MPU-laren nr 1 1982

Tidskrift för Privatdataklubben PD68

REDAKTIONEN

MPU-larenredaktionen består av

Roger Holmstrand (RH)
Rapsodivägen 168
142 00 TRÅNGSUND 08/7713852
(FLEX2,FLEX9,Papperstape)

Anders Karlsson (AK) Annonschef
Råbystigen 3
197 00 BRO Tel Arb 08/7307704
(FLEX2,FLEX9)

Bernt Svensson (BS)
Östgötag 31,4 tr
116 25 STOCKHOLM 08/448864

Lennart Uusitalo (LU)
Skrakvägen 20
141 72 HUDDINGE
08/466628

REDAKTIONEN BEHÖVER FÖRSTÄRKNING VISA LITE INTRESSE OCH STÄLL UPP ALLT DU BEHÖVER KUNNA ÄR ATT HANTERA SAXEN. DU DU BÖR DESSUTOM HA TILLGÅNG TILL DATOR MEN DET ÄR INTE AVGÖRANDE. (VI TRÄFFAS LITE DÅ OCH DÅ FÖR ATT GÅ IGENOM TIDNINGEN)

ANNONSER

Helsida	200:-
Halvsida	100:-
Kvartsida	50:-
Bilagor	300:-

Priserna gäller tryckfärdigt material.Radannonser för medlemmar är gratis om det inte gäller firma.Tidningens upplaga är för närvarande 425 ex.
För material till tidningen ring eller skriv gärna till någon i redaktionen ovan eller sänd detta till: PD68
BOX 1098 122 22 ENSKEDE

STYRELSEN

PD68 nuvarande styrelse.

Werner Gustafsson (WG) Ordförande
Klövervalsvägen 63C
183 44 TÄBY
(FLEX2,FLEX9)

Sven Erik Ringström (SER) V Ordf.
Högbyvägen 18
175 46 JÄRFÄLLA
(FLEX2)

Lennart Nilsson Sekreterare
Grusåsgränd 44
122 49 ENSKEDE

Bo Ljungblad (BLj) Kassör
Brännkyrkagatan 117
117 28 STOCKHOLM

Tommy Blath (TBL) Suppleant
Norrholmsvägen 106
132 00 SALTSJÖBOO (FLEX2,FLEX9,KCS)

Åke Holm (ÅH) Suppleant
Långsjövägen 15B
135 54 TYRESÖ

Tore Hall V Sekreterare
Sävstaholmsvägen 209
125 36 ÄLVSJÖ

Gunnar Lovius Suppleant
Ragnebergsvägen 21
136 67 HANDEN

MEDLEMSINTRÄDE

Vill DU bli medlem i PD68 ?
Sätt in 100:- på postgiro nr PG 960468-7 samt namn,adress o tele nr 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 TYVÄR ÄR DET INTE SÅ,MEN SKÄRPNING,ALLT ÄR VÄLKOMMET STORT SOM smått.MED HÄLSNING REDAKTIONEN

PS. DET ÄR FÖR OSS EN FÖRDEL OM NI KAN SÄNDA MATERIALET PÅ DISK (35 sp, ENKEL SIDIG,ENKEL DENSITET) ELLER TAPE. VI RETURNERAR SEDAN DISKEN ELLER TAPEN.

inLEDAREN

Återigen har ett årsmöte hållits, och jag (Verner Gustavsson) har blivit vald till ordförande efter Tommy Bladh, som nu resignerat till suppleant. Vi tackar Tommy för hans insatser som ordförande.

Föreningen är nu inne på sitt femte verksamhetsår, och har vuxit från en förening på cirka 50 medlemmar 1978 då klubben startades, till nästan 350 st (1982). Men även datorkraften bland våra medlemmar har vuxit från "enkla" kit med hexadecimala knappsatser till sofistikerade maskiner med en väldigt massa ´K´ och sofistikerade operativsystem. Därför är det viktigt att våra nya medlemmar får en god insikt i detta med bitar och bytes. Mer om detta nedan.

Man kan undra : Skriver inte våra medlemmar program som passar att publiceras i MPU-laren ? Rota fram lämpliga snuttar ur era gömmor och sänd in dessa till tidningen. Glöm dock ej att skicka med en enkel beskrivning hur dessa exelenta program funkar. Vem blir till exempel först med att publicera ett eget program skrivit i PASCAL ?
Mängden insänt material har glädjande nog ökat, men endast ett fåtal medlemmar är flitiga med insändandet.

Under hösten har det under Hans Degermans ledning hållits nybörjarkurser i maskinspråksprogrammering. Dessa kurser har blivit väldigt populära och kursverksamheten planeras att utökas, bland annat har förslag väcks att inköpa ett antal kit för utbildning.

En datordoktor har även utsetts. Det är Roger Holmstrand som med steto ..förlåt! oscilloskopets hjälp skall leta sig fram till fel på era krånglande hembyggen. Han kommer även att delegera ut problem till andra kompetenta personer . OBS! Frågor till D.D. ställes brevledes till föreningens postbox.
Under vissa möten (vilket meddelas i god tid före mötet) kommer CA-elektronik att ställa upp med en maskin för test och kontroll av problemmatiska hembyggda kort, detta gäller endast kort till den s.k RT-datorn.

Varm och trevlig sommar önskar

V. Gustavsson

SIDA	INNEHÅLL
4	STADGAR.
6	MUSIK&TAL MED DATOR.
7	FLEX9 UTILITES.
10	TALANDE MATTE PROGRAM.
13	REVISIONS-BERÄTTELSE 1981.
15	TALANDE MONITOR.
20	OM PROGRAMMEN PÅ SID 21-31.
21	REALTIDS-KLOCKA.
29	LOTTO-SYSTEM (UTÖKAT).
31	TBUG2C MONITORN.
54	FLEX-DISK BESKRIVNING.

PD68:s stadgar

§ 1 KLUBBENS UPPGIFT Klubbens ändamål är att ideellt verka för medlemmarna genom:
att sprida information om privatdatans möjligheter inom lagens ram,
att utbyta maskin- och programvarulösningar inom klubben.

§ 2 MEDLEMSKAP
Medlemskap kan erhållas genom ansökan och erläggande av medlemsavgift.
mom 1. Medlemskap upphör om ej medlemsavgift erlags inom mars månads utgång.
mom 2. Hedersmedlemmar väljes av styrelsen.

§ 3 STYRELSEN
Klubbens angelägenheter handhaves av en styrelse som äger handla å klubbens vägnar med de inskränkningar i dess befogenhet som förekomma i dessa stadgar.
Styrelsen består av fem ledarmöter valda för en tid av två år. Valet förrättas på ordinarie årsmöte. Första gången dessa stadgar är i kraft väljes tre ledarmöter för två år och två för ett år. För styrelsens ledamöter väljes tre suppleanter.

§ 4 STYRELSEVAL
Styrelsen utser inom sig ordförande, kassör, sekreterare, vice ordförande och vice sekreterare.

§ 5 STYRELSEBESLUT
Vid styrelsens sammanträden skall protokoll föras över dess beslut. Styrelsen är beslutsför då mer än halva antalet ledarmöter är närvarande. Såsom styrelsens beslut gäller den mening, varom de flesta röstande förena sig. Vid lika röstetal gäller den mening som bitäds av ordföranden.

§ 6 KLUBBENS TECKNARE
Styrelsen utser inom sig de två personer som i föreningen äger rätt att teckna klubbens ekonomiska transaktioner.

§ 7 STYRELSENS HEMVIST
Styrelsen har sin hemvist i Stockholm.

§ 8 KLUBBENS EKONOMI
Styrelsen tillser att betrygande försäkring finns för klubbens egendom och kassaförvaltning. Kontanta tillgångar placeras, i den mån de inte behövs för omedelbart förestående utbetalningar, på postgiro.

§ 9 RÄKENSKAPSÅR
Kalenderåret är räkenskapsår. Styrelsen upprättar balansräkning, vinst- och förlusträkning samt överlämnar dessa handlingar jämte protokoll och räkenskaper till revisorerna senast den 31 januari påföljande år.
Styrelsen skall föra förteckning över medlemmarna.
Styrelsen skall vid ordinarie årsmöte avgiva berättelse för det gågna verksamhetsåret.

§ 10 REVISION
För granskning av styrelsens förvaltning och räkenskaper utses på ordinarie årsmöte två revisorer och en revisorsuppleant. Mandatet gäller intill dess nästa ordinarie årsmöte hållits. Revisorerna skall senast tre veckor före ordinarie årsmöte till styrelsen överlämna revisionsberättelse.

§ 11 ÅRSMÖTE

Ordinarie årsmöte hålles årligen senast under februari månad. Extra årsmöte hålles då styrelsen finner så lämpligt eller då minst en tredjedel av klubbens medlemmar skriftligen, med angivande av skäl, så påfordrar.

§ 12 KALLELSE

Kallese till ordinarie årsmöte skall ske senast två veckor och till extra möte senast en vecka före mötets avhållande.

§ 13 ÅRSMÖTESFÖRHANDLINGAR

Vid ordinarie årsmöte skall följande ärenden tas upp till behandling:

1. Val orförande på mötet
2. Val av justeringsmän
3. Fråga om kallese bönhörligen skett
4. Anmälan om ej förhandsanmälda frågor samt beslut om dessa skall tas upp till behandling på mötet
5. Styrelsens redovisningsbehandlingar
6. Revisorernas berättelse
7. Fastställande av balansräkning
8. Fråga om ansvarsfrihet fär avgående styrelsen
9. Fråga om beslut i anledning av uppkommen vinst eller förlust enligt den fastställda balansräkningen
10. Styrelsens förslag till inkomst- och utgiftsstat samt fastställande av årsavgift
11. Val av:
    a) styrelseledamöter
    b) suppleanter
    c) revisorer
    d) revisorsuppleant
    e) valberedning
12. Inkomna motioner
13. Övriga frågor som under mötet under punkt 4 ovan beslutat skall tas upp till behandling
14. Avslutning

    Intill dess ordförande på mötet blivit utsedd förses ordet av den avgående styrelsens ordförande eller vice ordförande eller, vid förfall för båda dessa , den styrelsen därtill utser. Den avgående styrelsens sekreterare svarar för mötesprotokollet.

§ 14 MEDLEMSÄRENDEN

Ärenden som enskild medlem senast tre veckor före mötets avhållande anmält för behandling på mötet till styrelsen, skall styrelsen bereda och avge yttrande över före behandling på mötet. Ärenden som ej förhandsanmält skall tas upp till behandling på mötet där minst två tredjedelar av de närvarande röstberättigade medlemmarna så beslutar.

§ 15 STADGEÄNDRING

Ändring av dessa stadgar skall, för att bli gällande, antagas av två på varandra följande klubbmöten varav minst ett ordinarie.

§ 16 KLUBBENS UPPLÖSNING

Upplöses klubben, skall, sedan verksamheten lagligen avvecklats, eventuellt överskott tillfalla en av styrelsen godkänd ideell verksamhet.

* * * LÅT DATORN TALA OCH MUSICERA * * *

Av Åke Stenäng

Intresset för datoriserat tal är stort att döma av de telefonsamtal jag fått efter min förra artikel i MPU-laren, där jag nämnde lite om en talsynthesizer. Ska försöka utveckla detta lite mer, men först något om datorer och musik.

Datorer och musik: Efter en annons i 68 MICRO journal inköpte jag en editor för fyrstämmig musik som passade till ett gammalt SS-30 musikkort, från Newtech. Editorn är från Speech Systems U.S.A. och kostar $39.95. Den är mycket lätt att använda för den som är bekant med noter. Om man har en fyrstämmig sats är det bara att i stort sett skriva noternas namn, plus höjningar och sänkningar. Vidare bestämmer man i vilken oktav man vill ha resp ton. 7 oktavers omfång har editorn! Vill man komponera själv går det ju också bra, men det är faktiskt bekvämare att sätta sig vid pianot!? Musiken är förstås användbar i demonstrationssyfte och till div spel. Tonerna stämmer dock inte riktigt och någon större njutbarhet finns det knappast i musiken, men som sagt, det går att använda. Man skulle ju kunna utveckla det hela genom att koppla in en musik synthesizer och då få lite variation på den enformiga tonkvaliten. Tänker man på Ralf Lundstens fina elektronmusik kompositioner han gjort i sin elektronstudio Andromida, ja då är vägen mycket lång att gå!

Datorer och tal: Som jag nämnde i förra MPU-laren har jag inköpt en talsynthesizer "VS-1 Speaker" från Alford o Associates, U.S.A med tillhörande programvara. En komplett VS-1 kostar $189.95. SS-30 kortet består i stort sett av en PIA plus talkretsen SC-01A, som bara den kostar 65 dollar. En liten förstärkare för direkt drivning av en högtalare finns också. Man kan även koppla till några reläer att användas som automatisk telefonsvarare. Man programmerar med en kod som består av 64 amerikanska phonem och talkretsen omvandlar sedan dessa phonem till tal. Det går också att programmera svenskt tal, men då med en mycket tydlig amerikansk brytning. Ja, varför inte, kretsen kommer ju från U.S.A.!? Vill Du höra talprov, så slå en signal!

Använder även talkortet i demonstrationssyfte, men två av de medföljande programmen visar hur man kan utnyttja tal synthesizern till lite nytta. VOXMATTE.BAS liknar MATTE.BAS i förra numret, men nu utökat med tal. Alltså en träning i både engelska och matte för mina barn. En kväll förhörde jag grabben på OÄ och datorn förhörde flickan på multiplikation!! Som synes tar phonemen inte mer plats än annan text och passar därför mycket bra till BASIC program.

Lite då och då händer det att man vill kolla mot en lista vad som finns i minnet och då är VOXMEM.CMD mycket användbart. Man knappar bara in önskad startadress och sedan kan man lugnt sitta och lyssna och jämföra mot datalistan!

Editorn VOX-EDIT till talsynthesizern är mycket enkel och praktisk att använda. Man kan från editorn skriva ett ord och kolla hur det låter, skriva ut det på printer eller spara ord/meningar på diskfiler m.m. Manualen innehåller en lista på de 64 phonem som finns tillgänliga i SC-01 och med hjälp av denna lista tar vi ett litet exempel. Vi kanske vill åstadkomma ordet "fader", alltså "father" på engelska (med USA-uttal). Man hittar då i listan att följande phonem borde passa.

F AH1 THV ER

Sedan vill vi också få ordets betoning att låta som vi tänkt och då kommer det att se ut så här istället.

1/F 1/AH1 2/THV 3/ER

Ordet "father" bör nu låta nu som en fråga, "father?" och detta kan man omedelbart kontrollera. Vill man ändra på något finns det enkla kommandon också för detta, t.ex Insert, Delete osv.

När man ska transponera dessa phonem till svenska får man försöka ljuda och plocka ut de phonem man tror passar bäst och sedan är det ju bara att lyssna och ändra! En del ord går bättre och andra går sämre, men användbart är det definitivt.

Manualen är mycket utförlig och bra. Den beskriver först olika synthesize metoder och sedan följer en bra editorbeskrivning. Hur man initierar talkortet i både BASIC och assembler visas utförligt. En lång ordlista på många vanliga ord för användning i editorn finns också. Den tekniska beskrivningen är lätt att följa. Kretskortet är av mycket hög kvalitet och jag har inte haft några som helst problem med varken komponenter eller programvara. Så till något annat!

* * * FLEX 9 UTILITIES * * *

Lovade i förra MPU-laren att komma med några nya utilities från Allen Clark i USA. Här är då programmen, som är väl dokumenterade i textfilerna så någon ytterligare kommentar behövs knappast. Tyvärr finns ett fel i min förra artikel, MPU-laren nr 4 -81, sid 7 tredje stycket. Där står att det finns en odokumenterad FLEX-rutin i SEARCH.CMD men det skall vara SLIST.CMD. Eftersom denna rutin ligger på olika adresser i olika FLEX-versioner är det bättre att byta ut rutinen till följande:
Ta bort "PDATA EQU $CE83".
Ersätt "PRINT JMP PDATA", med

```
PRINT   LDA     ,X+
        CMPA    #4
        BNE     PRINT1
        RTS
PRINT1  JSR     PUTCHR
        BRA     PRINT
```

Av de tre programmen följer här endast den förklarande början av varje program. Resten får Du på disk!

DIR.CMD:

```
**********************************************************************
* DIR - PRINT DIRECTORY CONTENT
*       INCLUDING DELETED FILES.
*
* MODIFIED "DIR" UTILITY BY
*   ALLEN CLARK
*
* "DIR" UTILITY WILL PRINT COMPLETE PROFILE OF DISK FILES
*       FILE NO., BEGIN, END TRACK AND SECTOR NUMBERS AS
*       WELL AS FILE SIZE, CREATION DATE AND FILE ATTRIBUTES.
*       FINALLY, ANY DELETED FILES WILL BE FLAGGED WITH AN (*)
*       ASTERISK IN FIRST CHARACTER OF FILE NAME AND "DELETED"
*       MESSAGE WILL BE PRINTED AT END OF DIRECTORY LINE IF THE
```

```
*       "DELETED FILES DISPLAY OPTION" IS SELECTED BY PLACING A
*       ( + ) PLUS SYMBOL IN THE COMMAND LINE WHEN "DIR" IS
*       INVOKED.
*                 FORMAT:  DIR,+  ....FOR DISPLAY OF DELETED FILES.
*
* DIRECTORY OF DRIVE NUMBER 1
* DISK: PROV  #45    CREATED: 18-SEP-81
*
* FILE#   NAME    TYPE  R  BEGIN   END   SIZE    DATE      PRT
*
*
*     1  PCOPY   .CMD      01-01  01-05     5  16-AUG-81
*     2  UNDEL   .TXT      01-06  03-03    18  18-SEP-81
*     3  *UB8    .TXT      03-04  03-04     1  18-SEP-81   DELETED
*     4  SUB1    .TXT      08-06  08-09     4  18-SEP-81
*     5  *PEEDUP .TXT      08-0A  0A-06    17  17-SEP-81   DELETED
*
*      FILES=3, SECTORS=45, LARGEST=18, FREE=345
******************************************************************

UNDEL.CMD:
******************************************************************
* UNDELETE DISK FILE UTILITY
* FOR FLEX 9.0 6809.
*
* WRITTEN BY
* WALLACE A. WATSON
*
* "UNDELETE" DISK UTILITY ALLOWS RECOVERY OF A DELETED
* DISK FILE ON DISKETTE IN DRIVE NUMBER SPECIFIED IN
* DRIVE SPEC. ON COMMAND LINE.
*
* FORMAT: UNDEL(,DRIVE)
* DRIVE SPEC IS OPTIONAL
* DEFAULT IS TO WORKING DRIVE
* UNDELETE WILL PROMPT FOR A NEW FILE NAME.
******************************************************************

HEADER.CMD:
******************************************************************
*  WRITTEN BY:
* Wallace A. Watson
*
* HEADER - PRINT/CHANGE HEADER UTILITY
*          WILL PRINT DISK VOLUME NAME,
*          NUMBER, AND CREATION DATE AND
*          IF DESIRED ANY OR ALL ITEMS
*          MAY BE CHANGED FROM THE KEYBOARD.
*
* FORMAT:  HEADER<,DRIVE NO.>
* DEFAULT: DRIVE NUMBER IS OPTIONAL, IF NOT
*          ENTERED ON COMMAND LINE, DEFAULT
*          IS TO WORKING DRIVE.
*
*"PROMPTS" A PROMPT WILL BE MADE FOLLOWING
*          PRINTOUT OF VOLUME NAME, NUMBER
*          AND CREATION DATE.
*
* CHANGES (Y/N OR CR = REPEAT HEADER) ?
*
```

```
*           ANSWER (Y)ES IF ANY CHANGES DESIRED.
*           INDIVIDUAL PROMPTS FOR EACH PIECE OF
*           INFORMATION WILL BE MADE.
*           IF NO CHANGES ARE TO BE MADE, ANSWER
*           (N)O AND FLEX WILL BE REENTERED.
*            CR - WILL RE-DISPLAY HEADER INFORMATION.
*******************************************************************
```

För den som vill ha en kopia, skriver jag ut antal sektorer på varje fil, så kan Du räkna ut hur många disketter Du behöver skicka till mig. (Kanske från förra MPU-laren också). Single density och 35 eller 40 spår plus frankerat och adresserat kuvert! Glöm inte att skriva vad Du önskar. Obs! Allt för 6809!

```
DIR      .TXT   52        VOXMATTE.BAS   35
UNDEL    .TXT   19        VOXMEM  .TXT   26
HEADER   .TXT   26
```

