

FRÖGÅR

nittinian

ÅR 1984

202

SOMMAR
NUMMER

BILDEXTRA

Ordföranden ...	2
Nya freeware	2
Arsmötesprotokoll	3
Anders visar och berättar	4-5
Att minnas med magnetism	6
Kraft till separata diskdrivar	7
Att minnas mera och fortare	8-10
Triton Super XB	10-11
Nyttoprogram, vad är det?	12-13
Högtalar-konstruktion	14-15
Tillbaka till BASIC	16

Ordföranden ...

Jag blev vald till ordförande även för detta år trots stor tveksamhet från min sida. Det var mycket få medlemmar på årsmötet i mars och medlemsmötet i april. Du som vill att föreningen skall leva vidare måste visa ditt intresse genom att komma på mötena eller genom att skriva brev. Alla sorters brev är välkomna även om det är kritik eller förslag till förändringar. Du som inte kan skriva artiklar i tidningen kan sända dina frågor till föreningen.

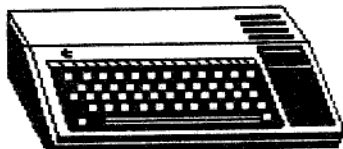
Föreningens verksamhet består idag i huvudsak av följande delar:

- Tidningen
- Programförmedling
- Medlemsmöten
- Rådgivning
- Datakommunikation
- Programmerbara räknare

Du som har synpunkter på dessa områden bör skriva och berätta hur du vill ha verksamheten. Vad är bra och dåligt och vill du ha mer eller mindre av något.

Jan Alexandersson

Jan Alexandersson
Springarvägen 5, 3 tr
142 61 TRÅNGSUND
Tel. 08/771 05 69



Nya Freeware

Archiver II version 2.4 av Barry Boone är ett mycket bra program som kan omvandla en hel skiva till en fil i INT/FIX 128-format. Den nya filen tar endast 60 % av det ursprungliga antalet sektorer. I praktiken kan du packa mer än så eftersom många halvfylla skivor kan lagras på samma skiva. Den ursprungliga skivan med alla skivnamn och filnamn kan när som helst återskapas med programmet. Nödvändigt är expansionsminne och någon av modulerna XB, EA, TW eller Funnelweb. 1 skiva.

Auto Dialer version 2.0 av John Birdwell kan endast användas tillsammans med FAST TERM. 1 skiva.

Ken Gilliland har sänt nya skivor till föreningen med nytt datum 01 JAN 1988. Alla skivor kan nu användas med skrivskyddet påklustrat. Nödvändigt är expansionsminne och Extended Basic. Följande har kommit:

- WAGNER 1 (2 skivor) ändrad
- WAGNER 2 (2 skivor) ny
- SOUTH PACIFIC (1 skiva) ändrad
- PATSY CLINE (1 skiva) ändrad
- MUSIC MAN (1 skiva) ändrad
- WIZARD OF OZ (1 skiva) ändrad

I redaktionen:

Redaktör Peter Odelryd
Utmaningsredaktör Anders Persson
Forth-redaktör Lars-Erik Svahn
Programförmedlare Börje Häll
Allt-i-alla Claes Schibler

Föreningens och redaktionens adress:

Föreningen Programbiten
c/o Schibler
Wahlbergsgatan 9, nb
12146 Johanneshov

Datainspektionens licensnummer: 82100488

Postgiro: 198300-6

Medlemsavgiften för 1987 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 1 000 SEK per helsida, 500 SEK per halv sida. För löslblad som skickas med tidningen gäller 1 000 SEK per blad.

För kommersiellt bruk gäller följande:

Mångfaldigande 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 mångfaldigande genom tryckning, duplicering, stencilering, bandinspelning, diskettinspelning etc.

Föreningens tillbehörsförsäljning

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

För TI-99/4A:

Användartips med Mini Memory	60:-
Nittinian T-tröja	40:-
99'er magazine nr 12/82, 1-5, 7-9/83 (per styck)	20:-
Programbiten/Nittinian 1986 (TI 59+99)	80:-
Programbiten/Nittinian 1985 (TI 59+99)	80:-
Programbiten/Nittinian 1984 (TI 59+99)	80:-
Nittinian, årgång 1983 (TI-99/4A)	60:-

För TI-59 mm

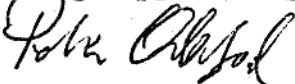
Programbiten, årgång 1983 (kalkylatorer)	40:-
Programbiten, årgång 1982 (kalkylatorer)	40:-
Programbiten, årgång 1981 (kalkylatorer)	40:-
Programbiten, årgång 1980 (kalkylatorer)	40:-
Programbiten, årgång 1978/79 (kalkylatorer)	40:-
Programbiten, fem årgångar 1978-83	160:-
Programbiten, sex årgångar 1978-84	220:-
Programbiten, sju årgångar 1978-85	300:-
Katalog med belgiska och engelska program för räknare TI-57, TI-58, TI-59	20:-
Föreningens programmeringsblanketter (TI-59), olika typer, block om 50 blanketter (se PB 83-1 sid 30) per block	11:-
Patenthandlingar TI-59	25:-
Tom magnetkortsplånbok	10:-

**K-TRYCK AB**

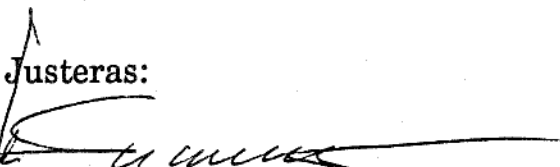
Protokoll fört vid årsmöte för Föreningen Programbiten den 12 mars 1988 på Militärhögskolan, Stockholm


1. Mötet öppnades av föreningens ordförande Jan Alexandersson.
2. Till mötesordförande valdes Lennart Agdeus. Till mötessekreterare valdes Peter Odelryd. Som justeringsmän valdes Sten Gunnarsson och Kent Edgardh.
3. Mötet konstaterades vara utlyst i stadgeenlig tid. 11 röstberättigade medlemmar var närvarande.
4. Dagordningen lästes upp och godkändes.
5. Verksamhetsberättelsen och den ekonomiska berättelsen föredrogs och godkändes.
6. Lennart Agdeus föredrog revisorernas berättelse för 1987. Han rekommenderade årsmötet att bevilja styrelsen ansvarsfrihet för 1987.
7. Årsmötet beviljade styrelsen ansvarsfrihet för 1987.
8. Val av styrelse för 1988. Åke Olsson föredrog valberedningens förslag. Efter en diskussion valde årsmötet följande styrelse:
ordförande: Jan Alexandersson
sekreterare: Kent Edgardh
kassör: Patric Eriksson
redaktör: Peter Odelryd
övriga ledamöter: Peter Olsson, Göran Nygren, Börje Häll, Claes Schibler och Åke Olsson.
9. Till revisorer valdes: Mikael Nordlin och Barbro Nygren. Till revisorssuppleant valdes Lennart Agdeus.
10. Till valberedning valdes Sten Gunnarsson och Mikael Nordlin (sammankallande).
11. Årsavgiften för 1988 fastställdes till 120 kronor.
12. Årsmötet avslutades.

Enligt uppfattning:


Peter Odelryd

Justeras:


Sten Gunnarsson


Kent Edgardh

Anders visar och berättar

av Anders Persson

Att min dator inte riktigt ser ut så som folket på Texas hade tänkt sig en gång i tiden, det har nog de flesta av er redan förstått. De som har bevistat träffarna i Staffanstorp och, någon gång, i Stockholm, har haft möjlighet att se en del av mina konstruktioner med egna ögon. Men för er andra ska jag nu kommentera några fotografier av vissa delar av anläggningen.

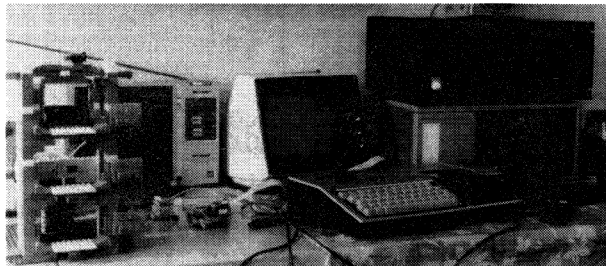
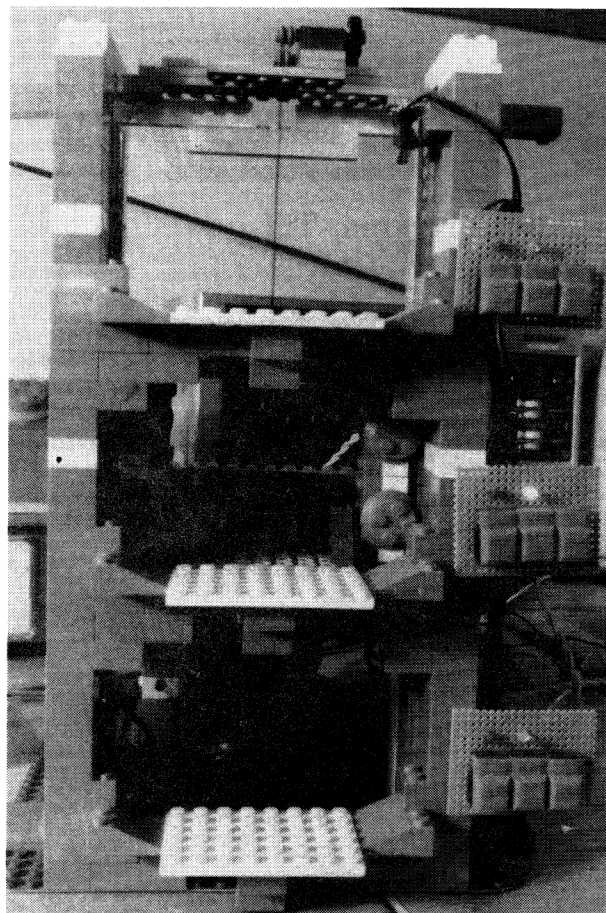


Bild 1 visar hela utrustningen, så när som på skrivaren. Till höger, ovanpå den vanliga expansionsboxen, syns ett hembygge för att härbärgera två extra drivverk och kraftaggregat till dessa.

Till vänster syns den hiss som jag satte ihop för att ha något intressantare att styra än bara lysdioder (se PB 87-1). Mellan hissen och datorn ligger det speciella anpassningskortet som tillåter styrning av hissen med PIO-porten.

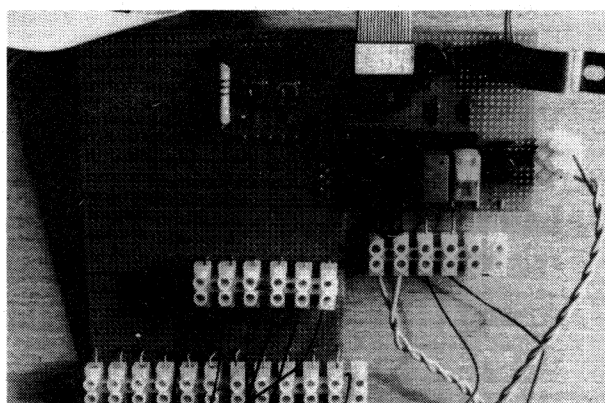


På **bild 2** syns hissen framifrån, med hisskorgen på andra våningen. Kraftaggregaten skymtar till vänster. Knapparna på hissen kan användas till olika saker, beroende på programvaran. Det har senare fallit mig in att det hade räckt med en knapp vid varje våning, och en uppsättning i hisskorgen. Den senare hade väl fått placeras utanför korgen av

praktiska skäl, men om knapparna verkligen följer korgen eller inte saknar principiell betydelse.

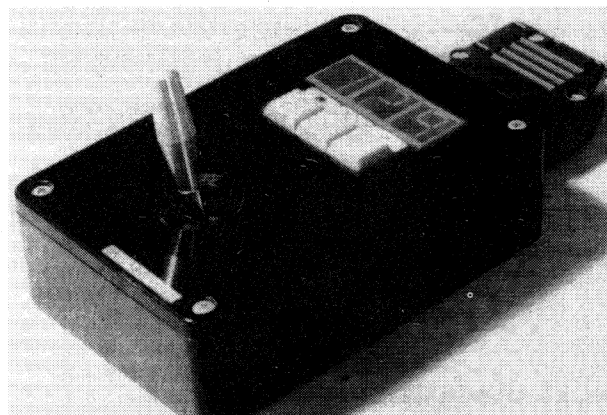
Hissen är tyvärr för liten för att vara något bra exempel att skriva programvara för. Med fler våningar att välja på skulle man kunna försöka att implementera olika, mer komplexa, strategier för hur begäran om transporter mellan våningarna skall utföras. De problem som uppstår då är helt analoga med de som man har när det gäller att tilldela resurser till olika processer i en dators operativsystem. Hissens storlek bestämdes dock av tillgången på byggnadsmaterial och dessutom av att det skulle vara lätt att ansluta den till PIO-porten.

