

# HACKER

nummer 4 \* årgång 3

NU KOMMER  
MEDLEMSSPELEN



**H**ej alla glada (?) medlemmar.

Nu sitter ni här med ett rykande färskt nummer av tidningen Svi & Msx Hacker. Med det här numret följer medlemsspelen till alla som inte fick dem och vårt fina medlemskort tillsammans med det förra numret.

Som ni säkert märkte var det förra numret starkt försenat men i och med att det här numret kommer ut så borde vi vara tillbaka i tiden. Det har gått en massa rykten om klubben och klubbens framtid och därför hade jag tänkt berätta lite om hur vi som jobbar med tidningen ser på framtiden. Att vi lever och mår bra är ganska tydligt och som det ser ut nu finns det ingen anledning att anta att vi inte skulle leva i samma välmående även nästa år. Till nästa år ska vi försöka hålla de utgivningstider för tidningen som vi har satt upp. Vi har till tidningen fått in vissa synpunkter på tidningens utseende och behandlingsområden. Dessa är ofta av starkt praktisk karaktär, typ att vi borde ha större text och liknande men de behandlade jag ju förra gången. Vad vi nu vill ha in är synpunkter på klubbens aktiviteter och framför allt synpunkter på vilka ämnen vi ska ta upp i tidningen och vilken svårighetsgrad ni vill att våra artiklar ska ha, för det är ju faktiskt ni medlemmar som ska säga vad ni vill ha. Glöm heller inte att vi gärna besvarar alla frågor ni skickar per brev eller när ni ringer och frågar någonting även små frågor som hur man "LOCATEAR I SCREEN 2 PÅ MSX" och stora som "VARFÖR FUNGERAR INTE MIN MASKINKODSRUTIN TILLSAMMANS MED MITT HEMBYGGDA INTERFACE I INTERRUPTMOD 2". Alla frågor besvaras med glädje,

almänintressanta frågor kommer att tas upp i tidningen i samma utsträckning som förut. Bland de brev vi har hunnit få in innan nästa nummers pressstopp tänker vi dra två stycken (en Msxägare och en Sviägare) som får ett "Decathlon" respektive "Megalone" (se tidigare nummer för recension) speloriginal. "Vinnarna" presenteras i nästa nummer av Svi & Msx Hacker.

Eftersom förra numret blev så försenat har jag ännu inte fått några download/upload rutiner till mitt kära komsystem (ja det stavas fatiskt med k och inte med c som det felaktigt stod i förra numret). Om jag inte får några sådana rutiner blir jag väl tvungen att skriva dessa själv, hur som helst så om inget oväntat händer så kanske vi till nästa år har fått en modem bas endast för de nu över tusen (!) medlemmarna i SSMK. Jag har ännu inte fått några synpunkter på om en modembas vore en vettig utveckling av klubbens aktiviteter. Som ni säkert vet är vi tacksamma och belönar alla införda bidrag till tidningen. Dessa tar vi helst emot på papper men har ni ingen printer så ring till oss så diskuterar vi oss fram till vilken ordbehandlare ni kan skriva på så att vi kan skriva ut den hos oss.

Till nästa års klubbarbete söker vi redan nu två medlemsspel som kan vara skrivet på både Msx och Spectravideo. Ett av dessa kan gärna vara ett adventure eller rent av ett nyttoprogram.

Vi som arbetar med klubben, d.v.s Anders och Daniel, hoppas att väntan på det här numret inte har varit för lång och att ni slipper vänta alltför länge på nästa nummer som blir ett dubbelnummer.

# INNEHÅLL

Nummer 4, årgång 3, 1986

Framsida ..... sid 1

Prolog om tidningens och klubbens framtid ..... sid 2

Innehåll mm ..... sid 3

Recension av Micro Technologys användarvänliga registerprogram Mt-Base ..... sid 5

Datadikt av anonym röst: Binärt Pater noster .... sid 7

En massa roliga adresser och rutiner i Msx-romet listas och förklaras grundligt under vinjetten "Msx-romet" .. sid 8

Recension av Aackosofts pratande flygspel Jet-Bomber ..... sid 10

Maskinkodsskolan går vidare. I det här numret förklarar Daniel skift och rotationer och några andra sorters bitmanipulationer ..... sid 11

Anders tar upp problemen med att spara de olika skärmarna på Msx datorn eftersom denna saknar ",s" efter CSAVE. Där finns också en del roliga rutiner att skriva av och lägga in i sina egna program ..... sid 17

Vinnarna i Dan Haggrens Megalone-tävling presenteras på ..... sid 19

Baksida ..... sid 20

Tidningen Msx & Svi Hacker ges ut 6 gånger per år och distribueras endast till medlemmarna i datorklubben SSMK (Svenska Spectravideo och Msx Klubben).

Ansvarig utgivare är Ingvar Ygeman.

Redaktörer är Anders Ygeman och Daniel Bergström.

Klubbens ekonomiske ansvarige är Håkan Bergström.

Annonser tas in i tidningen enligt följande priser:

800 kr per hel sida.

400 kr per halv sida.

200 kr per fjärdedels sida.

För annonser av annat format än de i Prislistan angivna tages 3 kr per kvadratcentimeter.

För till tidningen skickat ej beställt matriel ansvaras ej.

Artiklar eller recensioner inskickade av medlemmarna mottages med tacksamhet och premieras enligt nedstående:

Minst ett hundra svenska kronor per varje skriven A4.

