN 940
850 IF V$<>'N'THEN IF V$<>'NEJ'THEN GOTO 810
860 LET I2=I2+1
870 LET B=I2
880 PRINT
890 PRINT "HAN/HON HAR TILLDELATS NUMMER:"
900 PRINT
910 PRINT TAB(10);B
920 LSET B$=CVTF$(B)
930 GOTO 980
940 PRINT
950 PRINT "VAR SNÄLL OCH ANGE NUMRET:"
960 INPUT B
970 LSET B$=CVTF$(B)
980 PRINT:PRINT
990 PRINT "JAG BEHÖVER EN DEL DATA"
1000 PRINT "ANGE DATA I FÖLJANDE"
1010 PRINT "FORMAT:  DAG  MÅNAD  ÅR"
1020 PRINT "   T.EX.  23  APRIL 1949"
1030 PRINT "OM DATUM EJ ÄR BEKANT,SKRIV 'OBEKANT'"
1040 PRINT:PRINT
1050 PRINT "NÄR FÖDDES ";D$;"(D M ÅR)?"
1060 INPUT A1$
1070 LSET E$=A1$
1080 PRINT
1090 PRINT "VAR FÖDDES HAN/HON?"
1100 INPUT A2$
1110 LSET H$=A2$
1120 PRINT:PRINT
1130 PRINT "VAD HETTE ";D$;"S FADER?"
1140 PRINT "(OM DU EJ VET, SKRIV 'OBEKANT')"
1150 INPUT A3$
1160 LSET Q$=A3$
1170 IF Q$='OBEKANT' THEN GOTO 1370
1180 PRINT:PRINT
1190 PRINT "HAR ";Q$;" REDAN TILLDELATS NUMMER?"
1200 INPUT V$
```

```
1210 IF LEFT$(V$,1)="J" THEN 1320
1220 IF V$<>'N'THEN IF V$<>'NEJ'THEN GOTO 1180
1230 PRINT
1240 LET I2=I2+1
1250 LET R=I2
1260 PRINT "HAN HAR TILLDELATS NUMRET:"
1270 PRINT
1280 PRINT TAB(10);R
1290 LSET R$=CVTF$(R)
1300 PRINT:PRINT
1310 GOTO 1390
1320 PRINT
1330 PRINT "VAR SNÄLL, ANGE NUMRET:"
1340 INPUT R
1350 LSET R$=CVTF$(R)
1360 GOTO 1390
1370 LET R=0
1380 PRINT:PRINT
1390 PRINT "VAD HETTE ";D$;"S MOR?"
1400 PRINT "(ANVÄND OM MÖJLIGT FLICKNAMNET."
1410 PRINT "VET DU EJ NAMNET,SKRIV 'OBEKANT')"
1420 INPUT A4$
1430 LSET S$=A4$
1440 IF S$='OBEKANT' THEN GOTO 1620
1450 PRINT:PRINT
1460 PRINT "HAR HON REDAN TILLDELATS NUMMER?"
1470 INPUT V$
1480 IF LEFT$(V$,1)="J" THEN GOTO 1580
1490 IF V$<>'N'THEN IF V$<>'NEJ' THEN GOTO 1450
1500 PRINT
1510 LET I2=I2+1
1520 LET T=I2
1530 PRINT "HON HAR TILLDELATS NUMRET:"
1540 PRINT
1550 PRINT TAB(10);T
1560 LSET T$=CVTF$(T)
1570 GOTO 1640:   REM GO BACK FOR NUMBER
1580 PRINT
1590 PRINT "VAR SNÄLL, ANGE NUMRET:"
1600 INPUT T
1610 LSET T$=CVTF$(T)
1620 PRINT:PRINT
1630 LET T=0
1640 PRINT:PRINT
1650 PRINT "VARIFRÅN FICK DU INFO OM"
1660 PRINT D$;"S FÖDELSE OCH HÄRKOMST?"
1670 PRINT "SPECIFIERA (MAX 30 TECKEN)"
1680 INPUT A5$
1690 LSET D1$=A5$
1700 REM
1710 REM INFO FRÅN ÄKTENSSKAPSINTYG
1720 REM
1730 PRINT:PRINT
1740 PRINT "HUR MÅNGA GÅNGER VAR"
1750 PRINT D$;" GIFT?"
1760 PRINT "ANGE ETT TAL MELLAN 0 OCH 3"
1770 INPUT F
1780 IF F=0 THEN 2000
1790 LSET F$=CVTF$(F)
1800 FOR I=1 TO F
```

```
1810   PRINT:PRINT
1820   PRINT "ANGE DATUM FÖR GIFTERMÅLET#";I;"(DAG MÅN ÅR)"
1830 INPUT B1$(I)
1840 LSET C$(I)=B1$(I)
1850   PRINT
1860   PRINT "ANGE MAKENS (MAKANS) FULLSTÄNDIGA NAMN"
1870   PRINT "(ANVÄND FLICKNAMNET I TILLÄMPL FALL"
1880 INPUT B2$(I)
1890 LSET G$(I)=B2$(I)
1900   PRINT
1910   PRINT "VAR GIFTE DE SIG?"
1920 INPUT B3$(I)
1930 LSET K$(I)=B3$(I)
1940   PRINT:PRINT
1950 PRINT "VAR FICK DU INFO OM GIFTERMÅL # ";I; "?"
1960   PRINT "(MAX 30 TECKEN)"
1970    INPUT B4$(I)
1980   LSET D2$(I)=B4$(I)
1990 NEXT I
2000 IF F=0 THEN B1$(I)='OGIFT'
2010 LSET C$(I)=B1$(I)
2020 IF F=0 THEN B2$(I)='OGIFT'
2030 LSET $(I)=B2$(I)
2040 REM
2050 REM   DEATH CERTIFICATE INFORMATION
2060 REM
2070 PRINT:PRINT
2080 PRINT "NÄR DOG ";D$;" (DAG MÅN ÅR)?"
2090 PRINT "(OM DU EJ VET DET,SKRIV 'OBEKANT'."
2100 PRINT "OM HAN ÄR FORTFARANDE I LIVET, SKRIV 'LEVER'.)"
2110 INPUT A6$
2120 LSET J$=A6$
2130 IF A6$='LEVER' THEN GOTO 2230
2140 PRINT:PRINT
2150 PRINT "VAR DOG HAN/HON?"
2160 INPUT A7$
2170 LSET L$=A7$
2180 PRINT:PRINT
2190 PRINT "VAR FICK DU INFO OM HANS/HENNES DÖD?"
2200 INPUT A8$
2210 LSET D3$=A8$
2220 GOTO 2250
2230 IF A6$='LEVER' THEN A7$='TILLÄMPAS EJ'
2240 LSET L$=A7$
2250 PRINT:PRINT:PRINT
2260 REM
2270 PUT #1,RECORD E
2280 PUT #2,RECORD E
2290 RETURN
2300 REM
2310 REM      PROCEDURE FOR SUBROUTINE #2
2320 REM        READ AND REVISE FILE
2330 REM
2340 PRINT:PRINT
2350 PRINT "LÄS- OCH REVIDERINGSFILE."
2360 PRINT:PRINT
2370 REM
2380 PRINT "VILL DU SÖKA EFTER NAMN ELLER NUMMER?"
2390 INPUT W2$
2400 PRINT
```