Hissen var trots detta ett intressant objekt att skriva program för, speciellt när jag roade mig med att implementera riktig multitasking i p-systemet.



Den anpassning som behövs mellan PIO-porten och hissen framgår av **bild 3**. Flatkabeln går till parallellporten, medan övriga ledningar går till hissen eller kraftaggregaten.

Reläerna används för styrningen av hissmotorn, medan känsligheten hos de optiska givarna vid våningarna justeras med trimpotentiometrarna. De integrerade kretsar som syns på bilden används till en expansion av PIO-porten till åtta bitar både ut och in samtidigt. (Jämför PB 87-1, "Styr och ställ".)

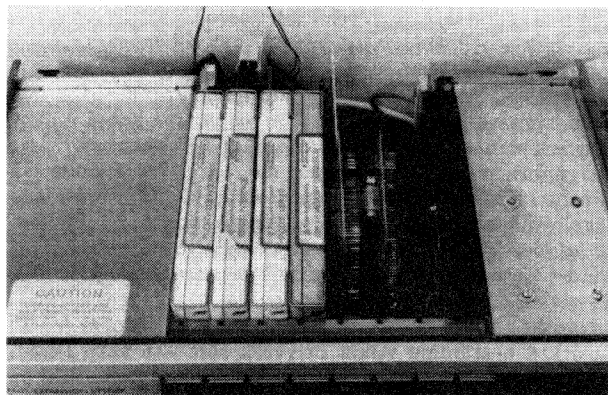


Nu något annat. **Bild 4** visar den analoga joystick som jag förfärdigat. En analog joystick är alltså steglös till sin natur, till skillnad från den digitala modell som används normalt till 99:an. Där finns det ju bara till eller från i en viss riktning.

Nu är det naturligtvis så att spakens position förvisso är steglös, men informationen kvantiseras innan den presenteras för datorn. Till detta använder jag en åttabitars A/D-omvandlare som sitter på ett kort i expansionsboxen. Åtta bitar ger 256 olika lägen. Då spakens slaglängd är knappt fem centimeter, innebär detta att ett steg blir ungefär 2/10 mm. Högre upplösning än så behövs knappast till något som flyttas för hand.

Knapparna och siffrorna kan användas till vad som helst. De styrs helt från programvaran. Jag har skrivit ett drivprogram till *TI-Artist*, som tillåter mig att använda den här joysticken som redskap att styra grafiken med. Där mot-

svarar en knapp den som vanligtvis sitter på en joystick, medan en annan motsvarar ett mellanslag på maskinen. Det är också möjligt att låta displayen visa vid vilken koordinat på skärmen som man för tillfället befinner sig, men det är något som jag inte testat ännu.



Att utseendet på mina egna kretskort skiljer sig något från det gängse framgår av **bild 5**. Till höger om fyra kort i originalutförande från TI, sitter I/O-kortet och klockkortet. Längst till höger diskkontrollkort från *CorComp* i en svart plåtkapsling. Batteriet som håller tidräkningen igång dygnet runt skymtar överst på klockkortet.

En ledig plats finns ännu, som synes. Minnesexpansionen är dock inte nödvändig längre, då jag numera har byggt in 64 Kbyte i konsolen, utöver de 16 Kbyte som finns där från början.

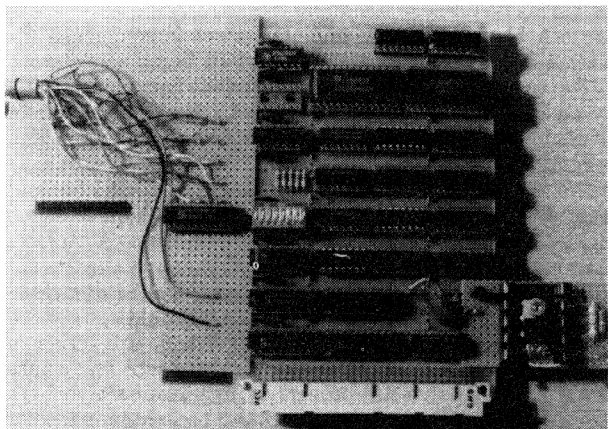
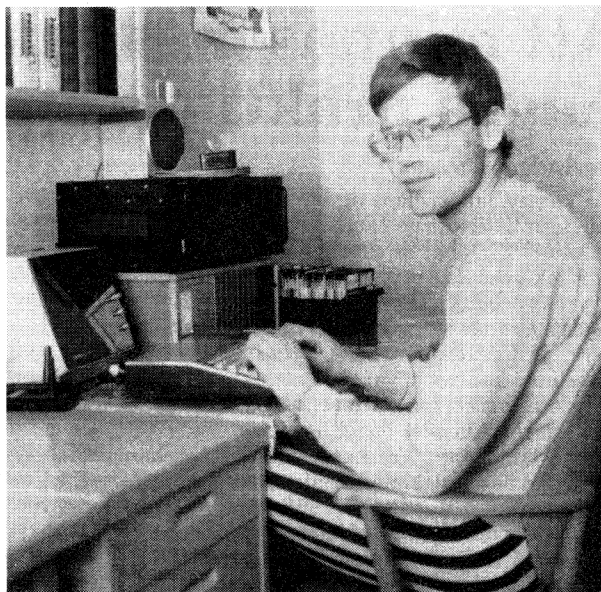


Bild 6 visar I/O-kortet. På bilden har kortet 40 digitala utgångar, 32 digitala ingångar och åtta analoga ingångar. Analoga utgångar saknas. Det hela är byggt på ett något avsågat Europakort (100x160 mm), som sedan är påbyggt både bakåt och framåt. Till vänster är kabeln till joystickens ansluten. När bilden togs hade det fortfarande inte kommit på något kontaktdon (det har det inte nu heller, för den delen), men tanken är att det ska bli möjligt att ta bort joystickens och koppla in något annat, om så önskas. Utan att lödpennan ska behöva tas till bara för anslutningens skull, alltså.

Uptill på kortet syns A/D-omvandlaren, där läget på joystickens spak omvandlas till något som datorn kan förstå, samt 2 Kbyte minne. Minnet kan användas till en liten DSR, om man behöver en sådan. Hittills har det bara kommit till nytta för att simulera en RAM-disk tillsammans med *Maximem* och en del annat minne. Utbyggnaden åt höger är till för att lysdioden ska räcka fram till "fönstret" i expansionsboxen.

Klockkortet har en liknande uppbyggnad. Detta av den enkla anledningen att jag använde erfarenheterna från det kortet som utgångspunkt för det här.



Slutligen visar den sjunde bilden utmaningsredaktören i "arbete", för dem som nu inte har haft möjlighet att se detta i verkligheten. Notera att hissen inte står framme i vardagslag. Den tar för mycket plats ...

Ny katalog från Texaments

Det har kommit en katalog på fyra sidor från:

Texaments
53 Center Street
Patchogue, NY 11772
USA

Den innehåller många grafikprogram som TI-Artist med många tilläggsskivor och GSGD (Character Sets and Graphics Design) med tilläggsskivor. I övrigt finns följande:

- Rapid Copy USD 15
- XB Detective USD 10
- Myarc Disk Controller DS/DD USD 150
- Myarc Hard Disk Controller USD 310
- Myarc RS232/PIO Card USD 110
- Myarc 512k Minne/RAM-disk USD 230
- Myarc Extended Basic II USD 65
- Myarc Geneve 9640 enhanced USD 480
- MY-Art och mus för Geneve USD 110

Till dessa priser tillkommer flygporto till utlandet enligt följande:

- Program USD 8 per order
- Hårdvara USD 16 per styck
- Geneve 9640 USD 70 per styck

Betalning kan göras med bankcheck (money order) eller med postgirots vanliga utlandsblankett. Om du inte anger något postgirokonto kommer den automatiskt att göras om till check för 23 SEK per betalning inklusive porto.

De tillkommer 5% tull och moms som du betalar i Sverige vid leveransen.

Om du vill ha en kopia på katalogen kan du sända 5 kr i frimärken till:

Jan Alexandersson
Springarvägen 5, 3 tr
142 61 TRÅNGSUND

Att minnas med magnetism — några praktiska synpunkter

av Anders Persson

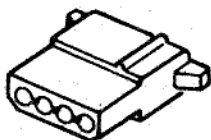
En kassetbandspelare är bättre än ingenting. Ett skivminne är bättre än en bandspelare. Två skivminnen är bättre än ett. Så långt är det inte heller några större problem rent tekniskt. Två drivverk går bra att stoppa in i en vanlig expansionsbox, under förutsättning att drivverken är i halvhöjdsutförande. Men vill man ha tre eller kanske fyra drivverk, då måste man ordna ström till dem på något sätt. Dessutom kan det ju vara bra att ha någon form av låda att montera dem i också, om man nu inte, som somliga, nöjer sig med en skolåda eller något liknande.

Om vi nu antar att du har löst den rent mekaniska monteringen så är väl strömförsörjningen nästa problem. En diskdrive behöver +5 VDC och +12 VDC. Som ett grovt riktmärke kan du utgå från en strömförbrukning på 0,8 A för 12 V och 0,5 A för 5 V. Moderna, halvhöga, enheter kräver normalt mindre ström än äldre fullhöjdare. Halvhöjdarna är ju avsedda att sitta två stycken på samma plats som tidigare upptogs av en enda. Belastningen på 12 V varierar ganska mycket beroende på om motorn går eller inte, medan förbrukningen på 5 V är jämnare. Speciellt när motorn ska starta blir det rusning på 12 V sidan, men den spänningen är å andra sidan inte så kritisk.

Nätaggregat kan man köpa färdiga på olika håll. När det här skrevs hade Jan Alexandersson sparat upp en modell, avsedd att försörja två drivverk. Dess kapacitet är dock tillräcklig för tre moderna drivverk av lågeffektutförande.

Ett annat alternativ är naturligtvis att bygga själv. Det som behövs är, räknat i tur och ordning från eluttaget: nätanslutning, säkring, brytare, transformator, likriktare, kondensator (stor), reglerkrets och kondensator (liten). Att tänka på är huvudsakligen att linjära reglerkretsar behöver ungefär tre volts marginal för att fungera. Transformatorn bör alltså leverera ungefär nio och femton volt om man ska få ut fem respektive tolv efter regleringen. Detta gäller speciellt om belastningen ligger nära transformatorns maximala kapacitet, eftersom den höjning av spänningen som likriktning och glättning ger upphov till då inte kan upprätthållas.

Anslutningen till drivverket sker med en fyrapolig kontakt. Den kan exempelvis köpas från *Elfa* i Solna, artikelnummer 25-891-25. Observera att kontakthylsor inte medföljer, utan säljs separat. De har nummer 25-891-41.



Nästa steg blir att koppla samman datasignalerna mellan drivverken och kontrollkortet. Till detta använder man 34-polig flatkabel och kortkantkontakter. Undantaget är anslutningen för interna drivverk på kontrollkortet. Där passar en hylskontakt i stället. Dessa kontaktdon kan också köpas från *Elfa*. 43-675-46 heter hylskontakten, medan den för kortkanterna kallas 43-684-78. Kabeln slutligen har artikelnummer 55-661-12.

Kabeln ansluter man till kontakterna genom att stoppa in den i kontakten och trycka till. Inte med tummarna precis, men i ett skruvstycke går det bra. Se till att den kommer rätt, bara. När anslutningen till drivverken ska göras, går det att vända kontaktdonen på två håll, varav endast det ena är rätt. Det är dock ganska lätt att se vilket som är vilket. På den externa kontakten på diskkontrollkortet kan man se att alla polerna på den ena sidan är jordade. Motsvarande gäller för drivverket. Anslut kabeln till kontrollkortet och kontrollera med en ohmmeter vilken sida av den eller de andra kontakterna som blir jordad. Vänd den mot den jordade sidan på drivverket.