Matriel avsett för införing i tidningen eller allmän klubbkommunikation + frågor skickas till följande adress:

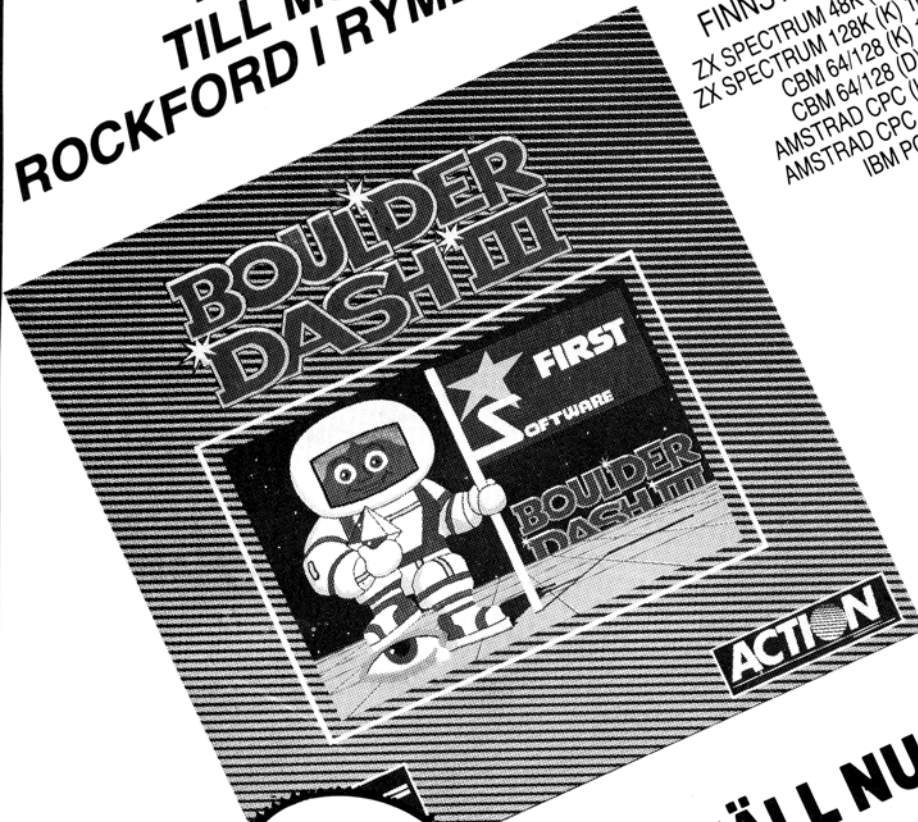
SSMK  
Box 5150  
162 05 Vällingby

Per telefon så når ni oss på följande telefonnummer:

08/47 31 21 (Till Anders, i första hand frågor som rör tidningen och Msx).

08/89 61 64 (Till Daniel, i första hand frågor rörande klubbens förvaltning och Svi).

**NYTT  
TILL MSX  
ROCKFORD I RYMDEN!!**



**FINNS ÄVEN TILL**  
ZX SPECTRUM 48K (K) 139:-  
ZX SPECTRUM 128K (K) 169:-  
CBM 64/128 (K) 139:-  
CBM 64/128 (D) 179:-  
AMSTRAD CPC (K) 149:-  
AMSTRAD CPC (D) 179:-  
IBM PC (D) 495:-

**NU  
TILL MSX  
169:-**

**RING & BESTÄLL NU!!**  
**08-776 11 12**  
PORTO INGÅR,  
INGET EXTRA TILLKOMMER!!

**BÄSTA BETYGI  
MSX HACKER**

**MOLTECH SOFTWARE**

Sveriges ledande postorderdistributör av hemdatorprogram!

HÄSTSKOVÄGEN 31 • 136 71 HANDEN

## Registerprogram för nybörjare?

Program: Mt-base  
Tillverkare: Micro Technology  
Återförsäljare: Eos computer  
Typ: Cartridge  
Pris: 369 kr  
Recenserad av: Anders Ygeman

Mt-base är ett mycket lätthanterligt och logiskt uppbyggt registerprogram som en femåring i princip ska kunna använda. Programmet är uppbyggt som en gammaldags registerlåda, där varje post i registret har ett kort. Formatet för kortet är det första man bestämmer. Detta görs på en speciell meny där man för varje underpost i programmet kan bestämma hur många bokstäver respektive siffror varje underpost ska ha.

Efter att ha gjort det här och satt upp ramarna för sitt register, så är det bara att börja skriva in sina poster. När man sen ska ta fram ett kort så finns det flera olika möjligheter för detta. En av dessa är att man skriver in huvudrubriken på den underpost man har vilket har den effekten att eftersökta kort kommer upp på skärmen.

Om det däremot finns fler kort på samma huvudrubrik, som om du till exempel har ett skivregister och har flera skivor med Kraftwerk (vem har inte det?), så får du först upp det kortet som står först i bokstavsordning och sedan en text som frågar om du vill fortsätta söka. Du kan också gå igenom hela registret post för post eller få fram vilka underposter som det finns fler av och en massa andra avancerade sökfunktioner.

Med programmet medföljer en 200-sidig manual som på ett genialiskt sätt är anpassat till programmet på det viset att det hela tiden finns två siffror uppe i det högra hörnet. Dessa siffror motsvarar den sida som det du håller på med står förklarad på i användarmanualen.

På detta sätt har tillverkaren på ett mycket bra sätt sett till att det i stort sett aldrig går att göra fel eller komma bort sig i programmet, om man inte har slarvat bort manualen förstås. Det här är faktiskt ett ganska bra kopieringsskydd eftersom det inte är så många som tar med sig en 200-sidig manual för att kopiera.

En annan av fördelarna med det här programmet som kanske kan förklara det höga priset är att hela programmet ligger på cartridge. Fördelarna och nackdelarna med cartridge är ju något som har diskuterats livligt på senaste tiden. Jag för min del tycker att fördelarna med cartridge klart överväger mot nackdelarna.

### COPY 328

COPY 328 är ett kopieringsprogram som kopierar alla program på kassetband till SV-328 eller SV-318+64k. Du kan nu snabbt och enkelt kopiera vilka program du vill!