Åke Stenäng, Ljungbergsv. 25, 564 00 Bankeryd, tel 036/724 91

```
10 REM ******************************************
20 REM *** VOXMATTE.BAS för VS-1 SPEECH SPEAKER ***
30 REM ***   från Alford & Associates, U.S.A.   ***
40 REM ***                                      ***
50 REM ***     För FLEX 9 och TSC XBASIC.       ***
60 REM ***              82-02-11                ***
70 REM ***                                      ***
80 REM *** Delar av programmet är från MICRO-   ***
90 REM *** COMPUTING april -81 och några av     ***
100 REM *** talrutinerna är från A & Associates ***
110 REM *** och Speech Systems. Programmet är   ***
120 REM *** sammansatt av Åke Stenäns, Bankeryd. ***
130 REM ******************************************
140 PT=HEX("E000"): REM VS-1 PORT ADDRESS
150 ON ERROR GOTO 3210
160 PRINT CHR$(12)
170 PRINT"* * *  NUMBERS GAME WITH SPEECH, by Å. Stenäns 1982 * * *"
180 PRINT:PRINT
190 PRINT "OK, LET'S PLAY A NUMBERS GAME!"
200 PRINT : PRINT
210 PRINT "I WILL GIVE YOU ADDITION, SUBTRACTION, MULTIPLICATION,"
220 PRINT "OR DIVISION PROBLEMS. A LITTLE BIT OF EACH IF YOU LIKE."
230 PRINT "SIMPLY:"
240 PRINT "   TYPE 1 FOR ADDITION"
250 PRINT "   TYPE 2 FOR SUBTRACTION"
260 PRINT "   TYPE 3 FOR MULTIPLICATION"
270 PRINT "   TYPE 4 FOR DIVISION"
280 PRINT "   TYPE 5 FOR A LITTLE OF EACH":PRINT
290 PRINT "WHEN YOU ARE DONE PLAYING, TYPE A NEGATIVE NUMBER WHEN I"
300 PRINT "GIVE YOU A PROBLEM AND I WILL GIVE YOU YOUR SCORE."
310 PRINT "DON'T FORGET TO DEPRESS THE 'RETURN' KEY AFTER EVERY ENTRY!"
320 PRINT:INPUT"WHAT IS YOUR PLEASURE (1,2,3,4 OR 5)";T
330 REM *** SPEECH PORT INIT ***
340 POKE PT+1,0:POKE PT+3,0
350 POKE PT,255:POKE PT+3,255:POKE PT+1,62
360 POKE PT+3,22:POKE PT+2,131
370 VO=62:GOSUB 1740
380 PRINT CHR$(12)
390 PRINT:INPUT"WHAT MAXIMUM VALUE SHOULD I GIVE YOU FOR ONE NUMBER ";N
400 PRINT
410 INPUT"WHAT MAXIMUM VALUE SHOULD I GIVE YOU FOR THE OTHER NUMBER ";P
420 DA=PEEK(HEX("CC0F"))
430 FOR I=1 TO DA
440 R1 = RND(0)
450 NEXT I
460 PRINT : PRINT
470 IF T < 5 THEN 500
480 G = 1
490 T=INT(4*RND(0)+1)
500 A = A + 1
510 X = INT((N+1)*RND(0))
520 Y = INT((P+1)*RND(0))
530 IF X+Y=0 THEN 510
540 ON T GOTO 560,680,840,960
550 REM ******ADDITION******
560 Z1 = X + Y
570 PRINT A;".  ";X;"+";Y;"=";
580 RESTORE 2110
590 GOSUB 1650
600 X9=X
610 GOSUB 2580
620 RESTORE 2150
630 GOSUB 1650
640 X9=Y
```

```
650 GOSUB 2580
660 GOTO 1110
670 REM ******SUBTRACTION******
680 IF X> Y THEN 720
690 X1 = Y
700 Y = X
710 X=X1
720 Z1 = X - Y
730 PRINT A;".  ";X;"-";Y;"=";
740 RESTORE 2110
750 GOSUB 1650
760 X9=X
770 GOSUB 2580
780 RESTORE 2170
790 GOSUB 1650
800 X9=Y
810 GOSUB 2580
820 GOTO 1110
830 REM ******MULTIPLICATION******
840 Z1 = X * Y
850 PRINT A;".  ";X;"X";Y;"=";
860 RESTORE 2110
870 GOSUB 1650
880 X9=X
890 GOSUB 2580
900 RESTORE 2190
910 GOSUB 1650
920 X9=Y
930 GOSUB 2580
940 GOTO 1110
950 REM ******DIVISION******
960 IF X > 0 THEN 990
970 X = Y
980 Y = 0
990 L = X * Y
1000 Z1 = Y
1010 PRINT A;".  ";L;"/";X;"=";
1020 RESTORE 2110
1030 GOSUB 1650
1040 X9=L
1050 GOSUB 2580
1060 RESTORE 2210
1070 GOSUB 1650
1080 X9=X
1090 GOSUB 2580
1100 REM ******INPUT THE ANSWER PORTION OF THE PROGRAM******
1110 INPUT Z
1120 IF Z<0 THEN 1470
1130 IF Z = Z1 THEN 1290
1140 R3=RND(0)*3+1
1150 ON R3 GOTO 1160,1190,1220
1160 RESTORE 1790
1170 GOSUB 1650
1180 GOTO 1240
1190 RESTORE 1820
1200 GOSUB 1650
1210 GOTO 1240
1220 RESTORE 1920
1230 GOSUB 1650
1240 F = 1
1250 ON T GOTO 570,730,850,1010
1260 REM *************************
1270 REM * THE ANSWER IS CORRECT *
1280 REM *************************
1290 R2=RND(0)*3+1
```

```
1300 ON R2 GOTO 1310,1350,1390
1310 RESTORE 1960
1320 GOSUB 1650
1330 PRINT
1340 GOTO 1420
1350 RESTORE 1990
1360 GOSUB 1650
1370 PRINT
1380 GOTO 1420
1390 RESTORE 2020
1400 GOSUB 1650
1410 PRINT
1420 IF F = 1 THEN 1440
1430 B = B + 1
1440 F = 0
1450 IF G = 1 THEN 490
1460 GOTO 500
1470 A = A - 1
1480 IF A=0 THEN 1610
1490 C = A - B
1500 D = 100 * B/A
1510 D = INT(D * 10 + .5)/ 10
1520 PRINT
1530 PRINT:PRINT"SCORE:"
1540 PRINT "   TOTAL NUMBER OF PROBLEMS =";A
1550 PRINT "   TOTAL NUMBER CORRECT     =";B
1560 PRINT "   TOTAL NUMBER WRONG       =";C
1570 PRINT "   PERCENTAGE CORRECT       =";D;"%"
1580 A=0:B=0:C=0:D=0
1590 PRINT:PRINT"DO YOU WANT TO PLAY AGAIN, (Y)es or (N)o ? ";
1600 IF INCH$(0)="Y" THEN PRINT CHR$(12):PRINT:GOTO 210
1610 RESTORE 2050
1620 GOSUB 1650
1630 END
1640 REM *** READ DATA ***
1650 READ VO
1660 IF VO=255 GOTO 1690
1670 GOSUB 1720
1680 GOTO 1650
1690 VO=62:GOSUB 1720
1700 RETURN
1710 REM *** OUTPUT PHONEME ***
1720 IF PEEK(PT+1)<128 GOTO 1720
1730 X4=PEEK(PT)
1740 POKE PT,VO
1750 POKE PT+1,54
1760 POKE PT+1,62
1770 RETURN
1780 REM SORRY TRY AGAIN
1790 DATA 159,149,113,122,105,195,254
1800 DATA 170,122,85,124
1810 DATA 114,156,123,77,255
1820 REM I KNOW YOU CAN DO BETTER THAN THAT
1830 DATA 85,64,73,105
1840 DATA 77,99,117,119
1850 DATA 169,118,104
1860 DATA 153,110,64,77
1870 DATA 222,118,104
1880 DATA 78,130,106,122
1890 DATA 184,111,64,77
1900 DATA 184,111,64,106,255
1910 REM TRY AND CONCENTRATE
1920 DATA 170,122,85,124
1930 DATA 111,64,77,94
1940 DATA 89,100,99,77,77,95,64,66,77,106,107,96,106,255
```

```
1950 REM CONGRATULATIONS
1960 DATA 89,100,99,77,77,92,107,111,64,106,98,118
1970 DATA 119,119,88,70,97,81,99,77,95,255
1980 REM VERY GOOD
1990 DATA 143,123,107,105
2000 DATA 92,86,86,94,255
2010 REM NICE GOING
2020 DATA 77,85,64,73,105,95
2030 DATA 92,86,116,119,103,84,255
2040 REM COME BACK AGAIN PLEASE, BYE BYE
2050 DATA 89,114,76
2060 DATA 78,175,64,89
2070 DATA 114,156,123,77
2080 DATA 165,152,124,124,82,254
2090 DATA 142,85,124,142,85,124,255
2100 REM HOW MUCH IS
2110 DATA 219,85,116,119
2120 DATA 140,114,106,80
2130 DATA 103,146,255
2140 REM PLUS
2150 DATA 101,88,114,113,95,255
2160 REM MINUS
2170 DATA 76,85,105,77,64,95,255
2180 REM TIMES
2190 DATA 170,85,124,76,82,255
2200 REM DIVIDED BY
2210 DATA 94,73,143,85,99,124,94,73,94
2220 DATA 142,85,124,255
2230 REM 0-9
2240 DATA 146,75,107,99,117,126,255
2250 DATA 109,114,113,77,77,255
2260 DATA 170,118,104,126,255
2270 DATA 185,171,124,105,126,255
2280 DATA 157,102,102,107,126,255
2290 DATA 93,99,85,124,79,126,255
2300 DATA 159,75,89,95,126,255
2310 DATA 159,66,79,64,77,126,255
2320 DATA 70,97,162,234,234,255
2330 DATA 141,85,73,105,77,126,255
2340 REM TEN - NINETEEN
2350 DATA 170,128,66,77,255
2360 DATA 99,152,66,79,64,77,255
2370 DATA 106,109,66,64,99,88,79,255
2380 DATA 185,122,107,106,124,124,77,255
2390 DATA 93,102,107,106,124,124,77
2400 DATA 93,75,157,170,124,124,77,255
2410 DATA 95,75,153,159,170,124,124,77,255
2420 DATA 159,66,79,64,141,170,124,124,77,255
2430 DATA 70,97,162,170,124,124,77,255
2440 DATA 77,85,64,105,77,170,124,124,77,255
2450 REM TWENTY - NINETY
2460 DATA 106,109,66,64,77,106,105,255
2470 DATA 121,122,107,94,105,255
2480 DATA 157,117,107,68,105,255
2490 DATA 157,139,93,106,105,255
2500 DATA 159,75,89,95,106,105,255
2510 DATA 159,66,79,64,77,106,105,255
2520 DATA 134,97,98,106,105,255
2530 DATA 77,85,64,105,77,106,105,255
2540 REM HUNDRED THOUSAND MILLION
2550 DATA 219,114,77,94,107,65,94,255
2560 DATA 121,85,99,119,82,64,77,94,255
2570 DATA 76,139,88,98,99,77,255
2580 IF X9=0 THEN RESTORE 2240:GOTO 3190
2590 IF X9<1000000 THEN 2660
```

Revisionsberättelse
för
Privat-dataklubben PD68

Undertecknad som utsetts att granska klubbens räkenskaper under räkenskapsåret 1981 får härmed avgiva följande berättelse:

Jag har grankat räkenskaper och andra handlingar som lämnar upplysningar om klubbens ekonomi samt granskat års-redovisningen.

Jag har därvid inte funnit någon anledning till anmärkning beträffande redovisningshandlingar bokföring eller förvaltning , och tillstyrker därför att mötet fastställer balans-räkningen för 1981 och beviljar styrelsen full ansvarsfrihet för det gågna verksamhets-året.

Stockholm den 25 feb. 1982

Lennart Uusitalo
Revisor

V. Gustavsson
revisor

```
2600 X8=X9
2610 X9=INT(X9/1000000)
2620 GOSUB 2660
2630 RESTORE 2570
2640 GOSUB 1650
2650 X9=X8-INT(X8/1000000)*1000000
2660 IF X9<1000 THEN 2730
2670 X7=X9
2680 X9=INT(X9/1000)
2690 GOSUB 2730
2700 RESTORE 2560
2710 GOSUB 1650
2720 X9=X7-INT(X7/1000)*1000
2730 IF X9<100 THEN 2800
2740 X6=X9
2750 X9=INT(X9/100)
2760 GOSUB 3090
2770 RESTORE 2550
2780 GOSUB 1650
2790 X9=X6-INT(X6/100)*100
2800 IF X9<10 THEN 3090
2810 IF X9<20 THEN 2960
2820 X5=X9
2830 X9=INT(X9/10)
2840 ON X9 GOTO 2840,2850,2860,2870,2880,2890,2900,2910,2920
2850 RESTORE 2460:GOTO 2930
2860 RESTORE 2470:GOTO 2930
2870 RESTORE 2480:GOTO 2930
2880 RESTORE 2490:GOTO 2930
2890 RESTORE 2500:GOTO 2930
2900 RESTORE 2510:GOTO 2930
2910 RESTORE 2520:GOTO 2930
2920 RESTORE 2530
2930 GOSUB 1650
2940 X9=X5-INT(X5/10)*10
2950 GOTO 3090
2960 ON X9-9 GOTO 2970,2980,2990,3000,3010,3020,3030,3040,3050,3060
2970 RESTORE 2350:GOTO 3070
2980 RESTORE 2360:GOTO 3070
2990 RESTORE 2370:GOTO 3070
3000 RESTORE 2380:GOTO 3070
3010 RESTORE 2390:GOTO 3070
3020 RESTORE 2400:GOTO 3070
3030 RESTORE 2410:GOTO 3070
3040 RESTORE 2420:GOTO 3070
3050 RESTORE 2430:GOTO 3070
3060 RESTORE 2440
3070 GOSUB 1650
3080 GOTO 3200
3090 ON X9+1 GOTO 3200,3100,3110,3120,3130,3140,3150,3160,3170,3180
3100 RESTORE 2250:GOTO 3190
3110 RESTORE 2260:GOTO 3190
3120 RESTORE 2270:GOTO 3190
3130 RESTORE 2280:GOTO 3190
3140 RESTORE 2290:GOTO 3190
3150 RESTORE 2300:GOTO 3190
3160 RESTORE 2310:GOTO 3190
3170 RESTORE 2320:GOTO 3190
3180 RESTORE 2330
3190 GOSUB 1650
3200 RETURN
3210 IF ERR<>30 THEN ON ERROR GOTO
3220 PRINT CHR$(7):PRINT"TYPE ONLY NUMBERS! TRY AGAIN!"
3230 RESUME
```

