

PROGRAM

nittinian

BITEN

99-9

Nya kort

MATRIKEL
NUMMER

PRK på djupet

Innehåll

Ordföranden	3
Rapid Copy	4
PRK-BASIC	5-8
Nya kort	8
Tips och tricks	9-10
Bokrecension	10
Matrikeln 1988	11-14
Utmaningen	15-17
Programbanken	18-19
Hårddiskkort	19
Dubbelspelet	20-21
Satellitbanor	22
Hårdvarumanual	22
Tips from the Tigercub	23-24

ISSN 0281-1146

Redaktören ...

5 år sedan "svarta fredagen"

En fredag i oktober 1983 kom meddelandet från Texas Instruments: Vi lägger ner tillverkningen av hemdatorer. 99:an skulle snart vara död trodde många. Trots att så lång tid gått sedan dess kan vi idag konstatera att fler och bättre program och tillbehör än någonsin tidigare tas fram för 99:an. I detta nummer presenterar vi några av de nya kort för expansionsboxen som kommit de senaste månaderna.

Föreningens telefon

Vi har flyttat föreningens telefon till ett nytt nummer. Det går hem till Peter Olsson som sitter med i förningsstyrelsen. Avsikten är att vi inom kort ska ha en databas igång på det telefonnumret. På de tider när modemmet inte är inkopplat kommer Peter ibland att svara själv. Du kan ställa frågor om föreningen och datorn till honom. Frågor han inte kan svara på själv kan ha vidarebefordra till andra. Telefonnumret är 0764/626 03.

Sänkt pris på freeware

För att uppmuntra till ökade inköp av freeware från medlemmarna har styrelsen beslutat att fr o m nu sänka priset per diskett till 30 kronor. Avsikten är ju inte att föreningen ska tjäna pengar på freeware. Däremot tar ju programförfattarna gärna emot bidrag. I detta nummer har vi en komplett lista över vad som finns. Vi har fått in hel del nyheter sen sist.

Matrikeln

I detta nummer publicerar vi föreningens medlemsmatrikel. Som vanligt vill vi uppmäna alla att titta efter om det finns några i närheten som Du inte har haft kontakt med. Ring upp och hör efter — kanske kan ni utbyta erfarenheter.

Möte i Helsingborg

Den 8:e oktober höll våra skånska kamrater en höstträff i Helsingborg. 3 danska vänner hade tagit sig över sundet också. En nyhet som de kunde berätta var att det finns 7 ägare av Geneve-datorer i Danmark.

Databas i Göteborg

Föreningen West 99 i Göteborg har nu startat en helt egen databas. Den körs på en 99:a med RAM-diskar. Man kan kommunicera med den både på 300 och 1200 baud. Första gången man ringer får man en begränsad behörighet, men sedan man blivit godkänd får man större möjligheter. Telefonnumret är 031/917004 och basen är normalt öppen mellan 17.00 och 23.00. Systemoperatör är Sten Gunnarsson.

Tips från Jim Peterson

Som utfyllnad har vi i det här numret lagt in lite material från Jim Peterson i USA. Hans "Tips from the Tigercub" är sedan flera år välkända för Texas-ägare över hela världen. De publiceras av många olika användarföreningar.

K-TRYCK AB

Söderarmsvägen 44 ☎ 59 29 00
121 54 Johanneshov ☎ 48 66 30

I redaktionen:

Redaktör	Peter Odelryd
Utmaningsredaktör	Anders Persson
Programförmedlare	Börje Häll
	Claes Schibler

Föreningens och redaktionens adress:

Föreningen Programbiten
c/o Schibler
Wahlbergsgatan 9, nb
121 46 Johanneshov

Föreningens telefon: 0764/626 03

Datainspektionens licensnummer: 82100488

Postgiro: 19 83 00-6

Medlemsavgiften för 1988 är 120:—

Annonser, insatta av enskild medlem (ej företag), som gäller försäljning av moduler eller andra tillbehör i enstaka exemplar är gratis.

Övriga annonser kostar 400 SEK per helsida, 200 SEK för halvsida. För lösblad som skickas med tidningen gäller 400 SEK per blad.

För kommersiellt bruk gäller följande:

Månfaldigande av innehållet i denna skrift, helt eller delvis, är enligt lag om upphovsrätt av den 30 december 1960 förbjudet utan medgivande av Föreningen Programbiten. Förbudet gäller varje form av månfaldigande genom tryckning, duplicering, stencilering, bandinspelning, diskettinspelning etc.

Föreningens tillbehörsförsäljning

Följande finns att köpa för medlemmar genom att motsvarande belopp insätts på postgiro 198300-6.

Användartips med Mini Memory	20:—
Nittinian T-tröja (se annons i detta nummer!)	40:—
99'er magazine nr 12/82, 1-5, 7-9/83 (per styck)	40:—
Gamla årgångar av Programbiten	50:—
TI-Forth manual	100:—

Ny Fairware

Följande nya skivor kan beställas från programbanken:
— FUNNELWEB 4.11 med en mycket förbättrad DM 1000 (2 skivor)
— TELCO 2.1 för datakommunikation (2 skivor)
— ARCHIVER 3.02 nu med bättre packning och packning/kompression i samma steg. Kompatibel med version 2.4.
— DISK UTILITIES 4.12 utan manual.
— MIAMI-BOOT ett menyprogram för XB + 32 K som liknar Horizons MENU. Menyn editeras med (BEGIN).

Home Computer Aktiv

Den tyska tidningen HCA (Home Computer Aktive) som även innehöll TI-Revue har slutat att komma ut. Det sista numret blev 8/88.

Kallelse till medlemsmöte

Lördagen den 26 november 1988 kl 13.00

Militärhögskolan, Valhallavägen 117, Stockholm. T-bana Stadion. Ingång från baksidan.

Vi kommer att ha minst en TI-99/4A med expansionsbox igång. Du som har något att visa eller har önskemål om att få se något speciellt bör höra av dig i god tid före mötet. Följande kommer vi att kunna visa:

- TI-BASE ett nytt databasprogram för telefonregister, skivsamlingar mm.
- EZ-KEYS ett nytt sätt att använda Extended BASIC. Det finns makroinstruktioner och mycket annat som underlättar programmering.
- Datakommunikation med TI-99/4A och *Telco, Fast-Term* mm.
- Några nya freeware kommer vi att kunna visa. Hör av dig i förväg om du har speciella önskemål.
- Det kommer att finnas möjlighet att kopiera några flexskivor som ej finns med i programbanken under mötet. Passa på och kom och utnyttja denna möjlighet. Ta med egna skivor.

Ordföranden ...

Antalet betalande medlemmar i föreningen har minskat till 120. Föreningen kommer att gå med förlust även i år. Det betyder att vi måste minska utgifterna genom att göra tidningen billigare men den kommer att komma ut fyra gånger per år även i fortsättningen. Passa på och beställ äldre nummer av tidningen eller program ur programbanken som ett stöd för vår ekonomi. Du som vill ordna ett lokalt möte måste se till så att kallelsen kan följa med ett ordinarie tidningsnummer.

Det är nu ännu viktigare än tidigare att alla medlemmar bidrar på något sätt genom att skriva artiklar, program eller ställa frågor som andra kan besvara. Om du har något specialintresse sätt in en gratisannons i Programbiten.

Vid köp av program bör du hålla dig till säljare som är seriösa både mot dig som köpare och mot programförfattaren. Från en mindre seriös säljare får du ofta ett piratkopierat program med en dåligt kopierad manual (om den inte saknas helt). Nästan alla amerikanska program har originalmanualen med mycket bra tryck som skiljer sig från fotokopiering. Jag har själv fått ett sådant piratkopierat program från en säljare där filerna på skivan var blandade på ett obegripligt sätt eftersom två program bytt namn med varandra. Säljaren hade tydligen ej kopierat originalskivan.

Det är viktigt att alla programförfattare får betalt för sitt jobb annars kommer de ej att skriva några nya program och 99:an att försvinna på grund av programtorka. Detta gäller även FAIRWARE(freeware)-författare som du bör sända 5–10 dollar till per program du använder.

Säljare som enligt min och andras erfarenhet är seriösa är TENEX och TEXAMENTS. TENEX har ett mycket stort heltäckande sortiment medan TEXAMENTS har specialiserat sig på ett fåtal saker. Jag har ej undersökt övriga firmor så noga att jag här vill peka ut vilka som är mindre seriösa.

Om program som ursprungligen har kommit ut på modul säljs på skiva så kan man misstänka piratkopiering. Det är inte troligt att detta är godkänt. FREEWARE får ej säljas med vinst eller levereras gratis tillsammans med någon annan produkt.

Föreningen Programbiten har ej gett tillstånd till försäljning av program ur programbanken till någon annan. Detta gäller både enstaka program och hela skivor som PB-FORTH, FORTH&99, FORIT och PERFECT PUSH. I princip har författaren möjlighet att sälja eller skänka bort kopieringsrätten. Jag har ej undersökt om detta har förekommit.

Kom gärna med frågor till föreningens adress eller till mig personligen. Jag svarar gärna på frågor även på telefon.

Jan Alexandersson
Springarvägen 5, 3 tr
S-142 61 TRÅNGSUND

Rapid scroll

av Anders Josefsson

Jurgen Switalski har skrivit *Rapid Scroll* som är ett program för läsning av textfiler i D/V 80-format. Och, som namnet antyder kan man scrolla eller bläddra i texten lite hur man vill. Det är ett mycket smidigt program när man ska läsa sina textfiler. Man scrollerar med piltangenterna, och texten rullar snabbt åt det håll man vill. Precis så snabbt att man kan skumma i texten om man letar efter något speciellt stycke.

Diskinnehåll

Rapid Scroll-disken innehåller också en fil så Du kan styra scrollningen med joystick! Då rullar texten något ryckigare men det är ju ganska smidigt att sitta med joystick.

Mera godis på disken! Menumaker, ett program Du kan använda till dina E/A-diskar. Filen heter Create-menu och den talar om vad Du ska göra. Du skriver in alla filerna (både program och D/F-80) och programmet sparar en fil på den disken i D/V 80-format som heter

PROGRAMS. Sedan finns en fil LOADMENU, som också läggs på nämnda disk. Den kan du sedan döpa om till UTIL, så händer allt automatiskt. Rutinen hämtar alla filnamnen, och ställer upp en meny och det är bara för dig att välja!

Sedan har vi SMANAGER, ett katalogprogram. Inga direkta konstigheter alls. Ett mycket enkelt katalogprogram som laddas in med E/A option 3. Jag har inte kollat, men det kan tänkas att det kan laddas in med XBasic-modulen med ett CALL LOAD. Tänkbart.

Sist men inte minst finns MAIL-LIST på disken. Ett XBasic-program med assembler-rutiner. Som namnet antyder är det ett registerprogram för adresser. Tyvärr amerikanskt, så det stämmer inte riktigt med postens anvisningar för hur en adress ska se ut. Men det har den snabbaste sorteringsrutinen jag har sett! Ett blink på skärmen, så är det klart. Och du kan sortera vad Du vill, adress, namn, postnummer, flagga. Flaggan bestämmer Du själv, kan vara t ex klubbnamn, ort eller vad Du vill.

Samtliga program är utvecklade av ovan nämnde Mr Switalski. Han nämner att dessa program var en del av hans utveckling i assembler-programmering. Rena övningar, alltså! Vad månader bliva av sådana grejor?

Alla program är dokumenterade på disken. Min slutliga bedömning blir mycket bra, en disk med mycket nyttiga program.

Betygen kommer här:

Utförande *****
Dokumentation *****
Användarvänlighet *****

Vänliga hälsningar från klubben West 99 Göteborg

Rapid Copy

av Jan Alexandersson

Att kopiera flexskivor kan vara ganska tidskrävande om man har många skivor och filer. Den *Disk Manager 2* som följer med diskkontrollkortet kopierar alltid en enstaka fil åt gången vilket kan kännas ganska jobbigt om du endast har en diskdrive. Med två diskdrivar får du en mycket stor förbättring.

Nästa stora nyhet var *Disk Manager 1000* (finns i programbanken) som kopierar hela skivan utan hänsyn till antalet filer. Detta görs i fyra pass för SS/SD något som du även har nytta av med endast en drive.

Det finns nu ett program kallat *Rapid Copy* av Barry Boone som säljs via Texaments för USD 15. Detta är fyra gånger snabbare än DM 1000 om man använder det på rätt sätt. Med flexskivan följer en 8 sidig manual. På samma skiva finns programmet i tre versioner för TI-, Corcomp- och Myarc-kontrollkort. Skivan innehåller även Myarc Disk Manager III som endast är avsedd för Myarc-kortet men många av rutinerna fungerar även med mitt TI-kort. Rapid Copy kräver 32 kb expansionsminne och kan laddas med Extended Basic eller någon laddare för maskinkod i PROGRAM-format som t ex Editor/Assembler, TI-Writer eller Funnelweb.

Formatera diskar

Det finns ett flertal olika program som kan formatera diskar. Med det TI-kontrollkort som jag har kan resultat bli på tre olika sätt som ej kan tas bort med mindre än att skivan formateras om. En sektoreditor hjälper inte i detta fall.

En SS/SD-skiva är uppdelad på 40 spår som vardera innehåller 9 sektorer. Dessa 9 sektor är inte lagrade i nummerordning i spåret utan närliggande sektornummer har ett bestämt avstånd kallat interlace som normalt är tre eller fyra.

Med hjälp av *Disk Hacker* (finns i programbanken) av Will McGovern kan man ta reda på interlace och mycket annat. Det visade sig att nästan alla program gav samma resultat som DM 2:

Sida 0 Spår 0 Sektor 0-7-5-3-1-8-6-4-2
Sida 0 Spår 1 Sektor 6-4-2-0-7-5-3-1-8
Sida 0 Spår 2 Sektor 3-1-8-6-4-2-0-7-5
(samma mönster upprepas för spår 3-5 osv)
Sida 1 Spår 0 Sektor 7-5-3-1-8-6-4-2-0
(detta spår är unikt)
Sida 1 Spår 1 Sektor 3-1-8-6-4-2-0-7-5
Sida 1 Spår 2 Sektor 6-4-2-0-7-5-3-1-8
Sida 1 Spår 3 Sektor 0-7-5-3-1-8-6-4-2
(samma mönster upprepas för spår 4-6 osv)

Om man tänker på att ett spår är en cirkel så ser man att det alltid är ett avstånd på fyra mellan sektorer med närliggande nummer dvs interlace = 4. Du ser även att spårerna startas något olika trots att de har samma interlace. På baksidan kommer dessa i omvänd ordning. Spår 0 på baksidan verkar vara något udda. De program som formaterar på detta sätt är DM 2, DM 1000, Disk Utilities och TI-Base.