```
2410 IF W2$='NAMN' THEN GOTO 2550
2420 IF W2$<>'NUMMER' THEN GOTO 2380
2430 PRINT "VAR SNÄLL, SKRIV NUMRET"
2440 INPUT U1
2450 PRINT
2460 LET Z5=0
2470 LET Z5=Z5+1
2480 IF Z5>I3 THEN PRINT "PERSON # ";U1;" FINNS EJ I FILEN"
2490 IF Z5>I3 THEN GOTO 3310
2500 LET E=U1
2510 GET #1,RECORD E
2520 GET #2,RECORD E
2530 IF U1<>CVT$F(B$) THEN 2470
2540 GOTO 2700
2550 PRINT "OM DU SÖKER NÅGON SÄRSKILD,"
2560 PRINT "ANGE HANS/HENNES NAMN. OM DU VILL"
2570 PRINT "LÄSA IGENOM HELA FILEN,SKRIV 'ALLA'."
2580 INPUT P1$
2590 PRINT
2600 LET P2$=P1$
2610 LET Z5=0
2620 LETZ5=Z5+1
2630 IF Z5+1>I3 THEN PRINT P1$;" FINNS EJ I FILEN."
2640 IF Z5+1>I3 THEN 3310
2650 LET E=Z5
2660 GET #1,RECORD E
2670 GET #2,RECORD E
2680 IF P1$='ALLA' THEN LET P2$=D$
2690 IF P2$<>D$ THEN GOTO 2620
2700 PRINT:PRINT
2710 PRINT "GENEALOGISK INFO FÖR:"
2720 PRINT D$
2730 PRINT "NUMMER: ";CVT$F(B$)
2740 PRINT
2750 PRINT "FÖDD ";E$
2760 PRINT "FÖDELSEORT: ";H$
2770 PRINT
2780 PRINT "FADER: ";Q$;" (NUMMER: ";CVT$F(R$);" )"
2790 PRINT "MODER: ";S$;" (NUMMER: ";CVT$F(T$);" )"
2800 PRINT
2810 LET F=CVT$F(F$)
2820 IF F=0 GOTO 2910
2830 PRINT "GIFT MED:"
2840 FOR I=1 TO F
2850 PRINT '    ';G$(I)
2860 PRINT '       DEN: ';C$(I)
2870 PRINT '       PLATS: ';K$(I)
2880 PRINT
2890 NEXT I
2900 PRINT:PRINT
2910 IF J$='LEVER' THEN GOTO 2940
2920 PRINT "DÖDSDATUM: ";J$
2930 PRINT "DÖDSORT: ";L$
2940 PRINT:PRINT
2950 PRINT "KÄLLUPPGIFT:"
2960 PRINT " FÖDELSE,HÄRSTAMN.: ";D1$
2970 FOR I=1 TOF
2980 IF F<>0 THEN PRINT " GIFTERMÅL #";I;":";D2$(I)
2990 NEXT I
3000 IF J$='LEVER' THEN 3020
```

```
3010 PRINT " DÖDSFALL: ";D3$
3020 PRINT:PRINT
3030 REM
3040 PRINT "VILL DU ÄNDRA NÅGONTING?"
3050 INPUT P3$
3060 IF LEFT$(P3$,1)="N" THEN GOTO 3290
3070 REM
3080 REM     MENU FOR REVISIONS
3090 REM
3100 PRINT
3110 PRINT "ANGE ÖNSKAT AVSNITT"
3120 PRINT
3130 PRINT '    1) NAMN,NUMMER'
3140 PRINT '    2) FÖDELSE,HÄRKOMST'
3150 PRINT '    3) INFO OM GIFTERMÅL'
3160 PRINT '    4) INFO OM DÖDSFALL'
3170 PRINT '    5) INGEN ÄNDRING'
3180 PRINT
3190 INPUT P4
3200 IF P4=1 THEN GOSUB 4600
3210 IF P4=2 THEN GOSUB 4720
3220 IF P4=3 THEN GOSUB 4910
3230 IF P4=4 THEN GOSUB 5170
3240 IF P4=5 THEN GOSUB 3310
3250 PRINT
3260 REM
3270 PRINT "VILL DU GÖRA NÅGON MER ÄNDRING?"
3280 GOTO 3050
3290 PUT #1,RECORD E
3300 PUT #2,RECORD E
3310 PRINT:PRINT
3320 REM
3330 PRINT "VILL DU SE FLERA UPPLÄGG?"
3340 INPUT P$
3350 IF LEFT$(P$,1)="J" THEN IF P1$<>'ALLA' THEN 2370
3360 IF LEFT$(P$,1)="J" THEN 2620
3370 REM
3380 RETURN
3390 REM PROCEDURE SUBROUTINE #3
3400 REM     LINK PEDIGREE BY NUMBER
3410 REM
3420 PRINT:PRINT
3430 PRINT "LISTA STAMTRÄDET"
3440 PRINT:PRINT
3450 LET I4=0
3460 PRINT "VEMS STAMTRÄD SKA JAG LISTA?"
3470 PRINT "(ANGE NUMRET)"
3480 INPUT X1
3490 PRINT:PRINT
3500 PRINT "'F','M' BETYDER 'FADER' RESP 'MODER'"
3510 PRINT:PRINT
3520 REM
3530 REM FIND PARENTS AND LIST
3540 REM
3550 LET I4=I4+1
3560 IF I4>I3 THEN PRINT "PERSON #";X1;" FINNS INTE I FILEN"
3570 IF I4>I3 THEN 4570
3580 LET E=X1
3590 GET #1,RECORD E
3600 GET #2,RECORD E
```