Skulle det bli fel brukar resultatet bara bli att drivverkens motorer går och går och att inget annat vill fungera. Då är det bara att vända på kontakten.

Koppla alla drivverken parallellt med varandra. Det går bra att pressa på flera kontaktdon på samma kabel. Vilket drivverk som är anslutet till vilken kabel saknar betydelse. Det går bra att ha alla på samma, men om du har ett eller två internt i boxen är det lämpligast att använda den interna kabeln till dem, och den externa till de som är utanför. Om du har den interna kabel som följde med Texas kontrollkort när det var nytt, kan du klämma på en extra kontakt på den. Då räcker den till två drivverk i boxen.

När du har bestämt dig för hur kablarna ska dras måste du också bestämma vilket drivverk som ska vara DSK1, vilket som ska vara DSK2 osv. Ofta finns det ett antal stift på drivverket, stift som kan kopplas ihop med en liten kortslutningsbygel. Det kan finnas ett flertal olika stift, med olika märkning. Det intressanta är de som betecknas DS0 (Drive Select 0), DS1, DS2 och DS3, eller något liknande.

Genom att placera bygeln mitt för önskat nummer bestämmer du vilket nummer drivverket ska ha. 99:an numrerar visserligen från ett till fyra i stället för från noll till tre, men den räkneoperationen klarar du säkert utan datorhjälp.

Andra varianter förekommer också. En är att det sitter hoplödade, fjädrande bleck som kortsluter samtliga alternativ från början. Då väljer man nummer på drivverket genom att värma de bleck som man inte vill ha. Lödningarna släpper och blecken fjädrar upp. Det förekommer också att blecken klipps av. Om du då har köpt ett begagnat drivverk av den typen kan det vara omöjligt att löda ihop blecket igen. Då får man bygga på något annat sätt. Liksom för all annan elektronik gäller att en viss försiktighet med värmen kan rekommenderas.

Har du otur finns det kanske bara en IC-sockel för bygging, utan någon märkning alls. Då krävs det kanske någon elektronikkunnig person för att räkna ut hur byglarna ska placeras.

Ytterligare en sak som kan välla huvudbry är den resistorpack som sitter i drivverken. Men det är faktiskt ganska enkelt. Resistorpacken ska sitta i det eller de drivverk som kommer sist på flatkabeln. Övriga plockas bort. Om du bara utnyttjar exempelvis den interna anslutningen, ska du bara ha en resistorpack i anläggningen. Om båda utgångarna är använda ska det finnas två.

Notabelt är att motståndsvärdet som används skiljer sig från olika fabrikat på drivverken. Det är alltså lämpligt att spara resistorpacken, så att den kan återställas om du händelsevis vill avyttra drivverket någon gång.

Ja, nu är det bara att slå till strömmen och sätta igång. Det är lämpligt att slå till strömmen till alla drivverken först, även om du inte har för avsikt att använda alla. Om inte annat så finns risken att datorn blir konfys av strömstöten när de extra drivverken kopplas in, om du nu i efterhand skulle komma på att de behövs trots allt.

Orsaken till den här artikeln är att jag fick några frågor i ett brev. Erfarenheten har lärt mig att om någon medlem undrar över något, så finns det nästan alltid fler som undrar över samma sak. Därför är det lika bra att svara i en artikel redan från början. Samma erfarenhet har visat att om man nu svarar på de ursprungliga frågorna, så hade andra intresserade personer ytterligare några frågor. Dessa blev inte besvarade, eftersom den ursprunglige frågeställaren inte ställde de frågorna. Detta beror i sin tur ibland på att han redan hade de svaren klart för sig, eller, oftare, på att han ännu inte insett att frågeställningen skulle komma att dyka upp.

Summan av det här blir, eller skulle åtminstone bli, att du kanske inte förstår allt, men då går det bra att ställa nya frågor. Jag brukar skicka en kopia av artikeln direkt till de som frågar, så under förutsättning att svaret snabbt kan ges, behöver du inte vänta på nästa nummer av tidningen.

Kraft till separata diskdrivar

av Jan Alexandersson

För drygt ett år sedan köpte jag två dubbelsidiga halvhöjdsdrivar från TEXCOMP som nu sänkt priset på dessa till USD 90. Jag har haft båda monterade i expansionsboxen medan den gamla TI-originaldriven (enkelsidig) legat på hyllan.

Datorhölje XT

Jag har nu skaffat ett datorhölje till en PC (500:—) och en kraftenhet (200:—) från Material Clearing Elektronik, Box 6157, 600 06 Norrköping. Datorhöljet är slut-sålt medan kraftenheten finns att få tag på men du måste köpa 5 st för att få ett bra pris (95 kr + moms + porto).

Datorhöljet har plats för 4 halvhöjdare så jag har enkelt kunnat montera mina två halvhöjdare och en helhöjdare. Mina tre drivar och kraftenhet tar 2/3 av utrymmet men 1/3 av volymen finns kvar för framtida utökningar. Locket som sitter på gångjärn är mycket lätt att öppna om man vill komma åt att ändra de interna kablarna eller strappningspacken på diskdrivarna.

Kraftenhet 5V/12V

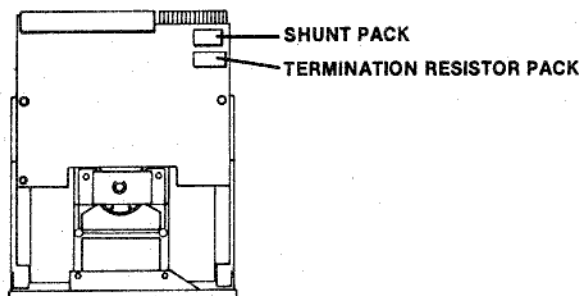
Kraftenheten BMJ 160 001/1 R1J (tillverkningsperiod 8419) är tillverkad för en PC och kan ge följande:

+ 5 V	4,5 A
+12 V	2,5 A
-12 V	0,1 A

Kraftenheten är 296x134x90 mm och väger 2,1 kg. Den har ett vanligt trepoligt jordat 220 V uttag och är helt kapslad så det är ingen risk att komma åt farlig spänning. Det är fullt möjligt att använda den som den är om inte utseendet stör men jag har stoppat in kraftenheten i datorhöljet.

Fläkt 9W/220V

Det finns även en fläkt EBM W2S075 AE 13-02 i kraftenheten vilket är mycket bra. Fläkten är på 9 W och matas från 220 V till skillnad från TI-expansionsboxen som har en 110 V fläkt. Från en andra kraftenhet har jag tagit bort fläkten och stoppat in den i expansionsboxen. Den nya fläkten är nämligen mycket tystare och det är faktiskt billigare att köpa en kraftenhet med fläkt än att köpa en ny fläkt på ELFA. Fläkten kan monteras i samma skruvhål som tidigare men är i övrigt kompaktare byggd och har många fler rotorblad. Motorn ligger inne i centrumnavet så luftströmmen kan passera utan de hindrande delar som TI-fläkten har.



Kraftkabel

Alla behövliga kontaktdon för hopkopplingen av +5V/+12V finns på ELFA men kan säkert fås på andra ställen. Jag betalade följande priser:

Kontakthus till kraft för drive	4:32
Hylsa till ovan	0:74
Kontakthus till kraftenhet	2:16
Kontaktelement till ovan	0:42
Kontaktelement för strömbrytare	0:62

Flatkabel

För att kunna ansluta tre yttre diskdrivar till mitt TI-diskkontrollkort var det nödvändigt att tillverka en ny flatkabel med 34-poliga kontaktdon som finns på ELFA till följande priser:

Kortkantkontakt 34-polig	49:38
Flatkabel 34-ledare per meter	14:32

Det var mycket enkelt att montera kontakterna. Det är bara att sätta kontakten i ett skruvstöd och stoppa in kabeln och sedan dra åt. Någon lödning görs således inte. Eftersom kontakterna är ganska dyra bör du tänka igenom lämplig kabellängd och antalet kontakter. Jag monterade en kontakt för kontrollkortet och tre kontakter för diskdrivar samt lämnade en liten kabelstump extra för en eventuell 4:e diskdrive (några dm kabel kostar inget). TI-kontrollkortet klarar endast tre drivar medan Corcomp och Myarc klarar fyra drivar.

Resistor-pack och shunt-pack

När du ökar antalet diskdrivar är det två saker till som måste ordnas utöver kraftkabel och flatkabel. Varje diskdrive har en resistorpack som måste tas bort på alla drivar utom den som sitter längst ut på flatkabeln. Detta behöver inte vara DSK3 utan du kan om det passar ha DSK1 längst bort men det är alltid den längst bort belägna driven som skall behålla resistorpacken.

Du måste även ställa in DSK-nummer på de olika drivarna så att kontrollkortet kan skilja dem åt. Detta görs genom att bygla önskat nummer. Dessa brukar vara markerade DS 0—3 vilket motsvarar DSK 1—4 dvs DSK2 byglas med DS1.

Min TI-original drive har shuntpacken färdigklippt för DSK1 och det går nog klippa och löda om den till annat nummer men det är bekvämare att fixa separata byglar (2 st per drive) med en böjd styv kopplingstråd. Det går även att köpa byglar (strappar) från ELFA för 2:28.

Ström till drivar

Hur många drivar kan jag nu mata kraft till med min nya kraftenhet. Den TI-originaldrive PHP 1250 (Shugart 400L R) som jag har kräver 0,4 A från 5V och 0,7 A från 12V (typiska värden). Jag har inga uppgifter om vad mina halvhöjds Fujitsu M2551A behöver. Enligt en kataloguppgift behöver en ny BASF-drive 0,5 A respektive 0,5 A. En äldre drive av typ Pertec kräver 1,6 A respektive 0,8 A (max-värden). Det ser ut att vara 12 V-matningen som blir dimensionerande och jag skulle gissa att man klarar tre eller fyra drivar.

Finns det någon medlem som har mera uppgifter om kraftmatning av diskdrivar eller du kanske har byggt en egen box för extra diskdrive. Skriv gärna till tidningen och berätta om detta. Du bör även läsa Hans Attersjös två sidor långa artikel i Nittinian 83-4: "Allt om Disk-Drivar". Enligt uppgifter där klarar TI-expansionsboxen av att leverera +5V/1A och +12V/1A.

Att minnas mera och fortare

av Anders Persson

En gång i tiden hade 99:an ett stort, inbyggt minne. Det har den inte längre. Inte för att minnet skulle ha krympt med åren, men måttstocken för vad som är stort har förändrats. 16 Kbyte var mycket i slutet av sjuttioalet. Idag är det ingen som höjer på ögonbrynen för någon megabyte eller två.

Långsam utbyggnad

Nu går det ju att bygga ut minnet. Det vanligaste steget är väl en expansion på 32 Kbyte. Den utgöres i regel av ett kort som monteras i expansionsboxen, även om det finns andra alternativ också. En nackdel med dessa kort, och allt annat som anslutes till I/O-porten på datorns sida, är att databussen bara är åtta bitar bred. Processorn TMS 9900 har en 16-bitars databuss, vilket gör att den kan läsa ett helt ord, två bytes, i taget. Men i 99:an utnyttjas detta bara till en liten del av minnet. ROM med GPL-interpretator, flyttalrutiner, monitor etc. (0000H—1FFFFH) och de 256 bytes RAM som finns inbyggt (8300H—83FFFH) är anslutet till denna 16-bitars buss. Allt annat minne utanför maskinen är åtta-bitars. Det gäller även allt som anslutes i modulporten. De 16 Kbyte som finns i konsolen kan CPU:n inte komma åt direkt överhuvudtaget, utan den måste gå omvägen över videoprocessorn.

För att det nu ändå ska fungera sitter det några kretsar i datorn som delar upp ett ord i två bytes när något ska lagras i minnet. Vid läsning slås i stället två bytes ihop till ett ord. Att allt det här manövrerandet tar tid är väl inte precis förvånande. Normalt gör 9900 en minnesaccess på två klockcykler. Härav kunde man kanske anta att det då tar fyra klockcykler att göra det i två omgångar, men faktum är att det tar sex. Ytterligare en klockcykel för varje byte ödas bort som ett så kallat vänteläge. Att läsa från VDP RAM ska vi bara inte tala om. Eftersom det minnet inte sitter på CPU:ns buss, går det åt flera instruktioner för att komma åt ett ord på godtycklig adress i VDP RAM. Ungefär 120 klockcykler för att läsa två bytes, och då har man ändå vanligtvis inte hunnit göra något med informationen som man fått tag på. Men det är en helt annan sak, som inte har med minnesexpansionen att göra.