Rapid Copy kan ställas in för valfritt interlace mellan 1 och 8 men för mitt kontrollkort är det endast 3 och 4 som är användbara.

Advanced Diagnostics och Rapid Copy med interlace 4 ger följande:

alla spår Sektor 0-7-5-3-1-8-6-4-2

Rapid Copy med interlace 3 ger detta:

alla spår Sektor 0-3-6-1-4-7-2-5-8

En titt på sektor 0 visar att samtliga formaterare ger samma sektor 0 utom Disk Utilities som även lagrar datum i sektor 0.

Tid för formatering

När du formaterar en skiva så går det ofta att välja om programmet skall verifiera att det inte finns felaktiga sekto-

rer. DM 2 har detta obligatoriskt. En test av formatering gav följande resultat för SS/SD 90 kb:

Disk Manager 2	53 sek
DM 1000 med verify	53 sek
DM 1000 utan verify	19 sek
Disk Utilities med verify	54 sek
Disk Utilities utan verify	19 sek
TI-BASE	21 sek
Advanced Diagnostics	59 sek
Rapid Copy med verify	26 sek
Rapid Copy utan verify	18 sek

En DS/SD 180 kb skiva ger följande:

Disk Manager 2	1 min 44 sek
Rapid Copy med verify	50 sek
Rapid Copy utan verify	34 sek

Tid för kopiering

Jag har jämfört kopiering med DM 1000 och Rapid Copy. Det är lätt att lura sig att ställa in ett för lågt interlace som man tror skall snabba upp kopieringen men som istället gör att kopieringen blir mycket långsammare än nödvändigt. Om du använder DM 1000 så blir inte skillnaden så stor men Rapid Copy mår inte bra av skivor formaterade med fel interlace. Jag har använt tre olika skivor:

- Formaterade med DM 1000
- Formaterade med Rapid Copy Interlace 4
- Formaterade med Rapid Copy Interlace 3

Alla kombinationer av dessa tre som original och kopia har provats dvs 9 olika fall.

Kopiering med DM 1000 sector copy 90 KB

----- KOPIA -----				
O	DM1000	Interl 4	Interl 3	
R	DM1000	2 m 24 s	2 m 30 s	2 m 22 s
I	Interl 4	2 m 30 s	2 m 35 s	2 m 26 s
G	Interl 3	2 m 22 s	2 m 27 s	2 m 18 s

Kopiering med Rapid Copy Verify 90 KB

----- KOPIA -----				
O	DM1000	Interl 4	Interl 3	
R	DM1000	32 s	37 s	1 m 01 s
I	Interl 4	37 s	41 s	1 m 06 s
G	Interl 3	1 m 24 s	1 m 22 s	42 s

Som du ser är inte DM 1000 så känslig för hur skivorna är formaterade och den vanliga formateringen som DM 1000 kan ordna själv duger mycket väl. Med Rapid Copy gäller det att ha samma interlace på både original och kopia. Det är något märkligt att Rapid Copy fungerar bäst med skivor formaterade av DM 1000.

Kopiering med DM 1000 sector copy 180 KB

----- KOPIA -----				
O	DM1000	Interl 4	Interl 3	
R	DM1000	4 m 45 s		
I	Interl 4		5 m 06 s	
G	Interl 3			4 m 35 s

Kopiering med Rapid Copy Verify 180 KB

----- KOPIA -----				
O	DM1000	Interl 4	Interl 3	
R	DM1000	1 m 10 s	1 m 05 s	2 m 03 s
I	Interl 4	1 m 10 s	1 m 05 s	2 m 02 s
G	Interl 3	2 m 36 s	2 m 34 s	1 m 05 s

Om man struntar i verify går kopieringen av 180 kb på 50 sek. Det är även möjligt att låta Rapid Copy formatera och kopiera samtidigt. Det tar då totalt 1 min 39 sek med verify av 180 kb och 1 min 22 sek utan verify. Om originalet är från DM 1000 så blir kopian enligt Rapid Copy Interlace 4.

Disk Manager 2 som kopierar filer kan kopiera en 180 kb-skiva som t ex Funnelweb på 9 m 51 s.

En underlighet som jag har märkt är att antalet spår endast räknas från >00 t o m >27 beroende av om 90 kb- eller 180 kb-skiva kopieras men kopian blir alltid riktigt.

Personal Record Keeping BASIC

av Jan Alexandersson

Om du har Personal Record Keeping modulen så har du även tillgång till ett stort antal nya CALL-kommandon:

- CALL D display
- CALL A accept
- CALL P prep
- CALL L load
- CALL S save
- CALL H header
- CALL G getput

Alla dessa kommandon med syntax finns beskrivna i Programbiten 85-2 med vissa tillägg i PB 85-4. Du kan med CALL A och D skriva och läsa var som helst på skärmen. CALL S och L kan spara och ladda en datafil mycket snabbare än den vanliga filhanteringen med OPEN och CLOSE. Detta är speciellt viktigt för dig med kassetbandspelare.

Jag har nu omvandlat modulens hela styrprogram till BASIC. Det blev ett 20 kb långt program som tar 81 sektorer på en flexskiva. Detta är för långt för att användas i BASIC men jag har delat upp det i några separata program som är fullt körbara i PRK-BASIC. Det blev ett CREATE-program som är 4 kb långt och ett EDIT-program som är 8 kb långt. Dessa program uppför sig som modulen med samma skärmbilder och programstyrning.

En överraskning var att det finns ytterligare 9 CALL-kommandon utöver de 7 som har beskrivits tidigare. Dessa har namn med mycket låga ASCII-koder mellan 4 och 12 dvs hexadecimalt >04 till >0C. Detta går inte att knappa in från tangentbordet men kan plockas från mina programfiler eller skapas på nytt med ett program i Extended Basic som skriver ett nytt program i MERGE-format.

Syntax för nya dolda CALL

CALL >04 MEMORY (rad 1382, 1558)

CALL >04(Z) där Z=maximalt antal möjliga poster.

CALL >05 PICTURE

CALL >05(BILD) där BILD är ett tal mellan 1 och 107. Med detta kommando skrivs alla skärmbilder. Vissa nummer ger kompletta bilder medan andra endast ger en liten komplettering till ett tidigare kommando.

Bild beskrivning

- 1 DEFINE FILE
- 2 LET'S BEGIN (definierar även pilarna)
- 3 PRINT PAGES OR REPORT (gul ram)
- 4 (QUIT)
- 5 PRINT PAGES OR REPORT (lila ram)
- 6 CREATE A FILE
- 7 DEFINE PAGE ITEMS
- 8 ITEM TYPE (rad 11-24)
- 9 MAX # OF DIGITS (rad 14-24)
- 10 INCLUDE DECIMAL POINTS (rad 14-24)
- 11 ERROR -SELECTED FILE EMPTY
- 12 DECIMAL PLACES (rad 14-24)
- 13 MAX # CHARACTERS (rad 14-24)
- 14 LOAD A FILE
- 15 ERROR -DATA FILE EMPTY
- 16 FILE NAME (rad 17-24)
- 17 FILE LOADED
- 18 ERROR CANNOT ACCESS THE DEVICE
- 19 ERROR INCORRECT DATA LOADED
- 20 MAIN INDEX
- 21 FILE STRUCTURE
- 22 DISPLAY PAGE (grön)
- 23 FORWARD, BACKWARD, PRINT, BACK (rad 21-24)
- 24 ANALYZE PAGES
- 25 ANALYSIS INDEX
- 26 DISPLAY PAGE (gul)
- 27 RE-ORDER PAGES

- 28 ASCENDING ORDER, DESCENDING ORDER (rad 22-23)
- 29 SORTING -PLEASE STAND BY (rad 23-24)
- 30 DEFINE SELECTION CRITERIA
- 31 LOW VALUE? (rad 23-24)
- 32 HIGH VALUE? (rad 23-24)
- 33 ANALYZE PAGE PASSING, ALL, ANY
- 34 ADD PAGE
- 35 CHANGE PAGES
- 36 CHANGE A PAGE
- 37 TRANSFORMATIONS INDEX 1
- 38 MATH TRANSFORMATIONS
- 39 WHICH ITEM IS A? (rad 23-24)
- 40 IS B A CONSTANT (Y OR N)? (rad 23-24)
- 41 WHICH ITEM IS B? (rad 23-24)
- 42 VALUE OF B? (rad 23-24)
- 43 IS C A CONSTANT (Y OR N)? (rad 23-24)
- 44 WHICH ITEM IS C? (rad 23-24)
- 45 VALUE OF C? (rad 23-24)
- 46 CORRECT (Y OR N)? (rad 24)
- 47 PRESS BACK TO STOP PRINTING, ENTER TO CONTINUE
- 48 UPDATING -PLEASE STAND BY
- 49 (sudda rad 6-20, kol 16-30)
- 50 DELETE A PAGE
- 51 ARE YOU SURE (Y OR N)? (rad 22)
- 52 PRINT FORMAT INDEX
- 53 PRINT SELECTED ITEMS
- 54 SAVE DATA
- 55 TRANSFORMATIONS INDEX 2
- 56 TRANSFORMATIONS INDEX 3
- 57 LABEL TYPE
- 58 ARE YOU THROUGH?
- 59 LINEAR FIT X ITEM?
- 60 Y ITEM? (rad 22)
- 61 ITEM STATISTICS WHICH ITEM?
- 62 SEARCH FOR A PAGE
- 63 PAGE, FORWARD, BACKWARD, PRINT, BACK (rad 4-24)
- 64-81 (matematiska transformationer (rad 21)
- 82 ITEM # FOR SELECTION? (rad 21, 23-24)
- 83 CHANGE PAGE IF IT MEETS, ALL, ANY
- 84 PRESS BACK WHEN FINISHED (rad 24)
- 85 PRESS ENTER TO CONTINUE (rad 24)
- 86 WHICH ITEM DO YOU WISH TO CHANGE? (rad 23-24)
- 87 FILE STRUCTURE
- 88 AID, PROC'D, PROC'D, BACK (rad 21-24)
- 89 (sudda rad 3-24)
- 90 ITEM STATISTICS
- 91 LINEAR FIT $Y=A*X+B$
- 92 CHARACTER ITEM UNACCEPTABLE (rad 24)
- 93 PRINTING DEVICE NAME? (rad 12)
- 94 CHARACTER WIDTH OF THE PRINTING DEVICE? (rad 16-17)
- 95 PRINTING HOLD DOWN ANY KEY TO STOP
- 96 WHAT IS THE TITLE FOR THIS PRINTOUT? (rad 20-21)
- 97 INSUFFICIENT DATA
- 98 PRINTING DEVICE
- 99 ASCENDING ORDER SORTING -PLEASE STAND BY (rad 22-24)
- 100 DESCENDING ORDER SORTING -PLEASE STAND BY (rad 22-24)
- 101 PRESS P TO PRINT SCREEN (rad 22-24)
- 102 PRESS PROC'D, BACK (rad 23-24)
- 103 PRESS PROC'D, ENTER, BACK (rad 22-24)
- 104 ANALYZING PLEASE STAND BY
- 105 (sudda rad 21-23)
- 106 PAGES AVAILABLE: (rad 22-24)
- 107 NO PAGES LEFT

CALL >06 DELETE (rad 3560)

CALL >06(P) där P är den post som skall suddas. Alla poster med högre nummer kommer då även att omnumreras.

CALL >07 SWAP (rad 20240, 21160, 21600)

CALL >07(POSTA,POSTB,MAXPOST) används vid sortering där POSTA och POSTB byter plats. Det är något osäkert om det tredje talet skall vara högsta använda postnumret eller detta plus 1 eftersom resultatet blir lika bra i båda fallen.

Ett lägre värde än största posten ger dock problem genom att en av posterna skrivs på två ställen.

CALL >08 PRINTER (rad 355, 4110)

CALL >08(PRINTER\$,TEST) undersöker om printer finns. TEST=0 vid misslyckat försök att öppna printer och TEST=1 vid lyckat försök dvs printer med beskrivning enligt PRINTER\$ finns. Observera att detta endast är en test så du måste sedan öppna printern med OPEN på vanligt sätt.

CALL >09 INITIALIZE

CALL >09 finns inte i det 20 kb långa huvudprogrammet utan i ett kort litet initieringsprogram med pre-scan och DIM. Jag har ej lyckats få det att fungera från BASIC.

CALL >0A TEXT

CALL >0A(NR,TEXT\$) ger olika texter till TEXT\$ beroende på NR där NR ligger mellan 0 och 21. Exempelvis **CALL >0A(1,TEXT\$)** ger TEXT\$="CHAR".

NR TEXT

0 PAGE #
1 CHAR
2 INT
3 DEC
4 SCI
5 INDEX
6 0 = PAGE #
7 FILE:
8 DATE:
9 TITLE:
10 ABS
11 LOG10
12 LOGE
13 EXP
14 ATAN
15 TAN
16 SIN
17 COS
18 INT
19 SGN
20 PI
21 RND

CALL >0B TRANSFORMATION (rad 3160, 3192)

CALL >0B(T) används i samband med matematiska transformationer där T är 0,1 eller 2. Detta kommando kräver att variablerna @ (alfaslang), A(30), B(30), C(15) och D(15) finns i BASIC-programmet. @ = antalet transformationer. A() och B() används för mellanlagring när transformationerna läses och skrives. T=0 för skriva och T=1 för läsa.

När du vill läsa med T=1 spelar de övriga variablernas värde ingen roll. Efter läsningen kommer dock A() och B() att innehålla transformationerna och @ = antalet transformationer.

När man skriver med T=0 är det viktigt att A(), B() och @ tilldelas rätta värden före transformationen. Om du gör fel här kan filen sedan krascha i modulen om du gör matematiska transformationer. T=2 skriver A() och B() men ej @ så jag vet inte vad man skall ha detta till.

A(), B() och @ kommer att lagras i datafilen så att värdena följer med vid SAVE och LOAD av filen. Dessa variabler kan endast skriva värden 0-99 till datafilen. Om du aldrig vill använda de matematiska transformationerna har du här möjlighet att lagra 61 olika tal mellan 0 och 99.

Tänk på att detta kommando ger dig möjlighet att sudda alla transformationer och lägga till transformationer. Om du använder modulen som TI har tänkt är detta ej möjligt utan om du definierar nya transformationer så kommer alla tidigare sådana att försvinna. Trots detta har TI ej tagit med möjligheten att sudda allt.

CALL >0C SCREENDUMP (rad 26010)

CALL >0C(RADER,PRINTER\$,TEST) skriver ut det önskade antalet rader från skärmen där RADER=1-24. Antalet rader räknas från övre delen av skärmen men endast 28 teckens bredd skrivs ut, dvs kolumn 1-2 och 31-32 kommer ej med. TEST=0 vid misslyckat försök och TEST=1 vid lyckat försök.