COPY 328 har många fördelar:

- + Autostart
- + Helautomatiskt, du behöver inte hålla reda på programnamn, adresser etc.
- + Obegränsat antal kopior av en inladdning
- + 100% maskinkod
- + Kopierar upp till 50 olika program samtidigt !!!
- + Specialmode för att kopiera Zaxxon, Wargames, De Cathlon, Donkey Kong mfl
- + Kopierar alla hittills kända originalprogram
- + Priset endast 49 kr inkl. manual

Sätt in pengarna på PG 418030-3 eller ring 0158-60530 Andreas Björnstad  
OBS! COPY 328 är kopieringsskyddat Copyright 1986 by ABD

Den största nackdelen med en cartridge är som sagt priset, men å andra sidan så tar det en hel sekund att ladda programmet (det vill säga den tid det tar att sticka ner cartridge i cartridgeporten). Den fördelen kanske inte verkar så övervägande för oss som har diskettstation men för de som har bandstation måste det vara en fröjd att få slippa de minuter det trots allt tar att ladda in ett program varje gång man ska kolla vilka skivor man har.

En annan fördel med cartridge är att programmet inte tar upp plats på någon diskett och att man därför inte behöver byta diskett varje gång man ska titta i en fil. Detta gör att man har en diskett mindre att leta på när man vill ha tag på sina kära Kraftwerkskivor.

Hela Mt-base är förberett för både kasset och diskett vilket är ovärderligt för oss diskettägare som har tröttnat på att den eviga standardtexten "Press play on tape" kommer upp på skärmen. Över huvudtaget är det här programmet ett av de absolut mest genomarbetade programmen jag sett.

Tillverkaren, Micro technology, verkar att ha tänkt på allt. Programmet är kompatibelt med allt och är därför ytterst användbart.

Alla printerfunktioner på det här programmet är värt ett helt kapitel i en bok men jag nöjer mig med en liten översikt. Du kan nämligen i stort sett skriva ut vad du vill, hur du vill. De är till och med anpassade till Micro technologys andra produkt, Mt-text, så att du kan skriva ett brev och sen var som helst i texten lägga in poster från Mt-base (precis som all outhärdlig reklam ni får i brevlådan).

Detta gör att du kan skriva ett brev till dina polare genom adressregistret och sedan låta brevet innehålla en massa andra uppgifter från andra register som bordsplacering och liknande. Kanske inte den mest användbara funktionen men ändå ett plus i kanten. Tankarna flyger direkt iväg mot betydligt dyrare program som till exempel Wordstar (som det här är skrivet med) som tillsammans med Mailmerge kan åstadkomma precis samma sak fast för ett betydligt högre pris.

Eftersom det här programmet bara har stora fördelar förutom priset, gör att jag lungt kan ge det vårt högsta betyg, en femma. Men för en produkt av den här klassen och dessutom på cartridge så kan det lite höga priset vara befogat. Jag kan verkligen rekommendera det här programmet till alla som behöver ett mycket kompetent registerprogram till ett överkomligt pris.

---

## SPEL TILL SPECTRAVIDEO!

### Zuper Znake!

Hinner du äta rent på skärmen innan tiden är slut? För SV 318/328

Pris inkl. manual

60:-

### Frosty (the Snowman)

Du ska hjälpa snögubben Frosty igenom olika farofyllda rum. SV 318/328

Pris inkl. manual

60:-

### Greve von Fikonstråle

Ta dig ut ur slottet innan midnatt, då greven förvandlas till en varulv. Adventurespel på svenska för SV 328. Pris

70:-

Beställ genom att sätta in pengarna på postgiro nr. 420 09 43-1 Arne Karlsson, eller genom att ringa 0753-70782 Peter Wastholm. Observera att om du beställer per telefon tillkommer postförskottsavgift!

# BINÄRT PATER NOSTER

"Fader dator som är i centralen  
helgad vare din skärm  
tillkomme ditt tangentbord  
ske din vilja så som i minnet  
så och på printern  
vår dagliga lista giv oss idag  
och förlåt oss våra misstag  
trots att vi icke förlåta dem  
som har programmerat fel  
låt om oss icke ha långa svarstider  
och fräls oss ifrån dumpar  
ty företaget är ditt  
och makten och personalen  
i evighet  
enter."

Anonym digital röst

## > ASSEMBLER <

Äntligen HÄR!!!  
Den "riktiga" assemblern,  
Zuper-80 Assembler (Z80A).  
Systemet består av 2 st  
program: 1 - Integrerad Editor  
med bla fullskärmsredigering  
och scroll funktion (scroll  
rutinen givetvis i maskinkod)  
både uppåt och nedåt. 2  
-Monitor med perfekt editte-  
ring och utrustad med massor  
av finnesser t.ex Move,  
Search, ASCII, Reg (man får  
se register innehållet), Di  
(disassemblerar, gör det till  
ett rent nöje att lista  
maskinkodsprog. Den översätt-  
er M-koderna till assembler.)  
och Relativ räknare.  
Du som ännu inte behärskar  
M-kod men tänker lära dig  
kommer att ha en väldig hjälp  
av en assembler.  
Men priset då!!!. Jodå, jag  
kan faktiskt överraska med  
ett otroligt lågt pris ENDAST  
129 kr. inkl. manual

## > ANDRA PROGRAM <

1. CRAZY TEETH. Det häftiga och supersnabba spelet som alla pratar om. Helt i M-KOD. Massor av ljudeff. Helt 17 nivåer. Med en 8-i-top-lista för de bästa resultaten. Erkänt bra av bla Jon Hätte. För 328/318 utan extra vinne. Pris ENDAST 70 kr.
  2. SCROLL M-KOD. Scroller (förskjuter) valfri area i screen 1 upp, höger, vänster eller ner. P 55 kr.
  3. FYRA I RAD. Typ luffarschack 9 olika svårigheter. P 30 kr
  4. PAXMAN. otroligt snabbt för att vara i basic. Pris 30 kr.
  5. MONZA. Bilspel. Pris 30 kr.
  6. JUMPMAN. En av favoriterna. Pris 30 kr
  7. BREAK Bra grafik och klart svårt. P 30 kr.
  8. Mind area. Du har till uppgift att desarmera gömda minor. Till din hjälp har du en vindetektor. Mycket spännande. P 25 kr
- ERBJUDANDE!!! Basicprogrammen (3-8) kan köpas tillsammans för ENDAST 95 kr. SÄTT in peng. på PG. 485 60 20-5 el. RING 0758/17987 DAN HAGGREN.  
SANTLIGA priser inkl. moms och porto.