Snabbare inbyggnad

Minnet utanför maskinen är alltså bara tredjedelen så snabbt som CPU:n tillåter. Enda sättet att få minnesexpansionen snabbare är att montera den inuti datorn. Inuti datorn? Det går väl inte? Nej, det går inte. Egentligen. Fast det går ändå. Visserligen var det ingen som gjorde något för att underlätta en sådan manöver när datorn konstruerades. Men den som är intresserad av att bygga på egen hand vägrar naturligtvis att låta sig hindras av sådana petitesseer. Tacknämligt nog är det så att lådan som datorn sitter i är tillräckligt stor för att det ska gå att bygga in lite av varje. Två olika principer för detta finns. Antingen bygger man ett kretskort som monteras inuti lådan och anslutes till datorns kretskort med exempelvis en bandkabel. Eller också monterar man kretsarna ovanpå de befintliga och gör anslutningarna med någon lämplig tråd som lödas direkt på benen på kretsarna. Med den sistnämnda metoden är det faktiskt möjligt att få in hela konstruktionen utan att det hindrar monteringen av den plåtåda som skärmar av störningar från datorn. Nackdelen är väl närmast att alla trådar i olika riktningar gör att konstruktionen liknar en väl tilltrampad grästuva. Dessutom kräver installationen god vana vid mjuklödning.

Hur det går till

Som du nog har förstått har jag själv provat att bygga ut minnet inuti konsolen. Jag hade dock inte tänkt låta det här bli en detaljerad anvisning om hur man ska gå tillväga. Det tycks finnas en uppsjö med olika förslag i olika tidningar världen över. Själv blev jag inspirerad av en artikel i kanadensiska "R/D Computing", men det förekommer också en hel del artiklar om det här i tyska alster. Om du vill ha närmare tips är du välkommen att kontakta mig.

Jag använde metoden att löda fast kretsarna ovanpå befintliga kretsar på datorns kretskort. Detta mest för att artikeln i "R/D Computing" förutsatte den metoden. Dessutom har jag byggt om och ändrat åtskilligt jämfört med den artikeln, för att få konstruktionen som jag ville ha den. Då är det behändigare att bara flytta några trådar, jämfört med att tillverka nya kretskort titt och tätt.

Vad man får

Beskrivningen i "R/D Computing" gällde 32 Kbyte snabb minnesexpansion inuti konsolen. För att förenkla bygget så mycket som möjligt, skulle man faktiskt installera 64 Kbyte, men det gick inte att använda mer än de 32 Kbyte som motsvarar en vanlig minnesexpansion. I tidningen sade de sig visserligen fundera på en konstruktion som skulle tillåta åtkomst av hela det inbyggda minnet, men den var inte färdig. Finessen så här långt var alltså att den vanliga minnesexpansionen inte behövdes, samt att det nya minnet är snabbare.

Men efter att ha följt instruktionen och konstaterat att det fungerade, började jag själv fundera över hur jag lämpligen skulle komma åt det resterande minnet också. De 32 Kbyte som motsvarar den vanliga minnesexpansionen finns på adresserna 2000H—3FFFH och A000H—FFFFH. Jag tyckte att det kunde vara bra om det gick att koppla in RAM även i resterande delar av adressområdet, dvs inom 0000H—1FFFFH och 4000H—9FFFH. Snart hade jag tänkt ut en konstruktion som tillät detta. Med hjälp av fyra CRU-bitar går det nu att koppla in RAM i block om åtta Kbyte. Varje sådant block överlappar då ett ordinarie block i maskinen. Om samtliga RAM-block kopplas in, får maskinen kontinuerligt RAM över hela adressområdet.

Vad är då vitsen med detta? För det första kan man på ett enklare sätt än normalt skriva assemblerprogram som behöver mycket utrymme för data och program.

För det andra kan man, i och med att RAM nu finns på samma adresser som bland annat GPL-interpretatorn, kopiera över denna till RAM och sedan modifiera efter behag. Här ges en del möjligheter för avancerade programmerare som inte är tillgängliga annars. Exempelvis kan man modifiera de vektorer som används vid avbrott, och därmed skriva helt egna avbrottsrutiner. På så sätt kan den timer som finns i 9901 användas till något mera än bara lagring på kassett.

För det tredje kan man kopiera det ROM som finns i vissa moduler till RAM. I och med att detta RAM är snabbare exekveras programmen på kortare tid. Tyvärr går detta inte utan vidare med Extended BASIC, då den modulen har 12 Kbyte inom adresserna 6000H—7FFFH.

Nu hade jag alltså fått ständig tillgång till 32 Kbyte snabb minnesexpansion och dessutom möjlighet att koppla in ytterligare 32 Kbyte, i block om åtta, så att maskinen totalt får 64 Kbyte RAM. Konsolens minneskapacitet hade alltså vuxit från 16 till 80 Kbyte. Dessutom hade det gamla kortet med 32 Kbyte RAM blivit överflödigt.

Då drabbades jag av tanken att det var något som var onödigt. Varför skulle man inte kunna använda det gamla minnet också? Med en CRU-bit skulle man ju kunna byta från intern till extern minnesexpansion, utan att detta påverkar de 32 Kbyte "extra" minne som nu fanns inbyggt. Efter ytterligare några modifikationer hade jag fått även denna "bankswitching" att fungera. Total kapacitet nu 112 Kbyte.

Snabb, snabbare, för snabb

Med ett snabbare minne går programmen snabbare att exekvera, naturligt nog. Men hur mycket? Tre gånger snabbare minne ger inte tre gånger så snabba program, tyvärr. Detta beror på att CPU:n inte ständigt och jämt hänger på låset till minnet. En addition av två register tar 14 klockcykler. Här inryms fyra minnesaccesser, en för att hämta instruktionen, två för att hämta de tal som ska adderas och en för att lagra resultatet. Eftersom varje minnesaccess tar två klockcykler, innebär detta att åtta av de 14 cykler som instruktionen totalt omfattar, utgöres av minnescykler. När då varje minnesaccess ökar från två till sex cykler, kräver instruktionen, med sina fyra minnesaccesser, 16 extra klockcykler. Totalt 30 alltså, jämfört med 14 som snabbast. Således en förbättring med över 110%.

Proportionerna är någorlunda lika för de mest frekventa instruktionerna, vilket medför att programmen exekveras ungefär dubbelt så snabbt. Men detta förutsätter då att både program och register (workspace) finns i långsamt respektive snabbt minne. Ofta använder program till 99:an RAM på adresserna 8300H—83FFH till registerna, då detta minne redan är 16-bitars. I exemplet ovan är det då bara läsningen av själva instruktionen som går fortare. Skillnaden blir 14 mot 18 klockcykler, eller knappt 30%.

För att nu krångla till det hela ytterligare, är det så att program i allmänhet arbetar med en hel del data som inte finns i register, utan på något annat ställe i — normalt — långsammare minne. Arkitekturen, "Memory-to-memory", hos TMS 9900 inbjuder till just detta. Blandningen är i regel sådan att den snabbare minnesexpansionen i praktiken ökar hastigheten med ungefär 50%.

Ett problem är att datorn faktiskt kan bli för snabb. Den snubblar så att säga över sina egna fötter. Problemet yttrar sig när man samtalar med videoprocessorn. Redan med ordinarie minne har TI angivit att en viss fördröjning ska införas i program som utnyttjar VDP RAM (E/A-manualen, 16.4). Men i och med att minnesexpansionen normalt är så långsam, kan man spara in vissa av de fördröjningarna. Detta utnyttjas också i somliga program. Men när då dessa program hamnar i ett snabbt minne, där de exekveras med upp till dubbla hastigheten, kan det bli problem. Detta visar sig exempelvis i spelet Tennis, som inte alls fungerar med en snabb minnesexpansion.

Spelet drabbas av totalt kaos, där spelarna bland annat delar sig på mitten och springer på två håll samtidigt. Med den konstruktion som jag nu har, kan jag naturligtvis koppla in den vanliga, långsamma minnesexpansionen, och köra programmet i den. Då fungerar det precis som vanligt. Men i vissa fall kan man ha nytta av den ökade snabbheten i övrigt, bara grafiken hänger med ändå. För att ordna detta byggde jag in ytterligare lite elektronik som fördröjer datorn när den vill läsa från VDP RAM. Lagringen i VDP RAM hänger med bra utan några speciella åtgärder.

För att kunna dra nytta av maximala prestanda när jag så önskar, gjorde jag även den här fördröjningen fränkopplingsbar med en CRU-bit. Många program klarar sig bra utan någon extra fördröjning.

Snabbheten i praktiken

Hur mycket snabbare blir då vanliga program? Ja, som vi redan har konstaterat beror det på i vilken utsträck-

ning de använder minnesexpansionen. TI BASIC påverkas inte alls, eftersom det språket inte kan utnyttja någon minnesexpansion. Extended BASIC ökar någon procent i snabbhet. Det mesta av X-BASIC:s verksamhet är av administrativ karaktär, och försiggår i GPL-interpretatorn eller i själva modulen. Pascal (p-systemet) kör huvudsakligen på p-kodskortet. Byråkratin är dock inte så omständig som för X-BASIC. Snabbheten ökar med ungefär 10%.

TI-Forth utnyttjar, liksom p-systemet, inbyggt RAM för den mest exekverade delen av systemet. Men då Forth inte har någon motsvarighet till p-kodskortet, utan utnyttjar minnesexpansionen i stället, ökar prestanda med lite drygt 30%. TI-Writer snabbas upp cirka 50%, vilket märks när man skriver. När man kommer till slutet på en rad, flyttas det ord som man håller på med automatiskt ner på nästa rad. I samband med den flyttningen tappar TI-Writer lätt bort ett tecken, om man trycker någorlunda snabbt på tangenterna. Det problemet försvinner nästan helt nu, åtminstone med den snabbhet som jag är kapabel att uppbrija när det gäller att skriva på maskin. Kommandon som formaterar inom givna marginaler och radering av rader går också märkbart fortare.

Explorer från MG är ett program som ställer stora krav på kapaciteten, samtidigt som det på grund av sin konstruktion måste ha både register och allt annat i minnesexpansionen. Här ökar snabbheten med ungefär 80% när man låter programmet uppdatera registernehåll och annat på skärmen under exekveringen. Kopplar man bort den funktionen, så att videoprocessorn inte bromsar lika mycket, kommer man nära den teoretiska gränsen på drygt 100%. Detta gäller även för egna assemblerprogram, om de är skrivna med E/A-modulen och använder en workspace i minnesexpansionen.

Diverse

Efter att nu ha använt det här ett tag, har jag praktiskt taget enbart positiva erfarenheter. Jag konstaterade ovan att maskinen fick 112 Kbyte RAM med den här utbyggnaden. Genom att stoppa in Maximem, och även räkna lite RAM på mina egna konstruktioner i expansionsboxen, har jag nu kommit upp i 172 Kbyte. Redan tidigare simulerade jag en enkel RAM-disk med hjälp av mitt I/O-kort och Maximem. Den simuleringen fick nu hastigt och lustigt 64 Kbyte till att ta vara på. Det hela fungerar bara tillsammans med Pascal, men det räcker för min del.

Ett problem är annars att en RAM-disk från Cor-Comp inte fungerar tillsammans med den här konstruktionen. Detta beroende på att den samtidigt utgör minnesexpansion. En sådan RAM-disk blir då förvisad till att vara en sekundär minnesexpansion, som kan kopplas in av ett program när det så önskar. Men den omkopplingen sker ju inte vid normala anrop av RAM-disken. Därför kan man inte komma åt den i egenskap av RAM-disk. Detta gäller även alla andra konstruktioner som fungerar som yttre minnesexpansion.

Däremot finns det inget hinder för en RAM-disk typ Horizon. Den utgör ingen vanlig minnesexpansion, och påverkas därför inte av den här utbyggnaden.