Testprogram för nya CALL

För att kunna testa de nya CALL-kommandona har jag skrivit ett kort CALLTEST-program där alla kommandona kan provas. Tryck på ENTER när du vill fortsätta. Om du trycker på P så skrivs skärmen ut på printer. Detta program går inte att knappa in direkt från tangentbordet. Knappa in alla rader som ej innehåller CALL >. Använd sedan CALL-MERGE-programmet i Extended Basic för att skriva dessa rader i MERGE-format. Ladda sedan in CALLTEST och skriv MERGE DSK1.MERGE så kommer du att få det kompletta programmet som sedan kan köras med Personal Record Keeping i Basic. Glöm inte att skriva CALL P(8000) och NEW i kommandomod innan du laddar in programmet. Du som har kassettspelare kan inte fixa detta själv men när programmet är rätt efter MERGE så kan du använda detta.

Om du sänder en kassett eller flexskiva till mig samt 5 kr i frimärken så får du en kopia på de fyra programmen: CREATE, EDIT, CALLTEST och CALLMERGE.

```
10 OPTION BASE 1
20 @=0
30 DIM A(30),B(30),C(15),D(15)
40 J=1
50 K=2
60 L=3
70 M=28
100 REM CALLTEST FOR PERSONAL RECORD KEEPING
110 REM JAN ALEXANDERSSON
120 REM SPRINGARVAGEN 5
130 REM S-142 61 TRANGSUND
140 REM SWEDEN
150 REM VERSION 1988-08-12
160 REM P=PRINT ENTER=CONTINUE
170 REM CALL P(8000) NEW
200 PS="PIO"
210 CALL CLEAR
220 CALL D(10,J,M,"CALL NR 4-12",12,J,M,"0=LO
AD 1=SAVE 2=DISP 3=END")
230 CALL A(10,14,K,T,N,Q,12)
240 IF T>J THEN 3000
250 ON N+J GOSUB 5030,5100,4000,3000,400,500,
600,700,800,900,1000,1100,1200
260 GOTO 210
400 CALL >04(Z)
410 CALL D(12,J,M,STR$(Z)&" MAX PAGES")
420 GOSUB 2000
430 RETURN
500 CALL D(12,J,M,"PICTURE 1-107")
510 CALL A(12,15,L,T,P,J,107)
520 CALL HCHAR(J,J,64,768)
530 CALL >05(P)
540 GOSUB 2000
550 CALL CLEAR
560 CALL SCREEN(8)
565 FOR W=J TO 8
570 CALL COLOR(W,K,8)
575 NEXT W
580 RETURN
600 GOSUB 5000
610 CALL D(12,J,M,"DELETE ITEM")
620 CALL A(12,13,L,T,P,J,Z)
630 CALL >06(P)
640 RETURN
700 CALL D(12,J,M,"CALL >07(W, X, Y)",14,17
,M,Z)
710 CALL A(14,10,L,T,W,J,Z)
720 CALL A(14,14,L,T,X,J,Z)
730 CALL A(14,18,L,T,Y,J,Z+J)
740 CALL >07(W,X,Y)
750 RETURN
800 CALL D(12,J,M,"PRINTER")
810 CALL A(12,9,M,T,PS)
820 CALL >08(PS,T)
830 IF T=Q THEN 810
840 CALL D(14,J,M,"PRINTER OK")
850 GOSUB 2000
860 RETURN
900 CALL D(12,J,M,"CALL >09 DON'T WORK IN BAS
IC")
```

```

910 GOSUB 2000
920 RETURN
1000 CALL D(12,J,M,"TEXT NR 0-21")
1010 CALL A(12,14,M,T,N,Q,21)
1020 CALL >0A(N,T$)
1030 CALL D(14,J,M,T$)
1040 GOSUB 2000
1050 RETURN
1100 GOSUB 5000
1105 CALL D(12,J,M,"TRANSFORMATION",14,J,M,"@
=",16,J,M,"CLEAR A() B() Y/N")
1110 CALL A(12,16,J,T,R,Q,K)
1115 CALL A(14,5,K,T,@,Q,99)
1120 CALL A(16,19,J,T,T$)
1125 IF T$<>"Y" THEN 1160
1130 CALL D(18,J,M,"FILL WITH 0")
1135 CALL A(18,11,K,T,S,Q,99)
1140 FOR W=J TO 30
1145 A(W)=S
1150 B(W)=S
1155 NEXT W
1160 CALL >0B(R)
1165 FOR W=J TO 30
1170 PRINT A(W);B(W),
1175 NEXT W
1180 PRINT : "@"="";@
1185 GOSUB 2000
1190 RETURN
1200 CALL D(12,J,M,"NUMBER OF ROWS")
1210 CALL A(12,16,L,T,R,J,24)
1220 FOR W=J TO 24
1230 CALL HCHAR(W,J,48+W,M)
1240 NEXT W
1250 CALL >0C(R,P$,T)
1260 RETURN
2000 CALL KEY(L,R,S)
2010 IF S<J THEN 2000
2020 IF R=80 THEN 2050
2030 R=24
2040 GOSUB 1250
2050 RETURN
3000 END
4000 CALL D(12,J,M,"SHOW ITEM")
4010 CALL A(12,11,L,T,S,J,U)
4020 GOSUB 5000
4030 FOR W=J TO Z
4040 CALL H(J,10,S,T)
4050 IF T=J THEN 4100
4060 CALL G(J,W,S,T,T$)
4070 IF T THEN 4130
4080 PRINT T$
4090 GOTO 4130
4100 CALL G(J,W,S,T,V)
4110 IF T THEN 4130
4120 PRINT V
4130 NEXT W
4140 GOSUB 2000
4150 RETURN
5000 CALL H(J,6,Q,Z)
5010 CALL H(J,5,Q,U)
5020 IF Z THEN 5090
5030 CALL CLEAR
5040 INPUT "LOAD FILE ":F$
5050 CALL L(F$,T)
5060 IF T=Q THEN 5040
5070 GOTO 5000
5080 CALL CLEAR
5090 RETURN
5100 CALL CLEAR
5110 INPUT "SAVE FILE ":F$
5120 CALL S(F$,T)
5130 IF T=Q THEN 5110
5140 RETURN

```

PRK-INIT

```

20 CALL >09
30 OPTION BASE 1
40 DIM @$(15),A$(5),B$(15),C$(5),D$(5)
50 E$=F$&G$&H$&I$&J$&K$
60 @=0
70 DIM A(30),B(30),C(15),D(15),E(5),F(5),G(5)
,H(5),I(15)
80 J=K+L+M+N+O+P+Q+R+S+T+U+V+W+X+Y+Z+[+]+A
@+AA+AB+AC+AD+AE
90 AF=AG+AH+AI+AJ+AK+AL+AM

```

CALLMERGE

```

100 REM CALLMERGE FOR PRK
110 REM JAN ALEXANDERSSON
120 REM VERSION 1988-08-12
130 REM EXTENDED BASIC
140 OPEN #1:"DSK1.MERGE",DISPLAY ,VARIABLE 16
3,OUTPUT
150 READ A
160 PRINT #1:CHR$(A);
170 IF A=0 THEN PRINT #1
180 IF A=255 THEN PRINT #1:CHR$(255):: CLOSE
#1 :: END
190 GOTO 150
200 DATA 1,144,157,200,1,4,183,90,182,0
210 DATA 2,18,157,200,1,5,183,80,182,0
220 DATA 2,118,157,200,1,6,183,80,182,0
230 DATA 2,228,157,200,1,7,183,87,179,88,179,
89,182,0
240 DATA 3,52,157,200,1,8,183,80,36,179,84,18
2,0
250 DATA 3,252,157,200,1,10,183,78,179,84,36,
182,0
260 DATA 4,136,157,200,1,11,183,82,182,0
270 DATA 4,226,157,200,1,12,183,82,179,80,36,
179,84,182,0
280 DATA 255,255

```

PRK-MAIN

```

164 J=1
165 K=2
166 L=3
171 M=9.999999999999999E+127
173 N=LOG(10)
220 CALL >05(K)
225 CALL A(13,16,K,O,P,J,31)
230 IF O>J THEN 225
240 CALL H(Q,L,Q,P)
255 CALL A(14,16,K,O,R,J,31)
258 IF O=4 THEN 220
260 IF O>J THEN 255
270 CALL H(Q,K,Q,R)
285 CALL A(15,18,K,O,S,79,99)
288 IF O=4 THEN 220
290 IF O>J THEN 285
300 CALL H(Q,4,Q,S)
310 CALL >05(98)
320 GOSUB 6300
325 ON T GOTO 330,390,320,320,220
330 CALL >05(93)
340 CALL A(13,J,28,O,E$)
345 ON O GOTO 355,340,340,310,340,340,220
355 CALL >08(E$,O)
360 IF O THEN 390
370 GOSUB 14300
375 GOTO 310
390 CALL >05(J)
425 GOSUB 10060
435 ON U GOTO 525,25005,310
525 GOSUB 6740
540 CALL >05(6)
550 CALL A(13,16,9,O,F$)
560 ON O GOTO 580,550,550,540,550,550,390
580 CALL H(Q,J,Q,F$)
620 V=J
640 FOR W=J TO 15
660 CALL >05(7)
670 CALL D(7,14,L,W)
673 IF V THEN 680
675 CALL H(J,9,W,F$)
677 CALL D(9,15,9,F$)
680 CALL A(9,15,9,O,F$)
690 ON O GOTO 780,680,680,660,680,680,720
720 IF W=J THEN 1330
760 IF V THEN 540 ELSE 1340
780 CALL H(Q,9,W,F$)
800 CALL >05(8)
802 IF V THEN 840
804 CALL H(J,10,W,U)
806 CALL >0A(U,F$)
808 CALL D(11,15,4,F$)
840 GOSUB 10068
845 IF U>X THEN 840
860 CALL H(Q,10,W,U)
865 CALL >0A(U,F$)
867 CALL D(11,15,4,F$)

```

```

880 ON U GOTO 910,990,1000,1130
910 CALL >05(13)
913 IF V THEN 920
915 CALL H(J,11,W,R)
917 CALL D(14,21,L,R)
920 CALL A(14,22,K,O,R,J,15)
930 ON O GOTO 950,920,920,660,920,920,920
950 CALL H(Q,11,W,R)
960 GOTO 1310
990 CALL >05(9)
995 GOTO 1010
1000 CALL >05(10)
1010 IF V THEN 1050
1030 CALL H(J,11,W,R)
1040 CALL D(14,21,L,R)
1050 CALL A(14,22,K,O,R,U-J,U+8)
1060 ON O GOTO 1090,1050,1050,660,1050,1050,10
50
1090 CALL H(Q,11,W,R)
1100 IF U=K THEN 1310
1130 CALL >05(12)
1133 IF V THEN 1140
1135 CALL H(J,12,W,S)
1137 CALL D(14,21,L,S)
1140 CALL A(14,22,K,O,S,4-U,-5*U+25)
1150 ON O GOTO 1165,1140,1140,660,1140,1140,11
40
1165 IF U=4 THEN 1270
1170 IF R<=S THEN 1140
1270 CALL H(Q,12,W,S)
1310 IF V=Q THEN 1340
1320 NEXT W
1330 CALL H(Q,5,W-J,W)
1340 CALL H(J,5,Q,Y)
1360 GOSUB 11400
1380 CALL >05(106)
1382 CALL >04(Z)
1384 CALL D(22,22,6,Z)
1400 GOSUB 6300
1420 ON T GOTO 1550,1450,1360,1400,1400
1450 V=Q
1470 CALL >05(86)
1480 CALL A(24,15,K,O,W,J,Y)
1490 ON O GOTO 660,1480,1480,1360,1480,1480,13
60
1550 CALL H(J,6,Q,[])
1552 CALL H(J,5,Q,Y)
1553 IF Y<16 THEN 1556
1554 Y=15
1556 \=J
1558 CALL >04(Z)
1570 CALL >05(20)
1590 X=8
1610 GOSUB 10080
1611 IF U>X THEN 1610
1612 IF (U<L)+(U>6)+([>Q]) THEN 1630
1614 CALL >05(15)
1616 GOSUB 10020
1618 IF U=13 THEN 1570 ELSE 1616
1630 ON U GOTO 1680,2030,1635,2570,20020,1638,
25400,25600
1635 GOSUB 2310
1636 GOTO 1550
1638 GOSUB 4040
1640 GOTO 1550

```

Annons

Säljes:

TI-99/4A med expansionsenhet innehållande 32K minneskort, diskettstation, diskkontrollkort, RS232-kort.

Moduler: Disk Manager 2, TI Extended BASIC, TI Invaders, TI Editor/Assembler inkl programhandboken med 2 disketter. Skrivare ITOH 8510 inkl alla anslutningskablar.

Pris: 5 000:—

Jonas Söderlund

Pl 2063

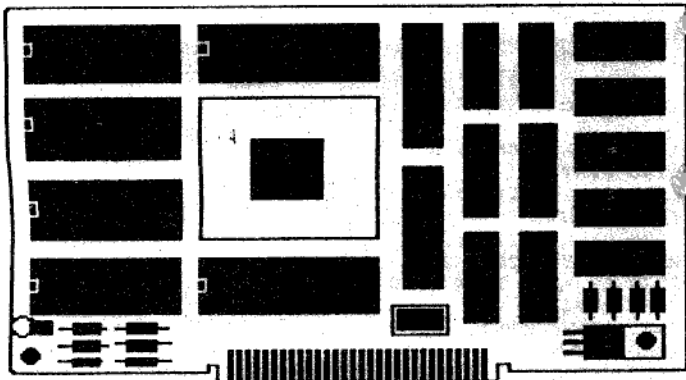
760 15 Gräddö, 0176/404 18

Nya kort

Forth-kort

McCann software i Ohama, USA, har tagit fram ett helt nytt kort för expansionsboxen. På kortet finns en NC4016-processor från Novix. Den är specialgjord för programspråket Forth och skall enligt uppgift kunna utföra 7 MIPS (miljoner instruktioner per sekund). Uppfinnaren av Forth-språket, Charles Moore, är också pappa till den nya processorn. En specialversion av Forth medföljer också kortet. Priset är 595 dollar, enligt tillverkaren billigt. Motsvarande kort finns också för PC-marknaden, men där uppges priset ligga över 1 000 dollar. Kortet heter *Avanti-99* och kommer från:

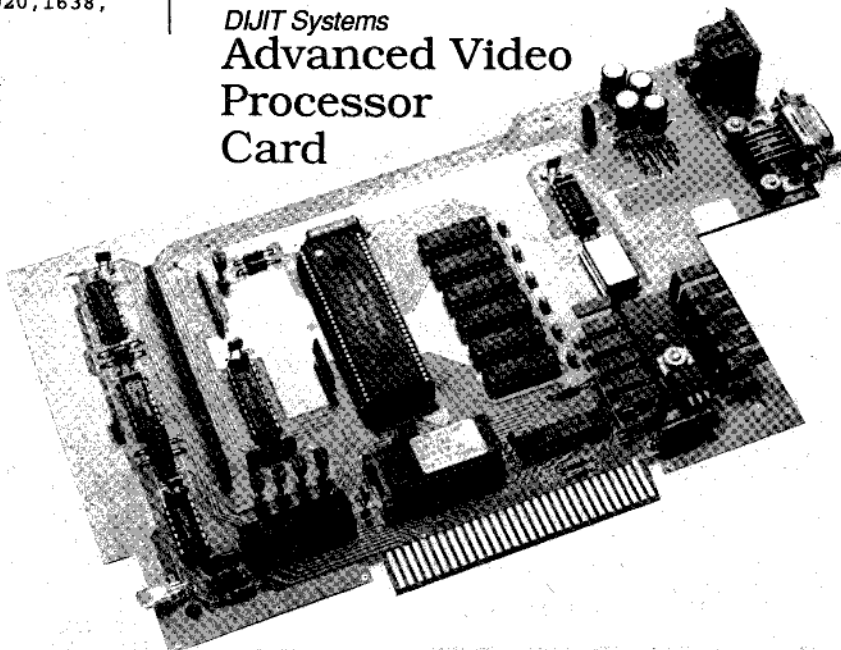
McCann Software
P.O. Box 34160
Omaha, NE 68134
USA



Nytt 80-kolumnskort

Den amerikanska firman *DIJIT Systems* har tagit fram ett nytt 80-kolumnskort för 99:an. Det ska vara kompatibelt med det kort som den tyska firman *Mechatronics* har tagit fram, men är i motsats till det tänkt att sitta i expansionsboxen. Se bilden här nedan. Priset är 220 eller 250 dollar, beroende på minnet på kortet.