# MSX-ROMET

Hej igen !

Nu är det dags för en massa roliga (?) adresser i Msxromet. Det här numrets adresser blir lite mer ostrukturerade än de i förra numret beroende på att jag då enbart skrev om videochipet. Det blir väl en del om videochipet den här gången också, men mest ett hopplöck av olika mer eller mindre användbara adresser i biosen till Msx.

Det första jag ska ta upp i det här avsnittet är hur man sätter upp de olika skärmarna på Msx. Detta gör du genom att kalla på följande rutiner:

0060...INITXT  
Initiera textskärmen med storleken 40 \* 24 tecken. (Basicens screen 0.)

006F...INIT32  
Initierar text & grafiskskärm med storleken 32 \* 24 tecken. (Basicens screen 1.)

0072...INIGRP  
Initierar grafiskskärmen med upplösningen 196 \* 256 pixels. Rutinen rensar hela skärmen, även alla spritedata. Den sätter också tillbaka alla basevärden genom standardrutinen SETGRP (se nedan). Motsvarar basicens screen 2.

007E...SETGRP  
Sätter alla vdp-värdena till normala värden för screen 2. Den här rutinen anropas av rutinen INIGRP.

0075...INIMLT  
Sätter upp "multi color mode" vilket betyder att skärmen får en mycket grov upplösning och att man sätter en pixel enbart genom att ange vilken färg denna ska ha. Använder rutinen SETGRP för att få alla vdp-värden normala (Basicens screen 3).

0081...SETMLT

Används av rutinen INIMLT för att sätta tillbaka vdp-värdena till normala för screen 1.

Nu vet ni hur man sätter upp de olika skärmarna på Msx. Nu kommer det lite andra roliga rutiner.

06A8...CLRSPR

Rutin för att rensa alla spritar i minnet. Hela 2 kilobytes spritedata rensas genom rutinen FILLVRM (se förra numret). Den här rutinen fyller också alla spritenummer med serien 0, 1, 2, 3...32 för 8 \* 8 spritar och med serien 0, 4, 8, 16...124 för 16 \* 16 spritar, beroende på om storleksbiten i vdp-register 1 är satt eller inte. Till slut fyller den också i färgbyten för varje sprite med värdet i FORCLR. Denna är från början initialiserad till 15 (vit).

06E4...CALPAT

Rutin som räknar ut adressen för starten för utseendet på en speciell sprite. Spritenumret ska ligga i ackumulatören och adressen kommer i HL. Register som påverkas är AF, DE och HL. Rutinen tar hänsyn till om storleksbiten i vdp register 1 är satt eller inte.

0704...GSPSIZ

Rutin som tar reda på hur många bytes varje sprite tar upp, genom att helt enkelt kolla om storleksbiten i vdp register 1 är satt. Resultatet kommer i ackumulatören och är antingen 1 för 8 \* 8 spritar eller 32 för 16 \* 16 spritar.



## 071E...Namnlös

Den här rutinen kopierar hela 2 kilobytes teckenutseende till videominnets start för bokstävernas utseende, vilken tas ifrån CGPBAS. (Den här rutinen kan vara bra att ha om man har råkat fylla hela området för bokstavsutseendet med nollor eller något annat smart...)

## 0777...Namnlös

Ännu en namnlös rutin men en mycket nyttig sådan. Den här rutinen rensar nämligen skärmen i alla olika vdp skärmar. I 40 \* 24 och 32 \* 24 textskärm så fylls hela skärminnet med chr\$(32) (mellanslag), och sedan sätts markören till positionen längst upp till höger och funktionstangentsvisningen återställs via rutinen FUNKSB.

I grafikskärmen sätts först ramfärgen via vdp register 7 (0832). Sedan fylls hela färgtabellen med bakgrundsfärgen tagen från rutinen BAKCLR för både 0 och 1 pixels. Till slut nollställs bokstävernas utseende.

I "multi color mode", d.v.s screen 3, sätts först ramfärgen genom vdp register 7 (0832), sedan fylls alla pixels med bakgrundsfärgen tagen från rutinen BAKCLR.

## 0848...CLS

Rutin som rensar skärmen i alla olika skärmar. Gör ingenting annat än att kalla rutinen på 0777.

## 084F...CHGMOD

Rutin för uppsättning av olika skärmar. Numret för den skärm som ska sättas (0, 1, 2, 3) ska finnas i acumulatorn.

## 085D...LPTOUT

Rutin för att skicka ett tecken till en skrivare genom centronicsinterfacet på datorn. Bokstavens ascii nummer ska finnas i acumulatorn när rutinen anropas. Printerns staus kollar kontinuerligt genom rutinen LPTSTT. Tänk på att ctrl/stop

tangenten kollas genom rutinen BREAKX. Om ctrl/stop är nedtryckt sänds chr\$(13) (vagn retur) till printern och carryflaggan sätts.

## 0884...LPTSTT

Rutin som känner av om printern för närvarande är upptagen. Det enda den gör är att läsa av första biten på port 90 och kolla om den är satt eller inte. Om printern är upptagen blir acumulatorn noll och flagga Z blir satt till noll.

## 088E...POSIT

Placerar markören var som helst på skärmen. Kolumnen ska finnas i H och raden i L. Härifrån läser också rutinen OUTDO positionen.

## 08BC...CHPUT

Rutin för att skriva ett tecken till screen 0 eller 1. Ascii värdet för tecknet som ska skrivas ut ska finnas i A. Rutinen kollar om någon av grafikmodarna är satta genom SCRMOD, och om så är fallet avslutas rutinen utan någon verkan och hoppar tillbaka.