Vissa spel kan däremot ställa till problem. De går helt enkelt så fort att det blir ännu värre än vanligt att hinna med ... Men om man har dem i modul, eller kan ladda in dem i Maximem eller något liknande, påverkas inte snabbheten. Dessutom är tydligen somliga spel så skrivna, att de utnyttjar videoprocessorn för att hålla räkning på tiden. Då har minnets snabbhet ingen betydelse, utom i det avseendet att spelet eventuellt kan fungera bättre än tidigare. Datorn kan ju hinna med mer på den tid som en viss manöver, exempelvis flytta en sprite en viss sträcka, tar att utföra.

Jag har för närvarande ingen annan möjlighet att välja inre eller yttre minnesexpansion än just med

CRU-bitar. Enda sättet att köra exempelvis Tennis i det yttre minnet, så att det inte går fortare än vanligt, är att först ladda in Mini Memory i Maximem, gå in i Easy Bug, slå om två CRU-bitar, trycka på QUIT, välja E/A på Maximem och ladda in Tennis. Det går inte utan vidare att göra med vanliga moduler i stället för Maximem, eftersom de ger en RESET av datorn när man byter modul. Då återställes även CRU-bitarna. För att komma undan från det här problemet har jag funderat på att sätta några omkopplare på datorn, så att jag kan göra den här omkopplingen även utan att använda något speciellt program.

Triton Super Extended BASIC

modul version 120

av Jan Alexandersson

I Programbiten 87-3 hade jag skrivit en artikel om Super-XB. När detta skrevs hade jag GK Utility I till min GRAM KRACKER samt Smart Programmer dec 1986 som beskrev modulen vilket betyder att allt jag gissade inte var helt sant. Super-XB är en mycket bra modul som helt ersätter TI XB version 110. Den levereras med tre manualer:

- Extended Basic version 100 till 99/4 (224 sidor)
- Extended Basic version 110 till 99/4A (16 sidor)
- Super Extended Basic version 120 (24 sidor)

De två första är helt identiska med vad som följer med TI XB så den nya modulen är helt komplett. Det tredje häftet beskriver alla ändringar och nyheter där den typografiska utformningen av sidor och omslag är identisk med TI-manualen. Man får faktiskt leta efter det finstilt i den innan man inser att Texas Instruments inte har gjort Super Extended Basic.

Vid årsmötet 12 mars hade Sten Gunnarsson från Göteborg en mycket intressant visning av den riktiga modulen. Jag har nu skaffat en egen modul från TEX-COMP för USD 60 + porto USD 10 + Master Card USD 2 = USD 72. Paketet var märkt "SOFTWARE" och Göteborgstullen (som även tittar i paket till Stockholm) stämplade det "Undantagen från tulldeklarering". Totalkostnaden blev 434 kr. Modulen är värd sitt pris och kan även vara något för dig som redan har TI-modulen. För dig som inte har Extended Basic är detta den modul som du bör skaffa men om du ändå köper den gamla modulen så borde du inte betala mer än 200 kr för den.

Observera att denna modul inte har något gemensamt med den Super Extended Basic från JKH Software på skiva som endast består av ett stort antal CALL LINK.

Den följande beskrivningen har ursprungligen gjorts av Sten Gunnarsson men har kompletterats med mina egna erfarenheter.

Ändrade kommandon

(jämfört med version 110)

- CALL VERSION ger 120 som svar mot tidigare 110.
- CALL INIT har modifierats så att den inte tar in mer data än nödvändigt till expansionsminnet.
- LIST kan bestämma vilken radlängd man vill ha på utskriften till printer eller disk.
- CALL LOAD fungerar utan CALL INIT så du kan nu göra POKE även utan expansionsminne.
- RES (tidigare RESEQUENCE) kan nu även göra omnumrering av delar av ditt program. Exempelvis RES 20,1,10-50 innebär att rad 10 till 50 omnumreras till rad 20 med en rads avstånd.

Texas Instruments
TI-99/4A Home Computer

SUPER Extended BASIC

FOR THE TI-99/4A HOME COMPUTER

CONTENTS: SUPER Extended BASIC Cartridge
TI Extended BASIC Manual
SUPER Extended BASIC Manual

- TRACE kan även skriva mot printer vilket är mycket bra vid alla program med skärm som inte får scrollas. Alla radnummer som innebär ett hopp som kan orsakas av GOTO och GOSUB mm kommer att skrivas med en stjärna.
- Cursor (markören) har ändrat utseende till ett diskret streck vilket känns behagligare för ögat.
- Felmeddelanden skrivs ut med små bokstäver.
- Auto-load som gör att Extended Basic alltid söker efter DSK1.LOAD kan man hoppa över genom att ha tangenten nedtryckt något längre än normalt. Med en normal tryckning kan du dock få den normala auto-load att fungera.
- Cursor kan snabbt flyttas vart som helst inom ett radnummer vid editering. Det är möjligt att flytta cursor mellan början och slut och mellan två närliggande skärmrader inom samma programrad. Det är även möjligt att göra ERASE av allt till vänster eller höger om cursorn. Som sista nyhet kan även cursorn hoppa 5 tecken per tryckning eller till nästa ord om så önskas. Alla dessa nyheter fungerar även med INPUT och ACCEPT i ditt XB-program. Här finns en liten bug som märks om du försöker trycka på "hoppa till slutet" när INPUT är tom. Cursor hoppar då till vänster och den första bokstav du då skriver läses ej av INPUT-satsen.

Nya kommandon

- COPY kopierar önskad del av programmet till annan plats. Exempelvis COPY 320-450,700,5 kopierar raderna 320 till 450 till rad 700 med ett mellanrum av 5. De ursprungliga raderna finns kvar.
- DEL raderar delar av ditt program. Exempelvis DEL 500-700 innebär att du raderar raderna 500 till 700.
- MOVE kan flytta flera rader vart du vill i programmet. Exempelvis MOVE 500-700,1000,5 flyttar raderna 500 till 570 till rad 1000 med ett mellanrum av 5. Detta kommando håller reda på GOTO och GOSUB mm som omnumreras på riktigt sätt.

Nya Call för skärm

- CALL ALL fyller hela skärmen med ett enda tecken. CALL ALL(42) fyller tex skärmen med *.
- CALL COLORS ger samma färger till alla teckenset 0-14 samtidigt. Den färgar faktiskt samtliga 32 möjliga teckenset för CHR 0-255.
- CALL SCROFF stänger av skärmen.
- CALL SCRON aktiverar skärmen.
- CALL STSPRT stoppar alla sprites samtidigt.
- CALL GOSPRRT startar alla sprites samtidigt.

Nya call för tangenter

- CALL ALOCK undersöker om ALPHA LOCK är nedtryckt men störs om FCTN-tangenten trycks ned.
- CALL SHIFT undersöker om SHIFT-tangenten är nedtryckt.

— CALL CTRL undersöker om CTRL-tangenten är nedtryckt.

— CALL FCTN undersöker om FCTN-tangenten är nedtryckt. Detta kommando har en bug som gör att det inte fungerar. Själva anropet ser ut att påverka tangentavläsningen så att det ibland visar att FCTN är nedtryckt trots att det inte är fallet. Även övriga tangentavläsningar som ALOCK, SHIFT och CTRL kommer att störas av CALL FCTN.

— CALL QUITOFF läser QUIT-tangenten vilket är mycket bra för dig som råkat ut för att trycka på FCTN = av misstag.

— CALL QUITON återställer QUIT-tangenten.

— CALL KEYS("lista",variabel) väntar tills du tryckt någon tangent i listan och ger som svar tangentens position i listan. Detta är således CALL KEY med validerte. Exempel i program:

```
10 CALL KEYS("1234",T) :: ON T GOTO 100,200,300,400
```

Varning för att detta kommando ej kan brytas med CLEAR så spara programmet innan det provköres.

Nya call för minne

— CALL PEEKV läser data från VDP-RAM. Om du läser av skärmen så tänk på att 96 adderas till ASCII-koden.

— CALL POKEV skriver data till VDP-RAM. Observera att du kan köra många program för Mini Memory som använder dessa POKEV med Super-XB. Detta gäller även VDP-register som man kan komma åt genom POKEV på höga adresser.

— CALL PEEKG läser data från GROM eller GRAM.
— CALL POKEG skriver data i GRAM. Observera att Super-XB saknar GRAM så du måste ha en GRAM KRACKER eller motsvarande för att kunna använda detta.

Nya call för ljud

— CALL BEEP ger ett positivt ljud.
— CALL HONK ger missljud.
— CALL CHIMES ljud som låter pling pling pling.

Nya call för programstyrning

— CALL BYE kan även användas i ett i program.
— CALL NEW ger NEW men kan även användas i program.

— CALL CAT katalogiserar angiven disk exempelvis CALL CAT("DSK1.").

— CALL RUNPROG fungerar som RUN men du kan nu även använda en strängvariabel. Med RUN måste du alltid ha en konstant tex RUN "DSK1.PROGRAM". Du kan nu skriva

```
A$="DSK1.PROGRAM":CALL RUNPROG(A$).
```

Detta är mycket användbart i menyprogram som skall kunna starta godtyckliga programnamn.

— CALL CLSALL stänger alla öppna filer.

Nya call för klocka

— CALL CLOCK klocka som ej går riktigt på grund av skillnaden mellan 50 och 60 Hz. Den är interruptdriven så den fungerar ändå ej exakt.

— CALL CLKOFF stänger av klockan.

Nya call för hopp

— CALL GOTO(numerisk variabel) har samma fördelar och nackdelar som CALL GOSUB.

— CALL GOSUB(numerisk variabel) exempel A=640 :: CALL GOSUB(A) utför underrutinen med start på rad 640. Varning för att använda RES och MOVE som ej numererar om variabeln A men väl den rad man vill hoppa till.

— CALL RESTORE(numerisk variabel) ger RESTORE av DATA-satser men du kan nu även använda en variabel.

CALL DRAWNPLOT

Det finns ett antal högupplösande grafikrutiner i bit-map mode som kan användas om du har 32 kb expansionsminne. Innan dessa aktiveras måste följande skrivas i kommandomod:

```
CALL FILES(2)
NEW
```

Detta innebär att du ej kan starta program som har DRAWNPLOT-rutiner med auto-load av DSK1.LOAD. Observera att nedanstående kommandon ej ritar på skärmen utan i expansionsminnet. Du kan endast se bilden med SHOW eller EDIT. Följande kommandon finns:

— CALL DRAWNPLOT laddar ut grafikrutinerna till den låga delen av expansionsminnet.

— CALL LINK("GCLEAR") rensar det minnesutrymme i höga delen av expansionsminnet som lagrar grafikbilden innan den visas.

— CALL LINK("MOVE",x-pos,y-pos) plaserar pennan på önskad position utan att rita.

— CALL LINK("DRAW",x-pos,y-pos)

— CALL LINK("CIRCLE",x-pos,y-pos,radie)

— CALL LINK("SQUARE",x-pos,y-pos,sidlängd)

— CALL LINK("LABEL",textsträng)

— CALL LINK("SHOW") visar bilden på skärmen genom att bilden flyttas från expansionsminnet till VDP-minnet. Det finns risk att detta förstör vissa strängvariabler.

— CALL LINK("EDIT") ger möjlighet att editera bilden med joystick samtidigt som du kan se den på skärmen.

— CALL LINK("GDUMP",printer) skriver bilden mot Epson-kompatibel printer.

— CALL LINK("LDUMP",printer) skriver en extra stor bild mot Epson-kompatibel printer.

— CALL LINK("PWDUMP",printer) skriver normal bild mot Prowriter.

— CALL LINK("PLDUMP",printer) skriver extra stor bild mot Prowriter.

— CALL LINK("GSAVE","DSK1.FILNAMN") sparar bilden på disk med 25 sektorer. Observera att kassett ej kan användas. Detta är samma format som TI-Artist.

— CALL LINK("GLOAD","DSK1.FILNAMN") laddar in bild från disk. Även TI-Artist bilder med _P på slutet kan laddas. Med den FREEWARE MAX-RLE som finns i programbanken kan alla GRAPHX och MAX-RLE bilder omvandlas till ett format som passar Super-XB.

Referenser

1. Programbiten 87-3: Super Extended Basic
2. Micropendium sept 1987: Super X-BASIC review
3. Micropendium maj 1985: Draw 'N Plot review

SOLID STATE CARTRIDGE

SUPER EXTENDED BASIC

BDAK

©1982 TI ©1987 TPC

Från styrelsen

Föreningsstyrelsen beslöt på sitt möte i januari att vi skulle skicka ett svenskt-engelskt/engelsk-svenskt lexikon till Tony McGovern i Australien. Han har ju försett oss med regelbundna utgåvor och uppdateringar av sitt *Funnelweb* program och även program från sonen Will. Tony har skrivit tidigare till oss om sina försök att läsa Programbiten (som vi sänder honom). Vi hittade ett bra lexikon på bokrean för en hundralapp och skickade ner det till honom.