DIJIT Systems
**Advanced Video
Processor
Card**



Tips och tricks

av Fredrik Nilsson

Här kommer 4 program som visar lite tips och tricks. De är:

- 1) POKEV & PEEKV
- 2) Radnummer 0 och listskydd
- 3) Ett sätt att styra "älvor"
- 4) Ett sätt att stoppa "älvor"

Det första programmet gör poke och peek i vdp. Det kräver expansionsminne och exbasic för att fungera. Programmet förklarar sig självt (tror jag). Men här kommer anropen:

```
CALL POKEV(VDPADDRESS,BYTE)
CALL PEEKV(VDPADDRESS,BYTE)
CALL WMBW(VDPADDRESS,BUFFERSTART I CPU,
ANTAL BYTES)
CALL WMBR(VDPADDRESS,BUFFERSTART,ANTAL)
```

Det här kan skrivas bättre, men det fungerar (även om det går lite långsamt).

WMBW kan användas till spel. Om du lagrar skärmar direkt i minnet och sen anropar WMBW får du ett skärmbyte som är mycket snabbare än vad du sett tidigare!

```
10 CALL CLEAR :: CALL INITIERA
20 CALL POKEV(-30971,0)
30 FOR I=1 TO 32 STEP 2 :: CALL VCHAR(I,I,4
2,24):: NEXT I
40 CALL WMBR(0,9478,768)
50 CALL CLEAR
60 FOR I=2 TO 24 STEP 2 :: CALL HCHAR(I,1,4
2,32):: NEXT I
70 CALL WMBR(0,10246,768)
80 CALL WMBW(0,9478,768)
90 CALL WMBW(0,10246,768)
100 GOTO 80
29900 !*****
29910 !* POKEV OCH PEEKV *
29920 !* AV F.NILSSON 1988 *
29930 !* EX + EM *
29940 !*****
30000 SUB INITIERA :: CALL INIT
30010 CALL LOAD(8196,63,248)! START ADRESS REF
TABELL
30020 CALL LOAD(16376,86,68,80,32,32,32,36,244
)! UNDERPROGRAM "VDP " STARTADRESS 946
0
30030 CALL LOAD(9460,2,0)! LI R0,0
30040 CALL LOAD(9462,0,0)!
30050 CALL LOAD(9464,2,1)! LI R1,0
30060 CALL LOAD(9466,0,0)!
30070 CALL LOAD(9468,2,2)! LI R2,1
30080 CALL LOAD(9470,0,1)!
30090 CALL LOAD(9472,4,32)! BLWP >WMBW
30100 CALL LOAD(9474,32,36)
30110 CALL LOAD(9476,4,91)! B *R11
30120 BUFF=768*2
30130 ADR=9478+BUFF :: A1=INT(ADR/256):: A2=AD
R-A1*256
30140 CALL LOAD(8194,A1,A2)! LADDA IN FÄRSTA L
EDIGA ADRESS I LIGA MINNET
30150 SUBEND
30160 SUB WMBW(VDPAD,CPUAD,ANTAL)
30170 VDPAD=VDPAD-(VDPAD<0)*65536
30180 CPUAD=CPUAD-(CPUAD<0)*65536
30190 V1=INT(VDPAD/256):: V2=VDPAD-V1*256
30200 C1=INT(CPUAD/256):: C2=CPUAD-C1*256
30210 A1=INT(ANTAL/256):: A2=ANTAL-A1*256
30220 CALL LOAD(9462,V1,V2)
30230 CALL LOAD(9466,C1,C2)
30240 CALL LOAD(9470,A1,A2)
30250 CALL LOAD(9474,32,36)
30260 CALL LINK("VDP")
30270 SUBEND
30280 SUB WMBR(VDPAD,CPUAD,ANTAL)
30290 VDPAD=VDPAD-(VDPAD<0)*65536
30300 CPUAD=CPUAD-(CPUAD<0)*65536
30310 V1=INT(VDPAD/256):: V2=VDPAD-V1*256
30320 C1=INT(CPUAD/256):: C2=CPUAD-C1*256
```

```
30330 A1=INT(ANTAL/256):: A2=ANTAL-A1*256
30340 CALL LOAD(9462,V1,V2)
30350 CALL LOAD(9466,C1,C2)
30360 CALL LOAD(9470,A1,A2)
30370 CALL LOAD(9474,32,44)
30380 CALL LINK("VDP")
30390 SUBEND
30400 SUB POKEV(VDPAD,BYTE)
30410 VDPAD=VDPAD-(VDPAD<0)*65536
30420 V1=INT(VDPAD/256):: V2=VDPAD-V1*256
30430 CALL LOAD(9462,V1,V2)
30440 CALL LOAD(9466,37,6)
30450 CALL LOAD(9470,0,1)
30460 CALL LOAD(9474,32,36)
30470 CALL LOAD(9478,BYTE)
30480 CALL LINK("VDP")
30490 SUBEND
30500 SUB PEEKV(VDPAD,BYTE)
30510 VDPAD=VDPAD-(VDPAD<0)*65536
30520 V1=INT(VDPAD/256):: V2=VDPAD-V1*256
30530 CALL LOAD(9462,V1,V2)
30540 CALL LOAD(9466,37,6)
30550 CALL LOAD(9470,0,1)
30560 CALL LOAD(9474,32,44)
30565 CALL LINK("VDP")
30570 CALL PEEK(9478,BYTE)
30580 SUBEND
```

Program två är ett program som ändrar i sig självt. Den som har läst Anders Perssons artikel i PB 85-1 sid 6 förstår att radnummertabellen ändras. Först kör du programmet som det står. När du sen listar ser du att rad 1 har blivit rad 0! Andra sedan rad 160 till IF RAD=2 THEN 190 och rad 190 till CALL LOAD(I,255,255). Kör sen programmet och försök lista det. Datorn kommer bara att skriva ut rad noll! Det här listskyddet är mycket enkelt att ta bort. Men det får du klura ut själv. Jag vet inte varför datorn slutar lista när den stöter på en rad som har ett högre värde än den efterföljande. Det kan kanske någon förklara? Vidare kan du ta testa att lägga in flera rader med samma nummer.

```
1 !** RAD 1 BLIR RAD 0 **
2 !** RAD 2 BLIR RAD 32767**
10 !*****
20 !* RAD NR 0 & LISTSKYDD *
30 !*
40 !* AV F.NILSSON 1988 *
50 !* EX + EM *
60 !*****
70 CALL CLEAR
80 CALL PEEK(-31952,ST1,ST2)! HJMTA STARTAD
RESS
90 CALL PEEK(-31950,SL1,SL2)! HJMTA SLUTADR
ESS
100 STA=ST1*256+ST2-65536
110 SLU=SL1*256+SL2-65536
120 FOR I=STA TO SLU STEP 4
130 CALL PEEK(I,RAD1,RAD2)! HJMTA RADNUMMER
I TABELL
140 RAD=RAD1*256+RAD2
150 PRINT RAD
160 IF RAD=1 THEN 190 ! JR DET RITT RAD? JA
->190
170 NEXT I
180 END
190 CALL LOAD(I,0,0)! SKRIV IN DET NYA RADNU
MRET
200 END
```

Program tre visar ett sätt att styra älvor (spritsar) enkelt, och ännu viktigare, snabbt! Det är gjort för exbasic. Nu väntar jag mig att alla ska skicka in massor med bidrag som använder den här idén. Det är lite knapert med det tycker jag.

```
10 !*****
20 !* JLVSTYRNING *
30 !*
40 !* AV F.NILSSON 1988 *
50 !* EXBASIC *
60 !*****
70 !
80 !
90 DIM RX(4,10),RY(4,10)
100 CALL CLEAR :: CALL SCREEN(2)
```

```

110 CALL CHAR(40,"01030F01010103030300030301
0101010C0E0F8C0E0C060208060E0E0C080C0C0")
120 CALL CHAR(44,"00010F030B237F252530301818
0D0F0700B0F0C0D0C4FE24240C0C0B1850F0E0")
130 CALL MAGNIFY(3)
140 FOR SP=1 TO 4
150 FOR I=1 TO 10
160 READ RY(SP,I),RX(SP,I)
170 NEXT I
180 NEXT SP
190 CALL SPRITE(#1,44,5,40,40,#2,44,7,40,200
,#3,44,12,160,40,#4,44,16,160,200,#5,40,
7,100,120)
200 R=0
210 !** HJR KOMMER DEN EGENTLIGA LOOPEN
220 CALL JOYST(1,X,Y):: R=1-(R<10)*R ! FORME
LN 1-(R<11)*R STEGAR R ETT OCH SJTTER R=
1 OM R>10
230 CALL MOTION(#5,-Y*3,X*3,#1,RY(1,R),RX(1,
R),#2,RY(2,R),RX(2,R),#3,RY(3,R),RX(3,R)
,#4,RY(4,R),RX(4,R))
240 GOTO 220
250 !** HJR KOMMER R\RELSE VARIABLERNA
260 DATA 0,40,0,20,0,10,0,5,0,0,0,0,-5,0,-
10,0,-20,0,-40
270 DATA 30,0,0,-30,-30,0,0,30,0,-30,30,0,0,
30,-30,0,-10,0,10,0
280 DATA 0,40,0,40,0,40,0,40,0,40,-40,0,40,0
,0,-80,0,-80,0,-40
290 DATA -50,20,0,20,50,20,0,0,0,0,-50,-20,0
,-20,50,-20,0,0,0,0

```

Program fyra startar och stoppar älvor samtidigt! Det är pokeningen (ursäkt svengelskan) i adress -31878 som gör det. Men akta dig för att skriva in något annat än de antal älvor du har på skärmen. För då garanterar jag inte vad som händer. Men för att få detta att fungera måste man ha expansionsminne eller Super Extended BASIC eller något annat som tillåter poke.

```

10 !*****
20 !*      JLVSTOPPNING      *
30 !*      *                  *
40 !* AV F.NILSSON 1988      *
50 !*      *                  *
60 !* EX + EM                *
70 !*****
80 CALL INIT :: CALL CLEAR :: CALL SCREEN(2
)
90 CALL CHAR(40,"FFA5A5A5A5A5A5FF")
100 FOR I=1 TO 28 :: CALL SPRITE(#1,40,5,I*8
-7,256):: NEXT I
110 FOR I=1 TO 28 :: CALL MOTION(#1,0,RND*90
+10):: NEXT I
120 CALL LOAD(-31878,28)
130 CALL KEY(0,K,S):: IF S<1 THEN 130
140 CALL LOAD(-31878,0)
150 CALL KEY(0,K,S):: IF S<1 THEN 150 ELSE 1
20

```

att lösas med hjälp av en dator. Vad boken kostar här i Sverige har jag inte kläm på, då jag har fått mitt exemplar direkt från USA. Där har den i alla fall ett rekommenderat pris på USD 12,95. Jag har dock sett den i något skyltfönster här i Lund också.

Problem med historia

Boken är indelad i tre delar. En problemdel, en del med tips för den som har problem med problemen och slutligen en del med lösningar. Det roliga med problemen är att de inte bara är uppräknade sida upp och sida ner, utan ingår som en naturlig del i en historia. Handlingen utspelar sig bland en brokig samling människor, som arbetar på I-Q Industries (Backa ett steg i alfabetet från initialerna I-Q, och det blir genast något annat...) i själva hjärtat av Kaliforniens datorindustri, Silicon Valley. Man får följa Spotswood Gilbert när han börjar arbeta på avdelningen för utveckling av Fortran-kompilatorer på I-Q. Personalen i Fortran-gruppen avdelar en hel del tid och datorkapacitet för att lösa olika problem som de kommer på efterhand. Det är dessa problem som läsaren också får försöka lösa.

Bokens handling är ganska underhållande. Själva är jag inte så väldigt förtjust i konstruerade problem av det här slaget, men det gick alldeles utmärkt att med behållning läsa själva historien, utan att bry sig om problemen. Dessa är exempelvis av typen: Hitta en serie konsekutiva heltal, vars summa blir exakt 10 000. $3\ 332 + 3\ 333 + 3\ 334 = 9\ 999$ är bra nära, men ändå inte riktigt rätt.

Hjälp på traven

Vissa av problemen är sådana att man snabbt ser vilken väg man ska gå för att lösa dem. Andra vill sig kanske inte alls. Då kan man ha god nytta av bokens andra avdelning. Där finns nämligen några ord på vägen för den villrådige. Inga lösningar, men ibland en antydning om hur man kan göra, och någon gång en om hur man inte ska göra.

Trefaldiga lösningar

Vare sig man kommer på någon egen lösning eller ej, kan det vara roligt att se hur författaren anser att det ska gå till. Lösningarna är speciellt intressanta, då de ges på tre olika språk. Till samtliga problem finns programexempel skrivna i GWBASIC, Microsoft C och Turbo Pascal. Vän av ordning frågar sig kanske vad det här då har med 99:an att göra?