VOXMEM.CMD:

```
                            ***********************************************
                            * FLEX 9 UTILITY                              *
                            *                                             *
                            * VOXMEM -- ETT PROGRAM SOM TALAR.            *
                            * DET HÄR PROGRAMMET UNDERSÖKER MINNET,       *
                            * OCH LÄSER UT BÅDE ADRESS OCH ÄVEN           *
                            * INNEHÅLLET I VARJE MINNESCELL.              *
                            *                                             *
                            * Vn 1.6 av Å. Stenäns 82-01-30.              *
                            * Några av rutinerna är från andra            *
                            * talprogram.                                 *
                            *                                             *
                            * Det här programmet är skrivet för 'VS-1     *
                            * SPEAKER BOARD' från ALFORD och ASSOCI-      *
                            * ATES, U.S.A.                                *
                            *                                             *
                            *  COMMANDS:                                  *
                            *  , - OUTPUT NEXT ADDRESS AND BYTE           *
                            *  . - OUTPUT NEXT BYTE                       *
                            *  / - REPEAT ADDRESS AND BYTE                *
                            *  N - NEW ADDRESS                            *
                            *  C - CONTINUOUS  BYTE-INFO, CR=STOP         *
                            *  ? - MENU + NEW STARTADDRESS                *
                            *  E - EXIT TO FLEX                           *
                            ***********************************************
                                   OPT     PAG

                            * I/O PORT ADDRESS
                  E000      PIA    EQU     $E000     VS-1 SPEAKER PORT
                  E004      ACIA   EQU     $E004     TERMINAL PORT

                            * EQUATES
                  CD03      WARMS  EQU     $CD03
                  CD15      GETCHR EQU     $CD15
                  CD18      PUTCHR EQU     $CD18
                  CD24      PCRLF  EQU     $CD24
                  CD1E      PSTRNG EQU     $CD1E

                            * PROGRAM START

C100                               ORG     $C100

C100 8E   E000              START  LDX     #PIA      POINT TO THE SPEAKER PORT
C103 6F   01                       CLR     1,X       RESET DIRECTION CONTROL REGS
C105 6F   03                       CLR     3,X
C107 86   FF                       LDA     #$FF
C109 A7   84                       STA     0,X       SET BOTH SIDES FOR OUTPUTS
C10B A7   02                       STA     2,X
C10D 86   3E                       LDA     #$3E      NOW SET A-SIDE CONTROL
C10F A7   01                       STA     1,X
C111 86   06                       LDA     #$06      AND B-SIDE COANTROL
C113 A7   03                       STA     3,X
C115 86   3E                       LDA     #$3E      SEND STOP CODE TO THE SPEAKER
C117 A7   84                       STA     0,X
C119 86   36                       LDA     #$36
C11B A7   01                       STA     1,X       STROBE
C11D 4A                     INIT1  DECA              INSURE GOOD STROBE DELAY
C11E 26   FD    C11D               BNE     INIT1     BY LOOPING OUT
C120 86   3E                       LDA     #$3E      THEN UNSTROBE
C122 A7   01                       STA     1,X
C124 86   83                       LDA     #$83      SET MID-FINE INFL/OUTS OFF
C126 A7   02                       STA     2,X
```

```
C128 A6   02                  LDA    2,X        CLEAR PENDING OPTO-INPUTS

C12A 30   8D 01D7     START2  LEAX   MSG1,PCR
C12E BD   CD1E                JSR    PSTRNG     PRINT MENU
C131 30   8D 02CF     START3  LEAX   MSG2,PCR
C135 BD   CD1E                JSR    PSTRNG     PRINT "STARTADDRESS"
C138 17   011B C256           LBSR   BADDR      GET STARTING ADDRESS

                      * RE-ENTRY IF ADDRESS IS DESIRED

C13B 1F   20          LOOP1   TFR    Y,D        MSB INTO A
C13D 17   00BD C1FD           LBSR   SAY2H      SAY LEFT 2 ADDR DIG
C140 1F   20                  TFR    Y,D        NEXT ADDR AGAIN
C142 1F   98                  TFR    B,A        LSB IN A
C144 17   00B6 C1FD           LBSR   SAY2H      SAY RIGHT 2 ADDR DIG
C147 8E   B000                LDX    #$B000
C14A 30   1F          WAIT1   LEAX   -1,X
C14C 26   FC   C14A           BNE    WAIT1

                      * RE-ENTRY ON CONTENTS ONLY

C14E BD   CD24        LOOP2   JSR    PCRLF
C151 1F   20                  TFR    Y,D        MSB INTO A
C153 17   00E9 C23F           LBSR   OUT2H      PRINT ADDRESS
C156 1F   20                  TFR    Y,D
C158 1F   98                  TFR    B,A        LSB INTO A
C15A 17   00E2 C23F           LBSR   OUT2H      PRINT LSB
C15D 86   20                  LDA    #$20       SPACE
C15F BD   CD18                JSR    PUTCHR
C162 A6   A4                  LDA    0,Y        GET CONTENTS
C164 17   00D8 C23F           LBSR   OUT2H      PRINT IT
C167 86   20                  LDA    #$20       SPACE
C169 BD   CD18                JSR    PUTCHR
C16C A6   A4                  LDA    0,Y        GET CONTENTS AGAIN
C16E 17   008C C1FD           LBSR   SAY2H      SAY IT
C171 31   21                  LEAY   1,Y        INCREMENT ADDRESS
C173 30   8D 02A2             LEAX   MSG3,PCR
C177 8D   28   C1A1           BSR    PSTRN
C179 BD   CD15                JSR    GETCHR     GET NEXT COMMAND
C17C 81   2C                  CMPA   #',        IS IT COMMA?
C17E 27   BB   C13B           BEQ    LOOP1      ADDRESS IF YES
C180 81   2E                  CMPA   #'.        IS IT PERIOD?
C182 27   CA   C14E           BEQ    LOOP2      DATA ONLY IF YES
C184 81   2F                  CMPA   #'/        IS IT SLASH?
C186 27   15   C19D           BEQ    BACKUP     YES
C188 81   4E                  CMPA   #'N        IS IT N?
C18A 27   A5   C131           BEQ    START3     GET NEW ADDRESS
C18C 81   3F                  CMPA   #'?        IS IT ?
C18E 27   9A   C12A           BEQ    START2     GET MENU AND NEW ADDRESS
C190 81   43                  CMPA   #'C        IS IT C?
C192 27   19   C1AD           BEQ    CONT       GET CONTINUOS BYTE
C194 81   45                  CMPA   #'E        IS IT E?
C196 27   02   C19A           BEQ    EXIT
C198 20   90   C12A           BRA    START2
C19A 7E   CD03        EXIT    JMP    WARMS
C19D 31   3F          BACKUP  LEAY   -1,Y       REPEAT
C19F 20   9A   C13B           BRA    LOOP1

                      * PRINT WITHOUT CR/LF

C1A1 A6   80          PSTRN   LDA    ,X+
C1A3 81   04                  CMPA   #4
C1A5 26   01   C1A8           BNE    PSTRN1
C1A7 39                       RTS
C1A8 BD   CD18        PSTRN1  JSR    PUTCHR
```

```
C1AB 20   F4   C1A1          BRA    PSTRN

                       * CONTINUOUS BYTES

C1AD BD   CD24        CONT   JSR    PCRLF
C1B0 1F   20                 TFR    Y,D       MSB INTO A
C1B2 17   008A C23F          LBSR   OUT2H     PRINT ADDRESS
C1B5 1F   20                 TFR    Y,D
C1B7 1F   98                 TFR    B,A       LSB INTO A
C1B9 17   0083 C23F          LBSR   OUT2H     PRINT LSB
C1BC 86   20                 LDA    #$20      SPACE
C1BE BD   CD18               JSR    PUTCHR
C1C1 A6   A4                 LDA    ,Y        GET CONTENTS
C1C3 8D   7A   C23F          BSR    OUT2H     PRINT IT
C1C5 86   20                 LDA    #$20
C1C7 BD   CD18               JSR    PUTCHR
C1CA A6   A4                 LDA    ,Y        GET CONTENTS AGAIN
C1CC 8D   2F   C1FD          BSR    SAY2H     SAY IT
C1CE 31   21                 LEAY   1,Y
C1D0 8E   B000               LDX    #$B000    DELAY
C1D3 30   1F          WAIT   LEAX   -1,X
C1D5 26   FC   C1D3          BNE    WAIT
C1D7 B6   E004               LDA    ACIA      CHECK IF ANY KEY'S BEEN HIT
C1DA 44                      LSRA
C1DB 24   D0   C1AD          BCC    CONT      NO, SAY MORE BYTE
C1DD B6   E005               LDA    ACIA+1    A KEY HAS BEEN HIT
C1E0 84   7F                 ANDA   #$7F      MASK PARITY
C1E2 81   0D                 CMPA   #$0D      IS IT CR?
C1E4 27   02   C1E8          BEQ    CONT2     YES, GO TO STOP
C1E6 20   C5   C1AD          BRA    CONT      SAY MORE BYTE
C1E8 30   8D 0239     CONT2  LEAX   MSG4,PCR
C1EC 8D   B3   C1A1          BSR    PSTRN
C1EE BD   CD15        GET    JSR    GETCHR    GET CHAR
C1F1 81   4E                 CMPA   #'N       IS IT N?
C1F3 1027 FF3A C131          LBEQ   START3    GET NEW ADDRESS
C1F7 81   45                 CMPA   #'E       IS IT E?
C1F9 27   9F   C19A          BEQ    EXIT      TO FLEX
C1FB 20   B0   C1AD          BRA    CONT

                       * SAY 2 HEX ROUTINE

C1FD 34   22          SAY2H  PSHS   A,Y       SAVE BYTE
C1FF 44                      LSRA
C200 44                      LSRA
C201 44                      LSRA             MOVE MSB RIGHT
C202 44                      LSRA
C203 8D   09   C20E          BSR    SAY1H     SAY LEFT DIGIT
C205 35   02                 PULS   A
C207 84   0F                 ANDA   #$0F      MASK RIGHT
C209 8D   03   C20E          BSR    SAY1H     SAY RIGHT DIGIT
C20B 35   20                 PULS   Y
C20D 39                      RTS

                       * SAY 1 HEX ROUTINE

C20E 48               SAY1H  ASLA             MULTIPLY BY 2
C20F 31   8D 0073            LEAY   WTABLE-HERE1,PC POINT TO WORD TABLE
C213 31   A6          HERE1  LEAY   A,Y       POINT TO WORD POINTER
C215 EC   A4                 LDD    0,Y       GET POINTER
C217 31   8D FEE5            LEAY   START-HERE2,PC POINT TO PGM START
C21B 31   AB          HERE2  LEAY   D,Y       POINT TO WORD

C21D A6   A0          OUT    LDA    0,Y+
C21F 81   FF                 CMPA   #$FF
C221 26   01   C224          BNE    OUT1
```

```
C223 39                         RTS
C224 8D   02   C228   OUT1      BSR    LOOP
C226 20   F5   C21D             BRA    OUT

                      * SEND PHONEME

C228 8E   E000        LOOP      LDX    #PIA
C22B 6D   01                    TST    1,X       CHECK FOR PORT READY
C22D 2A   F9   C228             BPL    LOOP      LOOP IF NOT
C22F 6D   84                    TST    0,X       ELSE CLEAR FLAG
C231 A7   84                    STA    0,X       SEND PHONEME
C233 86   36                    LDA    #$36      STROBE THE DATA
C235 A7   01                    STA    1,X
C237 4A               LOOPS     DECA             INSURE POSITIVE STROBE
C238 26   FD   C237             BNE    LOOPS
C23A 86   3E                    LDA    #$3E
C23C A7   01                    STA    1,X       UNSTROBE THE DATA
C23E 39                         RTS

                      * PRINT 2 HEX ROUTINE

C23F 34   02          OUT2H     PSHS   A         SAVE BYTE
C241 44                         LSRA
C242 44                         LSRA
C243 44                         LSRA             MOVE MSB RIGHT
C244 44                         LSRA
C245 8D   02   C249             BSR    OUT1H
C247 35   02                    PULS   A

C249 84   0F          OUT1H     ANDA   #$0F      MASK RIGHT
C24B 8B   30                    ADDA   #'0
C24D 81   39                    CMPA   #'9
C24F 23   02   C253             BLS    OUTEX     OUTPUT IF <10
C251 8B   07                    ADDA   #$7       CONVERT FOR A-F
C253 7E   CD18        OUTEX     JMP    PUTCHR

                      * BUILD ADDRESS ROUTINE

C256 8D   0B   C263   BADDR     BSR    BYTE      GET LS BYTE
C258 34   02                    PSHS   A
C25A 8D   07   C263             BSR    BYTE      GET LS BYTE
C25C 1F   89                    TFR    A,B
C25E 35   02                    PULS   A         COMBINE
C260 1F   02                    TFR    D,Y       MOVE INTO Y
C262 39                         RTS
                      * INPUT ONE BYTE
C263 8D   0B   C270   BYTE      BSR    INHEX     GET MS DIGIT
C265 48                         ASLA
C266 48                         ASLA
C267 48                         ASLA
C268 48                         ASLA
C269 34   02                    PSHS   A         SAVE DIGIT
C26B 8D   03   C270             BSR    INHEX     GET LS DIGIT
C26D AB   E0                    ADDA   0,S+      ADD B TO A
C26F 39                         RTS
                      * INPUT ONE DIGIT
C270 BD   CD15        INHEX     JSR    GETCHR    GET DIGIT
C273 80   30                    SUBA   #$30
C275 2B   F9   C270             BMI    INHEX     REPEAT IF NG
C277 81   09                    CMPA   #$9
C279 2F   0A   C285             BLE    IHEXIT
C27B 81   11                    CMPA   #$11
C27D 2B   F1   C270             BMI    INHEX     REPEAT IF NG
C27F 81   16                    CMPA   #$16
C281 2E   ED   C270             BGT    INHEX     REPEAT IF NG
```

```
C283 80   07                   SUBA  #7       CONVERT IF OK
C285 39               IHEXIT   RTS

                      * WORD STARTING ADDRESS TABLE

C286 01A6             WTABLE   FDB   ZEREO-START
C288 01AD                      FDB   ONE-START
C28A 01B3                      FDB   TWO-START
C28C 01B8                      FDB   THREE-START
C28E 01BE                      FDB   FOUR-START
C290 01C4                      FDB   FIVE-START
C292 01CB                      FDB   SIX-START
C294 01D1                      FDB   SEVEN-START
C296 01D8                      FDB   EIGHT-START
C298 01DF                      FDB   NINE-START
C29A 01E6                      FDB   A-START
C29C 01EB                      FDB   B-START
C29E 01F0                      FDB   C-START
C2A0 01F5                      FDB   D-START
C2A2 01FC                      FDB   E-START
C2A4 0200                      FDB   F-START

                      * WORD TABLE (PHONEME)
C2A6 92 4B 6B 63      ZEREO    FCB   $92,$4B,$6B,$63,$75,$7E,$FF
C2AD AD 73 4D 4D      ONE      FCB   $AD,$73,$4D,$4D,$7E,$FF
C2B3 AA 76 68 7E      TWO      FCB   $AA,$76,$68,$7E,$FF
C2B8 B9 AB 7C 69      THREE    FCB   $B9,$AB,$7C,$69,$7E,$FF
C2BE 9D 66 66 6B      FOUR     FCB   $9D,$66,$66,$6B,$7E,$FF
C2C4 5D 63 55 7C      FIVE     FCB   $5D,$63,$55,$7C,$4F,$7E,$FF
C2CB 9F 4B 59 5F      SIX      FCB   $9F,$4B,$59,$5F,$7E,$FF
C2D1 9F 42 4F 40      SEVEN    FCB   $9F,$42,$4F,$40,$4D,$7E,$FF
C2D8 46 61 A2 EA      EIGHT    FCB   $46,$61,$A2,$EA,$EA,$7E,$FF
C2DF 8D 55 49 69      NINE     FCB   $8D,$55,$49,$69,$4D,$7E,$FF
C2E6 80 46 61 7E      A        FCB   $80,$46,$61,$7E,$FF
C2EB 8E 7C 7C 7E      B        FCB   $8E,$7C,$7C,$7E,$FF
C2F0 9F 7C 7C 7E      C        FCB   $9F,$7C,$7C,$7E,$FF
C2F5 DE DE DE 7C      D        FCB   $DE,$DE,$DE,$7C,$7C,$7E,$FF
C2FC 6C 6C 7E FF      E        FCB   $6C,$6C,$7E,$FF
C300 46 5D 5D 7E      F        FCB   $46,$5D,$5D,$7E,$FF

                      * MESSAGE STRINGS
C305 0D 0A 2A 2A      MSG1     FCC   $D,$A,'*** VOXMEM  Vn 1.6 by Å. Stenäns 1982 ***',$D,$A
C332 0D 0A 43 4F               FCC   $D,$A,'COMMANDS:',$D,$A
C33F 20 20 2C 20               FCC   '  , - OUTPUT NEXT ADDRESS AND BYTE',$D,$A
C363 20 20 2E 20               FCC   '  . - OUTPUT NEXT BYTE',$D,$A
C37B 20 20 2F 20               FCC   '  / - REPEAT ADDRESS AND BYTE',$D,$A
C39A 20 20 4E 20               FCC   '  N - NEW ADDRESS',$D,$A
C3AD 20 20 43 20               FCC   '  C - CONTINUOUS BYTE-INFO, CR=STOP',$D,$A
C3D2 20 20 3F 20               FCC   '  ? - MENU + NEW STARTADDRESS',$D,$A
C3F1 20 20 45 20               FCC   '  E - EXIT TO FLEX',4
C404 0D 0A 0A 53      MSG2     FCC   $D,$A,$A,'STARTADDRESS ?',$D,$A
C417 3E 04                     FCC   '>',4
C419 20 43 4F 4D      MSG3     FCC   ' COMMAND ? ',4
C425 20 28 43 52      MSG4     FCC   ' (CR)=STOP-CONTINUE, (N)EW ADDRESS, (E)XIT TO FLEX ? ',4

                               END   START
0 ERROR(S) DETECTED
```

SäLJES :

ett antal kort fÖr SWTPC (CPU,MINNES,DISK,MODER mm)
2 st SHUGART SA-400

KöPES :

MPI-52 eller MPI-92

Avo Kask HÖdervÄgen 8 184 02 österskÄr
Tfn 0764/235 28

OM PATCHNINGAR

I artiklar som använder sig av patchningar till olika

befintliga program på systemnivå, saknar jag en mycket enkel uppgift : vilket VERSIONSnummer har programmet i fråga. Det torde ju vara ganska enkelt att trycka på tangenterna VER eller VERSION för att få fram vilken version det är frågan om. Det gör ju alla patchningar mycket enklare för oss andra. Alla vet väl att i fråga om FLEX.SYS så existerar för 6809 åtminstone 5 (fem) versioner : 2.4 2.5 2.6.5 2.7.3 samt 2.8.1. Alla kan väl inse att med denna flora så tar det lite tid att leta upp de korrekta adresserna.

Alltså alla ni, som refererar till något befintligt program : ange versionsnumret på programmet i fråga.

OM INNEHÅLLSFÖRTECKNING

Vad jag och flera andra saknar är en komplett innehålls-förteckning över samtliga tidigare tryckta artiklar i MPU-laren. Redaktörerna torde väl ha ganska stora möjligheter att till nästa nummer ge en förteckning över dessa, vilket ju underlättar för oss alla andra att söka efter en intressant artikel.

Detta anser jag vara en självklar service till medlemmarna.

——— : ———

Det tycker RED också avo men det hänger inte bara på redaktionen utan möjligen styrelsen el. medlemmarna. Vi är kraftigt under-bemannade och vill faktiskt (hur otroligt det än låter) ha lite fritid också. Tyvär är intresset för redaktionellt arbete rätt klent om inte uselt med tanke på att vi har uselt med material. Gammalt material ska vi på red försöa samla upp i ett arkiv för ev. utgivning utav god-bitarna till medlemmarna.

:

OM REAL-TIME-KLOCKOR

Det finns ju ett antal SWTPC-ägare också förutom RT-dator-Ägare i klubben som kanske skulle vara intresserade att koppla in en real-time-klocka till sitt system.

Åke Stenäng hade i nr 4/81 en dylik. Här kommer en annan, baserad likaså på OKI-kretsen, men av fabrikat COMPUWARE, med programvaran SETCLOCK samt FLEXPAT, dvs en patch till FLEX.SYS (ver. 2.6.5) som själv lägger in datumet till FLEX.SYS. Av programmet framgår att det finns ett antal olika klockrutiner att tillgå.

OM LOTTO

I en tidigare artikel (nr 3/81) gav jag en listning på ett 39-raders lotto-system med checkning. Efter påstötningar kommer här ett mer utökat reducerat system på 80 rader, där man väljer ut 20 nummer. Programmet ges också med checkning.

ÖVRIGT

Jag kan säga att för 6809-ägare finns ett superformidabelt och fantastiskt snabbt databas-register-program skrivet av Esko Antikainen, SA-TEKNIK. Ett i mitt tycke ett absolut förstklassigt och snabbt program.

Likaså för sändaramatörer finns ett eminent RTTY/ASCII/CW program med diskoperationer bl.a., allt skrivet i assembler av denne eminente Esko.