Vi har nu fått svar från Tony. Han skriver bla att han äntligen begripit sambandet mellan "nittinian" och ninety-nine. Han har också satt igång ett stort översättningsarbete — han kommer att översätta Sören Bernles artikel om Geneve till engelska för den lokala 99-klubben.

Han skriver också att en ny version (4.10) av *Funnelweb* är på gång inom kort.

Nyttoprogram, vad är det?

av Anders Persson

Nyttoprogram, de är sällsynta i Programbiten. Någon ynka gång finns en beskrivning eller en utskrift av något som är nyttigt, men för det mesta handlar tidningen om andra ting.

Provocerande uttalande? Ja, kanske det. Men det var en tanke som slog mig efter att ha bläddrat lite i några gamla tidningar. Samtidigt började jag fundera på vad det egentligen är som karakteriserar ett nyttoprogram. Vad är det som gör ett program nyttigt?

Multiplan anses vara ett typiskt nyttoprogram. Det kan man ju använda exempelvis för att ta fram ett offertunderlag eller något annat i den stilen. *TI-Writer* är väl också ett nyttoprogram? Ja, det beror på vad man menar. Om jag vill skriva ett brev använder jag naturligtvis hellre *TI-Writer* än en skrivmaskin. Ordbehandlaren gör samma arbete, fast bättre. Just det är nog den definitionen som de flesta vill använda för att avgöra om ett program är nyttigt eller inte. Men *TI-Writer* kan ju också användas till att mata in ett assemblerprogram med. Då är det ju inte nyttigt i den bemärkelsen att man åstadkommer något som utan dator hade fått göras på något annat, besvärligare sätt. Utan dator hade arbetsuppgiften ju inte funnits överhuvudtaget. Alltså har ordbehandlaren förvandlats till en editor, dvs ett program som är ett datormässigt självändamål.

Alla systemprogram till en dator är ju nyttiga så tillvida att man inte kan använda datorn till något annat, nyttigt eller ej, utan dem. Men trots det är det väl ingen som tänker på drivrutinen till tangentbordet som ett typiskt nyttoprogram.

Hur man än vrider på det, så kvarstår det faktum, att åtskilliga av de program som en dator exekverar är ett självändamål. Tursamt nog. Annars hade vi ju inte haft den här hobbyn, som leder både till intressant problemlösning och trevliga kontakter med andra intresserade. Det kan säkert vara väl så nyttigt.

Nåväl, nu tänkte jag berätta om två applikationer som en 99:a kan användas till. Åtminstone den ena kan anses vara klart nyttig, medan den andra förvisso är mer hobbymässig...

Att sätta fart på luft och annat

Många industrier är dammiga. Träindustrin, med sågverk och snickeriverkstäder, är ett typiskt exempel. Gemensamt för dem alla är att man vill bli av med dammet så bra som möjligt. Arbetsmiljön vinner ju inte på att dammet far runt i lokalerna. Dessutom utgör fin, torr sågspån i ett sågverk eller sockerdamm i ett sockerbruk en stor brandfara.

Ventilationsanläggningar för dammsug byggs ofta på gehör. Så verkar det i alla fall när man tittar på dem och deras funktion. Detta trots att det går att göra hyfsat bra teoretiska beräkningar i förväg. Lufthastighet och rördimensioner har stor betydelse för anläggningens prestanda, och bör alltså helst väljas någorlunda systematiskt och inte på måfå.

Beräkningarna är inte speciellt komplicerade, men däremot ganska omfattande. Ett system med en stamledning med enkla grenledningar kan beräknas rakt fram från ena änden till den andra, men om flera stamledningar med sina grenledningar går samman, måste de anpassas till varandra. Anpassningen kan leda till att lufthastigheten ändras så mycket att rördimensionerna också bör ändras, vilket i sin tur påverkar undertrycket och hela anpassningen. Ett utmärkt exempel på ett tråkigt arbete med massor av siffror, som alltså är synnerligen väl lämpat för en dator i stället.

Sven Lundgren, ägare till Process Ventilation i Staffanstorp, insåg detta för några år sedan. Han försökte sig på att skriva ett program i Extended BASIC, något som också lyckades till slut, med hjälp från några av oss andra i Programbiten.

När Sven så skaffade ett p-kodskort gavs möjligheten att

skriva om hela härligheten i Pascal. Efter en hel del utvecklande har programmet vuxit till en källkod på över 70 Kbyte. Kapaciteten har ökat till ungefär det dubbla jämfört med BASIC-versionen, samtidigt som snabbheten ökat ungefär sju gånger. Dessutom klarar programmet av hopkoppade stamledningar med dess anpassningsproblem, något som inte gick med BASIC programmet. Därmed går det nu att göra beräkningar på system med godtyckligt utseende. Pascalprogrammet är också försett med uppgifter om priser på olika rör och andra komponenter, och levererar raskt en sammanställning av materialkostnaderna. Materialåtgången, dvs rördimensioner, böjar, förgreningar m m, redovisas också, liksom tekniska data på anläggningen.

Ett större rörsystem, med kanske 100 olika delar, dimensioneras på ungefär tre minuter. När väl grunddata (rörens längd, antal böjar etc.) är inmatade, kan man sedan gå in och ändra på lufthastigheter och dimensioner och se vad resultatet av manipulationerna blir. Kostnaden för olika materialalv beräknas och skrivs ut på någon minut.

Flera av *UCSD Pascals* goda egenskaper har fått användas för att få med alla möjligheterna på en maskin av 99:ans format. Rörsystemet representeras med en dynamisk, fyrvägslänkad (faktiskt!) lista. Strukturen på denna är sådan att den lämpar sig bäst för bearbetning med rekursiva procedurer, något som också används till flera av programmets funktioner. Programmet är delat i sju olika separatkompilerade units, vilka i sin tur är segmenterade så att hela programmet består av 14 segment.

Att mäta tid i realtid

I samband med att en kusin till mig dammade av en ungefär tjugo år gammal bilbana, föll det mig in att man borde kunna installera datoriserad varvräkning och tidtagning. Klockan som jag använde till mitt klockkort, *Nationals MM58167A*, har en upplösning på en tusendels sekund, vilket ju torde vara tillräckligt. Genom att placera någon lämplig givare i banan, skulle det gå att visa mellan- och varvtider på skärmen när bilarna susar förbi.

Det här är ett typiskt exempel på realtidsprogrammering. I alla mer normala tillämpningar, som 99:an figurerar i, är det så att det är datorn som bestämmer arbetstakten. I ett kalkylprogram får du, som användare, vänta tills datorn har räknat om kalkylen. När ett program assembleras får du vackert vänta tills den är färdig. Det gör du också, helt enkelt därför att du inte har något val. Men när det gäller tidtagning på bilarna är det plötsligt så att datorn får se till att hålla jämna steg med verkligheten på den tid som finns att tillgå. När tävlingarna pågår som bäst väntas det definitivt ingenting!

Frågan var alltså om 99:an är snabb nog att hinna med om två bilar passerar tätt efter varandra. Processorn 9900 är förvisso snabb nog i sig. Om det hade gått att utnyttja dess möjligheter till avbrotts hantering fullt ut, skulle det gå att läsa av klockkretsen åtta gånger om på en tusendels sekund. (Med tanke på de som eventuellt funderar på en Geneve kan nämnas att TMS 9995 skulle klara av 20 avläsningar på samma tid.)

Men möjligheterna att generera ett avbrott är kraftigt begränsade i 99:an. Det går visserligen att begära avbrott utifrån, men de manövrer som operativsystemet har för sig för att fundera ut var avbrottet kommer ifrån är alltför vidlyftiga för att kunna accepteras i det här sammanhanget. Visserligen finns LOAD-ingången kvar, men ett avbrott via den måste synkroniseras med processorns inläsning av instruktioner, vilket kräver ytterligare elektronik.

Därför återstod endast möjligheten att i en slinga känna av givarna växelvis. Efter det att en givare påverkats, skall klockan läsas av, mellan- och varvtid räknas ut, samt omvandlas till text och visas på skärmen. Dessutom jämförs varje varvtid med den bästa, så att bästa varvtiden kontinuerligt kan uppdateras på skärmen. Av de olika momenten

är det framför allt omvandlingen av tiden till text som tar processornas kapacitet i anspråk.

Praktiska prov visade att datorn behövde tre tusendels sekunder för att läsa av klockan och visa mellan- och varvtid på skärmen. Bilarna, som är i skala 1/32, når en skalenlig toppfart av ungefär 340 km/h. Omräknat till verklig fart betyder det att bilarna på tre tusendels sekunder tillryggaläger ungefär en centimeter.

Den noggrannheten kan ju vara tillräcklig. Det största problemet hittills har visat sig vara att åstadkomma lämpliga givare. Vi ville helst undvika mekaniska sådana. Vissa av bilarna har tillräckligt kraftiga magneter för att det ska gå att känna av dem på magnetisk väg, men det går inte med alla. För närvarande funderar vi på hur en optisk givare lämpligen ska utformas.

Som vanligt är det problemlösandet som är intressantast...

Graden av nyttighet i det första exemplet är utan tvekan stor. Det går utmärkt att prova flera olika dimensioner på en anläggning, kontrollera deras prestanda och kostnad, och på så sätt snabbt skaffa ett gott underlag för en offert.

Nyttigheten i det senare exemplet kan säkert diskuteras, men det råder väl inget tvivel om att det i stort sett bara är fantasin som sätter gränsen för vad man kan ha en dator till. Kom gärna med fler exempel på allvarliga, roliga, nyttiga eller rent av häpnadsväckande exempel på vad just du använder datorn till.

Annonser

Säljes:

TI-99/4A komplett grundenhet + expansionsbox inkl: 32 K minnesexpansion, diskdrive, diskkontrollkort, RS232-kort, flexcable-kort + speech synthesizer + Maximem-modul med batteribackup + TI-joysticks + joystick-adapter.

Moduler: Disk manager 2, Personal Record Keeping + disketter för XB och Maximem; Master disk file, smash, personal record keeping, Spad XIII, Extended BASIC, Tunnels of Doom, Pole Position, Paint & Print, TI-Runner, TI-Writer, Karate, Par fore + ca 20-talet disketter.

Pris: 5 000 kr.

Anders Ljung
Fänestad
330 12 Forsheda
Tel. 0370/810 42

Säljes:

2 st diskdrivar TI-original 250:— st.
Micropal-disketter blanka DS/DD 10 st 75:—
MBX-system + Baseboll 600:—
TI Logo 2 400:—
TI Forth 4.5 150:—
Ca 10 Atari-moduler från 75:—/st.
Ca 15 TI-moduler från 50:—/st.

Jag önskar kontakt med Adventure-fanatiker. Jag själv är förmodligen en sådan. Jag har en hel del spel till Adventure-modulen och även en del från Infocom. Har även ca 10 st grafikadventure och ungefär 10 st spel till TOD. Är speciellt intresserad av att få kontakt med någon som har TOD-editorn, men Ni andra som känner er träffade tveka inte, ring!!!

Joacim Göransson
Industrigatan 13
595 00 Mjölby
tel. 0142/119 27

Horizon RAM-disk

av Jan Alexandersson

Jag har beskrivit Horizon RAM-disk i tidigare nummer av Programbiten 87-3, 87-4 och 88-1. Sedan dess har jag fått MENU version 7.3 från Miami TI Users Group daterad 15-01-88. Den finns nu i programbanken dels som programmet på en skiva och dels källkoden på två skivor.

MENU 7.3 har inga större förändringar jämfört med 7.1. De buggar jag beskrivit finns fortfarande kvar. Det enda nya jag hittat är att programmet kollar varje 2 kb-block vid initieringen av disken mot tidigare 8 kb-block. Detta har gett mig 6 kb extra på min 392 kb RAMDISK som nu kan delas på 96 + 1440 sektorer. En ny bug som jag hittat är att "HD" finns i operativsystemet utan att vara beskrivet i manualen. Anrop av RAM-disken med "HD.RAMDISK.PROGRAM" i stället för "DSK.RAMDISK.PROGRAM" kommer att krascha hela RAM-disken.