```
3610 IF CVT$F(B$)<>X1THEN 3550
3620 PRINT TAB(15);"STAMTRÄD FÖR PERSON #";CVT$F(B$)
3630 PRINT
3640 PRINT D$
3650 PRINT
3660 LET R=CVT$F(R$):LET T=CVT$F(T$)
3670 PRINT TAB(5);'F:';Q$;'(NUMBER:';R;')'
3680 PRINT TAB(5);'M:';S$;'(NUMMER:';T;')'
3690 PRINT
3700 REM
3710 REM  FIND GRANDPARENTS AND LIST
3720 REM
3730 LET X2=R
3740 LET X3=T
3750 LET I5=0
3760 LET IA=0
3770 IF X2<>0 THEN IA=IA+1
3780 IF X3<>0 THEN IA=IA+1
3790 IF IA=0 THEN 4570
3800 LET I4=I4+1
3810 LET E=I4
3820 GET #1,RECORD E
3830 GET #2,RECORD E
3840 IF X2=CVT$F(B$) THEN 3870
3850 IF X3=CVT$F(B$) THEN 3960
3860 GOTO 3800
3870 LET R=CVT$F(R$):LET T=CVT$F(T$)
3880 PRINT TAB(10);'FF:';Q$;'NUMMER:';R;')'
3890 PRINT TAB(10);'FM:';S$;'NUMMER:';T;')'
3900 PRINT
3910 LET X4=R
3920 LET X5=T
3930 LET I5=I5+1
3940 IF I5<IA THEN 3800
3950 IF I5>=IA THEN 4100
3960 LET R=CVT$F(R$):LET T=CVT$F(T$)
3970 PRINT TAB(10);'MF:';Q$;'NUMMER:';R;')'
3980 PRINT TAB(10);'MM:';S$;'NUMMER:';T;')'
3990 PRINT
4000 LET X6=R
4010 LET X7=T
4020 LET I5=I5+1
4030 IF I5<IA THEN 3800
4040 IF I5>=IA THEN 4100
4050 LET IB=0
4060 PRINT
4070 REM
4080 REM  FIND GREAT GRANDPARENTS AND LIST
4090 REM
4100 LET I5=0
4110 LET IA=0
4120 IF X4<>0 THEN IA=IA+1
4130 IF X5<>0 THEN IA=IA+1
4140 IF X6<>0 THEN IA=IA+1
4150 IF X7<>0 THEN IA=IA+1
4160 IF IA=0 THEN 4570
4170 LET I4=I4+1
4180 IF IA>4 THEN PRINT "STAMTRÄDET KAN EJ UTÖKAS MERA"
4190 IF IA>4 THEN 4570
4200 LET E=I4
```

```
4210 GET #1,RECORD E
4220 GET #2,RECORD E
4230 LET B=CVT$F(B$)
4240 IF X4=B THEN 4290
4250 IF X5=B THEN 4360
4260 IF X6=B THEN 4430
4270 IF X7=B THEN 4500
4280 GOTO 4170
4290 LET R=CVT$F(R$):LET T=CVT$F(T$)
4300 PRINT TAB(15);'FFF:';Q$;'(NUMMER:';R;')'
4310 PRINT TAB(15);'FFM:';S$;'(NUMMER:';T;')'
4320 PRINT
4330 LET I5=I5+1
4340 IF I5<IA THEN 4170
4350 IF I5>=IA THEN 4570
4360 LET R=CVT$F(R$):LET T=CVT$F(T$)
4370 PRINT TAB(15);'FMF:';Q$;'(NUMMER:';R;')'
4380 PRINT TAB(15);'FMM:';S$;'(NUMMER:';T;')'
4390 PRINT
4400 LET I5=I5+1
4410 IF I5<IA THEN 4170
4420 IF I5>=IA THEN 4570
4430 LET R=CVT$F(R$):T=CVT$F(T$)
4440 PRINT TAB(15);'MFF:';Q$;'(NUMMER:';R;')'
4450 PRINT TAB(15);'MFM:';S$;'(NUMMER:';T;')'
4460 PRINT
4470 LET I5=I5+1
4480 IF I5<IA THEN 4170
4490 IF I5>=IA THEN4570
4500 LET R=CVT$F(R$):LET T=CVT$F(T$)
4510 PRINT TAB(15);'MMF:';Q$;'(NUMMER:';R;')'
4520 PRINT TAB(15);'MMM:';S$;'(NUMMER:';T;')'
4530 PRINT
4540 LET I5=I5+1
4550 IF I5<IA THEN 4170
4560 REM
4570 RETURN
4580 REM
4590 REM     REVISION SUBROUTINE #1
4600 REM       CHANGE NAME AND NUMBER
4610 REM
4620 PRINT:PRINT
4630 INPUT "NAMN:";A$
4640 LSET D$=A$
4650 PRINT
4660 INPUT "NUMMER: ";B
4670 LSET B$=CVTF$(B)
4680 RETURN
4690 REM     REVISION SUBROUTINE #2
4700 REM       CHANGE BIRTH INFO
4710 REM
4720 REM
4730 PRINT:PRINT
4740 INPUT "FÖDELSEDATA:";A1$
4750 LSET E$=A1$
4760 INPUT "FÖDELSEORT: ";A2$
4770 LSET H$=A2$
4780 INPUT "FADERNS NAMN: ";A3$
4790 LSET Q$=A3$
4800 INPUT "FADERNS NUMMER:";R
```