Avo Kask
Hödervägen 8
184 02 österskär

```
                      * NEW VERSION OF FLEX PATCH. ADDS NEW ROUTINES
                      * TO ALLOW ANY PROGRAM TO PRINT TIME OR READ
                      * ANY CLOCK DATA. SITS ATOP OF THE USER AND
                      * PROTECTS ITSELF

                      * ENDAST FÖR FLEX 265 |||||||||||



                      * FLEX EQUATES *

                CC10  SYSYR   EQU    $CC10
                CCOF  SYSDAY  EQU    $CCOF
                CCOE  SYSMTH  EQU    $CCOE
                CD18  PUTCHR  EQU    $CD18
                CD1E  PSTRNG  EQU    $CD1E
                CD24  PCRLF   EQU    $CD24
                CE83  PTEXT   EQU    $CE83      *** FÖR FLEX24: $CE84
                CC2B  MEMEND  EQU    $CC2B

                      * CLOCK BOARD ADDRESS FOR PORT £4 *

                EO10  CLOCK   EQU    $EO10

                      * CURRENT TOP OF MEMORY       ADJUST IF NECESSARY *

                BFFF  TOP     EQU    $BFFF

                      * INDIRECT JUMP TABLE FOR CLOCK FUNCTIONS *
                      * PUT TABLE AT TOP OF UTILITY SPACE *

BFEC                          ORG    $BFEC

BFEC BEB2             CLOCK1  FDB    DATEH      PRINT DATE WITH LABEL
BFEE BEB9             CLOCK2  FDB    DATEN      PRINT DATE AS MM/DD/YY
BFFO BEF1             CLOCK3  FDB    DATEA      PRINT DATE AS DD-MMM-YY
BFF2 BFAO             CLOCK4  FDB    DATA       BLOCK OF 13 BYTES AS CLOCK DATA
BFF4 BF58             CLOCK5  FDB    GETTIM     GET NEW TIME TO BUFFER
BFF6 BF2E             CLOCK6  FDB    TIMEH      PRINT TIME WITH LABEL
BFF8 BF35             CLOCK7  FDB    TIMEN      PRINT PLAIN TIME AS XX:XX:XX
BFFA BF37             CLOCK8  FDB    LSTTIM     PRINT LAST CALLED TIME PLAIN
BFFC BEO1             CLOCK9  FDB    DOW        PRINTS DAY OF WEEK
BFFE BE5D             CLOCKA  FDB    GETDAT     GETS DATA FROM STT TO STP

                      * PATCH TO JUMP TO NEW STARTUP *

CA5E                          ORG    $CA5E
                      * FÖR FLEX24: $CA5O


CA5E 7E   BDAO        PATCH   JMP    START1
CA61 00 00 00 00      FILL    FCB    0,0,0,0,0,0
CA67 8D   5B          RETURN  BSR    $CAC4      RETUR TILL FLEX

BDAO                          ORG    $BDAO

BDAO CC   BD9F        START1  LDD    £START1-1
BDA3 FD   CC2B                STD    MEMEND
BDA6 4F               START2  CLRA              SETUP PIA
BDA7 B7   EO11                STA    CLOCK+1
```

```
BDAA B7    E013             STA    CLOCK+3
BDAD B7    E010             STA    CLOCK
BDB0 43                     COMA
BDB1 B7    E012             STA    CLOCK+2
BDB4 86    3E               LDA    £$3E
BDB6 B7    E011             STA    CLOCK+1
BDB9 B7    E013             STA    CLOCK+3
BDBC F6    E012             LDB    CLOCK+2
BDBF B1    E013             CMPA   CLOCK+3
BDC2 27    0C               BEQ    BRDOK      BOARD OK THEN GET DATA
BDC4 8E    CADE             LDX    £$CADE     FÖR FLEX24: $CAD0 START

                * THIS IS STANDARD START USED IF BOARD NOT OK

BDC7 BD    CE81             JSR    $CE81      ** FÖR FLEX24: $CE82
BDCA BD    CE2B             JSR    $CE2B      ** FÖR FLEX24: $CE2C
BDCD 7E    CA67             JMP    $CA67
BDD0 86    A0       BRDOK   LDA    £$A0       SET UP TO GET ALL DATA
BDD2 A7    8D 01E4          STA    STT,PCR
BDD6 86    AC               LDA    £$AC
BDD8 A7    8D 01DF          STA    STP,PCR
BDDC AD    9F BFFE          JSR    ÄCLOCKAÅ   GETDATA
BDE0 17    00AC             LBSR   DT2FLX     MOVE DATE TO FLEX
BDE3 30    8D 019E          LEAX   INTRO,PCR
BDE7 BD    CE83             JSR    PTEXT
BDEA AD    9F BFFC          JSR    ÄCLOCK9Å   PRINT CURRENT DAY OF WEEK
BDEE AD    9F BFEE          JSR    ÄCLOCK2Å   PRINT DATE
BDF2 86    20               LDA    £$20       SPACE
BDF4 BD    CD18             JSR    PUTCHR
BDF7 BD    CD18             JSR    PUTCHR
BDFA AD    9F BFF8          JSR    ÄCLOCK7Å   PRINT TIME
BDFE 7E    CA7F             JMP    $CA7F
BE01 34    16       DOW     PSHS   A,B,X      PRINT PLAIN DAY OF WEEK
BE03 30    8D 0199          LEAX   DATA,PCR
BE07 E6    06               LDB    6,X
BE09 86    0A               LDA    £10
BE0B 3D                     MUL
BE0C 30    8D 0007          LEAX   WKDYS,PCR
BE10 3A                     ABX
BE11 BD    CE83             JSR    PTEXT
BE14 35    16               PULS   A,B,X
BE16 39                     RTS

BE17 53 5C 6E 64    WKDYS   FCC    /SÖndag    /
BE20 04                     FCB    4
BE21 4D 5D 6E 64            FCC    /MÅndag    /
BE2A 04                     FCB    4
BE2B 54 69 73 64            FCC    /Tisdag    /
BE34 04                     FCB    4
BE35 4F 6E 73 64            FCC    /Onsdag    /
BE3E 04                     FCB    4
BE3F 54 6F 72 73            FCC    /Torsdag   /
BE48 04                     FCB    4
BE49 46 72 65 64            FCC    /Fredag    /
BE52 04                     FCB    4
BE53 4C 5C 72 64            FCC    /LÖrdag    /
BE5C 04                     FCB    4

BE5D 34    36       GETDAT  PSHS   A,B,X,Y    GET DATA FROM STT TO STP
BE5F 17    0105             LBSR   HOLDIN
BE62 30    8D 013A          LEAX   DATA,PCR
BE66 31    8D 0143          LEAY   LIMIT,PCR
```

```
 BE6A A6   8D 014C           LDA    STT,PCR
 BE6E B7   E012      DATA1   STA    CLOCK+2
§BE71 1021 FFFC              LBRN   *
§BE75 1021 FFFC              LBRN   *
 BE79 F6   E010              LDB    CLOCK
 BE7C E4   A0                ANDB   0,Y+
 BE7E E7   80                STB    0,X+
 BE80 A1   8D 0137           CMPA   STP,PCR
 BE84 27   03                BEQ    DATA2
 BE86 4C                     INCA
 BE87 20   E5                BRA    DATA1
 BE89 17   00F0      DATA2   LBSR   HLDOUT
 BE8C 35   36                PULS   A,B,X,Y
 BE8E 39                     RTS

 BE8F 30   8D 0119   DT2FLX  LEAX   DATA+12,PCR MOVE DATA FROM BUFFER TO FLEX
 BE93 A6   84                LDA    0,X
 BE95 C6   0A                LDB    £10
 BE97 3D                     MUL
 BE98 EB   82                ADDB   0,-X
 BE9A F7   CC10              STB    SYSYR
 BE9D A6   82                LDA    0,-X
 BE9F C6   0A                LDB    £10
 BEA1 3D                     MUL
 BEA2 EB   82                ADDB   0,-X
 BEA4 F7   CC0E              STB    SYSMTH
 BEA7 A6   82                LDA    0,-X
 BEA9 C6   0A                LDB    £10
 BEAB 3D                     MUL
 BEAC EB   82                ADDB   0,-X
 BEAE F7   CC0F              STB    SYSDAY
 BEB1 39                     RTS

 BEB2 30   8D 00DC   DATEH   LEAX   DATES,PCR PRINTS DATE WITH LABEL
 BEB6 BD   CE83              JSR    PTEXT
 BEB9 30   8D 00E3   DATEN   LEAX   DATA,PCR  PRINTS PLAIN DATE
 BEBD A6   08                LDA    8,X
 BEBF 8B   30                ADDA   £$30
 BEC1 BD   CD18              JSR    PUTCHR
 BEC4 A6   07                LDA    7,X
 BEC6 8B   30                ADDA   £$30
 BEC8 BD   CD18              JSR    PUTCHR
 BECB 86   2D                LDA    £'-
 BECD BD   CD18              JSR    PUTCHR
 BED0 A6   0A                LDA    10,X
 BED2 8B   30                ADDA   £$30
 BED4 BD   CD18              JSR    PUTCHR
 BED7 A6   09                LDA    9,X
 BED9 8B   30                ADDA   £$30
 BEDB BD   CD18              JSR    PUTCHR
 BEDE 86   2D                LDA    £'-
 BEE0 BD   CD18              JSR    PUTCHR
 BEE3 A6   0C                LDA    12,X
 BEE5 8B   30                ADDA   £$30
 BEE7 BD   CD18              JSR    PUTCHR
 BEEA A6   0B                LDA    11,X
 BEEC 8B   30                ADDA   £$30
 BEEE 7E   CD18              JMP    PUTCHR
 BEF1 30   8D 00AB   DATEA   LEAX   DATA,PCR  RINT ALPHA MONTH IN DATE
 BEF5 A6   08                LDA    8,X
 BEF7 8B   30                ADDA   £$30
 BEF9 BD   CD18              JSR    PUTCHR
```

```
BEFC A6   07                LDA   7,X
BEFE 8B   30                ADDA  £$30
BF00 BD   CD18              JSR   PUTCHR
BF03 86   2D                LDA   £'-
BF05 BD   CD18              JSR   PUTCHR
BF08 34   16                PSHS  A,B,X
BF0A 30   8D 00AE           LEAX  MONTHS,PCR
BF0E F6   CC0E              LDB   SYSMTH
BF11 5A                     DECB
BF12 86   04                LDA   £4
BF14 3D                     MUL
BF15 3A                     ABX
BF16 BD   CE83              JSR   PTEXT
BF19 86   2D                LDA   £'-
BF1B BD   CD18              JSR   PUTCHR
BF1E 35   16                PULS  A,B,X
BF20 A6   0C                LDA   12,X
BF22 8B   30                ADDA  £$30
BF24 BD   CD18              JSR   PUTCHR
BF27 A6   0B                LDA   11,X
BF29 8B   30                ADDA  £$30
BF2B 7E   CD18              JMP   PUTCHR
BF2E 30   8D 0067  TIMEH    LEAX  TIMES,PCR PRINT TIME WITH LABEL
BF32 BD   CE83              JSR   PTEXT
BF35 8D   21       TIMEN    BSR   GETTIM
BF37 30   8D 006A  LSTTIM   LEAX  DATA+5,PCR PRINT TIME IN BUFFER
BF3B C6   03                LDB   £3
BF3D A6   84       TIME1    LDA   0,X
BF3F 8B   30                ADDA  £$30
BF41 BD   CD18              JSR   PUTCHR
BF44 A6   82                LDA   0,-X
BF46 8B   30                ADDA  £$30
BF48 BD   CD18              JSR   PUTCHR
BF4B 30   1F                LEAX  -1,X
BF4D 5A                     DECB
BF4E 27   07                BEQ   TIME2
BF50 86   3A                LDA   £':
BF52 BD   CD18              JSR   PUTCHR
BF55 8D   E6                BSR   TIME1
BF57 39            TIME2    RTS

BF58 86   A0       GETTIM   LDA   £$A0
BF5A A7   8D 005C           STA   STT,PCR
BF5E 86   A5                LDA   £$A5
BF60 A7   8D 0057           STA   STP,PCR
BF64 16   FEF6              LBRA  GETDAT
BF67 34   02       HOLDIN   PSHS  A           HOLD FOR READ
BF69 86   80                LDA   £$80
BF6B B7   E012              STA   CLOCK+2
BF6E 86   23                LDA   £$23
§BF70 1021 FFFC    DLY1     LBRN  DLY1
BF74 4A                     DECA
BF75 27   02                BEQ   DLY2
BF77 20   F7                BRA   DLY1
BF79 35   02       DLY2     PULS  A
BF7B 39                     RTS

BF7C 34   02       HLDOUT   PSHS  A
BF7E 4F                     CLRA
BF7F B7   E012              STA   CLOCK+2
BF82 35   02                PULS  A
BF84 39                     RTS
```

```
BF85 00 00 00     INTRO   FCB    0,0,0
BF88 4C 6F 67 6F          FCC    /Logon :  /
BF91 04                   FCB    4
BF92 44 61 74 75  DATES   FCC    /Datum /
BF98 04                   FCB    4
BF99 54 69 64 20  TIMES   FCC    /Tid   /
BF9F 04                   FCB    4
BFA0              DATA    RMB    13
BFAD 0F 0F 0F 07  LIMIT   FCB    $F,$F,$F,7,$F,3,7,$F,3,$F,3,$F,$F
BFBA              STT     RMB    1
BFBB              STP     RMB    1
BFBC 4A 61 6E     MONTHS  FCC    /Jan/
BFBF 04                   FCB    4
BFC0 46 65 62             FCC    /Feb/
BFC3 04                   FCB    4
BFC4 4D 61 72             FCC    /Mar/
BFC7 04                   FCB    4
BFC8 41 70 72             FCC    /Apr/
BFCB 04                   FCB    4
BFCC 4D 61 6A             FCC    /Maj/
BFCF 04                   FCB    4
BFD0 4A 75 6E             FCC    /Jun/
BFD3 04                   FCB    4
BFD4 4A 75 6C             FCC    /Jul/
BFD7 04                   FCB    4
BFD8 41 75 67             FCC    /Aug/
BFDB 04                   FCB    4
BFDC 53 65 70             FCC    /Sep/
BFDF 04                   FCB    4
BFE0 4F 6B 74             FCC    /Okt/
BFE3 04                   FCB    4
BFE4 4E 6F 76             FCC    /Nov/
BFE7 04                   FCB    4
BFE8 44 65 63             FCC    /Dec/
BFEB 04                   FCB    4

                          END    $C600      *** FÖR FLEX24: $CD00

0 ERROR(S) DETECTED
```

```
                  * CLOCKKORTET I PORT £4
                  * EFTER MICROCOMPUTING 7-81
                  *         * FLEX ROUTINES *

            CD1E  PSTRNG  EQU    $CD1E
            CD15  GETCHR  EQU    $CD15
            CD03  WARMS   EQU    $CD03

                  *         * EXTERNAL EQUATES *

            0001  VERSION EQU    1
            E010  CLOCK   EQU    $E010     PORT £4


C100                      ORG    $C100

                  *         * MAIN PROGRAM *

C100 8D   01      CALSET  BSR    TIME1
C102 01                   FCB    VERSION
C103 17   00B4    TIME1   LBSR   INIT
C106 24   0A              BCC    OK2       IF SET THEN NO BOARD
C108 30   8D 011C         LEAX   NOBD,PCR
C10C BD   CD1E            JSR    PSTRNG    REPORT THIS
C10F 7E   CD03            JMP    WARMS     EXIT
C112 30   8D 0100 OK2     LEAX   INTRO,PCR
C116 BD   CD1E            JSR    PSTRNG
C119 31   8D 01C3         LEAY   BLOCK,PCR LIST OF MAX FOR EACH DIGIT
C11D 30   8D 0120         LEAX   YEAR,PCR
C121 BD   CD1E            JSR    PSTRNG
C124 8D   70              BSR    GET2
C126 30   8D 0120         LEAX   MONTH,PCR
C12A BD   CD1E            JSR    PSTRNG
C12D 8D   67              BSR    GET2
C12F 30   8D 0120         LEAX   DAY,PCR
C133 BD   CD1E            JSR    PSTRNG
C136 8D   5E              BSR    GET2
C138 30   8D 0121         LEAX   DOW,PCR
C13C BD   CD1E            JSR    PSTRNG
C13F 8D   51              BSR    GET1
C141 30   8D 012A         LEAX   HOUR,PCR
C145 BD   CD1E            JSR    PSTRNG
C148 8D   4C              BSR    GET2
C14A 30   8D 012B         LEAX   MINUTE,PCR
C14E BD   CD1E            JSR    PSTRNG
C151 8D   43              BSR    GET2
C153 30   8D 012C         LEAX   LPYR,PCR
C157 BD   CD1E            JSR    PSTRNG
C15A BD   CD15            JSR    GETCHR
C15D 81   59              CMPA   £'Y
C15F 26   08              BNE    ON1       IF LEAPYEAR SET CLOCK BIT
C161 B6   C2E4            LDA    DAY10
C164 8A   04              ORA    £$4       BIT IS MONTH DATA