Jo, 99:an kommer faktiskt in riktigt bra i sammanhanget. Programmen i GWBASIC är ganska lätta att översätta till Extended BASIC. X-BASIC är tillräckligt innehållsrikt för att klara det mesta av de BASIC-program som finns i boken.

Programmen i Turbo Pascal kan utan större besvär skrivas om för UCSD Pascal på 99:an. Turbo Pasc'99 är mer begränsad, vilket ställer till problem med vissa program.

Samma gäller för c99, som inte heller är en fullständig implementation av det språket.

Summering

På det hela taget en trevlig bok. För den som inte nödvändigtvis vill bekymra sig om problemen, går det bra att bara läsa den skönlitterära delen. Lösningarna på problemen är också speciellt intressanta för den som behärskar exempelvis Pascal men inte C. Däremot vill jag varna den som exempelvis kan BASIC, men inte Pascal, för att dra alltför långtgående slutsatser av lösningarna i boken. Programmen i Pascal och C är skrivna så att de ser ungefär likadana ut som deras motsvarigheter i BASIC. Därmed förlorar man en hel del av själva finessen med att använda något annat än BASIC. Exempelvis utnyttjas Pascals möjlighet att konstruera datastrukturer, som är anpassade till problemet i fråga, inte alls.

Hur man skaffar sig problem

av Anders Persson

Har man inte problem så skaffar man sig, sägs det. Den devisen kan gälla även i datorvärlden. Ett sätt att skaffa sig problem är att inhandla boken "Puzzled Programmers" av Michael Wiesenberg. Boken, som är utgiven av Microsoft Press, innehåller 15 problem som är avsedda

Utmaningen

av Anders Persson

En gång i tiden förklarade jag att om det inte dök upp några problem, så blev det ingen Utmaning heller. Därav förra årets lugn på den här fronten. Nu är det dock dags igen, även om det blir en mjukstart. Mest lösningar, med andra ord.

RAM-disk till p-systemet

När de första RAM-diskarna kom var det lite oklart om de fungerade ihop med p-systemet eller inte. Efterhand klarnade uppgifterna dock. Nu står det exempelvis uttryckligen i anvisningarna till Horizons modell att det kortet inte går att använda ihop med TI Pascal, som de kallar p-systemet. Jag undrade då lite över vad detta kunde bero på, dock utan någon intensivare forskning. Jag hade inte någon RAM-disk att prova med, så motivationen var väl inte så värst stor, heller.

Som framgick av PB 87-4 köpte Lennart Thelander sig en RAM-disk, av CorComps fabrikat. Den fungerade mycket riktigt inte tillsammans med p-systemet. Men när det nu fanns en inom räckhåll blev det naturligtvis genast intressantare att studera problemet. Det gick ju bra med nästan allt annat, varför då inte med p-systemet? Nyfikenheten var väckt, och dessutom gjorde väl Lennarts enträgen tjetande sitt, det också...

Nåväl, som Lennart också skrev i sin artikel, lyckades det till slut. Jag tänkte nu beskriva vad det är som gör att det inte går från början, och vad man ska göra åt det. Förhoppningsvis kan det vara intressant inte bara för dem som vill använda en RAM-disk tillsammans med p-systemet utan för alla som är intresserade av hur det kan komma sig att 99:an kan fungera rent allmänt.

Tillbehör i allmänhet

Sett ur elektronisk och programmessig synvinkel använder 99:an, till skillnad från en del andra datorer, ett mycket flexibelt system för anslutning av tillbehör. I princip kommer datorn åt RS232-kort, RAM-diskar eller klockor på samma sätt. Det enda som skiljer de olika enheterna från varandra är CRU-adressen. För att datorn vill varje tillfälle ska veta vad den ska leta efter, skapar man en PAB (Peripheral Access Block) i minnet. Den innehåller information om den DSR (Device Service Routine) som ska utnyttjas. Med hjälp av den informationen kan ett program, som brukar benämnas DSRLNK, länka ihop den begärda operationen med rätt tillbehör.

Konceptet tillåter inte bara operationer som har med filer att göra (OPEN, PRINT, CLOSE etc.), utan även SUB-program, avbrotts- och uppstartstrutiner m m.

Nackdelen med systemet är att länkningen mellan PAB och tillbehör utföres vid varje operation som berör tillbehöret i fråga. Systemets flexibilitet gör det möjligt att använda en RAM-disk, trots att det inte var meningen från början. Men det är inte gratis. Priset vi får betala är ineffektivitet.

Tillbehör till p-systemet

När p-systemet implementerades på 99:an gjorde TI ett försök att komma tillrätta med den dåliga effektiviteten utan att offra hela flexibiliteten. I p-systemet är varje PAB påbyggd med åtta bytes. Tillsammans blir detta ett PCB (Peripheral Control Block). Dessa extra bytes används till att lagra information om hur länkningen mellan PAB och tillbehör ska gå till. Bland annat lagras den CRU-bas som tillbehöret har, och även minnesadressen till det program man vill utnyttja. Den här informationen samlar systemet in vid uppstarten, genom att använda en modifierad DSRLNK. Finessen ligger alltså i det faktum att länkningen bara behöver

göras en gång. Därefter kan man gå rakt på den DSR som önskas. Allt enligt principen att det är bättre att veta än att leta.

Här ligger också anledningen till att p-systemet inte vill kännas vid en RAM-disk. Eftersom RAM-disken har en annorlunda CRU-bas än det ordinarie diskkontrollkortet, måste den också ha en annorlunda PCB, även om dess PAB är likadan. Men p-systemet var aldrig tänkt till att kunna utnyttja mer än ett kontrollkort för skivminne.

Åtgärder

Sådant är alltså problemet. Nu ska vi se vad man kan göra åt det.

Till att börja med måste man tala om för systemet att det finns mer än de tre skivminnen som det tror på från början. Normalt sett kan man utnyttja sex enheter i ett p-system, men eftersom Texas eget kontrollkort bara klarar tre, brydde man sig inte om att låta systemet känna till fler heller.

Detta är dock lätt att rätta till. Det finns en tabell som innehåller information om hur systemets olika I/O-enheter är beskaffade. Normalt finns där bara nollor på de platser som är reserverade för de tre skivminnen som inte kan utnyttjas. Genom att där lagra samma värde som de aktiva skivminnena har, väcks även de återstående tre till liv. Det här är det enda som behöver göras för att man ska kunna utnyttja fyra skivminnen tillsammans med ett kontrollkort från CorComp eller Myarc.

Men det räcker inte för en RAM-disk. Som vi redan konstaterat behöver den även en egen PCB. Hur ska den då se ut? Eftersom RAM-disken bär sig åt som om den vore ett vanligt kontrollkort, är det väl inte så överraskande att PCB:n kan likna den som används till skivminnena. Det enda som skiljer är CRU-adressen och adressen till det program som ska anropas. P-systemet har ju en alldeles egen filhantering, varför endast SUB-programmet för att läsa och skriva en sektor behövs.

CRU-adressen som RAM-disken har får operatören själv hålla reda på. Adressen till SUB-programmet kan man däremot låta ett program leta upp. När man väl känner dessa värden, är det lätt att sätta ihop en PCB som passar. Det som återstår är en fråga: Var ska man göra av den? För det första måste den, på grund av maskinens lustiga uppbyggnad, finnas i VDP RAM. De som finns där från början ligger på höga adresser, i det område som dokumentationen till assemblern anger som "Dynamic DSR allocations". Diverse försök att placera ytterligare en PCB i anslutning till dessa gav vid handen att det nog går i teorin, men tyvärr inte i praktiken. VDP RAM utnyttjas så hårt av vissa systemprogram, att det helt enkelt inte finns plats för något mer. Därmed är det naturligtvis inte bättre att utnyttja PME:ns minnesutrymme i andra änden av VDP RAM. Återstår då att inkräkta på exempelvis teckendefinitionerna eller i Sprite Motion Table. Så länge man kör med textmode är åtminstone den sistnämnda inte till någon annan nytta.

Systemet måste naturligtvis veta var PCB:n tagit vägen. Det finns en tabell som innehåller adresser till de olika PCB som används av olika I/O-enheter. Genom att lagra adressen till vår nya PCB där är den detaljen avklarad. Sedan finns det ytterligare datastrukturer som beskriver en I/O-enhet, men dessa kan systemet självt fylla i. Ett anrop av proceduren UNITCLEAR tar hand om den saken.

Avrundning

Då p-systemet har en egen filhantering, måste RAM-disken, liksom disketterna, förses med en filkatalog som systemet kan förstå. Det gör man lätt med kommandot Zero i Filern.

Eftersom systemprogrammen (editor, kompilator osv) är relativt stora, och därmed kraftigt segmentera-

de, är det en fördel att ha dessa på RAM-disken. För att det ska gå fortare att starta dessa program, finns det en tabell i systemet som talar om vad de har för filnamn. Om de ska placeras på RAM-disken måste den tabellen ändras, så att systemet verkligen laddar in dem därifrån. Enklart är att förutsätta att alla systemfiler finns på RAM-disken. Skulle så inte vara fallet, letar systemet ändå upp filerna där de nu är, och korregerar tabellen helt automatiskt. Det tar visserligen en stund, speciellt om man har fyra drivverk, men behöver bara göras en gång.

Konstigare är det inte, när man väl vet det, vill säga. Med kännedom om ovanstående kan man skriva ett program som sköter om installationen. Ett exempel på ett sådant har jag nu tillfört de disketter som ingår i det paket som jag åstadkom för att en SIG för Pascal skulle ha något att starta på. Det visade sig dock, men några få undantag, att de som påstod sig ha ett intresse för en sådan SIG, i själva verket visserligen var intresserade av att ta nytta av sådant som andra gjort, men mindre intresserade av att bidra med något själva. Men det hindrar ju inte att de ursprungliga disketterna fortfarande finns tillgängliga.

Programmet har visat sig fungera med CorComps RAM-disk och även med den enkla, simulerade RAM-disk som jag själv gjorde genom att förse mitt I/O-kort med ett lämpligt program (PB 88-2). Det finns förvisso fler typer av RAM-diskar, men dem har jag inte haft möjlighet att prova med ännu.

För att kunna skriva installationsprogrammet måste man naturligtvis veta inte bara vilka olika tabeller som finns i systemet, utan också var de är placerade. Jag har sammanställt det jag vet om detta i några textfiler, som också finns på disketterna till SIG Pascal.

Om lämpligt utrymme finns, kanske redaktören rent av fått med programmet i den här tidningen.

Myarc-disk till p-systemet

Ett annat problem med p-systemet är att det har svårt för att komma igång om datorn är försedd med ett disk-kontrollkort av Myarcs fabrikat. Det här upptäckte vi när vi laborerade med den första Geneven (se PB 87-1).

Här har dock botemedel infunnit sig. En herre vid namn Jerry Coffey, Virginia, USA, har forskat i problemet och åstadkommit en diskett som kan användas som systemskiva. Anledningen till att det inte fungerar utan speciella åtgärder är att Myarcs kontrollkort inte fungerar ordentligt om inte sektor noll på skivan lästs. Där finns information om hur skivan är formaterad, information som kortet behöver för att förstå resten av skivan. Kortet förutsätter att det program, som vill komma åt något på skivan, alltid självt beordrar läsning av sektor noll först. Även om program i allmänhet faktiskt gör så, visar det här klart vådan av att hänga upp "lyckan" i en alltför tunn tråd!

När p-systemet ska starta, brister tråden. Från p-systemet, som aldrig utnyttjar de fyra första sektorerna på en skiva, kommer aldrig någon order om att läsa sektor noll. Därmed begriper sig kontrollkortet heller inte på disketterna. Resultatet blir att inget fungerar. Nåja, nästan inget, då. Det första spåret kan man nämligen alltid komma åt, utan att ha läst sektor noll. Annars skulle det ju inte gå att läsa sektor noll heller.

Man måste alltså se till att sektor noll verkligen läses av på samtliga disketter. Det får då göras av programmet SYSTEM.STARTUP, så att det är gjort innan användaren kommer in i systemet. För att det nu ska gå att köra det programmet, måste det i sin helhet vara lagrat i det första spåret på systemskivan. Några yvigare manövrer finns det alltså inte plats till. Däremot går det bra att låta det första programmet starta ett annat, som kan finnas på godtycklig plats på skivan.

Men det här gäller bara så länge formatet på skivorna som är on-line (sitter i drivverken, alltså) är desamma. Om man i ett drivverk byter till en skiva med ett annat format, måste man köra programmet som lä-

ser sektor noll igen. Annars fortsätter kontrollkortet i gamla spår, vilket naturligtvis gör att datorn inte förstår sig på den nya disketten.

Detta innebär i sin tur att det måste finnas två program som ser till att sektor noll läses av. SYSTEM.STARTUP går ju inte att starta med Xecute. Dessutom vill man säkert inte länka in ett eventuellt större startprogram varje gång en skiva med annat format sätts in.

Om du behöver en systemskiva som löser det här problemet, går det bra att höra av sig. Det finns också några andra program och artiklar på skivorna som Coffey ordnat, bland annat en terminalemulator för p-systemet.

MERGE med kassettbandspelare

Det här är en kär (?) gammal bekant. Senast problemet diskuterades i Utmaningen var i PB 85-1. Trots att det mesta av den behövliga grundforskningen presenterades då, har ingen inom föreningen mig veterligt gjort något av det.

Men i det allra sista numret av tyska *Computer Kontakt* (tidningen är numera nedlagd), nr 2-3/88 finns en artikel med titeln "MERGE mit dem Cassettenrecorder". Programmet är av den enklare varianten, som kräver att maskinisten själv ser till att radnummerna i programmen inte kolliderar. Med lämpligt utnyttjande av RESEQUENCE är det ju inte alltför svårt.

Programmet kräver Extended BASIC och minnesexpansionen. MERGE är ju mest intressant i X-BASIC, då dess SUB program är speciellt lämpade att stoppas in i olika program. Arbetsgången är följande:

Det första programmet laddas in och sparas med CALL LINK("SAVE1").

Det andra programmet, som då ska ha samtliga radnummer högre än det första, laddas in och sparas med CALL LINK("SAVE2").

Programmen slås sedan samman med CALL LINK("MERGE").

Därefter kan hela härligheten lagras som ett program på vanligt vis. Dessutom finns en möjlighet att lagra program som är större än 12 288 bytes. Det vanliga kommandot SAVE kräver ju att programmet i sin helhet får plats i en buffert i VDP RAM, vilket effektivt begränsar lagringen av program som annars visst kan både matas in och köras. Speciellt med möjligheten till MERGE är ju risken överhängande för att programmen blir för stora för SAVE.

I Computer Kontakt finns både ett X-BASIC program med en massa DATA satser och källkod till programmet. Nu är det ju så att det inte utan vidare går att publicera sådant som funnits i andra tidningar. I det här fallet är det nog lika så gott det, eftersom källkoden ser alldeles bedräglig ut. Idén som sådan går däremot utmärkt att ta vara på. Eventuellt åstadkommer jag någon variant på deras program, som då förhoppningsvis ser vettigare ut. Det programmet kan då kanske publiceras i PB vid ett senare tillfälle.