```
4810 LSET R$=CVTF$(R)
4820 INPUT "MODERNS NAMN* ";A4$
4830 LSET S$=A4$
4840 INPUT "MODERNS NUMMER: ";T
4850 LSET T$=CVTF$(T)
4860 INPUT "INFORMATIONSKÄLLA: ";A5$
4870 LSET D1$=A5$
4880 RETURN
4890 REM
4900 REM   REVISION SUBROUTINE #3
4910 REM     CHANGE MARRIAGE INFO
4920 REM
4930 PRINT:PRINT
4940 INPUT "VILKET ÄKTENSKAP (ANGE NR) ?";I
4950 PRINT
4960 PRINT "DATUM FÖR GIFTERMÄL #";I;
4970 INPUT B1$(I)
4980 LSET C$(I)=B1$(I)
4990 PRINT
5000 INPUT "MAKENS (MAKANS) NAMN";B2$(I)
5010 LSET G$(I)=B2$(I)
5020 PRINT
5030 INPUT "PLATS FÖR GIFTERMÅL";B3$(I)
5040 LSET K$(I)=B3$(I)
5050 PRINT
5060 PRINT "INFORMATIONSKÄLLA?"
5070 INPUT B4$(I)
5080 LSET D2$(I)=B4$(I)
5090 PRINT
5100 PRINT "HUR MÅNGA AV DENNE PERSONS"
5110 PRINT "ÄKTENSKAP GER DENNA FIL INFO OM?"
5120 INPUT F
5130 LSET F$=CVTF$(F)
5140 RETURN
5150 REM
5160 REM   REVISION SUBROUTINE #4
5170 REM     CHANGE DEATH INFO
5180 REM
5190 PRINT:PRINT
5200 PRINT "DÖDSDATUM"
5210 PRINT "(OM EJ KÄNT,SKRIV 'OBEKANT'. OM"
5220 PRINT "VID LIV, SKRIV 'LEVER'.)"
5230 INPUT A6$
5240 LSET J$=A6$
5250 IF J$='LEVER' GOTO 5310
5260 INPUT "DÖDSPLATS: ";A7$
5270 LSET L$=A7$
5280 PRINT "INFORMATIONSKÄLLA?"
5290 INPUT A8$
5300 LSET D3$=A8$
5310 RETURN
5320 REM
5330 REM   PROC. SUBROUTINE #4
5340 REM     END OF PROGRAM
5350 CLOSE 1,2
5360 END
```

Bygge och programmering av ett Timesharingsystem

När man själv bygger sina kort så får man ibland en ide´ och i stort får man inte en lugn stund förrän ide´n är provad praktiskt.

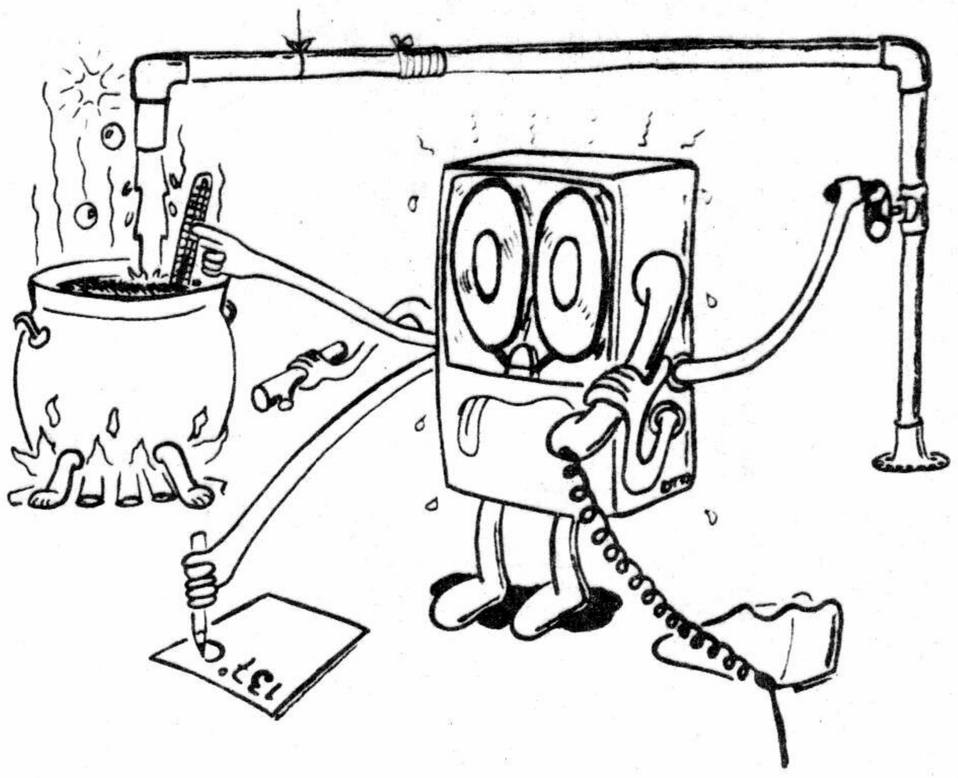
Under ett samtal med Erling Anderberg om klockprogram för snart ett år sedan kom vi in på att byta stackpekare och registerinnnehåll med hjälp av ett av 68ans interrupt. på detta sätt kan man då hålla igång flera användare samtidigt.

Hur det fungerar när det hela väl är igång var väl inte så svårt att förstå, men hur man skulle starta upp det hela var inte helt klart från början och den första ide´n byggde på att ha MIKBUG assemblerad till tre olika startadresser och tre olika ACIA- och stackadresser. Så småningom kom jag fram till att byta ACIA och stackRAM med hjälp av chip select styrd av en styrPIA, och ungefär vid årsskiftet började jag skriva de delar som MIKBUG behövde kompletteras med för att få det hela att snurra. Samtidigt byggde jag CPU-kortet, som skulle använda minavanliga RAMkort. Det har Timer på kortet för IRQ interrupten och för kommunikationsklockan samt en logik för att ge RESET vid rätt tidpunkt med hjälp av NMI. Dessutom fanns en växel för att skifta min terminal mellan användare 1 och 2 men den fick utgå under uppkörningen samtidigt som en extra chip select infördes till timern,som spårade ur.Så småningom fann jag att den saknade anslutning till R/W.

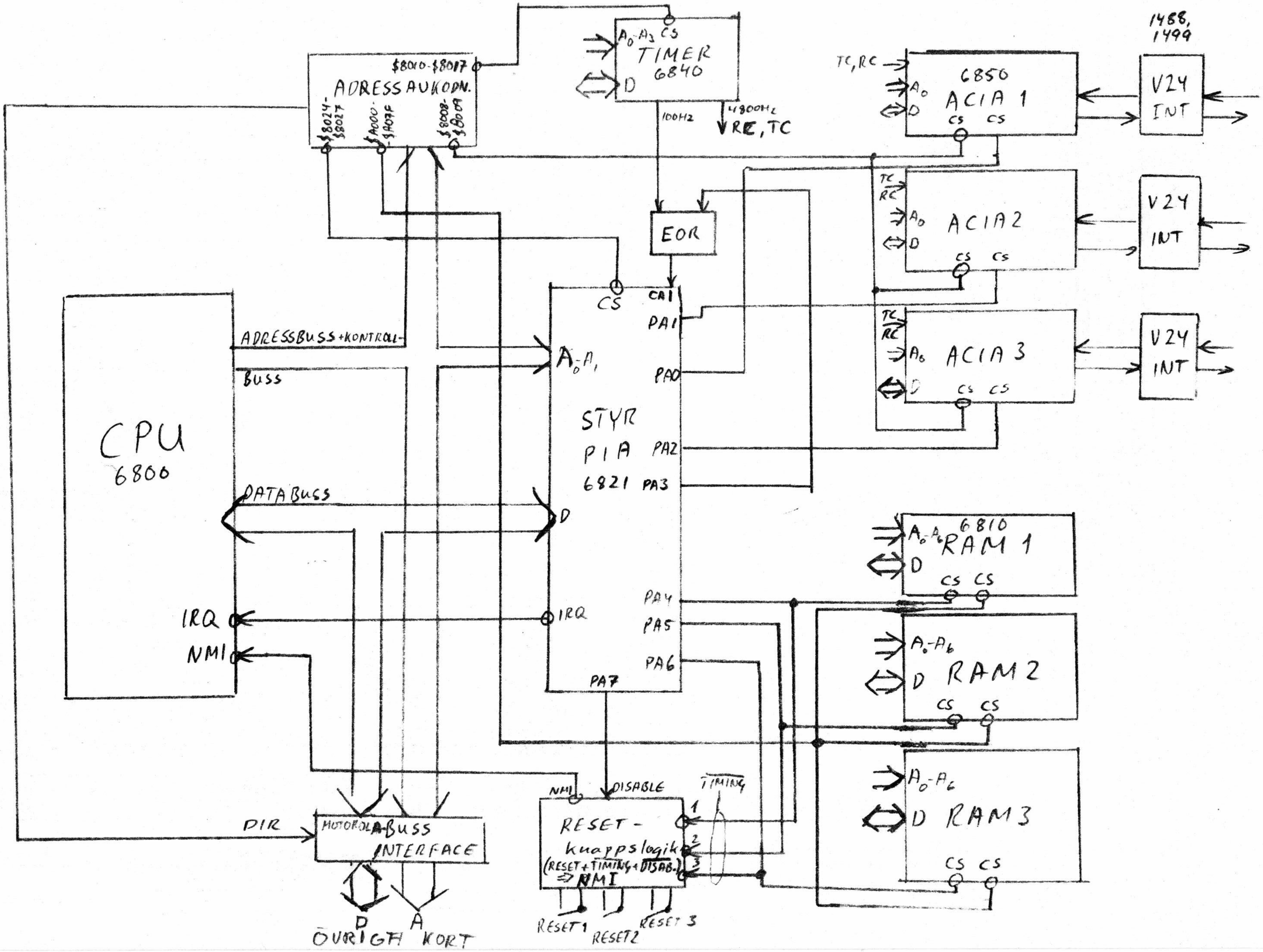
För att prova detta utan den PIAstyrda växeln byggde jag en burk med en omkopplare, så att jag kan byta mellan användare 1, 2 och 3. Innan detta prov sattes ibland interruptmasken vid start av program och detta uptäcktes ej om man inte kunde koppla om mellan de olika användarna.

Systemet var tänkt så att man skulle kunna printa eller mata ut till kassettband samtidigt som jag arbetade med min PROM-BASIC som arbetar mot MIKBUG. Tyvärr är det så att hela systemet går ned när jag försöker starta PROM-BASICen. Vad detta beror på vet jag ännu ej, men jag kan i alla fall hålla två olika enkla jobb igång samtidigt, så att jag har i alla fall visat att ide´n höll.

Carl Axel Bruno



SKISS ÖVER HÅRDVARAN TILL TIMESHARINGSYSTEMET