## 0F3D...CHGCAP

Rutin för att släcka och tända capslocklampan. Acumulatorn ska vara 0 om lampan ska tändas och någonting annat om den ska släckas. Den här rutinen har diskuterats om i tidningen otaliga gånger men man kan ju alltid använda den om man ska leka disco...

## 0F7A...CHGSND

Rutin för på- eller avsättning av tangentbordsklicket. Acumulatorn ska vara 0 om klicket ska sättas på och någonting annat om det ska stängas av.

## 1102...WRTSPG

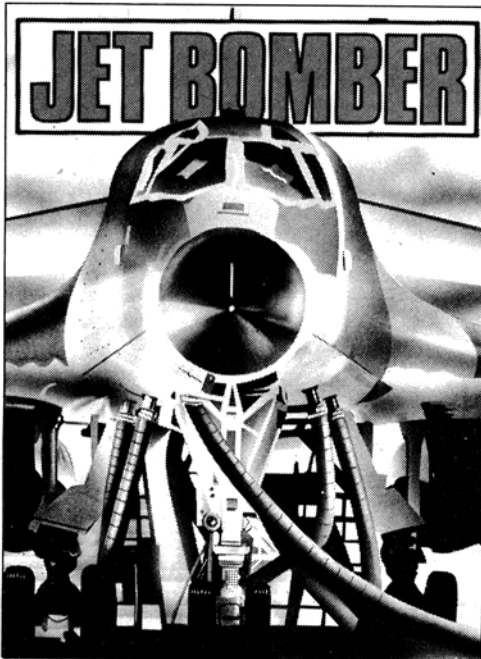
Rutin som skriver en byte till vilket som helst av de sexton olika psg-registren. Acumulatorn ska vara laddad med registret och register E med byten som ska skrivas till registret.

Det här var allt för den här gången. Ha det så bra.  
Text av Anders Ygeman

## ZAXXON ??

Program: Jet bomber  
Tillverkare: Aackosoft  
Pris: 139 kr  
Recenserad av: Anders Ygeman

Av fodralets bild och text leds man direkt till tron att det här är någon slags blandning mellan flyg-simulator och actionspel, men jag kan säga att det här bara är en billig "Zaxxon"-kopia.



Jet bombers handling: En revolution i ett land i mellanöstern. Hassan den grymme, ledaren för revolutionen, startar en stor terrorkampanj mot det multinationella oljebolaget Oxnan oil (läs Exxon oil = svenska Esso). Oxnan oil känner sig tvingade att ta situationen i egna händer och hyr en jetbombare för att nå en lösning:

En pilot med otrolig skicklighet och nerver av stål ska flyga in i landet förbi rebellerna och armen, och sedan förstöra rebellbasen. Det här är alltså vad det står på asken att man gör, men i själva verket är det som sagt en alldaglig Zaxxon kopia.

Spelet består av fem olika faser varav alla är ungefär lika tråkiga. Fas 1 består av första scenen på Zaxxon varken mer eller mindre. Fas 2 ungefär samma som fas 1 fast du måste flyga betydligt högre och bomba raketer som flyger under dig.

Fas 3 är ungefär samma sak som "rymden" på Zaxxon - vilket betyder att det kommer en massa flygplan (ser ut som rymdskepp) och skjuter på dig och du måste vara på samma höjd som dom för att kunna skjuta ner dom. Fas 4 är samma som tvåan fast med ett annat namn. Fas 5 förstör rebellbasen innan den sprängs och förstör dig.

Om de som gjorde det här spelet hade tänkt mer på att göra en bra handling och ett bra spel istället för att göra en kopia av Zaxxon som är så lik som möjligt i alla spelsituationer och så olik som möjligt i alla andra avseenden så skulle spelet med stor sannolikhet blivit bättre.

Trots allt har spelet blivit ganska spelvärt eftersom Zaxxon (originalet) egentligen är ganska roligt. En av de saker som talar till spelets fördel är att det innehåller syntetiskt tal (samplat) som faktist håller ganska hög kvalitet och där man hör allt som det är meningen att man ska höra att den säger.

En annan detalj som talar till spelets fördel är att det har en mycket fin laddningsbild som kommer upp efter varje avslutat spel, det ska också tilläggas att musiken håller en för msx mycket bra standard. Trots dessa positiva sidor ger jag inte spelet mer än en trea i betyg. Mest på grund av det faktum att idén är mycket gammal och mycket stulen.

# MASKINKODSSKOLAN

Denna gång, som alla andra får jag börja med en liten rättning av mig själv. Att ni inte fick mitt programexempel i förra numret att fungera berodde på att jag helt enkelt hade skrivit fel i Basicprogrammet. I DATA-satsen på rad 60 ska det stå ...40, 7, 205... istället för ...40, 5, 205... Med denna lilla ändring så ska allt fungera.

Nu över till de instruktioner som jag inte har presenterat än:

## Grupp 5. Skift och rotation.

Den här gruppen av instruktioner brukar vara de som är svårast att lära sig utantill och tar längst tid, så om ni inte förstår allt med en gång, läs igenom det en eller ett par gånger till så förstår ni nog.

En byte är uppbyggd av åtta stycken nollor och ettor som kallas bitar och vilka är numrerade 0-7. Dessa har värdena:

0 - 1	=2 <sup>0</sup>
1 - 2	=2 <sup>1</sup>
2 - 4	=2 <sup>2</sup>
3 - 8	=2 <sup>3</sup>
4 - 16	=2 <sup>4</sup>
5 - 32	=2 <sup>5</sup>
6 - 64	=2 <sup>6</sup>
7 - 128	=2 <sup>7</sup>

Detta medför att en byte kan ha värdet 0-255 (1+2+4+8+16+32+64+128). Med skift och rotation kan man flytta de åtta bitarna ett steg åt höger eller vänster. Detta gör att bytens värde i praktiken multipliceras eller divideras med 2. Undantag finns dock, vilket jag kommer att beskriva senare i artikeln.