C166 B7   C2E4            STA    DAY10
C169 B6   C2E7    ON1     LDA    H10       24 HOUR FORMAT ONLY
C16C 8A   08              ORA    £$8       SO SET BIT IN HOURS
C16E B7   C2E7            STA    H10
C171 30   8D 0150         LEAX   ERROR1,PCR WANT TO DO AGAIN
C175 BD   CD1E            JSR    PSTRNG
C178 BD   CD15            JSR    GETCHR
C17B 81   59              CMPA   £'Y
```

```
C17D 27    84                  BEQ    TIME1
C17F 8D    5E                  BSR    SET         PUT DATA IN CLOCK
C181 30    8D 0109             LEAX   START,PCR
C185 BD    CD1E                JSR    PSTRNG
C188 BD    CD15                JSR    GETCHR      DONT RELEASE CLOCK UNTIL TOLD TO
C18B 4F                        CLRA
C18C B7    E012                STA    CLOCK+2     RELEASE HOLD
C18F 7E    CD03                JMP    WARMS

                    *           * SUBROUTINES *

C192 C6    01       GET1       LDB    £1
C194 20    02                  BRA    GET
C196 C6    02       GET2       LDB    £2
C198 BD    CD15     GET        JSR    GETCHR      GET TWO NUMBERS AND VERIFY LIMITS
C19B 81    30                  CMPA   £$30        THEN STORE IN DATA BLOCK
C19D 2D    12                  BLT    G2ERR
C19F 81    39                  CMPA   £$39
C1A1 2E    0E                  BGT    G2ERR
C1A3 80    30                  SUBA   £$30
C1A5 A1    2B                  CMPA   11,Y        COMPARE TO LIMIT
C1A7 2E    08                  BGT    G2ERR
C1A9 A7    A0                  STA    ,Y+
C1AB 5A                        DECB
C1AC 27    02                  BEQ    RTN1
C1AE 20    E8                  BRA    GET
C1B0 39             RTN1       RTS

C1B1 30    8D 00FE  G2ERR      LEAX   ERRORS,PCR
C1B5 BD    CD1E                JSR    PSTRNG
C1B8 20    DE                  BRA    GET

                    *           * SETUP PIA AND CHECK FOR BOARD *

C1BA 4F             INIT       CLRA
C1BB B7    E011                STA    CLOCK+1
C1BE B7    E013                STA    CLOCK+3
C1C1 43                        COMA
C1C2 B7    E010                STA    CLOCK
C1C5 B7    E012                STA    CLOCK+2
C1C8 86    3E                  LDA    £$3E
C1CA B7    E011                STA    CLOCK+1
C1CD B7    E013                STA    CLOCK+3
C1D0 F6    E012                LDB    CLOCK+2
C1D3 B1    E013                CMPA   CLOCK+3
C1D6 24    04                  BCC    OK
C1D8 1A    01       NFG        ORCC   £1
C1DA 20    02                  BRA    RTN
C1DC 1C    FE       OK         ANDCC  £$FE
C1DE 39             RTN        RTS

                    *           * SET DATA IN CLOCK/CALENDAR *

C1DF 86    80       SET        LDA    £$80
C1E1 B7    E012                STA    CLOCK+2
C1E4 8D    20                  BSR    DELAY
C1E6 C6    8C       ON         LDB    £$8C
C1E8 30    8D 00F4             LEAX   BLOCK,PCR
C1EC F7    E012     DLY2       STB    CLOCK+2
C1EF A6    80                  LDA    ,X+
C1F1 B7    E010                STA    CLOCK
C1F4 CA    10                  ORB    £$10
```

```
 C1F6 F7   E012                STB    CLOCK+2
 C1F9 C4   8F                  ANDB   £$8F
 C1FB F7   E012                STB    CLOCK+2
 C1FE C1   80                  CMPB   £$80
 C200 27   03                  BEQ    ON3
 C202 5A                       DECB
 C203 20   E7                  BRA    DLY2
 C205 39            ON3        RTS

 C206 34   02       DELAY      PSHS   A
 C208 86   13                  LDA    £$13
§C20A 1021 FFF8     DELAY1     LBRN   DELAY
 C20E 4A                       DECA
 C20F 27   02                  BEQ    OVER
 C211 20   F7                  BRA    DELAY1
 C213 35   02       OVER       PULS   A
 C215 39                       RTS

                    *           * MESSAGES *

 C216 45 4E 54 45   INTRO      FCC    /ENTER DATE & TIME/
 C227 04                       FCB    4
 C228 4E 4F 20 43   NOBD       FCC    /NO CLOCK BOARD INSTALLED/
 C240 04                       FCB    4
 C241 59 45 41 52   YEAR       FCC    /YEAR? 19/
 C249 04                       FCB    4
 C24A 4D 4F 4E 54   MONTH      FCC    /MONTH?  /
 C252 04                       FCB    4
 C253 44 41 59 3F   DAY        FCC    /DAY?     /
 C25C 04                       FCB    4
 C25D 44 41 59 53   DOW        FCC    /DAYS FROM SUNDAY?/
 C26E 04                       FCB    4
 C26F 48 4F 55 52   HOUR       FCC    /HOUR?    /
 C278 04                       FCB    4
 C279 4D 49 4E 55   MINUTE     FCC    /MINUTE?  /
 C282 04                       FCB    4
 C283 4C 45 41 50   LPYR       FCC    /LEAP YEAR?/
 C28D 04                       FCB    4
 C28E 50 52 45 53   START      FCC    /PRESS ANY KEY TO START CLOCK §§§§§§§/
 C2B2 04                       FCB    4
 C2B3 45 52 52 4F   ERRORS     FCC    /ERROR RE-ENTER!!!/
 C2C4 04                       FCB    4
 C2C5 57 41 4E 54   ERROR1     FCC    /WANT TO REDO FOR MISTAKES?/
 C2DF 04                       FCB    4

             C2E0   BLOCK      EQU    *

 C2E0                          ORG    BLOCK

 C2E0               YR10       RMB    1
 C2E1               YR1        RMB    1
 C2E2               MO10       RMB    1
 C2E3               MO1        RMB    1
 C2E4               DAY10      RMB    1
 C2E5               DAY1       RMB    1
 C2E6               DOWK       RMB    1          DAY OF WEEK NUMBER FROM SUNDAY
 C2E7               H10        RMB    1
 C2E8               H1         RMB    1
 C2E9               MI10       RMB    1
 C2EA               MI1        RMB    1

 C2EB 09 09 01 09   MAX        FCB    9,9,1,9,3,9,6,2,9,5,9

                               END    CALSET
```

```
100 REM REDUCERAT LOTTOSYSTEM TILL 80 RADER
120 REM 20 VALFRIA NUMMER PÅ 80 RADER
140 REM VID 7 RÄTTA NUMMER : 94.6 % CHANS TILL MINST 5 RÄTT
160 REM VID 6 : 57.8 % CHANS TILL MINST 5 RÄTT
180 REM VID 5 : 10.8 % CHANS TILL MINST 5 RÄTT
200 DIM S(80,20),L(35),U(20)
220 PRINT CHR$(7)
240 INPUT "VILL DU TIPPA NYTT SYSTEM ",S1$
260 IF LEFT$(S1$,1)="J" THEN 400
280 PRINT CHR$(7)
300 PRINT "MATA IN GAMLA UTGÅNGSRADEN (20 NR) "
320 PRINT
340 FOR I=1 TO 20
360 INPUT U(I) :NEXT I
380 GOTO 820
400 FOR I=1 TO 35
420 LET L(I)=I
440 NEXT I
460 FOR I=35 TO 1 STEP -1
480 LET B=INT(RND(O)*I+1)
500 LET Q=L(I)
520 LET L(I)=L(B)
540 LET L(B)=Q
560 NEXT I
580 FOR I=1 TO 20
600 LET U(I)=L(I)
620 NEXT I
640 PRINT
660 FOR I=1 TO 19
680 FOR J=I+1 TO 20
700 IF U(I)|U(J) THEN 780
720 LET Q=U(I)
740 LET U(I)=U(J)
760 LET U(J)=Q
780 NEXT J
800 NEXT I
820 PRINT CHR$(7)
840 INPUT "TERMINAL ELLER PRINTER",R$
860 IF LEFT$(R$,1)="P" THEN OPEN "O.PRINT" AS O
880 PRINT CHR$(7)
900 PRINT CHR$(7):INPUT "HUR MÅNGA KOPIOR ",U
920 FOR UU=1 TO U
940 PRINT £O
960 PRINT £O,"80-RADERS REDUCERAT SYSTEM"
980 LET K=O
1000 FOR I=1 TO 20
1020 LET K=K+1
1040 IF U(I)|=9 THEN PRINT £O,TAB(K*4-2);U(I); ELSE PRINT £O,TAB(K*4-3);U(I
1060 NEXT I
1080 PRINT £O,
1100 PRINT £O:PRINT £O," KUPONG 1"
1120 FOR R=1 TO 80
1140 LET I=1
1160 FOR K=1 TO 20
1180 READ T
1200 IF T=O THEN 1260
1220 LET S(R,I)=U(K)
1240 LET I=I+1
1260 NEXT K
1280 NEXT R
1300 FOR R=1 TO 80
1320 IF LEFT$(S1$,1)="J" THEN 1500
```

```
3220 DATA 1,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,0,0,0,1
3240 DATA 0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,1,0,0,1,0
3260 DATA 0,0,1,0,0,0,1,0,0,0,0,1,0,1,1,1,0,1,0,0
3280 DATA 0,0,0,1,1,0,0,0,1,0,0,0,0,1,1,1,1,0,0,0
3300 DATA 1,0,0,0,0,1,0,0,0,0,1,0,1,0,0,0,1,1,1,0
3320 DATA 0,1,0,0,0,0,1,0,1,0,0,0,0,1,0,0,1,1,1,0
3340 DATA 0,0,1,0,1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,0
3360 DATA 0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,1,1,1,0
3380 DATA 1,0,0,0,1,0,0,0,0,0,0,1,0,1,0,0,1,1,0,1
3400 DATA 0,1,0,0,0,0,0,1,0,1,0,0,1,0,0,0,1,1,0,1
3420 REM
3440 DATA 0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,1,1,1,0,1
3460 DATA 0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,0,1,1,0,1
3480 DATA 1,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,1,0,1,1
3500 DATA 0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,1,0,1,1
3520 DATA 0,0,1,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,1,1
3540 DATA 0,0,0,1,0,1,0,0,0,1,0,0,0,1,0,0,1,0,1,1
3560 DATA 1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,1,1,1
3580 DATA 0,1,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0,1,1,1
3600 DATA 0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,1,1,1
3620 DATA 0,0,0,1,1,0,0,0,1,0,0,0,1,0,0,0,0,1,1,1
3640 END
```

```
1340 FOR I=1 TO 6
1360 FOR J=I+1 TO 7
1380 IF S(R,I)|S(R,J) THEN 1460
1400 LET P=S(R,I)
1420 LET S(R,I)=S(R,J)
1440 LET S(R,J)=P
1460 NEXT J
1480 NEXT I
1500 FOR K=1 TO 7
1520 IF S(R,K)|=9 THEN PRINT £0,TAB(K*4-2);S(R.K): ELSE PRINT
1540 NEXT K                                      £0,TAB(K*4-3)
1560 PRINT £0
1580 IF R/10=INT(R/10) THEN PRINT £0:PRINT £0," KUPONG ";R/10+1
1600 NEXT R
1620 FOR PA=1 TO 6:PRINT £0:NEXT PA:PRINT £0
1640 RESTORE
1660 NEXT UU
1680 PRINT £0, :PRINT :PRINT CHR$(07)
1700 INPUT "VILL DU SPARA SYSTEMET ",S1$
1720 IF LEFT$(S1$,1)="N" THEN 2620
1740 OPEN NEW "LOTT80" AS 1
1760 FOR R=1 TO 80
1780 FOR K=1 TO 7
1800 PRINT £1,S(R,K)
1820 NEXT K
1840 NEXT R
1860 CLOSE 1
1880 GOTO 3640
1900 DATA 1,1,1,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0
1920 DATA 1,1,1,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0
1940 DATA 1,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0
1960 DATA 1,1,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1
1980 DATA 1,1,0,1,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1
2000 DATA 1,1,0,1,0,1,0,0,1,0,0,0,0,0,0,1,0,0,1,0
2020 DATA 1,1,0,1,0,0,1,0,0,0,0,1,1,0,0,0,0,1,0,0
2040 DATA 1,1,0,1,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0
2060 DATA 1,0,1,1,1,0,0,0,0,0,1,0,0,0,0,1,0,1,0,0
2080 DATA 1,0,1,1,0,1,0,0,0,0,0,1,0,0,1,0,1,0,0,0
2100 REM
2120 DATA 1,0,1,1,0,0,1,0,1,0,0,0,0,1,0,0,0,0,0,1
2140 DATA 1,0,1,1,0,0,0,1,0,1,0,0,1,0,0,0,0,0,1,0
2160 DATA 0,1,1,1,1,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0
2180 DATA 0,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,0,0,1
2200 DATA 0,1,1,1,0,0,1,0,0,1,0,0,0,0,0,1,1,0,0,0
2220 DATA 0,1,1,1,0,0,0,1,1,0,0,0,0,0,1,0,0,1,0,0
2240 DATA 1,0,0,0,1,1,1,0,1,0,0,0,0,0,1,0,0,1,0,0
2260 DATA 0,1,0,0,1,1,1,0,0,1,0,0,1,0,0,0,0,0,1,0
2280 DATA 0,0,1,0,1,1,1,0,0,0,1,0,0,1,0,0,1,0,0,0
2300 DATA 0,0,0,1,1,1,1,0,0,0,0,1,0,0,0,1,0,0,0,1
2320 REM
2340 DATA 1,0,0,0,1,1,0,1,0,1,0,0,0,0,0,1,1,0,0,0
2360 DATA 0,1,0,0,1,1,0,1,1,0,0,0,0,1,0,0,0,0,0,1
2380 DATA 0,0,1,0,1,1,0,1,0,0,0,1,1,0,0,0,0,1,0,0
2400 DATA 0,0,0,1,1,1,0,1,0,0,1,0,0,0,1,0,0,0,1,0
2420 DATA 1,0,0,0,1,0,1,1,0,0,1,0,1,0,0,0,0,0,0,1
2440 DATA 0,1,0,0,1,0,1,1,0,0,0,1,0,0,1,0,1,0,0,0
2460 DATA 0,0,1,0,1,0,1,1,1,0,0,0,0,0,0,1,0,0,1,0
2480 DATA 0,0,0,1,1,0,1,1,0,1,0,0,0,1,0,0,0,1,0,0
2500 DATA 1,0,0,0,0,1,1,1,0,0,0,1,0,1,0,0,0,0,1,0
2520 DATA 0,1,0,0,0,1,1,1,0,0,1,0,0,0,0,1,0,1,0,0
2540 REM
2560 DATA 0,0,1,0,0,1,1,1,0,1,0,0,0,0,1,0,0,0,0,1
2580 DATA 0,0,0,1,0,1,1,1,1,0,0,0,1,0,0,0,1,0,0,0
2600 DATA 1,0,0,0,1,0,0,0,1,1,1,0,0,1,0,0,0,0,1,0
2620 DATA 0,1,0,0,0,1,0,0,1,1,1,0,0,0,1,0,1,0,0,0
2640 DATA 0,0,1,0,0,0,1,0,1,1,1,0,1,0,0,0,0,1,0,0
2660 DATA 0,0,0,1,0,0,0,1,1,1,1,0,0,0,0,1,0,0,0,1
2680 DATA 1,0,0,0,0,1,0,0,1,1,0,1,1,0,0,0,0,0,0,1
2700 DATA 0,1,0,0,1,0,0,0,1,1,0,1,0,0,0,1,0,1,0,0
2720 DATA 0,0,1,0,0,0,0,1,1,1,0,1,0,1,0,0,1,0,0,0
2740 DATA 0,0,0,1,0,0,1,0,1,1,0,1,0,0,1,0,0,0,1,0
2760 REM
2780 DATA 1,0,0,0,0,0,1,0,1,0,1,1,0,0,0,1,1,0,0,0
2800 DATA 0,1,0,0,0,0,0,1,1,0,1,1,1,0,0,0,0,0,1,0
2820 DATA 0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,0,0,0,1
2840 DATA 0,0,0,1,0,1,0,0,1,0,1,1,0,1,0,0,0,1,0,0
2860 DATA 1,0,0,0,0,0,0,1,0,1,1,1,0,0,1,0,0,1,0,0
2880 DATA 0,1,0,0,0,0,1,0,0,1,1,1,0,1,0,0,0,0,0,1
2900 DATA 0,0,1,0,0,1,0,0,0,1,1,1,0,0,0,1,0,0,1,0
2920 DATA 0,0,0,1,1,0,0,0,0,1,1,1,1,0,0,0,1,0,0,0
2940 DATA 1,0,0,0,0,0,1,0,0,1,0,0,1,1,1,0,1,0,0,0
2960 DATA 0,1,0,0,1,0,0,0,0,0,1,0,1,1,1,0,0,1,0,0
2980 REM
3000 DATA 0,0,1,0,0,1,0,0,1,0,0,0,1,1,1,0,0,0,1,0
3020 DATA 0,0,0,1,0,0,0,1,0,0,0,1,1,1,1,0,0,0,0,1
3040 DATA 1,0,0,0,0,0,0,1,1,0,0,0,1,1,0,1,0,1,0,0
3060 DATA 0,1,0,0,0,1,0,0,0,0,0,1,1,1,0,1,1,0,0,0
3080 DATA 0,0,1,0,1,0,0,0,0,1,0,0,1,1,0,1,0,0,0,1
3100 DATA 0,0,0,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0,1,0
3120 DATA 1,0,0,0,1,0,0,0,0,0,0,1,1,0,1,1,0,0,1,0
3140 DATA 0,1,0,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,0,1
3160 DATA 0,0,1,0,0,0,0,1,0,0,1,0,1,0,1,1,1,0,0,0
3180 DATA 0,0,0,1,0,1,0,0,0,1,0,0,1,0,1,1,0,1,0,0
3200 REM
```

```
10 PRINT " KOLL AV 80-RADERS LOTTO TIPS"
20 REM ** NAMN L80CHK.BAS **
30 DIM S(80,7),V(9)
40 OPEN OLD "LOTT80" AS 1
50 LET E=0:T=0:R4=0:R5=0:R6=0:R7=0:T7=0
60 PRINT "MATA IN RÄTT RAD I NUMMERORDNING ( 7 NR ) MED"
70 PRINT "KOMMA MELLAN NUMREN "
80 INPUT V(1),V(2),V(3),V(4),V(5),V(6),V(7)
90 INPUT "MATA IN TILLÄGGSUMMER ( 2 NR ) ",V(8),V(9)
100 PRINT " ... V  Ä  N  T  A ..."
110 FOR R=1 TO 80
120 FOR K=1 TO 7
130 INPUT £1,S(R,K)
140 NEXT K
150 NEXT R
160 FOR R=1 TO 80
170 LET E=0:T=0
180 FOR K=1 TO 7
190 FOR I=1 TO 7
200 IF S(R,K)=V(I) THEN E=E+1
210 NEXT I
220 NEXT K
230 IF E|4 THEN 260
240 GOSUB 360
250 GOSUB 400
260 NEXT R
270 PRINT CHR$(07)
280 PRINT "     R E S U L T A T L I S T A"
290 PRINT :PRINT
300 PRINT "7 RÄTT ",R7,"RADER"
310 PRINT "6+1 RÄTT ",T7,"RADER"
320 PRINT "6 RÄTT ",R6,"RADER"
330 PRINT "5 RÄTT ",R5,"RADER"
340 PRINT "4+1 RÄTT ",R4,"RADER"
350 CLOSE 1:GOTO 520
360 IF E=7 THEN R7=R7+1
370 IF E=6 THEN R6=R6+1
380 IF E=5 THEN R5=R5+1
390 RETURN
400 IF E=7 THEN 480
410 IF E=5 THEN 480
420 FOR K=1 TO 7
430 FOR I=8 TO 9
440 IF S(R,K)=V(I) THEN T=T+1
450 NEXT I
460 NEXT K
470 IF T§=1 THEN 490
480 RETURN
490 IF E=6 THEN R6=R6-1:T7=T7+1
500 IF E=4 THEN R4=R4+1
510 GOTO 480
520 END
```