Disketterrutiner

Kontrollkortet för skivminne från Myarc och CorComp har en del extra funktioner, som inte har ett skvatt med disketter att göra. PEEK och POKE, exempelvis. Även motsvarigheter till Load and Run finns, samt en del annat.

Men även du som har ett kontrollkort från TI har dolda tillgångar. Det finns sju subprogram, som tyvärr inte kan anropas direkt från BASIC, på alla kontrollkort. De utför dock nyttiga funktioner, som kan vara bra att känna till om du funderar på något program som har med filer på disketter att göra.

Orsaken till att programmen inte låter sig anropas med CALL är egentligen två. För det första är namnen på programmen så valda, att någon tangent som ger de koderna inte finns. För det andra ska informationen som programmen behöver för att fungera finnas i FAC (flyttalsackumulatören) vid anropet. Då BASIC saknar

mekanismer för att påverka FAC direkt, innebär detta gör att det bara går att anropa de här rutinerna från ett assemblerprogram. Det blir ganska vidlyftigt att redovisa alla detaljerna runt anropen, varför jag nöjer mig med att beskriva respektive rutins uppgift. Om intresse finns får en utförligare redogörelse komma i ett senare nummer.

En rutin kan läsa och skriva en godtycklig sektor på disketten. Den utnyttjas av många program som inte använder vanliga filer.

Formatera en diskett kan man också göra. Det går naturligtvis att ange hur många spår och sidor man vill ha formaterade, men även densitet för kontrollkort som klarar mer än enkel densitet. Med CorComps kort går det också att bestämma vilken ordning sektorerna ska hamna i, men det är något som inte var avsett från början. Därför görs detta utanför anropet av rutinen för formatering. Jag förmodar att Myarcs kort har motsvarande möjlighet, men det har jag ingen erfarenhet av.

Filer kan skyddas mot oavsiktlig radering. Genom att anropa en rutin för detta, går det att sätta dit eller ta bort detta skydd. Det finns också en rutin för att byta namn på filer.

Med det filsystem som 99:an har, kan filerna vara av många olika typer. DISPLAY och INTERNAL kan kombineras med VARIABLE och FIXED i olika längder. Dessutom finns typen PROGRAM. Men när man ska kopiera filerna ställer det till problem. En fil av en viss typ kan bara öppnas med ett kommando som specificerar just den typen av fil. Då vore det enklare om man kunde betrakta filerna som ett antal intressanta

sektorer, utan att för den skull behöva ta hänsyn till vad som egentligen finns lagrat i dessa sektorer.

Två rutiner på kontrollkortet ger just denna möjlighet. Med dessa kan man först få reda på hur stor en fil är, sedan skapa en ny sådan fil och slutligen läsa och skriva sektorer från originalet till kopian.

Slutligen finns det en rutin som motsvarar CALL FILES, men som alltså kan anropas från ett assemblerprogram.

Klockor och sånt (igen!)

I PB 88-1 redovisar Per Virving sina intryck av EDP från *Paragon Computing*. Han avslutade sin artikel med en utmaning om att någon kunnig assemblerprogrammerare skulle modifiera EDP:s klocka så att den går rätt även i datorer som är avsedda för 50 Hz. Visst, det går säkert bra för den som vill studera programmet ingående. Men Paragon tillåter inte distribution av sitt program om någon fil är ändrad på något sätt. En enklare metod vore kanske att utnyttja den i programmets instruktion utlovade möjligheten att påpeka eventuella fel och få en korrigerad version tillbaka. Det finns också en möjlighet att beställa en kommenterad källkod från Paragon Computing. Med den att tillgå är det naturligtvis betydligt lättare att rätta till klockan. Även om det väl inte innebär någon licens för föreningen, eller någon annan, att distribuera förändrade kopior av programmet, kan ju ingen förhindra en publicering av en beskrivning av hur modifikationen ska utföras.

Slut för den här gången. I vanlig ordning är alla välkomna med bidrag till Utmaningen, både problem och lösningar.

SISTA CHANSEN!!

Nu har Du absolut sista chansen att köpa föreningens tröjor som inte lär komma att tryckas upp någon mer gång. De finns i två olika utföranden samt i några olika storlekar. En typ är försedd med texten "Föreningen Programbiten" samt finns i storlek 4 (7 st), 5 (15 st) och 6 (8 st). Den andra typen av tröja är försedd med den kända Texas-bilden samt texten "Nittinian". Denna tröja finns i följande storlekar och antal: storlek 2/4=barnstorlek (4 st), S (3 st), M (30 st), L (1 st) och XL (5 st).

Eftersom antalet är mycket skiftande så gäller naturligtvis att den som beställer först har bästa chansen att välja. Kvaliteten på tröjorna är bra och de håller för åtskilliga tvättar.

Pris per tröja är 40:—/styck inklusive porto. Sätt in beloppet motsvarande det antal Du önskar på föreningens postgiro 198300-6 samt skriv storlek, typ och antal på talongen. Ange också ditt beställningsdatum.

Alltså, passa på att köpa dig ett trevligt plagg att ta på vid arbetet vid 99:an!

Hela skivor i Programbanken

av Jan Alexandersson

Detta är ett försök att beskriva de olika skivor som finns i programbanken med reservation för att det kan ha skett förändringar eller att jag missuppfattat något. Alla skivor kan beställas från föreningen för 30:- per styck. Följande förkortningar finns:

OP = operativsystem eller styrprogram
AL = Assembler
EM = Expansionsminne 32k
DS = Disk drive
XB = Extended Basic
EA = Editor/Assembler
GR3 = Editor/Assembler med GROM3
MM = Mini Memory
TW = TI-Writer
FW = Funnelweb körd med valfri modul
TE = Terminal Emulator II
DSK = antal diskar
* = består av program som kräver olika tillbehör

FAIRWARE (FREEWARE) 52 PROGRAM

NAMN	VERS	KÖRBART----MED									
		OP	EM	DS	XB	EA	MM	TW	FW	DSK	
MENU (för Horizon)	7.3	AL	EM	DS	-	EA	-	TW	FW	1	
J.Johnson+M.Ballmann, FL, USA											
MENU KÄLLKOD	7.3	AL	EM	DS	-	EA	-	-	FW	2	
J.Johnson+M.Ballmann, FL, USA											
DISK MANAGER 99	85-10	BA	EM	DS	XB	EA	MM	-	-	1	
Mike Dodd, Tenn, USA											
DISK MANAGER 1000 3.5(3.8)		AL	EM	DS	XB	EA	-	TW	FW	1	
Ottawa User Group, Canada											
3.5 mera felfri än 3.8											
DM 1000 KÄLLKOD	3.8	AL	EM	DS	-	EA	-	-	FW	2	
Ottawa Users Group, Canada											
ARCHIVER II	2.4	AL	EM	DS	XB	EA	-	TW	FW	1	
Barry Boone, OK, USA											
Disk Utilities	4.0A	AL	EM	DS	XB	EA	-	TW	FW	1	
John Birdwell, MN, USA											
DISK CAT	2.0	AL	EM	DS	XB	EA	MM	TW	FW	1	
Mack McCormick, USA											
DISK-AID		FH	EM	DS	XB	EA	-	-	FW	1	
Col Christensen, Australien											
DISK HACKER	2.0	AL	EM	DS	XB	EA	MM	-	FW	1	
Will McGovern, Australien											
COLIST	1.0	XB	EM	DS	XB	-	-	-	-	1	
Tony+Will McGovern, Australien											
STAR	1.0	XB	EM	-	XB	-	-	-	-	1	
Michael Riccio, PA, USA											
ENHANCED DISPLAY	2.1	XB	EM	-	XB	-	-	-	-	1	
Pargon Computing, NH, USA											
TEXTLOADER/EA5LOAD		XB	EM	DS	XB	-	-	-	-	1	
Pargon Computing, NH, USA											
PULSAR	1.2	AL	EM	DS	-	EA	-	-	FW	1	
Michael Amundsen, OH, USA											
XB*TOOLS	1.2	XB	EM	DS	XB	-	-	-	-	1	
Jim Swedlow, CA, USA											
NEATLIST	1.0	XB	EM	-	XB	-	-	-	-	1	
Danny Michael, AL, USA											
MACRO ASSEMBLER	6	AL	EM	DS	XB	EA	-	TW	FW	2	
R.A.Green, Canada											
LINKER	2	AL	EM	DS	XB	EA	-	TW	FW	1	
R.A.Green, Canada											

LINKER LIBRARIAN		AL	EM	DS	XB	EA	-	TW	FW	1	
Tom Bentley											
UNIVERSAL DISSASSEMBLER 2.3		FH	EM	DS	-	EA	MM	-	FW	1	
Rene LeBlanc, AZ, USA											
SUPERBUG II	2.1	AL	EM	-	XB	EA	MM	-	FW	1	
Edgar Dohmann, TX, USA											
TK-WRITER		AL	EM	DS	XB	EA	MM	-	-	1	
Tom Knight, FL, USA											
FUNNELWEB JUL/4/88	4.11	AL	EM	DS	XB	EA	MM	TW	-	2	
Tony+Will McGovern, Australien											
RAPID SCROLL	1.4+2.3	AL	EM	DS	XB*EA*MM*TW*FW*					1	
Jurgen Switalski, MI, USA											
PRBASE	2.0	AL	EM	DS	XB	EA	MM	TW	FW	2	
William Warren, CO, USA											
CREATIVE FILING SYST. 7.0		XB	EM	DS	XB	-	-	-	-	3	
Mark Beck, AR, USA											
BASIC SORT	1.0	BA	EM	DS	XB	EA	MM	-	-	1	
Andreas Dessooff, CT, USA											
SCREENDUMP	3.01	BA	EM	-	XB	EA	MM	-	-	1	
Danny Michael, AL, USA											
MAX-RLE/1 EA		AL	EM	DS	-	EA	MM	-	FW	1	
Travis Watford, USA											
MAX-RLE/13 XB		AL	EM	DS	XB	EA	-	TW	FW	1	
Travis Watford, USA											
MAX-RLE/11 (endast bilder)		MAX-RLE								1	
FRACTAL EXPLORER	2.00	AL	EM	DS	-	EA	MM	-	FW	2	
Steve Langguth, MO, USA											
S/FOSTER		XB	EM*DS*XB	-	-	-	-	-	-	1	
Stephen Foster, USA											
KNECHT-1		XB	-	-	XB	-	-	-	-	1	
Bill Knecht, USA											
KNECHT-2		XB	-	-	XB	-	-	-	-	1	
Bill Knecht, USA											
EMUSIC PREPROCESSOR	1.2	XB	EM	DS	XB	-	-	-	-	2	
Norm Sellers, PA, USA											
STARTREK		XB	EM	DS	XB	-	-	-	-	1	
Ken Gilliland, CA, USA											
WAGNER 1	88-01	XB	EM	DS	XB	-	-	-	-	2	
Ken Gilliland, CA, USA											
WAGNER 2	88-01	XB	EM	DS	XB	-	-	-	-	2	
Ken Gilliland, CA, USA											
SOUTH PACIFIC	88-01	XB	EM	DS	XB	-	-	-	-	1	
Ken Gilliland, CA, USA											
PATSY CLINE	88-01	XB	EM	DS	XB	-	-	-	-	1	
Ken Gilliland, CA, USA											
MUSIC MAN	88-01	XB	EM	DS	XB	-	-	-	-	1	
Ken Gilliland, CA, USA											
WIZARD OF OZ	88-01	XB	EM	DS	XB	-	-	-	-	1	
Ken Gilliland, CA, USA											
C99 RELEASE 3		AL	EM	DS	-	EA	-	-	FW	2	
Clint Pulley, Canada											
GR3											
C99 EXTRA		AL	EM	-	XB*EA*MM*	-	FW*			1	
Clint Pulley, Canada											
C-STUFF		AL	EM	-	-	EA	-	-	-	1	
Clint Pulley, Canada											
GR3											
PILOT 99		FH	EM	DS	-	EA	-	-	-	2	
Thomas Weithofer, USA											

FAST-TERM	1.16	AL EM DS XB EA MM TW FW	1
Paul Charlton, Virginia, USA			
TELCO	2.1	AL EM DS XB EA - TW FW	2
Charles Earl, Canada			
AUTO DIALER	2.0	FAST-TERM	1
John Birdwell, MN, USA			
AMNION HELPLINE		XB - - XB - - - -	8
(även som separata program)			

AV TI FRISLÄPPTA 6 PROGRAM

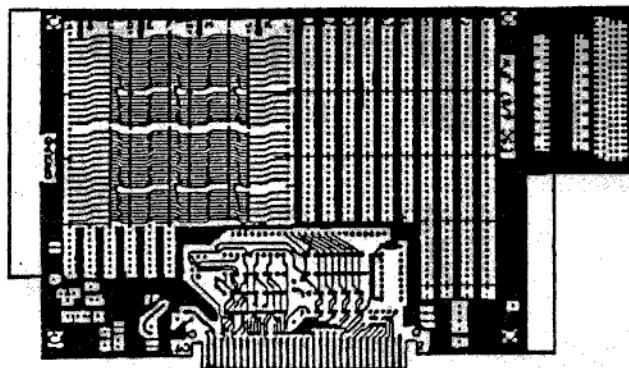
NAMN	VERS	OP EM DS XB EA MM TW FW DSK	
TI-FORTH	4	FH EM DS - EA - - FW	1
TI-FORTH XB	4	FH EM DS XB - - -	1
TI-FORTH Källkod		FH EM DS - EA - FW	3
WRITE/MULTI UPDATE		AL EM DS TW*	1
SUPERDEBUG	3.1	AL EM	1
TEST		XB+MM EM DS XB* - MM* - -	2

SKRIVNA AV MEDLEMMAR 9 PROGRAM

NAMN	VERS	OP EM DS XB EA MM TW FW DSK	
PB-FORTH	2.0	FH EM - XB EA MM TW FW	1
Björn Gustavsson			
PB-FORTH Källkod	2.1	FH EM - XB EA MM TW FW	3
Björn Gustavsson			
FORTH OCH 99		FH EM DS XB EA MM TW FW	3
Lars-Erik Svahn			
Game FORTH		FH	1
Lars-Erik Svahn			
PB-DISASSEMBLER		AL EM EA MM	1
Tony Rogvall			
FORIT		FH EM DS - EA - - FW	1
Bo Carleö			
CHECKLIST		XB EM - XB	1
Lars Thomasson			
PERFECT PUSH		AL EM DS	1
Martin Flodén			

SAMLINGSDISKAR 7 PROGRAM (samtliga även på samlingskasset)

NAMN	VERS	OP EM DS XB EA MM TW FW DSK	
NITTINIAN 1983			1
PROGRAMBITEN 1984			1
PROGRAMBITEN 1985			1
PROGRAMBITEN 1986			1
SPEL BASIC			1
SPEL EXTENDED BASIC			1
MUSIK EXTENDED BASIC			1



Prototypkort

För den som vill experimentera med egna byggen finns nu ett prototypkort för expansionsboxen. Kortet är förberett för de kretsar som behövs för interface till bussen och alla kontrollsignaler kan nås från kortet. Tanken är att den som vill bygga egna kort ska kunna utgå från detta och sen lägga till sina egna funktioner utan att behöva uppfinna hjulet på nytt. Kortet kostar 35 dollar + frakt och säljs bla: av

L.L. Conner Enterprise
1521 Ferry Street
Lafayette, IN 47904
USA

Styrkort för hårddiskar från Myarc

av Peter Odelryd

Winchesterdiskar till 99-an har det funnits länge. Företaget *Myarc* blev först kända inom 99-världen genom sina hårddiskar som introducerades i slutet av 1982. Den billigaste modellen kostade då 2599 dollar för en separat box med kontrollkort och en 5 Mb winchester. På senare tid har ju priset sjunkit enormt på hårddiskar. Idag ligger priset för en 20 Mbytes hårddisk på ungefär samma nivå som priset för 2 diskettstationer. I USA rör det sig runt 250 dollar. I Sverige runt 2500 + moms.