Med ett skift flyttas bitarna åt höger eller åt vänster och den bit som blir "utan" får värdet 0. Detta gäller med ett enda undantag, nämligen: **aritmetiskt högerskift, SRA.**

Detta beror på att negativa tal indikeras med en etta i den 7:e biten. Vid ett skift måste ju talet förbli negativt så om man skulle skifta åt höger och således lägga in en nolla där skulle talet bli positivt istället för negativt. För att registervärdet ska kunna förbli negativt om det var det innan operationen får den biten behålla det värde den hade från början.

Med rotation får den sista biten värdet på den biten som åkte ut på andra sidan alternativt värdet på Carryflaggan. Nedanstående figur visar alla möjligheter utom SLS vilket är ett specialfall. Den vanliga instruktionen SLA gör ju så att bit 0 får värdet noll, men SLS ger den biten värdet ett istället. SLA motsvarar ungefär en multiplikation med två, medan SLS adderar ett till detta resultat. Om man t.ex. hade 25 i A, så skulle det bli 50 med SLA och 51 med SLS.

Den här instruktionen fungerar dessutom inte på alla modeller av processorn Z80. Den finns därför vanligen inte på de assemblys som säljs i handeln. Alla instruktioner, som jag skrev i förra numret, representerades av ett till fyra tal mellan 0 och 255. För SLS gäller (hexadecimalt):

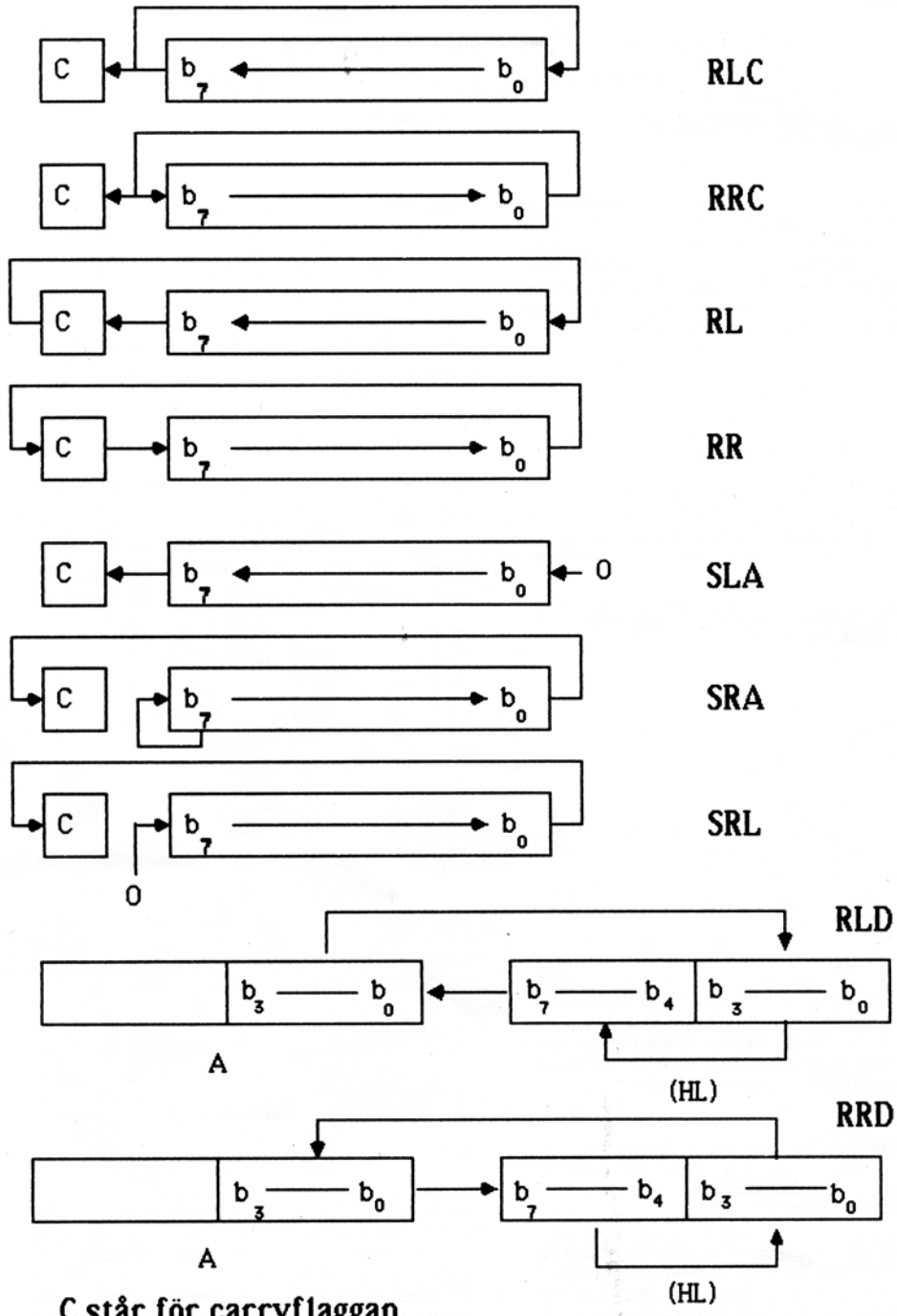
SLS A	CB 37
SLS B	CB 30
SLS C	CB 31
SLS D	CB 32
SLS E	CB 33
SLS H	CB 34
SLS L	CB 35
SLS (HL)	CB 36
SLS (IX+d)	DD CB d 36
SLS (IY+d)	FD CB d 36

Men kom ihåg: **De fungerar inte på alla processorer!**

Till alla instruktioner kan man använda operanderna A, B, C, D, E, H, L, (HL), (IX+d) och (IY+d). Att de tre sista står inom parentes betyder som vanligt att de är en adress i minnet.