```
1
2                    ************************************
3                    * THIS IS A MULTIUSER MIKBUG THAT USES THREE
4                    * ACIAS AND THREE STACKRAMS SWITCHED
5                    * BY A SPECIAL CONTROLPIA.DO NOT USE NMI WITHOUT
6                    * SPECIAL HARDWARE TO SYNCR SOFT RESET
7                    * MIKBUG IS A TRADEMARK OWNED BY MOTOROLA.
8                    * THIS CHANGED MIKBUG IS DONE BY CARL AXEL BRUNO
9                    * SOLLENTUNA SWEDEN
10                   * THANKS TO E ANDERBERG AND K ERIKSSON FOR
11                   * DISCUSSIONS ON THE MATTER
12                   * TIMER IS INTENDED TO BE RUN ON 1 MHZ
13                   ************************************
14                            NAM     T1MIK -800414
15   8010             TIMER   EQU     $8010
16   8024             PIAD    EQU     $8024
17   8009             ACIAD   EQU     $8009
18   8008             ACIAC   EQU     $8008
19   E000                     ORG     $E000
20   E000 FE A0 00    IO      LDX     IOV
21   E003 6E 00               JMP     0,X
22   E005 01                  NOP
23   E006 01                  NOP
24   E007 01                  NOP
25   E008 01                  NOP
26   E009 01                  NOP
27   E00A 86 15       LOAD    LDA A   #$15        * INIT ACIA (1,2 OR 3)
28   E00C B7 80 08            STA A   ACIAC
29   E00F 86 11               LDA A   #$11
30   E011 8D 62               BSR     OUTCH
31   E013 8D 63       LOAD3   BSR     INCH
32   E015 81 53               CMP A   #$53
33   E017 26 FA               BNE     LOAD3
34   E019 8D 5D               BSR     INCH
35   E01B 81 39               CMP A   #$39
36   E01D 27 25               BEQ     LOAD21
37   E01F 81 31               CMP A   #$31
38   E021 26 F0               BNE     LOAD3
39   E023 7F A0 0A            CLR     CKSM
40   E026 8D 2D               BSR     BYTE
41   E028 80 02               SUB A   #$02
42   E02A B7 A0 0B            STA A   BYTECT
43   E02D 8D 18               BSR     BADDR
44   E02F 8D 24       LOAD11  BSR     BYTE
45   E031 7A A0 0B            DEC     BYTECT
46   E034 27 05               BEQ     LOAD15
47   E036 A7 00               STA A   0,X
48   E038 08                  INX
49   E039 20 F4               BRA     LOAD11
50   E03B 7C A0 0A    LOAD15  INC     CKSM
51   E03E 27 D3               BEQ     LOAD3
52   E040 86 3F       LOAD19  LDA A   #$3F
53   E042 8D 31               BSR     OUTCH
54   E044 7E E0 E3    LOAD21  JMP     CONTRL
55   E047 8D 0C       BADDR   BSR     BYTE
56   E049 B7 A0 0C            STA A   XHI
57   E04C 8D 07               BSR     BYTE
58   E04E B7 A0 0D            STA A   XLOW
59   E051 FE A0 0C            LDX     XHI
```

```
60   E054 39              RTS
61   E055 8D 53   BYTE    BSR     INHEX
62   E057 48              ASL A
63   E058 48              ASL A
64   E059 48              ASL A
65   E05A 48              ASL A
66   E05B 16              TAB
67   E05C 8D 4C           BSR     INHEX
68   E05E 1B              ABA
69   E05F 16              TAB
70   E060 FB A0 0A        ADD B   CKSM
71   E063 F7 A0 0A        STA B   CKSM
72   E066 39              RTS
73   E067 44      OUTHL   LSR A
74   E068 44              LSR A
75   E069 44              LSR A
76   E06A 44              LSR A
77   E06B 84 0F   OUTHR   AND A   #$0F
78   E06D 8B 30           ADD A   #$30
79   E06F 81 39           CMP A   #$39
80   E071 23 02           BLS     OUTCH
81   E073 8B 07           ADD A   #$07
82   E075 7E E1 D1 OUTCH  JMP     OUTEEE
83   E078 7E E1 AC INCH   JMP     INEEE
84   E07B 8D F8   PDATA2  BSR     OUTCH
85   E07D 08              INX
86   E07E A6 00   PDATA1  LDA A   0,X
87   E080 81 04           CMP A   #$04
88   E082 26 F7           BNE     PDATA2
89   E084 39              RTS
90   E085 8D C0   CHANGE  BSR     BADDR
91   E087 CE E1 9D CHA51  LDX     #MCL
92   E08A 8D F2           BSR     PDATA1
93   E08C CE A0 0C        LDX     #XHI
94   E08F 8D 37           BSR     OUT4HS
95   E091 FE A0 0C        LDX     XHI
96   E094 8D 34           BSR     OUT2HS
97   E096 FF A0 0C        STX     XHI
98   E099 8D DD           BSR     INCH
99   E09B 81 20           CMP A   #$20
100  E09D 26 E8           BNE     CHA51
101  E09F 8D B4           BSR     BYTE
102  E0A109               DEX
103  E0A2 A7 00           STA A   0,X
104  E0A4 A1 00           CMP A   0,X
105  E0A6 27 DF           BEQ     CHA51
106  E0A8 20 96           BRA     LOAD19
107  E0AA 8D CC   INHEX   BSR     INCH
108  E0AC 80 30           SUB A   #$30
109  E0AE 2B 94           BMI     LOAD21
110  E0B0 81 09           CMP A   #$09
111  E0B2 2F 0A           BLE     IN1HG
112  E0B4 81 11           CMP A   #$11
113  E0B6 2B 8C           BMI     LOAD21
114  E0B8 81 16           CMP A   #$16
115  E0BA 2E 88           BGT     LOAD21
116  E0BC 80 07           SUB A   #$07
117  E0BE 39      IN1HG   RTS
118  E0BF A6 00   OUT2H   LDA A   0,X
119  E0C1 8D A4   OUT2HA  BSR     OUTHL
```