Försäljningen av Horizon RAM-disk har tagits över av:

Bud Mills Services
166 Dartmouth Drive
Toledo Ohio 43614
USA

Denna firma har tidigare varit huvudleverantör för alla komponenter som har använts i RAM-disken.

RAM-disken säljs nu med 32 kb kretsar 62256LP12 och kallas då "HRD+ RAMDISK". Detta betyder att det är lättare att bygga själv när kretsarna inte behöver staplas ovanpå varandra vid normala storlekar på RAM-disken. Följande alternativ finns:

- Tomt kretskort (utan manual/skiva) USD 38
- HRD+RAM 96k SSSD pris USD 140
- HRD+RAM 192k DSDD pris USD 165
- HRD+RAM 384k DSDD pris USD 225
- HRD+RAM 512k pris USD 265
- HRD+RAM 800k pris USD 375
- HRD+RAM 1Mb pris USD 435

Alla dessa priser gäller kompletta byggsatser inklusive manual och flexskivor. Om du önskar kortet färdigt tillkommer USD 60 per kort. Som operativsystem används MENU version 7.3. Detta innebär att man inte kan initiera större disk än 1440 sektorer (360 kb) men eftersom man samtidigt kan initiera två RAM-diskar på samma kort med olika DSK-nummer går det utmärkt med ett dubbelt så stort kort. Du bör dock tänka på att det är praktiskt att ha samma storlek på dina fysiska drivrar som du initierar RAM-disken till dvs att om du har 180 kb drivrar bör du köpa en 384 kb RAM-disk.

Den slutsats man kan dra är att det normala är att satsa på 384 kb men att även 800 kb är användbart. Jag tror att det är direkt olämpligt att satsa på 1 Mb eftersom denna storlek gör att du måste dela upp kortet i tre RAM-diskar där den första kan anropas med nummer och de två sista måste anropas med SKIV-NAMN eftersom de alltid får samma DSK-nummer. Om du har råd att köpa 1 Mb så satsa på två RAM-diskar som tillsammans kan initieras med 4 DSK-nummer. Operativsystemet medger DSK-nummer 1—9.

Instruktioner för Loudspeaker construction

copyright: Snake computer software
pieere 1987

LSPEAK kan användas för att beräkna basreflex- och slutna lådors storlek för ett visst högtalarelement. De data som är nödvändiga för att programmet ska fungera är:

Speker resonance — högtalarens resonans frekvens
Membrane area — högtalarens membran area i cm^2
Membrane weight — högtalarens membran vikt i gram
Total Q-value — högtalarens TOTALA q värde

För beräkning av bas reflex behövs även:
Vent diameter — bas reflex tubens DIAMETER

Exempel

Vi säger att du vill använda en 8" högtalare i en sluten låda.

Högtalarens data är:
Resonans frekvens: 40 Hz
Membran area: 238 cm^2
Membran vikt: 18.5 g.
Q-värde: 0.45

Detta ger:

Låd volym (box volume): 46.8 liter
Systemets resonans frekvens: 44 Hz

Säg att vi vill använda samma högtalare i en bas reflex låda:

Diameter på basreflex röret: 7 cm

Detta ger:

Låd volym : 38.2 liter
systemets resonans frekvens: 36 Hz
basreflexrörets längd: 18.2 cm

OBS!! Q-värdet på högtalaren måste vara mellan 0.2 och 0.6 vid beräkning av bas reflexlådor (andra högtalare är olämpliga).

Man kan efter att ha räknat ut basreflex låda eller sluten låda trycka på PROCEED och få ut resultatet på skrivare.

Skrivar-definitionen sätts genom att trycka på AID när man befinner sig i huvudmenyn.

Utprintningen är i DIS/VAR 80 format och alltså även spara beräkningar på diskett (eller kasset).

Passiva delningsfilter

LSPEAK kan även räkna ut passiva delnings filter för högtalare (OBS!! man kan ej få ut detta på skrivare)

De olika filtren är antingen för:

Bas- och diskant högtalare med brantheten 6 eller 12 db/oktav

Bas-, mellan- och diskant högtalare med branthet 6 eller 12 db/oktav.

Data som behövs:

Högtalarens impedans i ohm
Delnings frekvenserna

Vid problem eller frågor:

skriv till S.c.s., Pl 709, 741 00 Knivsta
eller Per Virving 018-38 52 01

```
100 CALL CLEAR :: CALL SCREEN(13):: FOR A=0
    TO 14 :: CALL COLOR(A,16,1):: NEXT A
110 CALL C
115 ON WARNING NEXT
120 LN$=" "&RPT$(CHR$(143),26)
130 DEF CTR$(X$)=SEG$("          ",1,14-LEN
    (X$)/2)&X$
140 DISPLAY AT(5,1):LN$ :: DISPLAY AT(8,1):C
    TR$("loudspeaker"): :CTR$("construction"
    )
150 DISPLAY AT(13,1):LN$
155 DISPLAY AT(24,1):CTR$("initializing,plea
    se wait")
160 DISPLAY AT(16,1):CTR$("by p.virving 1987
    "):CTR$("rel. 1.2"): :CTR$("é snake comp
    uter software")
170 DIM QL(40)
175 HC$="RS232"
180 SCS$="sp.cons.          éscs.pieere-87"
190 RESTORE 20000
200 FOR A=1 TO 40 :: READ L :: QL(A)=L :: NE
    XT A
300 CALL CLEAR :: DISPLAY AT(1,1):SCS$:LN$
320 DISPLAY AT(6,5):"1..bass reflex": : "
    2..closed box": : "      3..filter 6db lf-h
    f": : "      4..filter 6db lf-mf-hf"
322 DISPLAY AT(14,5):"5..filter 12db lf-hf":
    : "      6..filter 12db lf-mf-hf"
330 DISPLAY AT(20,10):"your choice:" :: NR=6
    :: CALL ANSWER(NR,20,24)
335 IF NR=-47 THEN GOSUB 9000 :: GOTO 300
340 ON NR GOTO 1000,2000,3000,4000,5000,6000
1000 BR=1 :: CALL CLEAR :: DISPLAY AT(1,1):SC
    S$ :: DISPLAY AT(2,1):LN$
1005 DISPLAY AT(4,1):CTR$("b a s s r e f l e
    x"):CTR$(RPT$(CHR$(143),16))
1010 CALL IN(FO,SD,MO,QO)
1015 IF (QO<0.2)OR(QO>0.6)THEN DISPLAY AT(21,
    1):CTR$("Q value must be between"):CTR$(
    "0.2 and 0.6"):: CALL PETC(A):: IF A=0 T
    HEN 300 ELSE 1010
1020 DISPLAY AT(11,1):"ö vent diameter (cm)
    ": : ACCEPT AT(11,25)VALIDATE(DIGIT,"."
    )SIZE(4):DV
1030 VB=(355*(SD/PI)Ü2)/(2*QL(QO*100-20)*FOÜ2
    *MO)
1040 FC=QL(QO*100-19)*FO
1050 IP=((2.36*10000*DVÜ2)/(FCÜ2*VB))-0.74*DV
1060 DISPLAY AT(14,1):"box vol. (l)":;VB;"ve
    nt len.(cm)":;IP;"syst.res.(hz)":;FC
1070 CALL PETC(A):: IF A=0 THEN 300 ELSE IF A
    =2 THEN 8000 ELSE 1000
2000 BR=0 :: CALL CLEAR :: DISPLAY AT(1,1):SC
    S$:LN$ :: DISPLAY AT(4,1):CTR$("c l o s
    e d b o x"):CTR$(RPT$(CHR$(143),14))::
    CALL IN(FO,SD,MO,QO)
2010 VB=(355*(SD/PI)Ü2)/(MO*(0.5/QOÜ2-1)*(FOÜ
    2))
2020 FC=SQR((29*SDÜ2)/(VB*MO))
2030 DISPLAY AT(14,1):"box vol. (l)":;VB;"sy
    st.res.(hz)":;FC
2040 CALL PETC(A):: IF A=0 THEN 300 ELSE IF A
    =2 THEN 8000 ELSE 2000
3000 R=0 :: CALL CLEAR :: DISPLAY AT(1,1):SCS
    $:LN$ :: DISPLAY AT(4,1):CTR$("f i l t e
    r 6 db lf - hf"):CTR$(RPT$(CHR$(143),
    22))
3010 RESTORE 10000
3020 GOSUB 9500
3030 DISPLAY AT(12,1):"cut freq.(hz)": :: ACC
    EPT AT(12,16)VALIDATE(DIGIT)SIZE(5):F
3040 DISPLAY AT(13,1):"loudsp.imp(O)": :: ACC
    EPT AT(13,16)VALIDATE(DIGIT,".")SIZE(3):
    RO
3050 C1=1/(6.28*F*RO)
3060 L1=RO/(6.28*F)
3070 DISPLAY AT(16,1):"c1=";C1*1000000;"uf":
    "11=";L1*1000;"mh"
```

```

3080 CALL PETC(A):: IF A THEN 3000 ELSE 300
4000 CALL CLEAR :: DISPLAY AT(1,1):SCS$:LN$ :
: DISPLAY AT(4,1):CTR$("f i l t e r 6db
1f-mf-hf"):CTR$(RPT$(CHR$(143),22))
4010 R=0 :: RESTORE 10100 :: GOSUB 9500
4020 DISPLAY AT(13,1):"low cut freq(hz)"::"h
igh cut freq(hz)"::"speaker imp. (0)":
4030 ACCEPT AT(13,20)VALIDATE(DIGIT)SIZE(5):F
1 :: ACCEPT AT(14,20)VALIDATE(DIGIT)SIZE
(5):F2 :: ACCEPT AT(15,20)VALIDATE(DIGIT
, ".")SIZE(3):RO
4040 C1=1/(6.28*F1*RO)
4050 C2=1/(6.28*F2*RO)
4060 L1=RO/(6.28*F1)
4070 L2=RO/(6.28*F2)
4080 DISPLAY AT(17,1):"c1=";C1*1000000;"Uf":
"c2=";C2*1000000;"Uf": "11=";L1*1000;"Mh":
"12=";L2*1000;"Mh"
4090 CALL PETC(A):: IF A THEN 4000 ELSE 300
5000 CALL CLEAR :: DISPLAY AT(1,1):SCS$:LN$ :
: DISPLAY AT(4,1):CTR$("f i l t e r 12d
b 1f-mf-hf"):CTR$(RPT$(CHR$(143),20))
5010 R=0 :: RESTORE 10200 :: GOSUB 9500
5020 DISPLAY AT(13,1):"cut freq(hz)"::"sp. i
mp.(0)": : ACCEPT AT(13,15)VALIDATE(DIG
IT)SIZE(5):F :: ACCEPT AT(14,15)VALIDATE
(DIGIT, ".")SIZE(3):RO
5030 L1=RO*.23/F
5040 C1=1/(8.88*F*RO)
5050 DISPLAY AT(16,1):"c1=c2=";C1*1000000;"Uf
": "11=12=";L1*1000;"Mh"
5060 CALL PETC(A):: IF A THEN 5000 ELSE 300
6000 CALL CLEAR :: R=0 :: DISPLAY AT(1,1):SCS
$:LN$
6010 DISPLAY AT(4,1):CTR$("f i l t e r 12db
1f-mf-hf"):CTR$(RPT$(CHR$(143),22))
6020 RESTORE 10500 :: GOSUB 9500
6030 DISPLAY AT(15,1):"low cut freq(hz)"::"h
igh cut freq(hz)"::"speaker imp. (0)":
6040 ACCEPT AT(15,20)VALIDATE(DIGIT)SIZE(5):F
1 :: ACCEPT AT(16,20)VALIDATE(DIGIT)SIZE
(5):F2 :: ACCEPT AT(17,20)VALIDATE(DIGIT
, ".")SIZE(3):RO
6050 L1=RO*.23/F1
6060 L2=RO*.23/F2
6070 C1=1/(8.88*F1*RO)
6080 C2=1/(8.88*F2*RO)
6090 DISPLAY AT(19,1):"11=13=";L1*1000;"Mh":
"12=14=";L2*1000;"Mh": "c1=c3=";C1*1000000
;"Uf": "c2=c4=";C2*1000000;"Uf"
6100 CALL PETC(A):: IF A THEN 6000 ELSE 300
8000 OPEN #1:HC$
8010 PRINT #1:TAB(10);"==LOUDSPEAKER CONSTRUC
TION==" :TAB(10);"====
===="
8020 PRINT #1 : :TAB(10);"Speaker resonance:
";FO:TAB(10);"Membrane area ";SD
8030 PRINT #1:TAB(10);"Membrane weight ";MO
:TAB(10);"Total Q value ";QO
8040 IF BR THEN PRINT #1:TAB(10);"vent diamet
er ";DV
8050 PRINT #1 : : :
8060 PRINT #1:TAB(10);"Box volume ";VB:TAB(1
0);"System res.":FC
8070 IF BR THEN PRINT #1:TAB(10);"vent len.
":IP
8080 PRINT #1 : :TAB(10);"=====
=====
====="
8090 CLOSE #1
8100 GOTO 300
9000 CALL CLEAR :: CALL CS :: DISPLAY AT(1,1)
:SCS$:LN$ :: DISPLAY AT(4,1):CTR$("print
er description"):CTR$(RPT$(CHR$(143),16
))
9010 DISPLAY AT(12,1):HC$ :: ACCEPT AT(12,1)S
IZE(-28):HC$
9020 CALL CLEAR :: CALL C :: RETURN
9500 R=R+1 :: READ P$ :: IF P$="XXX" THEN RET
URN ELSE DISPLAY AT(6+R,1):CTR$(P$):: GO
TO 9500