Sen är det så finurligt ordnat, att om man skall rotera A åt något håll, kan man, istället för t.ex. RR A skriva: RRA som en enda instruktion istället för en instruktion med en operand. Detta sätt att skriva på gör att det tar upp hälften så stor plats i minnet och går dubbelt så fort.

Det finns ett annat sätt att spara tal på, BCD-kod. Då sparar man 2 siffror i varje byte. En siffra tar då upp 4 bitar, vilket har både fördelar och nackdelar. Den stora fördelen är att det är lättare att använda om man ska skriva ut talet decimalt, eftersom man annars måste översätta till det decimala talsystemet från det hexadecimala vilket kan vara ganska jobbigt. Men, eftersom varje nibble (ett kortare namn för 4 bitar) kan ha ett värde på upp till 15, får man problem vid



C står för carryflaggan.

additioner och subtraktioner. Detta beror på att varje nibble bara får ha värdena 0-9. Lyckligtvis är det så att det finns en liten "räkningsinstruktion" för detta. Den heter DAA och ändrar talet i A beroende på föregående instruktion. En annan nackdel med BCD-kod är att man bara kan ha talen 0-99 i en byte, mot 0-255 på vanligt sätt, vilket gör att det tar större plats.

Och, som ni säkert redan har listat ut, eftersom jag tog upp det här nu, finns det två roteringsinstruktioner specialgjorda för BCD-kod. De heter RRD och RLD, vilket står för rotera höger decimalt resp. vänster. Dessa är en rotation mellan talet på adressen som HL pekar på och de lägsta fyra bitarna i A. De roterar fyra bitar i talet till skillnad från de andra som bara roterar eller skiftar en bit.

Det finns två sätt till att manipulera med bitar på. Det ena är med instruktionerna AND, OR och XOR, vilka jag kommer att beskriva härnäst, samt med SET, RES och BIT. Dessa kommer jag att beskriva efteråt.

Alla tre instruktionerna AND, OR och XOR, har alltid A som ena operand. Instruktionen AND arbetar så att om samma bit i de två operanderna är 1, får den biten i A också värdet 1. Sanningstabellen ser ut så här:

AND	0	1
0	0	0
1	0	1

Antag att vi vill nollställa de tre bitarna längst till höger i ett register. Detta görs med följande program:

```
LD A,r Ladda A med värdet i något reg.
AND 11111000B 11111000B är mask.
```

Om tex. A fick värdet 00110011B från början, skulle resultatet bli 00110000B. Anledningen till att det står ett B efter betyder att det är i binär form.

AND kan man med fördel använda för att nollställa vissa bitar i A.

OR fungerar på samma sätt fast efter denna sanningstabell:

OR	0	1
0	0	1
1	1	1

OR innebär alltså att om det finns en etta i någon av operanderna blir resultatet 1.

OR använder man ofta då man vill sätta någon eller några bitar i A.

Slutligen har vi då XOR, vilket står för eXclusive OR. Här blir resultatet 1 om en, och endast en, av operanderna är 1. Om båda är 0 eller 1 blir resultatet således 0. Sanningstabellen ser alltså ut så här:

XOR	0	1
0	0	1
1	1	0

XOR kan också användas för att nollställa A. Detta går dubbelt så snabbt och tar upp hälften så mycket plats som det vanliga:

```
LD A,0
```

Alla tre AND, OR och XOR nollställer carryflaggan om man nu skulle vilja det. Annars kan man ändra tillbaka värdet med CCF (komplementera carryflaggan, C=1-C) och SCF (sätt carryflaggan till ett).

De tre instruktionerna SET, RES och BIT är mycket bra att ha. De används nämligen för att sätta, SET, nollställa, RES, och testa bitar i något register eller minnesadress, BIT. Alla tre arbetar med samma operander som skift och rotationsinstruktionerna, A, B, C, D, E, H, L, (HL), (IX+d) och (IY+d). Alla åtta bitarna kan sättas och nollställas. Med instruktionen BIT sätts carryflaggan om biten var 1, annars nollställs den.

För A finns det två små instruktioner till som kan vara användbara, CPL och NEG. CPL gör alla bitar i A till det de inte är, d.v.s. alla satta bitar nollställs och alla nollor blir ettor. NEG gör samma sak men adderar ett till resultatet.

De som vill använda grafik och ljud i maskinkod behöver inte vänta längre. Allt görs med de små behändiga instruktionerna IN och OUT. De används när man skall skriva text, rita grafik, göra spritar, flytta dessa, känna av om två spritar kolliderar, samt spela musik eller göra andra ljudeffekter. De kan också användas för att skriva till printern, om man inte anropar en liten rutin i BASIC-tolken för detta. Kort sagt så kan man säga att de används för att kommunicera med alla tillbehör till datorn.

I ett program står de så här:

```
IN  r,(C)
OUT (C),r      eller:
OUT (n),A
```

Den lilla bokstaven r kan här vara något av de 7 registren A, B, C, D, E, H och L. I den

andra OUT-varianten är n porten som avses. I de övriga fallen finns portnumret i register C.

Om man vill föra över hela block använder man instruktionerna IND, INDR, INI, INIR, OUTD, OTDR, OUTI eller OTIR. Alla använder HL som pekare till var data ska hämtas eller

lämnas, C innehåller portnumret och B används som räknare.

IND tar in ett tecken från port C, sparar det på den plats som HL pekar på, räknar ner HL, och räknar ner B. INDR repeterar detta till B är noll.

INI gör samma sak fast räknar upp HL istället. INIR repeterar.

OUTD tar data från adressen HL pekar på, skickar ut det på port C, och räknar ner HL. OTDR repeterar. OUTI ökar HL ett steg istället och OTIR repeterar OUTI tills B blir noll.