```
120  E0C3 A6 00          LDA A  0,X
121  E0C5 08             INX
122  E0C6 20 A3          BRA    OUTHR
123  E0C8 8D F5  OUT4HS  BSR    OUT2H
124  E0CA 8D F3  OUT2HS  BSR    OUT2H
125  E0CC 86 20  OUTS    LDA A  #$20
126  E0CE 20 A5          BRA    OUTCH
127  E0D0 F6 80 24  START  LDA B  PIAD      * RESET REALTIME INTERRUPT
128  E0D3 C4 7F          AND B  #$7F        * AND DISABLE NMI - RESTART
129  E0D5 F7 80 24       STA B  PIAD
130  E0D8 8E A0 42       LDS    #STAKTO
131  E0DB BF A0 08       STS    SP
132  E0DE 86 03          LDA A  #$03        * INIT ACIA (1,2 OR 3)
133  E0E0 B7 80 08       STA A  ACIAC
134  E0E3 86 55  CONTRL  LDA A  #$55        * INIT ACIA FOR 8 BITS AND 1 STOP
135  E0E5 B7 80 08       STA A  ACIAC
136  E0E8 01             NOP
137  E0E9 0E             CLI                * OTHERWISE SWI WOULD LOCK
138  E0EA 01             NOP
139  E0EB 8E A0 42       LDS    #STAKTO
140  E0EE CE E1 9C       LDX    #MCLOFF
141  E0F1 8D 8B          BSR    PDATA1
142  E0F3 8D 83          BSR    INCH
143  E0F5 16             TAB
144  E0F6 8D D4          BSR    OUTS
145  E0F8 C1 4C          CMP B  #$4C
146  E0FA 26 03          BNE    *+5
147  E0FC 7E E0 0A       JMP    LOAD
148  E0FF C1 4D          CMP B  #$4D
149  E101 27 82          BEQ    CHANGE
150  E103 C1 52          CMP B  #$52
151  E105 27 18          BEQ    PRINT
152  E107 C1 50          CMP B  #$50
153  E109 27 32          BEQ    PUNCH
154  E10B C1 47          CMP B  #$47
155  E10D 26 D4          BNE    CONTRL
156  E10F BE A0 08       LDS    SP
157  E112 3B             RTI
158  E113 BF A0 08  SFE  STS    SP
159  E116 30             TSX
160  E117 6D 06          TST    6,X
161  E119 26 02          BNE    *+4
162  E11B 6A 05          DEC    5,X
163  E11D 6A 06          DEC    6,X
164  E11F FE A0 08  PRINT  LDX  SP
165  E122 08             INX
166  E123 8D A5          BSR    OUT2HS
167  E125 8D A3          BSR    OUT2HS
168  E127 8D A1          BSR    OUT2HS
169  E129 8D 9D          BSR    OUT4HS
170  E12B 8D 9B          BSR    OUT4HS
171  E12D CE A0 08       LDX    #SP
172  E130 8D 96          BSR    OUT4HS
173  E132 20 AF     C2   BRA    CONTRL
174  E134 0D        MTAPE1  FCB  $0D
175  E135 0A             FCB    $0A
176  E136 00             FCB    $00
177  E137 00             FCB    $00
178  E138 00             FCB    $00
179  E139 00             FCB    $00
```

```
180   E13A 53                    FCB     $53
181   E13B 31                    FCB     $31
182   E13C 04                    FCB     $04
183   E13D 86 12       PUNCH     LDA A   #$12
184   E13F BD E0 75              JSR     OUTCH
185   E142 FE A0 02              LDX     BEGA
186   E145 FF A0 0F              STX     TW
187   E148 B6 A0 05    PUN11     LDA A   ENDA+1
188   E14B B0 A0 10              SUB A   TW+1
189   E14E F6 A0 04              LDA B   ENDA
190   E151 F2 A0 0F              SBC B   TW
191   E154 26 04                 BNE     PUN22
192   E156 81 10                 CMP A   #$10
193   E158 25 02                 BCS     PUN23
194   E15A 86 0F       PUN22     LDA A   #$0F
195   E15C 8B 04       PUN23     ADD A   #$04
196   E15E B7 A0 11              STA A   MCONT
197   E161 80 03                 SUB A   #$03
198   E163 B7 A0 0E              STA A   TEMP
199   E166 CE E1 34              LDX     #MTAPE1
200   E169 BD E0 7E              JSR     PDATA1
201   E16C 5F                    CLR B
202   E16D CE A0 11              LDX     #MCONT
203   E170 8D 25                 BSR     PUNT2
204   E172 CE A0 0F              LDX     #TW
205   E175 8D 20                 BSR     PUNT2
206   E177 8D 1E                 BSR     PUNT2
207   E179 FE A0 0F              LDX     TW
208   E17C 8D 19       PUN32     BSR     PUNT2
209   E17E 7A A0 0E              DEC     TEMP
210   E181 26 F9                 BNE     PUN32
211   E183 FF A0 0F              STX     TW
212   E186 53                    COM B
213   E187 37                    PSH B
214   E188 30                    TSX
215   E189 8D 0C                 BSR     PUNT2
216   E18B 33                    PUL B
217   E18C FE A0 0F              LDX     TW
218   E18F 09                    DEX
219   E190 BC A0 04              CPX     ENDA
220   E193 26 B3                 BNE     PUN11
221   E195 20 9B                 BRA     C2
222   E197 EB 00       PUNT2     ADD B   0,X
223   E199 7E E0 BF              JMP     OUT2H
224   E19C 13          MCLOFF    FCB     $13
225   E19D 0D          MCL       FCB     $0D
226   E19E 0A                    FCB     $0A
227   E19F 14                    FCB     $14
228   E1A0 00                    FCB     $00
229   E1A1 00                    FCB     $00
230   E1A2 00                    FCB     $00
231   E1A3 2A                    FCB     $2A
232   E1A4 04                    FCB     $04
233   E1AC                       ORG     $E1AC       * MIKBUGCOMPATIBIL
234   E1AC 37          INEEE     PSH B
235   E1AD F6 80 08    INE1      LDA B   ACIAC
236   E1B0 C4 01                 AND B   #$01
237   E1B2 26 05                 BNE     INE2        * IF REC BUFFER FU
238   E1B4 BD E2 DF              JSR     SETINT      * SET A REAL TIME
239   E1B7 20 F4                 BRA     INE1
```