TBUG

På begäran följer här en lista utav TBUG2C , en monitor med single-stepp plus en hel del nyttiga funktioner. Ett stort TACK säger vi till TOMMY och Åke.

```
                    NAM    TBUG 2C
              *
              * 1980-10-16 TOMMY BLADH OCH ÅKE HOLM
              * COPYRIGHT BY TOMMY BLADH        OCH ÅKE HOLM
              *             BÅSTADSVÄGEN 7          LÅNGSJÖVÄGEN 15B
              *             121 51 JOHANNESHOV      135 54 TYRESÖ
              *
              *
              * KOMMANDON I TBUG2C
              * É VARMSTART FLEX 2.0
              * A
              * B DISPLAY BREAKPOINTS
              * C CONTINUE
              * D DUMPA I HEX OCH ASCII 8 BYTES/RAD
              * E EXTENDED INSTRUKTIONSSÖKNING
              * F GÅ TILL MINIDISK-BOOT
              * G GÅ TILL PROGRAMADRESS OCH EXEKVERA
              * H BERÄKNA CHECKSUMMA AAAA-BBBB
              * I
              * J GÅ TILL SUBRUTIN OCH SKRIV UT REGISTREN
              * K KOPIERA MINNESINNEHÅLL
              * L LADDA MINNET I S1-FORMAT MED EV. OFFSET
              * M MINNESMANIPULERING
              *   - GER FÖREGÅENDE MINNESPOSITION
              *   . GER NÄSTA MINNESPOSITION
              *   OAAAA OFFSETBERÄKNING
              * N NÄSTA INSTRUKTION
              * O FYLL ETT VISST MINNESBLOCK MED EN BYTE
              * P PUNCH I S1-FORMAT MED RUBRIK
              * Q MÄT RAM-STORLEKEN
              * R PRINTA REGISTREN
              * S SÖK 2 BYTES ADRESS
              * T
              * U TA BORT BREAKPOINTS
              * V SÄTT IN BREAKPOINTS MAX 5 ST
              * W MINNESTEST
              * X EXCHANGE AA NNNN MOT AA MMMM
              * Y VERIFIERING
              * Z DUMPA I HEX OCH ASCII 16 BYTES/RAD
              * Ä
              * Ö
              * Å
              * Ü
              *
              *
8008          ACIAT   EQU    $8008     TERMINAL ACIA
80D0          PIASAH  EQU    $80D0
80D1          PIASAL  EQU    $80D1
80D2          PIACRA  EQU    $80D2
80D3          PIACRB  EQU    $80D3
8008          PORT    EQU    $8008
8014          DRVREG  EQU    $8014
8018          COMREG  EQU    $8018
801A          SECREG  EQU    $801A
801B          DATREG  EQU    $801B
0085          SKIP1   EQU    $85       SKIPPAR 1 INSTRUKTION
008C          SKIP2   EQU    $8C       SKIPPAR 2 BYTE INSTR
              *
              * DATA AREOR
              *
A000                  ORG    $A000
```

```
                *
A000            IOV     RMB    2        IRQ VEKTOR
A002            BEGA    RMB    2        START ADRESS /PRINT/PUNCH
A004            ENDA    RMB    2        SLUT ADRESS
A006            NIO     RMB    2        NMI VEKTOR
A008            SP      RMB    2        ANVÄNDARENS STACK PEKARE
A00A            CKSM    RMB    1        CHECKSUMMA
A00B            BYTECT  RMB    1        RÄKNAREKNARE
A00C            XHI     RMB    1        TEMP LAGRING X-REG HÖG BYTE
A00D            XLOW    RMB    1        LAGRING LÅG BYTE
A00E            MCONT   RMB    1
A00F            SWIVEK  RMB    2        EGEN SWI VEKTOR
A011            EKO     RMB    1        EKO FLAGGA
A012            DMASK   RMB    1
A013            SWIFLX  RMB    2        SWIVEKTOR FÖR FLEX
A015                    RMB    43       STACK AREA
A040            STACK   RMB    1        HÄR BÖRJAR STACKEN
A041            VFLAG   RMB    1        ANTAL INSATTA BRKPT'S
A042            BPTAB   RMB    15       5 ST BREAK POINTS
A051            BUFF    RMB    50       BUFFERT AREA
A083                    RMB    1        IN BUFFERT
A000            TILL    EQU    IOV
A00B            SLASK   EQU    BYTECT
A000            LOFFST  EQU    IOV
                *
E000                    ORG    $E000
                *
E000 FE A0 00   IO      LDX    IOV      IRQ
E003 6E 00              JMP    X        TAG HAND OM AVBROTTET
                *
                * NMI SERVICE RUTIN
                *
E005 FE A0 06   POWDWN  LDX    NIO      NMI
E008 6E 00              JMP    X
                *
                * SWI SERVICE RUTIN
                *
E00A FE A0 0F   SWISER  LDX    SWIVEK
E00D 6E 00              JMP    X
                *
                * HAR PA 'L' KOMMANDO
                *
                *
                * LOAD
                *
E00F 7C A0 11   LOAD    INC    EKO
E012 86 11      LOADL   LDA A  £$11     DC1
E014 8D 5F              BSR    OUTCH
                *
E016 8D 60      LOAD3   BSR    INCH     SÖK RECORD TYP
E018 81 53              CMP A  £'S
E01A 26 FA              BNE    LOAD3
E01C 8D 5A              BSR    INCH
E01E 81 39              CMP A  £'9      EOF
E020 27 22              BEQ    LOAD21
E022 81 31              CMP A  £'1      DATA RECORD
E024 26 F0              BNE    LOAD3
E026 7F A0 0A           CLR    CKSM     ATT BYGGA CHECKSUMMA I
E029 8D 2A              BSR    BYTE     LÄS IN BYTE RÄKNAREN
E02B 80 02              SUB A  £2       KOMMANDE ADRESS= 2 BYTE
E02D B7 A0 0B           STA A  BYTECT
E030 4A                 DEC A
```

```
E031 B7 A0 53          STA A  BUFF+2
                *
                * TILLVERKA ADRESS
                *
E034 8D 11             BSR    BADDR
E036 BD E2 DC          JSR    ADRMOD    TILLV SLUTL LADDADRESS
E039 CE A0 54          LDX    £BUFF+3   TEMP BUFFERT ADRESS
                *
E03C 7E E2 55          JMP    LOAD10
                *
E040                   ORG    $E040
E040 BD E2 91   LOAD19 JSR    PIIP      PIP OCH FRÅGETECKEN
E043 01                NOP
E044            LOAD21 EQU    *
E044 7E E0 D5   C1     JMP    CONTRL
                *
                * BYGG ADRESS
                *
E047 8D 0C      BADDR  BSR    BYTE      HÖGSTA 8 BITARNA
E049 B7 A0 0C          STA A  XHI
E04C 8D 07             BSR    BYTE
E04E B7 A0 0D          STA A  XLOW
E051 FE A0 0C          LDX    XHI       HELA ADRESSEN
E054 39                RTS
                *
                * TAG IN TVÅ TECKEN. GÖR EN DATA BYTE
                *
E055 8D 53      BYTE   BSR    INHEX     HEXADECIMALT TECKEN
E057 48         BYTE2  ASL A
E058 48                ASL A
E059 48                ASL A
E05A 48                ASL A            HHHH 0000
E05B 16                TAB
E05C 8D 4C             BSR    INHEX
E05E 1B                ABA              HHHH LLLL
E05F 16                TAB
E060 FB A0 0A          ADD B  CKSM      ADDERA TILL CHECKSUMMAN
E063 F7 A0 0A          STA B  CKSM
E066 39                RTS
                *
E067 44         OUTHL  LSR A
E068 44                LSR A
E069 44                LSR A
E06A 44                LSR A
                *
E06B 84 0F      OUTHR  AND A  £$0F      MASKA BORT HÖG DELEN
E06D 8B 30             ADD A  £$30      TILLVERKA ASCII
E06F 81 39             CMP A  £$39      STÖRRE ÄN 9
E071 23 02             BLS    OUTCH
E073 8B 07             ADD A  £$7       FÖR A-F
                *
                * MATA UT ETT TECKEN.
                *
E075 7E E1 D1   OUTCH  JMP    OUTEEE
E078 7E E1 AC   INCH   JMP    INEEE     TAG IN ETT TECKEN
                *
                * SKRIV UT TEXT STRÄNG.
                * X-REG PEKAR PÅ TECKNEN.
                * AVSLUT MED 'EOT' (04).
                *
E07B 8D F8      PDATA2 BSR    OUTCH
```

```
E07D 08                    INX
E07E A6 00      PDATA1     LDA A  X
E080 81 04                 CMP A  £$4         'EOT'
E082 26 F7                 BNE    PDATA2
E084 39                    RTS
                *
E085 8D F1      INCH7      BSR    INCH        HÄMTA TECKEN
E087 81 5F                 CMP A  £$5F        KOLLA OM SMÅ BOKSTÄVER
E089 23 02                 BLS    INCH8
E08B 80 20                 SUB A  £$20
E08D 39         INCH8      RTS
                *
E08E 43         TEXT06     FCC    'Cksm:'
E093 04                    FCB    4
                *
E094 BD E1 DE   IOINIT     JSR    BEGENS      START OCH SLUTADRESSER
E097 86 12      DC2OUT     LDA A  £$12        DC2 LÄSARE PÅ
E099 20 DA                 BRA    OUTCH       + RTS
                *
E09B                       ORG    $E09B
                *
E09B 8D 09      BEGEND     BSR    BADDRS
E09D FF A0 02              STX    BEGA
E0A0 8D 04      ENDADR     BSR    BADDRS
E0A2 FF A0 04              STX    ENDA
E0A5 39                    RTS
                *
E0A6 8D 9F      BADDRS     BSR    BADDR
E0A8 20 22                 BRA    OUTS        SP+RTS
                *
                *
E0AA                       ORG    $E0AA
                *
                * TAG IN HEXADECIMALT TECKEN.
                * OMVANDLA ASCII 0-F TILL BINÄRT
                * 0-F
                *
E0AA 8D D9      INHEX      BSR    INCH7
E0AC 80 30      INHEX2     SUB A  £$30
E0AE 2B 94                 BMI    C1          HOPPA OM EJ HEX
E0B0 81 09                 CMP A  £$09
E0B2 2F 0A                 BLE    IN1HG       DECIMALT ?
E0B4 81 11                 CMP A  £$11
E0B6 2B 8C                 BMI    C1          HOPPA OM EJ HEX
E0B8 81 16                 CMP A  £$16
E0BA 2E 88                 BGT    C1          HOPPA OM EJ HEX
E0BC 80 07                 SUB A  £$7
E0BE 39         IN1HG      RTS
                *
E0BF A6 00      OUT2H      LDA A  X           PRINTA 2 HEX TECKEN
E0C1 08         OUT2HA     INX
E0C2 36         OUT2HT     PSH A
E0C3 8D A2                 BSR    OUTHL       OMVANDLA TILL ASCII
E0C5 32                    PUL A
E0C6 20 A3                 BRA    OUTHR       PRINTA HÖGRA BYTE HALVAN
                *
E0C8 8D F5      OUT4HS     BSR    OUT2H       PRINTA 4 HEX TECKEN + BLANK
E0CA 8D F3      OUT2HS     BSR    OUT2H       PRINTA 2 HEX TECKEN + BLANK
E0CC 86 20      OUTS       LDA A  £$20        MELLANSLAG
E0CE 20 A5                 BRA    OUTCH       (RETURN I OUTCH)
                *
                * VARM -START
```

```
                  *
E0D0 CE E1 04     START   LDX     £STTBUG     'TBUG 2'
E0D3 8D A9        STARTS  BSR     PDATA1
                  *
E0D5 8E A0 38     CONTRL  LDS     £STACK-8    INITIERA STACKEN
E0D8 01                   NOP
E0D9 01                   NOP
E0DA 7F A0 11             CLR     EKO         FULL DUPLEX
E0DD CE E1 9C             LDX     £MCLOFF     INITIERINGS STRÄNG
E0E0 8D 9C                BSR     PDATA1
                  *
                  * TAG IN KOMMANDO I A-ACC
                  * SÖK SEDAN INGÅNG TILL
                  * BEGÄRD RUTIN I KOMTAB.
                  *
                  * ENTRY-PUNKT FOR MIKBUG/CONTRL
                  *
E0E2 8C                   FCB     SKIP2
E0E3 20 F0        MIKCON  BRA     CONTRL
                  *
E0E5 8D 91                BSR     INCH
E0E7 16                   TAB                 SPARA UNDAN
E0E8 8D E2                BSR     OUTS        BLANK
E0EA C1 60                CMP B   £$60        GÖR OM SMÅ BOKSTÄVER
E0EC 25 02                BLO     CON50       TILL STORA
E0EE C0 20                SUB B   £$20
                  *
E0F0 C0 3F        CON50   SUB B   £'É-1       B ÄR NU 1 FÖR STOR
E0F2 23 E1                BLS     CONTRL      FEL DÅ B| É
E0F4 C1 1E                CMP B   £$1E
E0F6 22 DD                BHI     CONTRL      FEL DÅ B VAR §Ü
                  *
                  * INMATAT TECKEN OMGJORD TILL ADRESS
                  * I KOMMANDOTABELLEN
                  *
E0F8 CE E4 03             LDX     £KOMTAB-2
E0FB 08           NESTA   INX
E0FC 08                   INX
E0FD 5A                   DEC B
E0FE 26 FB                BNE     NESTA
E100 EE 00                LDX     X           INHOPPSADRESS TILL RUTIN
E102 6E 00                JMP     X           PYS DIT & JOBBA
                  *
E104 0D           STTBUG  FCB     $D,$A,0,0
E108 54                   FCC     'TBUG 2C'
E10F 04                   FCB     4
                  *
                  * NMI AVBROTT - SINGLE STEP
                  *
E110 BF A0 08     NONMSK  STS     SP          SPARA STACKPEKAREN
E113 BD E4 43             JSR     DISNMI
                  *
                  * HÄR PÅ 'R' KOMMANDO.
                  *
E116 8D 02        PRINT   BSR     PRREG
E118 20 BB        C2      BRA     CONTRL
                  *
                  * SKRIV UT STACKENS INNEHÅLL
                  *
E11A 8D 15        PRREG   BSR     PCRLF
E11C FE A0 08             LDX     SP
E11F 08                   INX
```

```
E120 86 43                LDA A  £'C
E122 BD E2 97             JSR    UTMIN
E125 8D A3                BSR    OUT2HS     CC
E127 86 42                LDA A  £'B
E129 BD E2 97             JSR    UTMIN
E12C 8D 9C                BSR    OUT2HS     ACC-B
E12E 7E E6 EC             JMP    PRREG9
                *
                *
E131 CE E1 A6   PCRLF     LDX    £SCRLF     CR+LF
E134 7E E0 7E   PDAT1     JMP    PDATA1     STRENG + RTS
                *
                *  PUNCH DUMP I S1 FORMAT
                *
E137 0D         MTAPE1    FCB    $D,$A,0,0,0,0,'S,'1,4 PUNCH FORMATET
                *
                * HÄR PÅ 'P' KOMMANDO.
                *
E140 BD E0 94   PUNCH     JSR    IOINIT
                *
E143 8D 3E      PUN11     BSR    CMPADR     JAMFOR START/STOPP ADR
E145 26 04                BNE    PUN22
E147 81 10                CMP A  £16
E149 25 02                BCS    PUN23
                *
                * LÄGG UT 16 MINNESBYTE/RECORD
                *
E14B 86 0F      PUN22     LDA A  £15
E14D 8B 04      PUN23     ADD A  £4         INKLUDERA ADRESS I BYTE RÄKNAREN
E14F B7 A0 0E             STA A  MCONT
E152 80 03                SUB A  £3
E154 B7 A0 51             STA A  BUFF       DETTA RECORS BYTE ANTAL
                *
                * PUNCHA CR,LF,NULL,S,1
                *
E157 CE E1 37             LDX    £MTAPE1
E15A 8D D8                BSR    PDAT1
E15C 5F                   CLR B             CHECKSUMMA
                *
                * PUNCHA RECORDETS BYTE RÄKNARE
                *
E15D CE A0 0E             LDX    £MCONT
E160 8D 2E                BSR    PUNT2      2 HEX
                *
                * PUNCHA ADRESSEN
                *
E162 CE A0 02             LDX    £BEGA
E165 8D 29                BSR    PUNT2
E167 8D 27                BSR    PUNT2
                *
                * DATA
                *
E169 FE A0 02             LDX    BEGA
E16C 8D 22      PUN32     BSR    PUNT2      2 ASCII TECKEN
E16E 7A A0 51             DEC    BUFF       BYTE RÄKNAREN
E171 26 F9                BNE    PUN32
                *
                * KLAR
E173 53                   COM B             CHECKSUMMAN INVERTERAS
E174 17                   TBA
E175 BD E0 C2             JSR    OUT2HT     2 HEX TECKEN
E178 FF A0 02             STX    BEGA
```

```
E17B 09                DEX
E17C BC A0 04          CPX    ENDA        KLAR MED HELA DUMPEN
E17F 26 C2             BNE    PUN11
E181 20 12             BRA    S9REC       JA KLAR
               *
               * JÄMFOR START/STOPP ADRESSER
               *
E183 B6 A0 05  CMPADR  LDA A  ENDA+1
E186 B0 A0 03          SUB A  BEGA+1
E189 F6 A0 04          LDA B  ENDA
E18C F2 A0 02          SBC B  BEGA
E18F 39                RTS
               *
               * PUNCHA 2 HEX, UPPDATERA CHECKSUMMAN
               *
E190 EB 00     PUNT2   ADD B  X
E192 7E E0 BF          JMP    OUT2H       PUNCHA 2 TECKEN OCH RTS
               *
               * AVSLUTA MED S9
               *
E195 CE E1 CD  S9REC   LDX    £S9STR      'S901'
E198 7E E6 E6          JMP    S9PAUS
               *
E19C                   ORG    $E19C
               *
               *
E19C 13        MCLOFF  FCB    $13         READER RELAY OFF
E19D 0D        MCL     FCB    $D,$A,0,0,0,$14,0,'*,4
               *
E1A6 0D        SCRLF   FCB    $D,$A,0,0,0,4
               *
E1AC                   ORG    $E1AC
               *
               * TAG IN ETT TECKEN I ACC A
               *
E1AC 8D 0C     INEEE   BSR    INCHP
E1AE 84 7F             AND A  £$7F        MASKA PARITETEN
E1B0 81 7F             CMP A  £$7F        IGNORERA 'RUBOUT'
E1B2 27 F8             BEQ    INEEE
E1B4 7D A0 11          TST    EKO         SKALL TECKNET EKAS UT
E1B7 27 18             BEQ    OUTEEE      HOPPA OM JA
E1B9 39                RTS
               *
E1BA B6 80 08  INCHP   LDA A  ACIAT       FINNS TECKEN
E1BD 47                ASR A
E1BE 24 FA             BCC    INCHP
E1C0 B6 80 09          LDA A  ACIAT+1     DATA
E1C3 39                RTS
               *
               * NÄSTA INSTR
               *
E1C4 BE A0 08  NEST    LDS    SP
E1C7 86 34             LDA A  £$34
E1C9 B7 80 D2          STA A  PIACRA
E1CC 3B                RTI
               *
E1CD 00        S9STR   FCB    0
E1CE 53                FCC    'S9'
E1D0 04                FCB    4
               *
               *
               *
```

```
E1D1                   ORG    $E1D1
                *
                * MATA UT ETT TECKEN FRÅN A-ACC
                *
E1D1 37         OUTEEE PSH B
                *
E1D2 F6 80 08   OUTO   LDA B  ACIAT     KONTROLLORDET
E1D5 57                ASR B
E1D6 57                ASR B
E1D7 24 F9             BCC    OUTO
E1D9 B7 80 09          STA A  ACIAT+1   SÄND TECKNET
E1DC 33                PUL B
E1DD 39                RTS
                *
                *
E1DE CE E7 AF   BEGENS LDX    £TEXT04
E1E1 BD E5 2B          JSR    PDATA3
E1E4 BD E0 9B          JSR    BEGEND    FIXA START &STOPP
E1E7 8D 9A             BSR    CMPADR
E1E9 25 F3             BCS    BEGENS
E1EB 39                RTS
                *
E1EC 8D F0      BEGENT BSR    BEGENS
E1EE 08                INX
E1EF FF A0 04          STX    ENDA
E1F2 39                RTS
                *
E1F3 4F         TEXT07 FCC    'Opcode:'
E1FA 04                FCB    4
                *
                * HÄR PÅ 'M' KOMMANDO.
                *
                *  MINNES MANIPULERING.
                *
E1FB BD E0 A6   CHANGE JSR    BADDRS    BYGG ADRESS , PRINTA BLANK
E1FE FE A0 0C   CHANG  LDX    XHI
E201 BD E0 CA          JSR    OUT2HS    PRINTA GAMLA INNEHÅLLET
E204 FF A0 02          STX    BEGA
E207 09                DEX
E208 BD E0 85   CHA1   JSR    INCH7     SMÅ EL STORA BOKST.
E20B 81 2E             CMP A  £'.       LINE FEED-ERSÄTTNING
E20D 27 30             BEQ    LF
E20F 81 2D             CMP A  £'-       Ü
E211 27 2A             BEQ    UA
E213 81 4F             CMP A  £'O       BERÄKNA OFFSET
E215 26 18             BNE    NYDATA
E217 BD E0 A0          JSR    ENDADR    TAG IN DESTINATIONS ADRESSEN
E21A BD E1 83          JSR    CMPADR    JAMFOR ADRESSERNA
                *
                * OM B=00, INOM RECKHÅLL FRAMMÅT
                *          OM A|80
                * OM B=FF, INOM RECKHÅLL BAKÅT
                *
E21D 4D                TST A            OFFSET FRAMMÅT |80 ÄR OK
E21E 2B 04             BMI    OK
E220 5D                TST B
E221 26 2E             BNE    CHOFEL
E223 5A                DEC B            B=FF NU
E224 5C         OK     INC B
E225 26 2A             BNE    CHOFEL    FEL I OFFSET BAKÅT
E227 BD E0 C2          JSR    OUT2HT    PRINTA OFFSET (A-ACC)
                *
```

```
E22A FE A0 02  CHOCON  LDX    BEGA       URSPRUNGLIGA OPCODE ADRESS
E22D 20 0E             BRA    UA
E22F BD E0 AC  NYDATA  JSR    INHEX2
E232 BD E0 57          JSR    BYTE2
E235 A7 00             STA A  X
E237 A1 00             CMP A  X
E239 26 16             BNE    CHOFEL
E23B 08                INX
E23C 08                INX
               *
E23D 09        UA      DEX
E23E 85                FCB    SKIP1      HOPPA ÖVER NÄSTA INSTRUKTION
E23F 08        LF      INX
E240 FF A0 0C          STX    XHI
E243 CE E1 9D          LDX    £MCL       LF+CR,NULLS
E246 BD E0 7E          JSR    PDATA1
E249 CE A0 0C          LDX    £XHI
E24C BD E0 C8          JSR    OUT4HS     SKRIV DATA ADRESSEN
E24F 20 AD             BRA    CHANG      OM IGEN
               *
E251 8D 3E     CHOFEL  BSR    PIIP
E253 20 D5             BRA    CHOCON
               *
               *
               * FORTSÄTTNING PÅ LOAD
               *
E255 FF A0 02  LOAD10  STX    BEGA
               *
E258 BD E0 55  LOAD11  JSR    BYTE
E25B 7A A0 0B          DEC    BYTECT     ANTAL BYTES I RECORDET
E25E 27 0B             BEQ    LOAD80     HOPPA OM KLAR
E260 A7 00             STA A  X          LAGRA I TEMP. BUFFER
E262 08                INX
E263 8C A0 83          CPX    £BUFF+50   SLUT PÅ BUFFERT
E266 26 F0             BNE    LOAD11
E268 7E E0 40  LOAD70  JMP    LOAD19
               *
               * KOLLA CHECKSUMMAN
               *
E26B 7C A0 0A  LOAD80  INC    CKSM
E26E 26 1C             BNE    LOADA0
               *
               * TAG DATA FRÅN TEMPORÄRA BUFFERTERN
               * OCH FLYTTA TILL MINNET
               *
E270 FE A0 02  LOAD90  LDX    BEGA
E273 A6 00             LDA A  X
E275 08                INX
E276 FF A0 02          STX    BEGA
               *
E279 FE A0 51          LDX    BUFF       ADRESS TILL MINNET
E27C A7 00             STA A  X
E27E A1 00             CMP A  X          FASTNADE DET
E280 26 E6             BNE    LOAD70
               *
E282 08                INX
E283 FF A0 51          STX    BUFF
E286 7A A0 53          DEC    BUFF+2     BYTE RÄKNARE
E289 26 E5             BNE    LOAD90
               *
E28B 8C                FCB    SKIP2
```

```
E28C 8D 03     LOADAO  BSR     PIIP
E28E 7E E0 16          JMP     LOAD3
                *
E291 86 07     PIIP    LDA A   £$07
E293 8D 06             BSR     OUTEE
E295 86 3F             LDA A   £$3F
E297 8D 02     UTMIN   BSR     OUTEE
E299 86 3D     UTMIN2  LDA A   £'=
                *
E29B 7E E1 D1  OUTEE   JMP     OUTEEE      INSTRUKTIONSSPARARE
                *
E29E 7F A0 41  KILL    CLR     VFLAG       TA BORT ALLA BRKPOINTS
E2A1 20 26     BIL300  BRA     CKSM30      MELLANHOPP
                *
                *
                *
                *
                * C H K S M
                *
E2A3 BD E1 EC  CHKSM   JSR     BEGENT
E2A6 4F                CLR A
E2A7 5F                CLR B
E2A8 FE A0 02          LDX     BEGA
                *
E2AB BC A0 04  CKSM10  CPX     ENDA
E2AE 27 07             BEQ     CKSM20
E2B0 AB 00             ADD A   X
E2B2 C9 00             ADC B   £0
E2B4 08                INX
E2B5 20 F4             BRA     CKSM10
                *
E2B7 F7 A0 02  CKSM20  STA B   BEGA
E2BA B7 A0 03          STA A   BEGA+1
E2BD CE E0 8E          LDX     £TEXT06     "CKSM"
E2C0 BD E5 2B          JSR     PDATA3
E2C3 CE A0 02          LDX     £BEGA
E2C6 BD E0 C8          JSR     OUT4HS
E2C9 20 7B     CKSM30  BRA     BRKP30