Problemet är kontrollkortet. Myarc har nyligen släppt sitt nya kontrollkort. Det kan styra både hårddiskar och upp till fyra diskettstationer. Man kan också koppla in en bandstation för streamerkassetter för att göra backup på hårddisken. Upp till 3 hårddiskar kan anslutas, vardera på upp till 135 Mb. Det blir 400 Mb totalt...

Den disktyp som kan anslutas är den med ST 506/412-interface, dvs samma som till vanliga PC. Enligt en artikel i aprilnumret av *MICROpendium* så organiserar man filerna på hårddisken i en huvudkatalog och därefter i upp till 114 underkataloger. Varje sådan kan sen innehålla upp till 127 filer. På det sättet blir det lätt att hitta det man söker efter. Man anropar en fil med exempelvis följande namn: WDSL.TEXTER.BREV.BOSSE8804. En klocka på kortet märker filerna med datum.

Myarc verkar ha tänkt till en del när det gäller hur man kan använda disken också. Program som kräver att startas från DSK1 fungerar också genom att systemet först letar på hårddisken efter en underkatalog med namnet DSK1. Multiplan skall också kunna fungera om man har en katalog med namnet DSK.TIMP.

Kortet kostar runt 310 dollar i USA. Enligt en artikel i *MICROpendiums* maj-nummer passar kortet för närvarande bara till 99/4A. Tydligt är Myarc inte klara med anpassningen till Geneve ännu.

En utförligare artikel om detta kort kommer i nästa nummer av Programbiten.

av Fjellstedt

```

620 CALL DELSPRITE(#3):: BO=BO+1 :: SKOTT=0
  :: CALL HCHAR(24,2,41,BO)
630 CALL COLOR(#2,7):: CALL PATTERN(#2,116)::
  : FOR I=1 TO 50 :: NEXT I :: CALL COLOR(
  #2,16):: CALL PATTERN(#2,44):: CALL COLO
  R(#2,16)
640 CALL SOUND(100,-5,0):: CALL SOUND(400,-7
  ,0):: FOR I=1 TO 20 :: CALL SOUND(-92,16
  0-I,0):: NEXT I
650 CALL SOUND(500,-7,0)
660 CALL COLOR(#2,2):: CALL PATTERN(#2,48)::
  FOR I=1 TO 5 :: NEXT I
670 CALL DELSPRITE(#2)
680 CALL DISTANCE(#2,166,128,D):: PO=PO+5*IN
  T(10000/SQR(D)):: GOTO 510
690 IF SKOTT=0 THEN 520 ELSE CALL COINC(#3,#
  2,9,TR):: IF TR THEN 620
700 GOTO 520
710 CALL JOYST(1,X,Y):: CALL MOTION(#1,-Y*9,
  X*9):: RETURN
720 CALL DELSPRITE(ALL)
730 CALL SOUND(200,523,0,392,0,330,0)
740 CALL CLEAR
750 CALL MAGNIFY(3)
760 CALL CHARSET
770 REM STRIDSA6N
780 CALL CHAR(104,"0000804020100805030301FFF
  F7F3F0F00000000000000000E0E0FBE0FFFFFEFCF
  0")
790 REM SIKTE
800 CALL CHAR(108,"000000000000000000000000
  000000000000000000000000180000999900001B"
  )
810 REM eldsлага
820 CALL CHAR(96,"0000010103030FFF3F0F030301
  01010180808080C0C0FOFFFCFC0C0C0808000000")
830 REM MYNNINGSFLAMMA
840 CALL CHAR(116,"00000000000000000000000000
  0000000000000000000020101020000000000")
850 REM FLYGPLAN
860 CALL CHAR(128,"00C0C0C0C0FFFC06020100000
  0000000000003844FC82337BFEEFC103838100000"
  )
870 REM rokmoln
880 CALL CHAR(92,"000000000F0808F8FBFFFFF0F67
  F7F060C06030188CB868383FFFFFFFFFF0E6EFOF06")
890 CALL CHAR(124,"0C3F3F1F0F0F1F3F3F7F7F7FF
  3E1600000008080CCFEFFFFFEFCF0F8FCFCF860"
  )
900 PO=0 :: TIME=100 :: CALL SCREEN(2):: FOR
  X=0 TO 8 :: CALL COLOR(X,2,8):: NEXT X
  :: CALL COLOR(9,4,4)
910 FOR X=11 TO 24 :: CALL HCHAR(X,1,96,32)::
  : NEXT X
920 P=0 :: CALL SPRITE(#2,108,2,200,200)
930 CALL SPRITE(#5,92,2,INT(RND*40)+71,INT(R
  ND*256)+1,0,-INT(RND*32)+(-2))
940 CALL SPRITE(#8,104,9,INT(RND*72)+71,INT(
  RND*256)+1,0,-INT(RND*50)+1)
950 CALL SPRITE(#6,104,2,INT(RND*72)+71,INT(
  RND*256)+1,0,-INT(RND*55)+(-1))
960 CALL SPRITE(#7,104,2,INT(RND*72)+71,INT(
  RND*256)+1,0,-INT(RND*35)+(-1))
970 CALL SPRITE(#3,120,2,57,173,0,30):: CALL
  SPRITE(#4,120,2,45,99,0,50)
980 CALL JOYST(1,X,Y)
990 TIME=TIME-1 :: DISPLAY AT(2,1):"TID: ";T
  IME :: IF TIME<1 THEN 1190
000 CALL MOTION(#2,0,0)
010 CALL MOTION(#2,-Y*6,X*8)
020 CALL KEY(1,K,S):: IF S<>0 THEN 1040
030 GOTO 980
040 CALL POSITION(#2,Q,W)
050 C=0
060 CALL SPRITE(#9,118,10,Q,W)
070 CALL SOUND(180,-7,0)
080 FOR I=1 TO 20 :: NEXT I :: CALL SOUND(40
  0,-5,10)
090 FOR Q=3 TO 8
100 CALL COINC(#9,#0,10,C):: IF C<>-1 THEN 1
  140 ELSE CALL DELSPRITE(#0,#9):: CALL SP
  RITE(#10,96,9,Q,W):: CALL SOUND(400,-5,2
  )
110 CALL SOUND(900,-7,0)
120 CALL SPRITE(#10,124,12,Q,W):: FOR I=1 TO
  150 :: NEXT I :: CALL DELSPRITE(#10)
130 GOTO 980
140 NEXT Q
150 CALL DELSPRITE(#9)

```

21

TI-Artist

av Jan Alexandersson

Jag har nu skaffat TI-Artist från Texaments för USD 20. Programmet består av en flexskiva och en 24 sidor tjock manual. Den har nu version 2.01G och fungerar mycket bra med Horizons RAM-disk. Hela programmet kan lagras på RAM-disken vilket är mycket praktiskt eftersom TI-Artist består av flera olika delar som laddas in vart efter de behövs. Med RAM-disk blir väntetiden mycket liten.

TI-Artist är ett grafikprogram för att skapa bilder. Varje färdig bild lagras som två filer för pixel (FILNAMN_P) respektive färger (FILNAMN_C). Den färdiga bilden kan även användas av MAX-RLE (finns i programbanken) eller Super Extended Basic.

Det är möjligt att ladda programmet från Extended Basic, Mini Memory, Editor/Assembler, TI-Writer eller Funnelweb. Programladdningen som även kräver 32 kb görs av tre olika laddningsprogram:

- LOAD som är XB-program
- ARTIST som är DIS/FIX 80
- ARTIST1 som är PROGRAM-fil

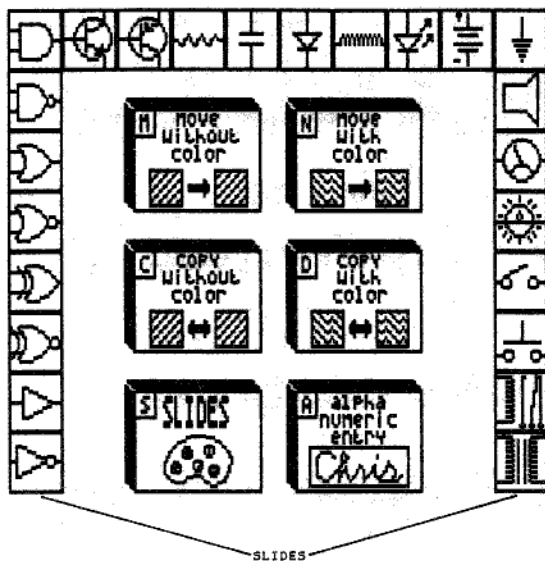
LOAD laddar ARTIST som i sin tur laddar ARTIST1 som sedan laddar in de programdelar som skall användas. Du har dock full frihet att starta med vilket som helst av dessa tre laddningsprogram.

LOAD-programmet kan editeras (på 2 ställen) så att laddningen kan göras från godtycklig disk (DSK1—DSK9) även RAM-disk oberoende av skivnamn.

ARTIST-filen som är i DIS/FIX 80 kan patchas med ett BASIC-program så att man kan ladda från alla disknummer (OBS onödigt för XB) oberoende av skivnamn. Det bästa är att göra om den till PROGRAM-fil med t ex LINKER (finns i programbanken).

ARTIST1-filen i PROGRAM-format laddar efter skivnamn DSK.INSCEBOT. Detta kan vara användbart för DSK1 eller godtycklig RAM-disk på CRU > 1000 om du döper skivan/RAM-disken till INSCEBOT. Det är ej möjligt att söka efter skivnamn på RAM-disk på CRU > 1200 och högre. Att söka efter skivnamn för DSK2 och DSK3

ENHANCEMENT

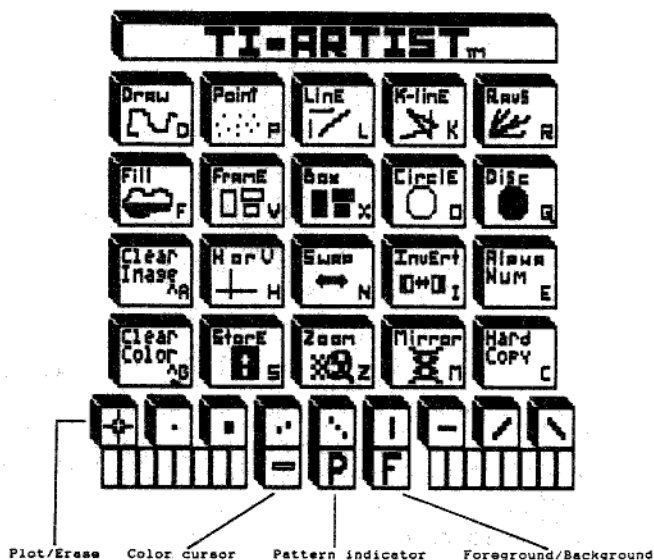


kan vara besvärligt eftersom programmet först startar DSK1 osv innan övriga nummer uppsöks både vid start och hopp mellan olika programsnittar.

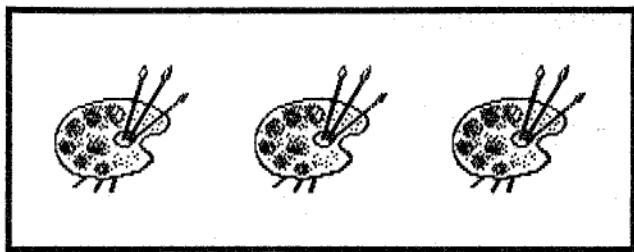
Allt har fungerat bra på RAM-disk både på CRU 1000 och CRU > 1200 utom "Index picture" (ett katalogprogram som visar vilka bilder m m som finns) som ej fungerar på CRU > 1200.

Det finns tre olika DSR-rutiner för att flytta pennan ("cursor") på skärmen varav en för joystick och en för Mechatronic mus (inkl källkod). Vad den tredje DSR-rutinen gör vet jag ej eftersom den ser ut att fungera lika bra som joystick-rutinen.

TI-ARTIST



TI-Artist



for creating, editing, printing

graphics, drawings, artwork

INSCEBOT

P.O. Box 291610
Port Orange, FL 32029

TIPS FROM THE TIGERCUB

#46

Copyright 1987

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, NOW REDUCED TO JUST \$1.00 EACH!, plus \$1.50 per order for cassette or disk and PP&M. Minimum order of \$10.00. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette. Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!

TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH. VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPIES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to \$15.00 postpaid.

NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$15 postpaid.

* NUTS & BOLTS #3 is now *
* ready, another full disk *
* of 140 new merge-format *
* utility subprograms, all *

* compatible with the pre- *
* vious. With 11 pages of *
* documentation, \$15 ppd. *

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to \$10 ppd.

TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$10

TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tips and routines from Nos. 25 through 32, \$10 postpaid.

TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also \$10 postpaid.

TIGERCUB CARE DISKS #1, #2 & #3, three full disks of text files, mostly of lessons on programming in XBasic, \$5 per disk postpaid.