```
240  E1B9 B6 80 09  INE2    LDA A  ACIAD
241  E1BC 84 7F             AND A  #$7F
242  E1BE F6 80 08  INE3    LDA B  ACIAC     * WAIT FOR OUTPUT
243  E1C1 C4 02             AND B  #$02
244  E1C3 27 F9             BEQ    INE3
245  E1C5 B7 80 09          STA A  ACIAD
246  E1C8 33                PUL B            * RETURN B
247  E1C9 39                RTS
248  E1D1                   ORG    $E1D1
249  E1D1 37        OUTEEE  PSH B
250  E1D2 84 7F             AND A  #$7F
251  E1D4 F6 80 08  OUTE1   LDA B  ACIAC     * WAIT FOR END OF SEND
252  E1D7 C4 02             AND B  #$02
253  E1D9 26 05             BNE    OUTE2     * IF TR BUFFER EMPTY
254  E1DB BD E2 DF          JSR    SETINT    * SET REALTIME INTERRUPT
255  E1DE 20 F4             BRA    OUTE1
256  E1E0 B7 80 09  OUTE2   STA A  ACIAD     * SEND
257  E1E3 33                PUL B
258  E1E4 39                RTS
259  E1E5 0F        STARTA  SEI
260  E1E6 8E A0 42          LDS    #$A042    * SET STACKPOINTER
261  E1E9 CE FF 07          LDX    #$FF07
262  E1EC FF 80 24          STX    PIAD      * INITIALISE CONTROLPIA
263  E1EF 86 69             LDA A  #$69      * INITIALISE TERMINAL1
264                    *(PA0-2 HAVE HIGH SIGN,PA4-6 HAVE LOW SIGN,PA3 NEEDS
265                    * BE HIGH TO ALLOWOW RESET VIA NMI)
266  E1F1 B7 80 24          STA A  PIAD
267  E1F4 CE A0 40          LDX    #$A040
268  E1F7 86 00             LDA A  #$00      * RESET INTERRUPTMASK ON ST
269  E1F9 A7 00     STA1    STA A  0,X
270  E1FB 08                INX
271  E1FC 8C A0 48          CPX    #$A048
272  E1FF 26 F8             BNE    STA1
273  E201 CE E0 D0          LDX    #START
274  E204 FF A0 48          STX    $A048
275  E207 CE A0 42          LDX    #$A042    * SET STACKPOINTER
276  E20A FF A0 06          STX    NIO       * YOU COULD USE STS NIO ALS
277  E20D CE E2 AD          LDX    #INT2     * FROM TERM 1 TO TERM 2
278  E210 FF A0 00          STX    $A000
279  E213 86 5A             LDA A  #$5A      * INIT TERMINAL 2
280  E215 B7 80 24          STA A  PIAD
281  E218 CE A0 40          LDX    #$A040
282  E21B 86 00             LDA A  #$00      * RESET INTERRUPTMASK ON ST
283  E21D A7 00     STA2    STA A  0,X
284  E21F 08                INX
285  E220 8C A0 48          CPX    #$A048
286  E223 26 F8             BNE    STA2
287  E225 CE E0 D0          LDX    #START
288  E228 FF A0 48          STX    $A048
289  E22B CE A0 42          LDX    #$A042
290  E22E FF A0 06          STX    NIO
291  E231 CE E2 C6          LDX    #INT3
292  E234 FF A0 00          STX    $A000
293  E237 86 3C             LDA A  #$3C      INITIALISE TERMINAL 3
294  E239 B7 80 24          STA A  PIAD
295  E23C CE A0 40          LDX    #$A040
296  E23F 86 00             LDA A  #$00      * RESET INTERRUPTMASK ON ST
297  E241 A7 00     STA3    STA A  0,X
298  E243 08                INX
299  E244 8C A0 48          CPX    #$A048
```

```
300  E247 26 F8          BNE    STA3
301  E249 CE E0 D0       LDX    #START
302  E24C FF A0 48       STX    $A048
303  E24F CE A0 42       LDX    #$A042
304  E252 FF A0 06       STX    NIO
305  E255 CE E2 94       LDX    #INT1        * FROM TERMINAL 3
306  E258 FF A0 00       STX    $A000
307  E25B CE FF 04       LDX    #$FF04
308  E25E FF 80 26       STX    PIAD+2       * INIT PIAB
309  E261 86 86          LDA A  #$86         * ENABLE PRINTER
310  E263 B7 80 26       STA A  PIAD+2       * AND TIMER CHIP
311  E266 BD E2 6B       JSR    TIMINI       * INIT TIMER
312  E269 01             NOP
313  E26A 3B             RTI                 * START TERMINAL
314  E26B 86 03  TIMINI  LDA A  #$03         * OPEN CONTROLREG
315  E26D B7 80 11       STA. A TIMER+1
316  E270 86 03          LDA A  #$03
317  E272 B7 80 10       STA A  TIMER
318  E275 86 82          LDA A  #$82         * TIMER 1 CONTROL
319  E277 C6 82          LDA B  #$82         * TIMER 2 CONTROL
320  E279 B7 80 10       STA A  TIMER
321  E27C F780 11        STA B  TIMER+1
322  E27F CE 13 88       LDX    #5000        * TIMER 1 FOR REA
323  E282 FF 80 12       STX    TIMER+2
324  E285 CE 00 67       LDX    #103         * TIMER 2 FOR TR
325  E288 FF 80 14       STX    TIMER+4
326  E28B 86 06          LDA A  #$06         * DISABLE TIMER C
327  E28D B7 80 26       STA A  PIAD+2
328  E290 39             RTS
329  E294                ORG    $E294
330  E294 86 3C  INT1    LDA A  #$3C
331  E296 B7 80 24       STA A  PIAD         * DISABLE NMI-RES
332  E299 BF A006        STS    NIO          * SAVE SP NO 3
333  E29C B6 80 24       LDA A  PIAD
334  E29F 86 69          LDA A  #$69
335  E2A1 B7 80 24       STA A  PIAD
336  E2A4 BE A0 06       LDS    NIO          * GET SP FOR NO 1
337  E2A7 86 E9          LDA A  #$E9
338  E2A9 B7 80 24       STA A  PIAD
339  E2AC 3B             RTI
340  E2AD 86 69  INT2    LDA A  #$69
341  E2AF B7 80 24       STA A  PIAD
342  E2B2 BF A0 06       STS    NIO          * SAVE SP FOR NO
343  E2B5 B6 80 24       LDA A  PIAD
344  E2B8 86 5A          LDA A  #$5A
345  E2BA B7 80 24       STA A  PIAD
346  E2BD BE A0 06       LDS    NIO          * GET SP FOR NO 2
347  E2C0 86 DA          LDA A  #$DA
348  E2C2 B7 80 24       STA A  PIAD
349  E2C5 3B             RTI
350  E2C6 86 5A  INT3    LDA A  #$5A
351  E2C8 B7 80 24       STA A  PIAD
352  E2CB BF A0 06       STS    NIO          * SAVE SP FOR NO
353  E2CE B6 80 24       LDA A  PIAD         * RESET ITERRUPT
354  E2D1 86 3C          LDA A  #$3C
355  E2D3 B7 80 24       STA A  PIAD
356  E2D6 BE A0 06       LDS    NIO          * GETSP FOR NO 3
357  E2D9 86 BC          LDA A  #$BC
358  E2DB B7 80 24       STA A  PIAD
359  E2DE 3B             RTI
```