                *
E2CB CE E7 C6  LOADOF  LDX     £TEXT09     'SUBTRAHERA'
E2CE BD E5 2B          JSR     PDATA3
E2D1 BD E0 55          JSR     BYTE
E2D4 85                FCB     SKIP1
E2D5 4F        LOADNO  CLR A   EJ          OFFSET
E2D6 B7 A0 00          STA A   LOFFST
E2D9 7E E0 12          JMP     LOADL
                *
                * TILLVERKA SLUTLIG LADDADRESS
                *
E2DC FF A0 51  ADRMOD  STX     BUFF        INMATAD ADRESS
E2DF B6 A0 0C          LDA A   XHI         HÖG BYTE
E2E2 B0 A0 00          SUB A   LOFFST      EVENTUELL MODIFIERING
E2E5 B7 A0 51          STA A   BUFF
E2E8 39                RTS
                *
E2E9 52        TEXT11  FCC     'Rubrik:'
E2F0 04                FCB     4
                *
E2F1                   ORG     $E2F1
                *
                *
                * SWI SERVICE RUTIN (BREAK POINTS)
```

```
                    *
E2F1 BF A0 08  SFE      STS    SP
E2F4 BD E4 43           JSR    DISNMI
E2F7 30                 TSX
E2F8 6D 06              TST    6,X      DECREMENTERA PC-VÄRDET
E2FA 26 02              BNE    *+4
E2FC 6A 05              DEC    5,X
E2FE 6A 06              DEC    6,X
E300 8D 1E              BSR    BRKP40   FINNS BREAK POINTS
E302 27 14              BEQ    BRKP20   PRINTA REGISTREN
               *
               * TAG BORT BREAK POINTS NÄR
               * VI ÄR I TBUG MONITORN
               *
E304 FF A0 0C  BRKP00   STX    XHI      SKYDDA MOT FLERA
E307 A6 02              LDA A  2,X      BRKPT'S PÄ SAMMA ADRESS
E309 81 3F              CMP A  £$3F     MULTIDEFINIERADE BRKPT'S
E30B 27 07              BEQ    BRKP10
E30D EE 00              LDX    X
E30F A7 00              STA A  X
E311 FE A0 0C           LDX    XHI
E314 8D 05     BRKP10   BSR    ADD3X
E316 26 EC              BNE    BRKP00
E318 7E E1 16  BRKP20   JMP    PRINT
               *
E31B 08        ADD3X    INX
E31C 08                 INX
E31D 08                 INX
E31E 5A                 DEC B
E31F 39                 RTS
               *
E320 CE A0 42  BRKP40   LDX    £BPTAB   TABELL ÖVER BREAK POINTS, OPCODER
E323 F6 A0 41           LDA B  VFLAG    ANTAL INSTALLERADE BRKPT'S
E326 39                 RTS
               *
               * HAR PA 'V' KOMMANDO
               *
               *
               * SÄTT IN EN BRKPT.
               *
E327 BD E7 78  SETBR    JSR    SWIINS
               *
E32A 8D F4              BSR    BRKP40   FINN LEDIG PLATS
E32C 27 0B              BEQ    BRKP60   OM LEDIG, STOPPA IN BRKPT.
E32E C1 05              CMP B  £5       MAX FEM PLATSER
E330 25 03              BCS    BRKP50
E332 7E E0 40           JMP    LOAD19
               *
E335 8D E4     BRKP50   BSR    ADD3X
E337 26 FC              BNE    BRKP50
               *
E339 7C A0 41  BRKP60   INC    VFLAG
E33C B6 A0 0C           LDA A  XHI      ADRESS TILL MINNE
E33F A7 00              STA A  X        IN I TABELLEN
E341 B6 A0 0D           LDA A  XLOW
E344 A7 01              STA A  1,X
E346 7E E0 D5  BRKP30   JMP    CONTRL
               *
               *
               * STOPPA IN ALLA BRKPTS I MINNET
               * VID 'G' ELLER 'C'
               *
```

```
E349 8D D5      BRKPIN  BSR     BRKP40
E34B 26 01      BRKPRT  BNE     *+3
E34D 39         PDKLAR  RTS
                *
E34E FF A0 0C   TGB     STX     XHI
E351 EE 00              LDX     X
E353 A6 00              LDA A   X
E355 36                 PSH A
E356 86 3F              LDA A   £$3F      SWI OPCODE
E358 A7 00              STA A   X
E35A FE A0 0C           LDX     XHI
E35D 32                 PUL A
E35E A7 02              STA A   2,X       SPARAD OPCODE
E360 8D B9              BSR     ADD3X
E362 20 E7              BRA     BRKPRT
                *
                * HÄR PÅ 'G' KOMMANDO.
                *
                *
                * GÅ OCH EXEKVERA
                *
E364 BD E0 47   GOEXEC  JSR     BADDR     START ADRESS
E367 FE A0 08           LDX     SP
E36A A7 07              STA A   7,X       LÅG ADRESS
E36C B6 A0 0C           LDA A   XHI       MSB ADRESS HALVA
E36F A7 06              STA A   6,X       HÖG ADRESS
                *
                * HÄR PÅ 'C' KOMMANDO.
                *
                *
E371 8D D6      CONTIN  BSR     BRKPIN    STOPPA IN EV BRKPT'S
E373 BE A0 08           LDS     SP
E376 3B                 RTI
                *
                *
                * HÄR PÅ 'D' KOMMANDO.
                *
                * DUMP RUTIN.
                * UTLAGD SOM SUBRUTIN FOR ATT KUNNA
                * ANVÄNDAS I AVLUSNINGSSYFTE I EGNA
                * PROGRAM. DÅ MÅSTE BEGA/ENDA LADDAS
                * MED START RESP. STOPPADRESSER.
                * BEGA MODIFIERAS AV RUTINEN.
                *
E377 C6 08      DUMP    LDA B   £8
E379 F7 A0 12           STA B   DMASK
E37C BD E1 DE           JSR     BEGENS    LAGG UPP START/STOPP
E37F 8D 03              BSR     PDUMP
E381 20 C3              BRA     BRKP30
                *
E384                    ORG     $E384
                *
E384 B6 A0 03   PDUMP   LDA A   BEGA+1
E387 84 F8              AND A   £$F8      MASKA TILL FULL RAD
E389 B7 A0 03   PDUMP5  STA A   BEGA+1
                *
E38C BD E1 31   PD1000  JSR     PCRLF     RADBYTE
                *
                * BÖRJA DUMPA
                * FORMAT: AAAA BB BB BB BB BB BB BB BB .ASCII..
                *
E38F CE A0 02           LDX     £BEGA
```

```
E392 BD E0 C8          JSR    OUT4HS     PRINTA ADRESSEN
E395 FE A0 02          LDX    BEGA
E398 F6 A0 12          LDA B  DMASK      8 BYTES/RAD
E39B BD E0 CA  PD3000  JSR    OUT2HS
E39E 5A                DEC B
E39F 26 FA             BNE    PD3000
               *
               * PRINTA SAMMA BYTES MEN I
               * EKVIVALENT ASCII.
               *
E3A1 BD E0 CC          JSR    OUTS       SPACE
E3A4 F6 A0 12          LDA B  DMASK
E3A7 FE A0 02          LDX    BEGA
E3AA A6 00     PD4000  LDA A  X
E3AC 84 7F             AND A  £$7F       MASKA BIT 8
               *
               * SKRIV EJ UT KONTROLLTECKEN
               * MEN SMÅ BOKSTÄVER,
               * DVS OM A|$20
               * ELLER  A§$7F
               * BYT DÅ MOT '.'
               *
E3AE 81 20             CMP A  £$20
E3B0 2D 04             BLT    PD5000
E3B2 81 7F             CMP A  £$7F
E3B4 2D 02             BLT    PD6000
E3B6 86 2E     PD5000  LDA A  £'.
E3B8 BD E0 75  PD6000  JSR    OUTCH
E3BB 08                INX
E3BC 5A                DEC B
E3BD 26 EB             BNE    PD4000
               *
E3BF FF A0 02          STX    BEGA
E3C2 27 89             BEQ    PDKLAR     KOLLA OM KLAR
               *
               * KOMPARERA START/STOPP ADRESSER
               *
E3C4 BD E1 83          JSR    CMPADR
E3C7 25 84             BCS    PDKLAR
E3C9 BD E4 9D          JSR    BREAK
E3CC 20 BE             BRA    PD1000     1 RADFRAMMAT
               *
               * KALLSTART
               *
E3CE           CSTART  EQU    *
E3CE 8E A0 38          LDS    £STACK-8   STACKPOINTER INITIERING
E3D1 BF A0 08          STS    SP
E3D4 86 03             LDA A  £3         RESET ACIA
E3D6 B7 80 08          STA A  ACIAT
E3D9 86 FF             LDA A  £$FF       PIA UTGÅNGAR
E3DB B7 80 D0          STA A  PIASAH
E3DE B7 80 D1          STA A  PIASAL     LSB-HALVA
E3E1 BD E4 43          JSR    DISNMI     SET CA2 HÖG
E3E4 86 34             LDA A  £$34
E3E6 B7 80 D3          STA A  PIACRB     SET CB2 LÅG
E3E9 7F A0 41          CLR    VFLAG      O ANTAL BRKPT'S INSATTA
E3EC CE E1 10          LDX    £NONMSK    PSEUDO NMI VEKTOR
E3EF FF A0 06          STX    NIO        GER NU SINGLE STEP FUNKTION
E3F2 BD E7 7B          JSR    SWIIN2
E3F5 86 15             LDA A  £$15       8 BIT,EJ PAR.,RTS LÅG, 1STOPPB.
E3F7 B7 80 08          STA A  ACIAT
E3FA 7F A0 11          CLR    EKO        FULL DUPLEX
```

```
E3FD 86 08             LDA A  £8
E3FF B7 A0 12          STA A  DMASK
E402 7E E0 D0          JMP    START     TILL RESTEN AV STARTEN
                *
                * KOMMANDOTABELL
                *
E405 E7 88      KOMTAB FDB    FLEXO     VARMSTART FLEX 2.0
E407 E0 D5             FDB    CONTRL
E409 E6 9D             FDB    DISBRK
E40B E3 71             FDB    CONTIN
E40D E3 77             FDB    DUMP
E40F E5 0A             FDB    EXTE      EXTENDED SÖKNING
E411 E7 23             FDB    BOOTO     MINIDISKBOOT
E413 E3 64             FDB    GOEXEC
E415 E2 A3             FDB    CHKSM
E417 E0 D5             FDB    CONTRL
E419 E4 49             FDB    SUBR
E41B E4 52             FDB    COPY
E41D E2 CB             FDB    LOADOF    LADDA MED EV. OFFSET
E41F E1 FB             FDB    CHANGE    MINNESMANIPULERING
E421 E1 C4             FDB    NEST
E423 E4 F5             FDB    NOLLFF
E425 E5 9D             FDB    HEADER
E427 E6 7A             FDB    METRAM
E429 E1 16             FDB    PRINT
E42B E6 B4             FDB    SOK2B
E42D E0 D5             FDB    CONTRL
E42F E2 9E             FDB    KILL
E431 E3 27             FDB    SETBR
E433 E5 D2             FDB    MEMTST
E435 E5 37             FDB    XCHA      EXCHANGE
E437 E4 C6             FDB    VERIF
E439 E7 0C             FDB    DUMP16
E43B E0 D5             FDB    CONTRL    Ä
E43D E0 D5             FDB    CONTRL    Ö
E43F E0 D5             FDB    CONTRL    Å
E441 E0 D5             FDB    CONTRL    Ü
E443            KOMTEN EQU    *
                *
                * DISABLA NMI
                *
E443 86 3C      DISNMI LDA A  £$3C
E445 B7 80 D2          STA A  PIACRA
E448 39                RTS
                *
E449 BD E0 47   SUBR   JSR    BADDR     SUBRUTINTEST
E44C 8E A0 40          LDS    £STACK
E44F AD 00             JSR    0,X       HOPPA DIT
E451 3F                SWI              SKRIV REGISTER OCH ÅTERGÅ
                **************
                *
                * K O P I E R I N G *
                ************************
                *
E452 8D 26      COPY   BSR    BEGEN3
E454 8D 65      COPY10 BSR    CBEND
E456 A6 00             LDA A  X
E458 08                INX
E459 FF A0 02          STX    BEGA      DATA HÄMTAT
                *
E45C FE A0 00          LDX    TILL
E45F A7 00             STA A  X         LAGRA
```

```
E461 6D 00            TST     X          TIME OUT
E463 A1 00            CMP A   X          FASTNADE
E465 26 05            BNE     COPY20
E467 08               INX
E468 8D 20            BSR     STXRUT
E46A 20 E8            BRA     COPY10     FORTSÄTT
              *
              * INGET MINNE I
              *
E46C CE E7 A4 COPY20  LDX     £TEXT03    'EJ ÄNDRAT'
E46F BD E5 2B         JSR     PDATA3
E472 CE A0 00         LDX     £TILL
E475 BD E0 C8         JSR     OUT4HS
E478 20 49            BRA     CTRL10
              *
E47A CE E7 91 BEGEN3  LDX     £TEXT02
E47D BD E5 2B         JSR     PDATA3
E480 BD E0 9B         JSR     BEGEND
E483 08               INX
E484 FF A0 04         STX     ENDA
E487 BD E0 A6         JSR     BADDRS
E48A FF A0 00 STXRUT  STX     TILL
E48D 39       BREART  RTS
              *
              *****
              * HJÄLPRUTINER
              ******
              *
E48E BD E1 31 ADDRDA  JSR     PCRLF
E491 CE A0 02         LDX     £BEGA
E494 BD E0 C8         JSR     OUT4HS     NNNN DD
E497 FE A0 02         LDX     BEGA
E49A 7E E0 CA UT2HS1  JMP     OUT2HS
              *
E49D B6 80 08 BREAK   LDA A   PORT
E4A0 47               ASR A
E4A1 24 EA            BCC     BREART
E4A3 B6 80 09         LDA A   PORT+1
E4A6 84 7F    BRE12   AND A   £$7F
E4A8 81 30            CMP A   £'0        PAUS ?
E4AA 27 0A            BEQ     BREKQ
E4AC 81 71    BRQ     CMP A   £'q        ÄVEN LILLA Q
E4AE 27 04            BEQ     BREAQ
E4B0 81 51            CMP A   £'Q
E4B2 26 D9            BNE     BREART
E4B4 20 0D    BREAQ   BRA     CTRL10
              *
E4B6 BD E1 BA BREKQ   JSR     INCHP
E4B9 20 EB            BRA     BRE12
              *
E4BB FE A0 02 CBEND   LDX     BEGA
E4BE BC A0 04 CBEND1  CPX     ENDA
E4C1 26 CA            BNE     BREART
E4C3 7E E0 D5 CTRL10  JMP     CONTRL
              *
              *********************
              *
              * V E R I F Y
              *
              *********************
              *
E4C6 8D B2    VERIF   BSR     BEGEN3
```

```
E4C8 8D F1      VERI10  BSR     CBEND
E4CA A6 00              LDA A   X
E4CC FE A0 00           LDX     TILL
E4CF A1 00              CMP A   X           VERIFIERA
E4D1 27 10              BEQ     VERI20      HOPPA OM SAMMA
                *
E4D3 8D B9              BSR     ADDRDA      NNNN DD
E4D5 BD E0 CC           JSR     OUTS        SP
E4D8 CE A0 00           LDX     £TILL
E4DB BD E0 C8           JSR     OUT4HS      MMMM
E4DE FE A0 00           LDX     TILL
E4E1 8D B7              BSR     UT2HS1      DD
                *
E4E3 FE A0 02   VERI20  LDX     BEGA
E4E6 08                 INX
E4E7 FF A0 02           STX     BEGA
E4EA FE A0 00           LDX     TILL
E4ED 08                 INX
E4EE FF A0 00           STX     TILL
E4F1 8D AA              BSR     BREAK
E4F3 20 D3              BRA     VERI10      OM IGEN
                *
                *
                ********************
                *
                * N O L L A  /  F F - S T Ä L L
                *
                ********************
                *
                *
E4F5 BD E1 EC   NOLLFF  JSR     BEGENT
                *
E4F8 CE E7 BD           LDX     £TEXT05     'TECKEN'
E4FB 8D 2E              BSR     PDATA3
E4FD BD E0 55           JSR     BYTE
E500 FE A0 02           LDX     BEGA
E503 8D B9      NOLL10  BSR     CBEND1
E505 A7 00              STA A   X
E507 08                 INX
E508 20 F9              BRA     NOLL10
                *
                *
                ****************************
                *
                * E X T EN D E D   A D R .   T E S T
                *
                ****************************
                *
                *
E50A BD E1 EC   EXTE    JSR     BEGENT
E50D CE E1 F3           LDX     £TEXT07     'OP-CODE'
E510 8D 19              BSR     PDATA3
E512 BD E0 55           JSR     BYTE
E515 B7 A0 0B           STA A   SLASK
                *
E518 8D A1      EXTE10  BSR     CBEND
E51A A6 00              LDA A   X
E51C B1 A0 0B           CMP A   SLASK       MATCHAR ?
E51F 26 06              BNE     EXTE20
                *
                * SKRIV NNNN 00 PPPP
                *
```

```
E521 BD E4 8E          JSR    ADDRDA
E524 BD E0 C8          JSR    OUT4HS
E527 8D 6A     EXTE20  BSR    BEGINC
E529 20 ED             BRA    EXTE10
               *
E52B FF A0 51  PDATA3  STX    BUFF
E52E BD E1 31          JSR    PCRLF
E531 FE A0 51          LDX    BUFF
E534 7E E0 7E          JMP    PDATA1
               *
               * RUTIN FÖR ATT BYTA UT ADRESSER
               * I EXT. INSTRUKTIONER
E537 BD E1 EC  XCHA    JSR    BEGENT     START+SLUT
E53A CE E1 F3          LDX    £TEXT07    'OPCODE'
E53D 8D EC             BSR    PDATA3
E53F BD E0 55          JSR    BYTE
E542 B7 A0 0B          STA A  SLASK
E545 CE E7 DD          LDX    £TEXT12+3 'ADRESS'
E548 8D E1             BSR    PDATA3
E54A BD E0 47          JSR    BADDR      IN MED AAAA
E54D FF A0 53          STX    BUFF+2     SPARA ADRESEN
E550 CE E7 DA          LDX    £TEXT12    'NY ADRESS'
E553 8D D6             BSR    PDATA3
E555 BD E0 47          JSR    BADDR      FIXA NYA ADRESSEN
E558 FF A0 55          STX    BUFF+4     SPARA DEN
E55B BD E4 BB  XCHA30  JSR    CBEND      KOLLA OM BEG=END
E55E A6 00             LDA A  0,X        1:A BYTEN
E560 B1 A0 0B          CMP A  SLASK
E563 26 2A             BNE    XCHA90
E565 A6 01             LDA A  1,X        2:A BYTEN
E567 B1 A0 53          CMP A  BUFF+2
E56A 26 23             BNE    XCHA90
E56C A6 02             LDA A  2,X
E56E B1 A0 54          CMP A  BUFF+3
E571 26 1C             BNE    XCHA90
E573 B6 A0 55          LDA A  BUFF+4     BYT 1:A
E576 F6 A0 56          LDA B  BUFF+5
E579 A7 01             STA A  1,X
E57B E7 02             STA B  2,X
E57D BD E4 8E          JSR    ADDRDA
E580 CE A0 53          LDX    £BUFF+2
E583 BD E0 C8          JSR    OUT4HS
E586 BD E2 99          JSR    UTMIN2
E589 CE A0 55          LDX    £BUFF+4
E58C BD E0 C8          JSR    OUT4HS
E58F 8D 02     XCHA90  BSR    BEGINC
E591 20 C8             BRA    XCHA30
               *
               *
E593 FE A0 02  BEGINC  LDX    BEGA
E596 08                INX
E597 FF A0 02          STX    BEGA
E59A 7E E4 9D          JMP    BREAK      +RTS
               *
               * PUNCHA MED KLARTEXTRUBRIK
               *
E59D BD E1 DE  HEADER  JSR    BEGENS     START/STOPP
E5A0 CE E2 E9          LDX    £TEXT11    'RUBRIK:'
E5A3 8D 86             BSR    PDATA3
E5A5 CE A0 53          LDX    £BUFF+2    TEMP BUFFER
E5A8 86 04             LDA A  £4
E5AA A7 00             STA A  0,X        LÄGG IN EOT
```

```
E5AC BD E1 AC  HEAD10  JSR     INEEE      TAG IN TECKEN
E5AF 81 1B             CMP A   £$1B       OM ESC SKREV FEL
E5B1 27 EA             BEQ     HEADER
E5B3 81 0D             CMP A   £$D        'CR'
E5B5 27 0F             BEQ     HEAD20     KLAR
E5B7 A7 00             STA A   X          LAGRA I BUFF
E5B9 08                INX
E5BA 86 04             LDA A   £4         EOF-TECKEN
E5BC A7 00             STA A   X
E5BE 8C A0 82          CPX     £BUFF+49   MAX TECKENANTAL
E5C1 26 E9             BNE     HEAD10
                       *
E5C3 7E E0 40  HEAD30  JMP     LOAD19
               *
               * KLAR MED RUBRIKEN
               *
E5C6 BD E0 97  HEAD20  JSR     DC2OUT
E5C9 CE A0 53          LDX     £BUFF+2
E5CC BD E5 2B          JSR     PDATA3     UT MED RUBRIKEN
E5CF 7E E1 43          JMP     PUN11      PUNCHA DATA
               *
               * MINNESTTEST
               *
E5D2           MEMTST  EQU     *
               *
E5D2 BD E1 DE          JSR     BEGENS
E5D5 8D 74             BSR     PCRLFW
E5D7 5F                CLR B
E5D8 37        LOOP    PSH B
E5D9 FE A0 02          LDX     BEGA
E5DC 09                DEX
E5DD 08        INITLP  INX
E5DE E7 00             STA B   X
E5E0 5C                INC B
E5E1 BC A0 04          CPX     ENDA
E5E4 26 F7             BNE     INITLP
               *
               *
E5E6 FF A0 53  BUBBLE  STX     BUFF+2
E5E9 FE A0 02          LDX     BEGA
E5EC FF A0 51  EXCHG   STX     BUFF
E5EF A6 00             LDA A   X
E5F1 FE A0 53          LDX     BUFF+2
E5F4 E6 00             LDA B   X
E5F6 A7 00             STA A   X
E5F8 09                DEX
E5F9 FF A0 53          STX     BUFF+2
E5FC FE A0 51          LDX     BUFF
E5FF E7 00             STA B   X
E601 BC A0 53          CPX     BUFF+2
E604 27 06             BEQ     CHKL
E606 08                INX
E607 BC A0 53          CPX     BUFF+2
E60A 26 E0             BNE     EXCHG
E60C 32        CHKL    PUL A
E60D 36                PSH A
E60E 4A                DEC A
E60F FE A0 04          LDX     ENDA
E612 FF A0 53          STX     BUFF+2
E615 FE A0 02          LDX     BEGA
E618 FF A0 51          STX     BUFF
               *
```

```
E61B FE A0 53  CHECK   LDX     BUFF+2
E61E 4C                INC A
E61F A1 00             CMP A   X
E621 26 30             BNE     ERROR
E623 BC A0 02  CONTIW  CPX     BEGA
E626 27 10             BEQ     GOOD
E628 FE A0 51          LDX     BUFF
E62B 08                INX
E62C FF A0 51          STX     BUFF
E62F FE A0 53          LDX     BUFF+2
E632 09                DEX
E633 FF A0 53          STX     BUFF+2
E636 20 E3             BRA     CHECK
               *
               *
E638 86 21     GOOD    LDA A   £'!
E63A BD E0 75          JSR     OUTCH
E63D BD E4 9D          JSR     BREAK
E640 7A A0 0A          DEC     CKSM
E643 26 02             BNE     PASS
E645 8D 04             BSR     PCRLFW
E647 33        PASS    PUL B
E648 5C                INC B
E649 20 8D             BRA     LOOP
               *
E64B 86 20     PCRLFW  LDA A   £32
E64D B7 A0 0A          STA A   CKSM
E650 7E E1 31          JMP     PCRLF
               *
E653 E6 00     ERROR   LDA B   X
E655 37                PSH B
E656 36                PSH A
E657 BD E1 31          JSR     PCRLF
E65A B6 A0 53          LDA A   BUFF+2
E65D BD E0 C2          JSR     OUT2HT
E660 B6 A0 54          LDA A   BUFF+3
E663 8D 0D             BSR     WUT2HS
E665 32                PUL A
E666 16                TAB
E667 8D 09             BSR     WUT2HS
```

```
E669 32                PUL A
E66A 8D 06             BSR   WUT2HS
E66C 17                TBA
E66D FE A0 53          LDX   BUFF+2
E670 20 B1             BRA   CONTIW
                *
E672 BD E0 C2   WUT2HS JSR   OUT2HT
E675 86 20             LDA A £$20
E677 7E E0 75   WOUTCH JMP   OUTCH
                *
                * MÄT RAMSTORLEK
                *
E67A            METRAM EQU   *
E67A CE E7 E6          LDX   £STRRAM
E67D BD E5 2B          JSR   PDATA3
E680 CE 00 00          LDX   £0
E683 08         MET10  INX
E684 86 55             LDA A £$55
E686 E6 00             LDA B 0,X
E688 A7 00             STA A 0,X
E68A A6 00             LDA A 0,X
E68C E7 00             STA B 0,X
E68E 81 55             CMP A £$55
E690 27 F1             BEQ   MET10
                *
E692 09         MET80  DEX
E693 FF A0 0C          STX   XHI        SPARA MAXADRESSEN
E696 CE A0 0C          LDX   £XHI
E699 8D 6B             BSR   AUT4HS
E69B 20 4C             BRA   S9P95
                *
                * DISPLAY BREAKPONITS
                *
E69D F6 A0 41   DISBRK LDA B VFLAG      HÄMTA ANTAL
E6A0 27 0C             BEQ   DISB00
E6A2 CE A0 42          LDX   £BPTAB     HÄMTA TABELLEN
E6A5 BD E0 C8   DISB50 JSR   OUT4HS
E6A8 08                INX              HOPPA ÖVER OPKOD
E6A9 5A                DEC B
E6AA 26 F9             BNE   DISB50
E6AC 20 3B             BRA   S9P95
                *
E6AE 86 30      DISB00 LDA A £$30       SÄND EN NOLLA
E6B0 8D C5             BSR   WOUTCH
E6B2 20 35             BRA   S9P95
                *
                * SÖK 2 BYTES ADRESS
                *
E6B4 BD E1 EC   SOK2B  JSR   BEGENT
E6B7 CE E7 DD          LDX   £TEXT12+3 "ADRESS"
E6BA BD E5 2B          JSR   PDATA3
E6BD BD E0 47          JSR   BADDR      FIXA 4 TECKEN
E6C0 FE A0 02          LDX   BEGA
E6C3 BD E4 BE   SOK2B4 JSR   CBEND1
E6C6 A6 00             LDA A 0,X        KOLLA FÖRSTA BYTEN
E6C8 B1 A0 0C          CMP A XHI
E6CB 26 14             BNE   SOK2B7
E6CD E6 01             LDA B 1,X
E6CF F1 A0 0D          CMP B XHI+1
E6D2 26 0D             BNE   SOK2B7
E6D4 BD E1 31          JSR   PCRLF
```