This one is explained in lines 180-190. I think that it will run on any Gemini printer.

```
100 DIM B(25,12),B$(25),CH$(12),L$(12)
110 GOTO 150
120 S,K,T$,C$,V,J,A,CH$( ),X,X$,B$( ),B(X,J),T,M,Q$,L$( ),C,C$,C2$,L,M$
130 CALL CLEAR :: CALL COLOR :: CALL SCREEN :: CALL CHAR :: CALL KEY :: CALL NUMTH
140 !@P-
150 !SEGMENTED BAR GRAPH
    by Jim Peterson 10/87
160 CALL CLEAR :: FOR S=1 TO 12 :: CALL COLOR(S,2,8):: N
EXT S :: CALL SCREEN(5):: DI
SPLAY AT(3,10):"TIGERCUB" ::
    DISPLAY AT(5,6):"SEGMENTED BAR GRAPH"
170 CALL CHAR(95,"3C4299A1A199423C"):: DISPLAY AT(7,12):
" 1987" :: DISPLAY AT(9,2):
"For free distribution but n
o":"price or copying fee may
be":"charged."
180 DISPLAY AT(14,2):" Will
output to a Gemini":"printer
a horizontal bar-":"graph o
f up to 25 bars, each":"segm
ented into up to 12"
190 DISPLAY AT(18,1):"values
, with a title for":"each an
d optionally with a":"table
of identification of":"the s
egment symbols."
200 DISPLAY AT(24,8):" " :: D
ISPLAY AT(24,8):"PRESS ANY K
EY" :: CALL KEY(0,K,S):: IF
S=0 THEN 200
210 ON WARNING NEXT
220 DISPLAY AT(12,1)ERASE AL
L:"GRAPH TITLE?" :: ACCEPT A
T(14,1):T$: TS=RPT$( " ",17
-LEN(TS)/2)&T$: CS=CHR$(27
)
230 DISPLAY AT(16,1):"HOW MA
NY SEGMENTS PER BAR?" :: ACC
EPT AT(16,27)VALIDATE(DIGIT)
SIZE(2):V :: IF V=0 OR V>12
THEN 230
240 !@P+
250 DATA 239,229,168,251,173
,175,184,236,169,250,160,207
260 !@P-
```

```
270 FOR J=1 TO V :: READ A :
: CH$(J)=CHR$(A):: NEXT J
280 DISPLAY AT(3,1)ERASE ALL
:"Type END when finished"
290 X=X+1 :: IF X>25 THEN 33
0
300 CALL NUMTH(X,X$):: DISPL
AY AT(12,1):"Title of "&X$&
"bar?" :: ACCEPT AT(14,1):B$(
X):: IF B$(X)="END" OR B$(X
)="end" THEN 330
310 FOR J=1 TO V :: CALL NUM
TH(J,X$):: DISPLAY AT(16,1):
X$&" segment value?" :: ACCE
PT AT(18,1)VALIDATE(NUMERIC)
:B(X,J):: T=T+B(X,J):: NEXT
J
320 M=MAX(M,T):: T=0 :: GOTO
290
330 X=X-1 :: DISPLAY AT(20,1
):"Print labels? Y/N" :: ACC
EPT AT(20,19)VALIDATE("YN")S
IZE(1):Q$: IF Q$="N" THEN
350
340 FOR J=1 TO V :: CALL NUM
TH(J,X$):: DISPLAY AT(22,1):
X$&" label?" :: ACCEPT AT(24
,1):L$(J):: NEXT J
350 C=120/M :: C1$=C$&"B"&CH
R$(1)&C$&"G"&C$&"E" :: C2$=C
$&"B"&CHR$(3)
360 OPEN #1:"PIO",VARIABLE 2
55 :: PRINT #1:C$&"@" :: PRI
NT #1:C$&"E"&C$&"G"&C$&"M"&C
HR$(6)
370 PRINT #1:CHR$(14)&T$&CHR
$(20):"":RPTS(CHR$(229),70):
: :: PRINT #1:C$&"3"&CHR$(1
0)
380 FOR J=1 TO X :: PRINT #1
:B$(J)&C2$ :: FOR L=1 TO V :
: M$=M$&RPTS(CHR$(L),INT(B(J
,L)*C+.5)):: NEXT L
390 PRINT #1:RPTS(CHR$(232),
LEN(M$)):: PRINT #1:M$ :: PR
INT #1:M$ :: PRINT #1:RPTS(C
HR$(231),LEN(M$))
400 M$="" :: PRINT #1:C1$::
NEXT J :: IF Q$="N" THEN ST
OP
410 PRINT #1:"":
420 FOR J=1 TO V :: PRINT #1
:C2$&RPTS(CHR$(232),10):: PR
INT #1:RPTS(CHR$(J),10)&C1$&
"&L$(J):: PRINT #1:C2$&
RPTS(CHR$(J),10):: PRINT #1:R
PTS(CHR$(231),10):: NEXT J
430 !@P+
440 SUB NUMTH(N,N$):: IF FLA
G=1 THEN 520 :: FLAG=1 :: RE
STORE 480
450 GOTO 480
460 J,ONES(),TEENS(),TENS(),
N,N$
470 !@P-
480 DATA first,second,third,
fourth,fifth,sixth,seventh,e
ighth,ninth,tenth
490 DATA eleventh,twelfth,th
irteenth,fourteenth,fifteenth
h,sixteenth,seventeenth,eigh
teenth,nineteenth
500 DATA twenty,THIRTY,FORTY
,FIFTY,SIXTY,SEVENTY,EIGHTY,
NINETY
510 FOR J=1 TO 10 :: READ ON
ES(J):: NEXT J :: FOR J=1 TO
9 :: READ TEENS(J):: NEXT J
:: FOR J=2 TO 9 :: READ TEN
$(J):: NEXT J
520 IF N<11 THEN N$=ONES(N):
: SUBEXIT
530 IF N<20 THEN N$=TEENS(N-
10):: SUBEXIT
540 IF N/10=INT(N/10)THEN N$
=SEGS(TEENS(N/10),1,LEN(TENS(
N/10))-1)&"ieth" :: SUBEXIT
550 N$=TENS(INT(N/10))&"-"&O
NES((N/10-INT(N/10))*10)
```



```
560 !@P+
570 SUBEND
```

And a little something educational -

```
100 DIM M$(100)
110 GOTO 150
120 S,J,M$( ),A$,Z$,K,W$( ),X,
Y,ADV$,A,Q$
130 CALL CLEAR :: CALL COLOR
:: CALL SCREEN :: CALL CHAR
:: CALL KEY :: CALL ADVERB
:: CALL SOUND
140 !@P-
150 CALL CLEAR :: FOR S=0 TO
12 :: CALL COLOR(S,2,8):: N
EXT S :: CALL SCREEN(5):: DI
SPLAY AT(3,2):"ADJECTIVE TO
ADVERB V.1.3"
160 CALL CHAR(64,"3C4299A1A1
99423C"):: DISPLAY AT(5,6):"
© Tiger Cub Software":: For
free distribution with no
charge or copying fee."
170 FOR J=1 TO 100 :: READ M
$(J):: A$=A$&CHR$(J):: NEXT
J :: Z$=A$ :: CALL KEY(3,K,S
)
180 W$(1)=" If adjective end
s in Y, change the Y to
ILY." :: W$(2)=" If adjectiv
e ends in C, add ALLY."
190 W$(3)=" If adjective end
s in LL, just add Y."
200 W$(4)=" If adjective end
s in LE, preceded by a con
sonant, drop the E and ad
d Y."
210 W$(5)=" If the word ends
in E preceded by a con
sonant, preceded by a vow
el, just add LY."
220 W$(6)=" This word is an
exception to the rule - the
adverb is WHOLLY."
230 W$(7)=" If the adjective
does not end in C,E,LL or
Y, always just add LY."
240 W$(8)=" This is an excep
tion to the rule. The prefer
red adverb form is DRYLY."
250 W$(9)=" If the adjective
ends in E preceded by a vo
wel, drop the E and add LY
."
260 W$(10)=" If the adjectiv
e ends in E preceded by a co
nsonant other than L, ad
d LY."
270 RANDOMIZE :: X=INT(RND*L
EN(Z$)+1):: Y=ASC(SEG$(Z$,X,
1)):: Z$=SEG$(Z$,1,X-1)&SEG$
(Z$,X+1,255):: IF LEN(Z$)=0
THEN Z$=A$
280 ACCEPT AT(24,1):M$(Y)
290 CALL ADVERB(M$(Y),ADV$,A
)
300 DISPLAY AT(12,1):" Type
the adverb form of -" :: DIS
```

```
Y AT(18,10):"" :: ACCEPT AT(
15,15)BEEP:QS
310 IF Q$=ADV$ THEN DISPLAY
AT(18,10):"CORRECT!" :: GOTO
240
320 CALL SOUND(100,110,5,-4,
5):: DISPLAY AT(20,1):W$(A):
"" :: GOTO 300
330 !@P+
340 DATA DUE,COOL,SOLE,STOIC
,FRANTIC,COMIC,ABLE,FULL,POO
R,HANDY,SORE,SOCIAL,PENAL,SL
OW,HIGH,LOW
350 !@P-
360 DATA FRISKY,PLAYFUL,HEAL
THY,ROUGH,BUSY,SILLY,SICK,SM
ART,SORE,FAIR,ANGRY,BARE,TIR
ED,WISHFUL,ACTUAL
370 DATA HASTY,LONE,HECTIC,O
FFICIAL,MAGIC,MAGICAL,MATHEM
ATIC,LOGIC,TRAGIC,PATHETIC,T
RAUMATIC
380 DATA DRAMATIC,AUTOMATIC,
AROMATIC,EQUAL,SERIAL,BASIC,
USUAL,FAVORABLE,UNSTABLE,LEG
IBLE
390 DATA HECTIC,LIVE,WARY,VI
SIBLE,TERRIBLE,HORRIBLE,VIVI
D,FANCY,EASY,VILE,WICKED,BLO
ODY,SHODDY
400 DATA NOBLE,HAPPY,LEGAL,M
ERRY,JOLLY,CRAZY,CASUAL,CARE
FUL,FOOLISH,FAMOUS,GAY,GUILT
Y
410 DATA HOPEFUL,HATEFUL,TIM
ID,BRAVE,BEAUTIFUL,DRY,NICE,
LARGE,PAINFUL,SINFUL,SORROWF
UL,SIMPLE,WILLFUL
420 DATA MENTAL,MORAL,PALE,W
HOLE,HUNGRY,FINAL,FORMAL,TRU
E,AMPLE,DOUBLE
430 !@P+
440 SUB ADVERB(M$,ADV$,A)::
L=LEN(M$):: E$=SEG$(M$,L,1):
F$=SEG$(M$,L-1,2):: G$=SEG
$(M$,L-1,1):: P$=SEG$(M$,1,L
-1):: H$=SEG$(M$,L-2,1)
450 IF ASC(SEG$(M$,1,1))<97
THEN A$="ALLY" :: IS="ILY" ::
L$="LY" :: Y$="Y" :: V$="A
EIOU" ELSE A$="ally" :: IS="
ily" :: L$="ly" :: Y$="y" ::
460 IF M$="WHOLE" THEN ADV$=
"WHOLLY" :: A=6 :: SUBEXIT
470 IF M$="DRY" THEN ADV$="D
RYLY" :: A=8 :: SUBEXIT ELSE
IF F$="LL" OR F$="ll" THEN
ADV$=M$&Y$ :: A=3 :: SUBEXIT
480 IF E$="C" OR E$="c" THEN
ADV$=M$&A$ :: A=2 :: SUBEXI
T ELSE IF E$="Y" OR E$="y" T
HEN ADV$=P$&I$ :: A=1 :: SUB
EXIT
490 IF E$<>"E" AND E$<>"e" T
HEN 530
500 IF G$="L" OR G$="l" THEN
IF POS(V$,H$,1)<>0 THEN ADV
$=M$&L$ :: A=5 :: SUBEXIT EL
SE ADV$=P$&Y$ :: A=4 :: SUBE
XIT
510 IF POS(V$,G$,1)<>0 THEN
ADV$=P$&L$ :: A=9 :: SUBEXIT
```

```
520 IF POS(V$,SEG$(M$,L-2,1)
,1)=0 THEN ADV$=M$&L$ :: A=1
0 :: SUBEXIT ELSE ADV$=M$&L$
:: A=5 :: SUBEXIT
530 ADV$=M$&L$ :: A=7 :: SUB
END
```

```
100 !MOCKINGBIRD TINYGRAM by
Jim Peterson. Tap your
tune on the 1 to 0 keys
(tuned A through C)
110 !Then press any other
key to hear it repeated
120 DATA 220,247,262,294,330
,349,392,440,494,523
130 FOR J=1 TO 10 :: READ N(
J):: NEXT J :: J=0 :: DIM T(
50,2)
140 CALL KEY(5,K,S):: IF S=0
THEN 140
150 ON ERROR 190
160 CALL KEY(5,K,S):: IF K=-
1 THEN 160 :: K=K-(K=48)*10
:: T(J,1)=N(K-48):: CALL SOU
ND(-999,T(J,1),0)
170 IF K=K2 THEN T(J,2)=T(J,
2)+1 :: GOTO 160
180 K2=K :: J=J+1 :: GOTO 16
0
190 FOR X=0 TO J-1 :: CALL S
OUND((T(X,2)+1)*400,T(X,1),0
,T(X,1)*1.01,0):: NEXT X ::
J=0 :: GOTO 140
```

A little subprogram to add a bit of variety to your "PRESS ANY KEY" routine.

```
1 CALL CLEAR :: CALL PRESSKE
Y(24)
30000 SUB PRESSKEY(R)
30001 C=C+1 :: IF C=16 THEN
30002 :: DISPLAY AT(R,1):""
:: DISPLAY AT(R,C):"PRESS AN
Y KEY" :: DISPLAY AT(R,C):"p
ress any key" :: CALL KEY(0,
K,S):: IF S=0 THEN 30001 EL
S 30003
30002 C=C-1 :: IF C=0 THEN 3
0001 :: DISPLAY AT(R,1):"" ::
: DISPLAY AT(R,C):"PRESS ANY
KEY" :: DISPLAY AT(R,C):"pr
ess any key" :: CALL KEY(0,K
,S):: IF S=0 THEN 30002
30003 DISPLAY AT(R,1):"" ::
SUBEND
```

And a new way to wipe the screen -

```
1 CALL CORNERWIPE(30)
29000 SUB CORNERWIPE(CH):: F
OR T=1 TO 24 :: CALL HCHAR(T
,3,CH,T+4):: CALL HCHAR(25-T
,32-T,CH,T):: NEXT T :: CALL
CLEAR :: SUBEND
```

MEMORY FULL

Jim Peterson



"Some Enchanted Evening"

I PB 87-3 publicerade vi en del bilder som gjorts av Kim Gilliland. Som vi skrev i förra numret har vi fått nyligen fått nya skivor från honom. Bilderna härintill kommer från de skivorna och är utskrivna med hjälp av programmet MAX-RLE.

Peter Olsson har varit i kontakt med en amerikansk databas och hämtat hem ytterligare MAX-RLE bilder via modem. Vi hoppas kunna publicera en del av dem nästa gång.