```
360   E2DF 0F          SETINT  SEI
361   E2E0 F6 80 24            LDA B  PIAD
362   E2E3 C4 F7               AND B  #$F7
363   E2E5 F7 80 24            STA B  PIAD
364   E2E8 CA 08               ORA B  #$08
365   E2EA F780 24           STA B  PIAD
366                  * USE LOGIC THAT ENABLES BOTH CLOCK AND PIAINTERRUPTS
367   E2ED 0E                  CLI
368   E2EE 01                  NOP
369   E2EF 39                  RTS
370   E2F0 8E A0 3A  STARTE    LDS    #$A03A     * DO NOT DESTROY OLD STARTAD
371   E2F3 F6 80 24            LDA B  PIAD
372   E2F6 C4 7F               AND B  #$7F
373   E2F8 F7 80 24            STA B  PIAD
374   E2FB CE F4 00            LDX    #$F400     * ABOUT 40 TO 150 MSECS DELA
375   E2FE 0E                  CLI
376   E2FF 08        STARTI    INX
377   E300 8C FF FF            CPX    #$FFFF
378   E303 26 FA               BNE    STARTI
379   E305 7E E0 D0            JMP    START
380   E3F8                     ORG    $E3F8
381   E3F8 E0 00               FDB    IO
382   E3FA E1 13               FDB    SFE
383   E3FC E2 F0               FDB    STARTE
384   E3FE E1E5              FDB    STARTA
385   A000                     ORG    $A000
386   A000           IOV       RMB    2
387   A002           BEGA      RMB    2
388   A004           ENDA      RMB    2
389   A006           NIO       RMB    2          * AS NMI INTERRUPTS ARE USED
390   A008           SP        RMB    1
391   A009                     RMB    1
392   A00A           CKSM      RMB    1
393   A00B           BYTECT    RMB    1
394   A00C           XHI       RMB    1
395   A00D           XLOW      RMB    1
396   A00E           TEMP      RMB    1
397   A00F           TW        RMB    2
398   A011         MCONT     RMB    1
399   A012           XTEMP     RMB    2
400   A014           STACK     RMB    46
401   A042           STAKTO    RMB    1
402   A043           STAKK     RMB    7
404                            END
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

```
ACIAC   8008    ACIAD   8009    BADDR   E047    BEGA    A002    BYTE    E055
BYTECT  A00B    C2      E132    CHA51   E087    CHANGE  E085    CKSM    A00A
CONTRL  E0E3    ENDA    A004    IN1HG   E0BE    INCH    E078    INE1    E1AD
INE2    E1B9    INE3    E1BE    INEEE   E1AC    INHEX   E0AA    INT1    E294
INT2    E2AD    INT3    E2C6    IO      E000    IOV     A000    LOAD    E00A
LOAD11  E02F    LOAD15  E03B    LOAD19  E040    LOAD21  E044    LOAD3   E013
MCL     E19D    MCLOFF  E19C    MCONT   A011    MTAPE1  E134    NIO     A006
OUT2H   E0BF    OUT2HA  E0C1    OUT2HS  E0CA    OUT4HS  E0C8    OUTCH   E075
OUTE1   E1D4    OUTE2   E1E0    OUTEEE  E1D1    OUTHL   E067    OUTHR   E06B
OUTS    E0CC    PDATA1  E07E    PDATA2  E07B    PIAD    8024    PRINT   E11F
PUN11   E148    PUN22   E15A    PUN23   E15C    PUN32   E17C    PUNCH   E13D
```

KLUBBDATOR

Mellan medlemmar och på möten har små diskusioner förekommit om en klubbdator. Dessa har varit lösa och trevanden.Detta, plus de tillsynes praktiska svårigheterna som lokalitet, vem skall sköta den och ett stort bundet kapital, har nog verkat att frågan har lagts på is.

Nu på vårkanten verkar det som isen har smält och frågan har blivit aktuell igen. Vid årsmötet i mars lade klubstyrelsen fram ett förslag till en klubbdator. Dess huvudide var en dator med två till fyra minidiskar som skall användas via modem. Där kan medlemmarna hämta befintliga program och sända in eget. Detta skulle även gällainformation och medelanden.

Förslaget diskuterades under mötet och andra ideer kom fram.Till slut bestämdes att en kommit Ö skulle tillsättas och vad den skulle göra.

Först skall den ha kontakt och sammla in synpunkter från medlemmarna om dess vara eller icke vara "Shakespeare",och eventuel funktioner. För det andra kolla system, kort och program kontra kostnad. Till slut lägga fram ett eller några förslag till styrelsen.

Denna fråga berör alla medlemmar oavsett om man har ett nybörjarkit eller system med disk. Det jag därför vill är att alla ställer frågor och synpunkter om detta till er själva. Dessa skall ni skicka till kommiten.

Var skall datorn stå? Vem vilka skall skötaden? Har vi några "experter" på hårdvara och mjukvara inom klubben? Vad får man för de cirka 10 000 tillgängligt kapital? Vilka fördelar kan någon utan modem få ut av en klubbdator?

Ja det finns massor av frågor och jag skall inte ställa dom själv. Välkommern med synpunkter till kommiten.

Den består av.

Tommy Riboe Robert Lundin Jason King Nisse Ohlson

BS

SAMKÖP

Det finns printrar från MOTOROLA och SAVEN som kan vara attraktiva för oss iklubben. De kostar ca 4500-4800 +moms men kan man gå samman så att man kan köpa 25 st så är priset ca 3000+moms.De som är itresserade kan sända in namn och address till PD68.Och om det är tillräckligt stort intresse så skaffar klubben datablad och sänder till de som harsänt in sitt intresse.

BO LJUNGBLAD