```

```

10000 DATA "ACBCCCD "
10002 DATA "11E c1F "
10004 DATA " L1f Lhf"
10006 DATA "ACHCCCG "
10008 DATA XXX
10100 DATA "ABCCCBCCCD "
10102 DATA " N c1F c2F "
10104 DATA " E11 E12 N "
10106 DATA " L1f Lmf Lhf"
10108 DATA "AHCCCHCCCG "
10110 DATA "XXX"
10200 DATA "ACBCCCCCD "
10202 DATA "11E c1F "
10204 DATA " ID ID "
10206 DATA "c2FL1f 12ELhf"
10208 DATA "ACHHCCCCCHG "
10210 DATA "XXX"
10500 DATA "ACBCCCCCCCCBCCCCCCCCCD "
10502 DATA " E11 c1F f2F "
10504 DATA " N 12E N "
10506 DATA " ID ICCD ID "
10508 DATA " NN KHD N NN "
10510 DATA "f3FL1f 13E Fc4Lmf 14ELhf"
10512 DATA "ACHHCCCCCHCHCCHCCCCCHG "
10514 DATA "XXX"
19999 GOTO 19999
20000 DATA 1.8,1.73,1.68,1.63,1.56,1.50,1.43,1
.38,1.32,1.28,1.24,1.22,1.21,1.19,1.17,1
.14,1.12,1.09
20010 DATA 1.07,1.05,1.03,1.,.94,.92,.89,.88,.8
6,.84,.82,.80,.77,.75,.73,.71,.70,.69,.6
8,.67,.66,.65
20100 SUB CS
20110 CALL CHARSET :: FOR A=0 TO 14 :: CALL CO
LDR(A,16,1):: NEXT A
20120 CALL CHAR(143,"0000FFFF")
20130 SUBEND
20200 SUB C
20210 CALL CHAR(65,"00000B170B",66,"000000FF0B
0B0B0B000000FF",68,"000000F0B0B0B0B0B")
20220 CALL CHAR(69,"0C10100C10100C0B0B0B3E003E
0B0B0B0B0B0B0BFB",72,"0B0B0BFF",73,"0B0B0B
0F0B0B0B0B0B0B0B0B")
20230 CALL CHAR(75,"0000000F0B0B0B0B",76,"0903
071F1F070309")
20240 CALL CHAR(79,"18244242422446600",77,"0000
80E692929292",85,"0000121212122C40",78,"
0B0B0B0B0B0B0B0B")
20250 CALL CHAR(143,"0000FFFF000000"):: CALL C
HAR(64,"3C429DA1A19D423C",124,"013A444A5
2225C80")
20260 SUBEND
25000 SUB PETC(A)
25010 DISPLAY AT(24,1):" press redo or back
"
25020 CALL KEY(0,K,S):: IF S=0 THEN 25020
25025 IF K=12 THEN A=2 :: SUBEXIT
25030 IF K=6 THEN A=1 :: SUBEXIT
25040 IF K=15 THEN A=0 :: SUBEXIT
25050 GOTO 25020 :: SUBEND
30000 SUB ANSWER(NR,R,C)
30010 DISPLAY AT(R,C)SIZE(1):"1"
30020 COUNT=COUNT+1 :: IF COUNT=10 THEN CALL H
CHAR(R,C+2,30)
30030 IF COUNT=20 THEN DISPLAY AT(R,C):"1" ::
COUNT=0
30040 CALL KEY(0,K,S):: IF S THEN 30090
30050 GOTO 30020
30090 IF K=13 THEN K=49 :: GOTO 30110
30095 IF K=1 THEN 30110
30100 IF (K<49)OR(K>48+NR)THEN 30020
30110 NR=K-48
30120 SUBEND
32000 SUB IN(FO,SD,MO,QO)
32010 DISPLAY AT(7,1):"resonance freq. (hz):
": "membrane area (cm2)": "membrane we
ight (g)": "total Q value "
32020 FOR A=1 TO 4 :: ACCEPT AT(6+A,25)VALIDAT
E(DIGIT, ".")SIZE(4):S(A):: NEXT A
32030 FO=S(1):: SD=S(2):: MO=S(3):: QO=S(4)
32040 SUBEND

```

Tillbaka till BASIC

av Jan Alexandersson

Jag skall försöka att ta upp några frågor som kan vara av intresse för dig som har TI-99/4A med kassettbandspelare och använder BASIC.

Fördelar med TI-99/4A BASIC

Till att börja med vill jag påstå att BASIC till TI-99/4A inte är så dålig som många tror. Du kan få automatisk radnumrering med NUM och omnumrering av rader med RES. ALPHA LOCK låser endast bokstavstangenterna men inte siffertangenterna. Datorn räknar faktiskt med 14 decimaler trots att inte alla syns på skärmen med PRINT. Du kan använda tiopotenser upp till 127 trots att endast upp till 99 syns på skärmen med PRINT. Du har även möjlighet att använda långa variabelnamn med upp till 15 tecken där även ÅÅOE_ kan användas.

PRINT med 14 siffror

Jag har gjort ett litet kort BASIC-program som visar att datorn verkligen använder 14 siffror. Alla tal lagras i ett format som kallas RADIX 100 med 8 bytes per tal. Sju bytes används för att lagra 14 siffror som hundrapotenser (0-99 per byte) och en byte används för att sätta ut decimalpunkten (64 hundrapotenser = 128 tiopotenser) samt markera +/- före talet och +/- före potensen.

```
100 TAL=SIN(1)
110 PRINT STR$(TAL)
120 GOSUB 150
130 PRINT TAL$
140 END
150 REM ALLA DECIMALER
160 REM TAL=.1 TILL <1E9
170 REM JAN ALEXANDERSSON
180 REM 1988-04-02
190 DEF LOG10(X)=LOG(X)/LOG(10)
200 POTENS=INT(LOG10(TAL))
210 TAL1=INT(TAL*10^(7-POTENS))
220 TAL2=INT((TAL-TAL1/(10^(7-POTENS)))*10^(14-POTENS))
230 TAL$=STR$(TAL1)&STR$(TAL2)
240 TAL$=SEG$(TAL$,1,POTENS+1)&"."&SEG$(TAL$,POTENS+2,14-POTENS-1)
250 RETURN
```

Du kan prova att byta TAL=SIN(1) mot TAL=SIN(1) 10/10 och observera skillnaden. Du ser att datorn använder 13 eller 14 siffror beroende på om det är udda eller jämn tiopotens. Det finns inget behov att skriva ut så många siffror i ett verkligt program men det känns bra att datorn räknar med så stor noggrannhet så att man slipper avrundningsfel.

I praktiken har du nog behov att kapa bort decimaler. Du kan göra så här med avrundning av sista decimalen:

```
100 TAL=4 ATN(1)
110 TAL=INT(TAL 1000+.5)/1000
120 PRINT TAL
```

Om du har Extended Basic kan du prova antalet decimaler och avrundning med PRINT USING.

Tiopotenser upp till 127

Jag har även gjort ett kort BASIC-program för att visa att TI-99/4A klarar mycket stora potenser vilket inte är så vanligt bland hemdatorer. XB-ägare kan även prova med PRINT USING.

```
100 TAL=1.1234E123
110 PRINT STR$(TAL)
120 GOSUB 150
130 PRINT TAL$
140 END
150 REM ALLA POTENSER
160 REM JAN ALEXANDERSSON
170 REM 1988-04-02
180 DEF LOG10(X)=LOG(X)/LOG(10)
```

```
190 POTENS=INT(LOG10(ABS(TAL)))
200 DECIMALER=TAL/10^POTENS
210 TAL$=STR$(DECIMALER)&"E"&STR$(POTENS)
220 RETURN
```

EDIT (ändra programrad)

När du har ett program som du vill ändra så måste du editera berörd programrad. I BASIC har du kanske lärt dig att skriva EDIT 160 om du vill ändra rad 160. Eftersom detta inte finns i Extended Basic bör du redan från början vänja dig vid att skriva 160 FCTN X (pil ner) eller 160 FCTN E (pil upp). När du ändrat raden färdigt så fortsätt till nästa rad med "pil ner" eller "pil upp". Du kommer ur editeringsläget genom att trycka på ENTER. Varning skriv inte radnummer ENTER om du inte vill sudda denna rad.

Det är även möjligt att editera med NUM så du kan skriva NUM 160. För säkerhets skull bör du göra RES först eftersom NUM endast visar var 10:e rad. Mellanliggande rader visas ej. Du kan inte heller göra ERASE av en hel rad om NUM används.

Om du någon gång får problem med att knappa in en lång programrad från en tidning så att datorn vägrar att ta emot flera tecken så gör så här. Avsluta raden var som helst men se till att du inte får SYNTAX ERROR. Om det är en textsträng så avsluta med " eller sätt ut parenteser som behövs. När du fått en riktig rad tryck sedan på ENTER och tag sedan tillbaka raden för editering. Du upptäcker nu att du kan skriva en mycket längre programrad.

Kassettbandspelare

Den bandspelare du använder bör ha diskantkontroll och diskanten skall vridas på maximalt. Volymen bör vridas på relativt mycket men inte maximalt vid avspelning. Om du får felmeddelande "NO DATA FOUND" bör du öka volymen något och om du får felmeddelande "ERROR IN DATA" så minska volymen något.

Din kassettspelare bör även ha fjärrkontroll (den svar-ta sladden med den tunnare pluggen). Det finns ingen standard på denna kontroll så vissa kassettspelare måste ha en inverteringsplugg mellan kassettkabeln och kassettspelaren. Fjärrkontrollen är nödvändig vid filhantering med kassett. Att spara och ladda program går dock bra utan fjärrkontroll även om det kanske är något bekvämare.

De äldre TI-99/4A har möjlighet till både CS1 och CS2 medan nyare endast klarar CS1. Denna CS2 kan dock bara användas för att lagra med SAVE eller PRINT#. Dessutom har TI tagit bort CS2 från nya kassettkablar. Det är nog inte säkert att alla TI-99/4A-ägare har den optimala sladden eftersom dessa två ändringar gjorts oberoende av varandra.

Som du redan vet används SAVE och OLD för att spara respektive ladda program. Om du skulle ångra dig efter att du tryckt på ENTER så tryck på E för EXIT så avbryter datorn det du hade börjat med. Observera att detta fungerar endast när bandet står still.

På liknande sätt kan du även jämföra två bandinspelningar med varandra och ta reda på om programmen är lika. Ladda först in med OLD CS1 från första bandet och skriv sedan SAVE CS1 (ENTER). Tryck sedan på C för CHECK och ladda in från det andra bandet. Datorn kommer nu att jämföra de två versionerna av programmet men kan inte visa vad som skiljer utan endast att något skiljer.

Mer att läsa

1. Nittinian 83-2: Att använda grafik
2. Programbiten 84-3: BASIC programmering
3. Programbiten 84-4: Tips och tricks
4. Programbiten 85-3: Kommandon i Basic och XB
5. Programbiten 85-4: CALL KEY