Forts från sid 55

MONITOR
ETT PROGRAM ELLER EN BILDSKÄRM, PROGRAMMET ANVÄNDS FÖR ATT HANTERA EN DATORS GRUND- FUNKTIONER. (EX TBUG2C I DETTA NUMMER).

RECORD
EN RECORD ÄR EN BENÄMNING PÅ EN INPELNING UTAV EN BEGRÄNSAD LÄNGD (TEX. SEKTOR). MED RECORD- FILER ELLER SK. RANDOM -FILER MENAS EN FIL VARS DATA ÄR ORDNAT SÅ ATT MAN LÄTT KAN NÅ EN 'RECORD' MED ANGIVANDE UTAV ETT NUMMER. DET ÖKAR SNABBHETEN I ATT HANTERA DATA.

```
E6D7 CE A0 02          LDX    £BEGA      HÄMTA ADRESSEN
E6DA 8D 2A             BSR    AUT4HS
E6DC CE A0 0C          LDX    £XHI
E6DF 8D 25             BSR    AUT4HS     LÄGG UT DATAT
E6E1 BD E5 93 SOK2B7   JSR    BEGINC
E6E4 20 DD             BRA    SOK2B4
              *
E6E6 BD E5 2B S9PAUS   JSR    PDATA3
E6E9 7E E0 D5 S9P95    JMP    CONTRL
              *
              * FORTSÄTTNING PÅ REG-PRINT
              *
E6EC 86 41    PRREG9   LDA A  £'A
E6EE 8D 19             BSR    UTMIN8
E6F0 BD E0 CA          JSR    OUT2HS     A-ACK
E6F3 86 58             LDA A  £'X
E6F5 8D 12             BSR    UTMIN8
E6F7 8D 0D             BSR    AUT4HS     X-REG
E6F9 86 50             LDA A  £'P
E6FB 8D 0C             BSR    UTMIN8
E6FD 8D 07             BSR    AUT4HS
E6FF 86 53             LDA A  £'S
E701 8D 06             BSR    UTMIN8
E703 CE A0 08          LDX    £SP
E706 7E E0 C8 AUT4HS   JMP    OUT4HS     SP +RTS
              *
E709 7E E2 97 UTMIN8   JMP    UTMIN
              *
              *
E70C C6 10    DUMP16   LDA B  £16
E70E F7 A0 12          STA B  DMASK
E711 BD E1 DE          JSR    BEGENS     FIXA START & STOPP
E714 B6 A0 03          LDA A  BEGA+1
E717 84 F0             AND A  £$F0       FULL RAD
E719 BD E3 89          JSR    PDUMP5
E71C 86 08             LDA A  £8
E71E B7 A0 12          STA A  DMASK
E721 20 C6             BRA    S9P95
              *
E723 86 7E    BOOT0    LDA A  £$7E       FIXA ÅTERSTART
E725 B7 24 00          STA A  $2400
E728 CE E7 33          LDX    £BOOT2
E72B FF 24 01          STX    $2401
E72E C6 D0             LDA B  £$D0       FORCE INTERRUPT
E730 F7 80 18          STA B  COMREG
E733 4F       BOOT2    CLR A
E734 CE 70 00          LDX    £$7000
E737 B7 80 14 BOOT3    STA A  DRVREG     DRIVE £0 MOTOR ON
E73A 08                INX
E73B 26 FA             BNE    BOOT3
E73D C6 0F    BOOT4    LDA B  £$0F       RESTORE
E73F F7 80 18          STA B  COMREG
E742 8D 31             BSR    BOTRTS
E744 F6 80 18 LOOP1    LDA B  COMREG
E747 C5 01             BIT B  £1
E749 26 F9             BNE    LOOP1
E74B 7F 80 1A          CLR    SECREG
E74E 8D 25             BSR    BOTRTS
E750 C6 9C             LDA B  £$9C       READ W LOAD
E752 F7 80 18          STA B  COMREG
E755 8D 1E             BSR    BOTRTS
E757 CE 24 00          LDX    £$2400
```

```
E75A C5 02     LOOP2   BIT B  £2         DRQ?
E75C 27 06             BEQ    LOOP3
E75E B6 80 1B          LDA A  DATREG
E761 A7 00             STA A  0,X
E763 08                INX
E764 F6 80 18  LOOP3   LDA B  COMREG
E767 C5 01             BIT B  £1         BUSY ?
E769 26 EF             BNE    LOOP2
E76B F6 80 18          LDA B  COMREG     CRC FEL ?
E76E C4 0C             AND B  £$0C
E770 26 CB             BNE    BOOT4
E772 7E 24 00          JMP    $2400
               *
E775 8D 00     BOTRTS  BSR    BOOTRT
E777 39        BOOTRT  RTS
               *
E778           SWIINS  EQU    *
E778 BD E0 47          JSR    BADDR
E77B           SWIIN2  EQU    *
E77B FE A0 0F          LDX    SWIVEK
E77E FF A0 13          STX    SWIFLX
E781 CE E2 F1          LDX    £SFE       LÄGG IN TBUGS VEKTOR
E784 FF A0 0F  SWI90   STX    SWIVEK
E787 39                RTS
               *
E788 FE A0 13  FLEX0   LDX    SWIFLX     LÄGG TILLBAKS
E78B FF A0 0F          STX    SWIVEK
E78E 7E AD 03          JMP    $AD03      GÅ TILL FLEX VARMSTART
               *
E791 53        TEXT02  FCC    'Start Slut Till'
E7A0 0D                FCB    $D,$A,'§,4
               *
E7A4 45        TEXT03  FCC    'Ej ändrat:'
E7AE 04                FCB    4
               *
E7AF 53        TEXT04  FCC    'Start Slut'
E7B9 0D                FCB    $D,$A,'§,4
               *
E7BD 54        TEXT05  FCC    'Tecken: '
E7C5 04                FCB    4
               *
E7C6 53        TEXT09  FCC    'Subtrahera (00-FF):'
E7D9 04                FCB    4
               *
E7DA 4E        TEXT12  FCC    'Ny Adress: '
E7E5 04                FCB    4
               *
E7E6 52        STRRAM  FCC    'Ram slut='
E7EF 04                FCB    4
               *
               * VEKTORER
               *
E7F8                   ORG    $E7F8
               *
E7F8 E0 00             FDB    IO         IRQ VEKTOR
E7FA E0 0A             FDB    SWISER     SWI VEKTOR
E7FC E0 05             FDB    POWDWN     NMI VEKTOR
E7FE E3 CE             FDB    CSTART     RESTART VEKTOR KALL START
               *
                       END

NO ERROR(S) DETECTED
```

FLEX-DISK-BESKRIVNING

JAG HAR UNDER EN TID FÖRSÖKT ATT STUDERA HUR EN FLEX-DISKETT ÄR UPP-BYGGD, OCH DET HÄR ÄR VAD JAG HAR KOMMIT FRAM TILL. JAG HOPPAS ATT JAG INTE MISSAT ALLTFÖR MYCKET, MEN ÄR DET PÅ DET VISET SÅ HOPPAS JAG ATT NÅGON VÄNLIG SJÄL VILL TALA OM VAD JAG HAR MISSAT.

SPÅR 00 BESKRIVNING :

SEKTOR 0=(6800),1=(6809) :
  DETTA ÄR FLEX BOOT LADDARE
  BYTE 0 :
    START ADRESS $C000
    EFTER NED HÄMTNING
    AV MONITORNS BOOTL.
  BYTE 1-4:
    DATA VARIABLER
  BYTE 5,6 = DEN FÖRSTA SEKTORN (LINK-ADRESS) FÖR FLEX-OPERATIV.SYS-FILEN
  BYTE 7-255 :
    BOOT PROGRAM
  BYTE 256-
    VARIABLER
    UTANFÖR SEKTORN.

SEKTOR 2 :
  OANVÄND

SEKTOR 3 :
  SYSTEM INFO RECORD
  SE NEDAN!

SEKTOR 4 :
  OANVÄND!

SEKTOR 5-SLUT :
  DIRECTORY SEKTORER
  VARJE SEKTOR INNEHÅLLER
  10 ST DIRECTORY-RECORDS
  ENLIGT NEDAN. OM FILNAMNEN
  ÖVERSKRIDER 50 ST TAGES EN
  ELLER FLERA LEDIGA SEKTORER
  I ANSPRÅK FÖR DIRECTORYT.

SPÅR 1-SLUT BESKRIVNING :

SEKTOR 1-SLUT :
  FIL-DATA SEKTORER
  BYTE 0-1:
    LÄNK ADRESS TILL
    NÄSTA I FILEN FÖRE-
    KOMMANDE SEKTOR. =0
    OM SLUT PÅ FILEN.
  BYTE 2-3:
    RECORD NUMMER (1-$FFFF)
    =0 OM ALDRIG ANVÄND
  BYTE 4-255:
    ANVÄNDAR-DATA

BÖRJAN AV SEKTOR 00-03

```
00  00 00 00 00 00 00 00 00
08  00 00 00 00 00 00 00 00

10  44 49 53 4B 4E 41 4D 4E  DISKNAMN
    (----------1----------)
18  00 00 00 00 02 08 07 06
                (-2-) (-3-) (-
20  0A 01 3A 0B 0B 50 22 0A
    4) (-5-) (--6---) (-7-)
```

1/ NAMNET PÅ DISKEN (MAX 8 CHAR.).

2/ DISKENS VERSIONS-NUMMER (0-65535).

3/ DISK-ADRESSEN FÖR DEN FÖRSTA LEDIGA SEKTORN PÅ DISKEN (SPÅR,SEKTOR).

4/ DISK-ADRESSEN FÖR DEN SISTA LEDIGA SEKTORN PÅ DISKEN (SPÅR,SEKTOR).

5/ ANTALET LEDIGA SEKTORER PÅ DISKEN (0-65535)

6/ DATUMET FÖR SKAPANDET UTAV DISKEN (MÅN,DAG,ÅR)

7/ DEN HÖGSTA DISK-ADRESSEN SOM FÅR FÖREKOMMA PÅ DISKEN.

BÖRJAN AV SEKTOR 00-05

```
00  00 06 00 00 00 00 00 00
    (-1-)
08  00 00 00 00 00 00 00 00

10  46 49 4C 4E 41 4D 4E 31  FILNAMN1
    (----------2----------)
18  42 49 4E 00 00 01 01 01  BIN
    (--3---)          (-4-) (-
20  01 00 01 00 00 0B 0B 50
    5) (-6-) (7)    (--8---)

28  46 49 4C 4E 41 4D 4E 32  FILNAMN2
    (----------2----------)
30  54 58 54 00 00 01 02 01  TXT
    (--3---)          (-4-) (-
38  02 00 01 00 00 0B 0B 50
    5) (-6-) (7)    (--8---)
```

1/ DETTA ÄR LÄNKEN TILL NÄSTA BIBLIOTEKS SEKTOR (DIRECTORY-SEKTOR).

2/ FILENS NAMN (8 CHAR MAX). FÖRSTA BYTEN BIT 7 =1 OM DEL. FÖRSTA BYTEN =0 OM SISTA DIR. INFO RECORDEN.

3/ FILENS 'EXTENSION' (3 CHAR.).

4/ FILENS FÖRSTA DISK ADRESS (TRACK,SEKTOR).

5/ FILENS SISTA DISK ADRESS (TRACK,SEKTOR).

6/ ANTALET SEKTORER I FILEN.

7/ FIL-TYP (RECORD =02).

8/ DATUMET FÖR SKAPANDET UTAV FILEN (MÅNAD DAG ÅR).

VAD ÄR EN SEKTOR?

SEKTOR = EN 'TÅRT-BIT' UTAV DISKETTEN. SEKTORERNA PÅ EN DISK KAN VARA ANTINGEN MJUK - ELLER HÅRDVARU FORMATERADE. DET INNEBÄR ATT MAN I HÅRDVARU-FALLET HAR EN HEL MASSA HÅL I DISKETTEN DÄR VARJE HÅL MOTSVARAR EN SEKTORS BÖRJAN. DESSUTOM HAR MAN ETT TILL HÅL SOM TALAR OM VAR FÖRSTA SEKTORN FINNS. I MJUKVARU FALLET SÅ HAR MAN BARA ETT HÅL SOM TALAR OM VAR SPÅRET BÖRGAR, OCH SEDAN SÅ ÄR HELA SPÅRET EN SERIE UTAV INSPELNINGAR GRUPPERADE SÅ ATT DE BILDAR SEKTORER. MAN TALAR ISTÄLLET OM VAR VARJE SEKTOR BÖRGAR MED EN SPECIELL INSPELNING PÅ SKIVAN SOM BARA INNEHÅLLER INFO. OM VILKEN SEKTOR SOM KOMMER EFTER INSPELINGEN. DEN INSPELNINGEN KALLAS ÄVEN ADRESS-MARK (AM), OCH TJÄNSTGÖR SOM ETT HÅL PÅ DISKEN. ETT SPÅR PÅ DISKEN KAN SE UT SÅ HÄR:

§HÅL§GAP§IM§GAP
§AM§CRC§GAP§DAM§GAP§DATA§CRC
§AM§CRC§GAP§DAM§GAP§DATA§CRC
OSV. DÄR:

HÅL = HÅLET PÅ DISKEN.
GAP = TOMMRUM FÖR ATT KONT-ROLLERN (EX. 1771) SKA KUNNA HINNA MED. ÄVEN SYNKRONISERING SKER HÄR.
IM = INDEX MARK, MARKERAR BÖRGAN AV SPÅRET (AN-VÄNDES OFTAST EJ I 5" DISKAR.
AM = ADRESS MARK DVS. ETT INFO. OM VILKET SEKTOR-NUMMER SOM KOMMA SKALL.
CRC = EN 16-BIT CHECK-SUMMA FÖR KONTROLL AV ATT DATA LÄST I AM ELLER DATA ÄR OK.
DAM = DATA ADRESS MARK, MARKERAR BÖRGAN UTAV INSPELAT DATA, INNEHÅLLER ÄVEN UPPGIFT OM ANTAL BYTES I SEKTORN.
DATA = SEKTORNS DATA (I FLEX 256 BYTES).

ALL DENNA INFORMATION BEHÖVS FÖR ATT HÅLLA REDA PÅ DATA ORDNINGEN PÅ DISKEN. DEN SKRIVS IN PÅ DISKEN NÄR DU KÖR NEWDISK (SK. FORMATERING.).

LU

ORDLISTAN

BOOT-LOADER
ÄVEN 'BOOT STRAP LOADER', ETT PROGRAM SOM FINNS I DIN MONITOR, SOM LADDAR NER ETT SK. BOOT-PROGRAM OCH STARTAR DETTA. BOOT- PROGRAMMET FINNS PÅ SPÅR 0 SEKTOR 0 (FÖR 6809 SEKTOR 1). DETTA PROGRAM LADDAR I SIN TUR NER HELA OPERATIV- SYSTEMET.

DIRECTORY
EN DATA- AREA SOM HÅLLER REDA PÅ FILERS START & STOP ETC.

EXTENSION
EN I FLEX 3 BOKSTÄVER LÅNG FIL-TYPS ANGIVELSE. EN FIL KAN VARA TEXT- ORINETERAD ELLER BINÄR- ORIENTERAD (.TXT .BIN) DET KAN VARA ETT BASIC PROGRAM (.BAS) DU KANN NOG KOMMA PÅ EGNA EXTENSIONS OM DET BEHÖVS.

FLEX
ETT DISK OPERATIV- SYSTEM IFRÅN USA FRÅN FIRMAN TSC (TECHNICAL SYSTEMS CONSULTANTS INC)

LINK ADRESS
EN ADRESS SOM TALAR OM VAR (PÅ DISKEN) INFORMA- TIONEN FORTSÄTTER VID EN FIL UPP-BYGGNAD.

Forts. sid 51