Om man nu vill skriva i videominnet (rita grafik eller flytta spritar t.ex.) gör man som följer:

Dela upp adressen i två delar, med hjälp av den hexadecimala representationen. Den får man genom att skriva:

```
PRINT HEX$(talet)
```

De två siffrorna (eller bokstäverna) längst till höger utgör den lägre delen. De två övriga utgör alltså den högre. De kan sedan omvandlas till normal decimal form igen genom att helt enkelt skriva:

```
PRINT &H talet
```

Den lägre delen ska sedan skickas ut på port 129 (81H) för Spectravideo, eller port 153 (99H) för Msx. Sedan skickar du ut den högre delen + 64 på samma port. Observera att detta måste göras från ett program, maskinkod eller Basic.

När nu detta är klart så är det dags att skicka ut data som ska bestå av tal mellan 0 och 255. Dessa ska skickas ut på port 128 (80H) för Svi eller 152 (98H) för Msx.

Om du ska skriva flera bytes efter varandra är det bara att skicka ut värdena ett efter ett. Adresspekaren ställer om sig självt till att peka på nästa adress. Detta gäller både om man skriver eller läser i minnet. Om man ska läsa tar man in värdet från port 132 (84H) för Svi respektive 153 (99H) för Msx.

När man sen vill spela musik, eller bara föra ett allmänt oväsen, (vilket dock inte brukar vara lika populärt), använder man sig av samma princip som med Basicens SOUND-kommando. Detta innebär, precis som med SOUND, att man tar reda på vilket av ljudchipets register man ska skriva till, och vilket värde som ska dit. Registernumret skickar man ut på port 136 (88H) resp. 160 (A0H). Sedan är det bara att skicka ut data på port 140 (8CH) resp. 161 (A1H). Vilka registernumrena är och vad de gör o.s.v står i din manual vid beskrivningen av kommandot SOUND.

För att uppehålla datorn en stund, om man bara spelar musik, finns det en liten behändig instruktion som kort och gott heter HALT. Den stoppar processorn arbete upp till en 50:e dels sekund! Detta hänger ihop med att det 50 gånger per sekund kommer en liten signal till processorn som gör att programmet hoppar till en subrutin på adress 38H. Den rutinen kollar om någon tangent har blivit nedtryckt bl. a. Instruktionen HALT stannar programmet och väntar på en sådan signal.

Men om man istället vill att datorn ska vänta en viss tid får man göra en normal liten loop av typen:

```
FOR I=1 TO 2000:NEXT I
```

I maskinkod ser ovanstående program ut så här: (Självklart kan man göra det på många andra sätt också, men jag brukar göra så här)

```

PUSH AF
PUSH BC
LD BC,0
IGEN: DEC BC
LD A,B
OR C
JR NZ,IGEN
POP BC
POP AF
RET

```

Först sparar man alltså de register som skall användas. Det är en bra regel som man ska försöka följa så ofta det går. Om ovanstående rutin t.ex. skulle anropas från ett större program kommer annars registervärdena gå förlorade. Det är **inte** så bra. Sedan laddas ett register med noll. Detta för att få en så lång fördröjning som möjligt. Registret miskas med ett (DEC BC) och den högre delen laddas till A. En logisk OR utförs sedan mellan A och den lägre delen av registret. Om nu någon bit i något register är satt, (registervärdet ej noll), så blir A inte heller noll. Inte förrän båda registerdelarna är noll, blir A noll och rutinen avslutas med att ta tillbaka registervärdena och hoppa tillbaka till där programmet var förut. Den här rutinen skapar en fördröjning på ca en halv sekund. Som jämförelse kan jag ju säga att ett liknande program i Basic (loop mellan noll och 65536) skulle ta ungefär en dryg minut på sig.

Om man laddar loopräknaren med ett tal under 32768 (8000H) kan man istället för raderna 5-6 skriva:

```
BIT 7,B
```

vilket tar mindre plats. Plats är nämligen det räddaste man ska vara om när det gäller programmering. I småprogram spelar det ofta inte någon större roll om en rutin tar upp 18 eller 19 bytes, men när man är uppe i ett par hundra sådana rutiner och det börjar bli trångt om utrymme i minnet kan det vara bra att kunna några "platsminskningsknep".

# MEGALONE

DET NYA  
SPELET!

100%  
M-KOD

- \* 1000 rum
- \* 82 olika föremål
- \* 31 olika fiendetyper
- \* över 100 ljudeffekter
- \* Patrullerande vakter
- \* 55 Kb maskinkodsprogram
- \* 10 våningar
- \* Arcade-Adventure (är både ett adventure och ett actionspel)
- \* Mängder med grafik
- \* 2 års utvecklingsarbete
- \* Ett paradys för upptäckare
- \* Det i särklass mest avancerade spelet för Spectra-video 32B/318+64Kb
- \* Med omfattande manual



Tejnr för frågor  
och beställningar  
Johan Axelman  
0758/15528  
Dan Haggren  
0758/17987

## STORY

En lycklig pojke öppnar dörren för sina kamrater på sitt 7-årskalas. Festen började bra men ju längre tiden led desto konstigare blev stämningen, för många såg Det, men sa inget. Gary kände att något var fel. På kvällen frågade han sina föräldrar. Paret Dovfisher svarade: "Du är trött. Vi måste till labbet. Sov nu!" När dörren stängdes grep den kusliga stämningen tag i Gary på allvar. Han drogs in i föräldrarnas mörka rum. Endast ett kallt månljus föll på det enorma bordet. Något drev honom mot det. Framme såg han en bok, "Electronic humans", på bordet. Han bläddrade mekaniskt. På sid 1000 läste han om människor som utsatts för vidriga experiment. De hade tvärt förvandlats till styrbar elektronik och använts av Megalomaniac för att förtrvcka världen.

På himlen skymtade Satelliten hastigt till. När Gary såg den stack det till i hjärtat. "Vad är det med mig", tänkte han. Han gick till spegeln. Nu förstod han. EXPERIMENTET! Hans yttre hade förändrats brutalt...

Detta är upptakten till Garys äventyr. Experimentet hade pga Garys starka person en fördröjd effekt på psyket. Han förstod att hans enda chans var att ta ett nytt namn, Megalone, och med hjälp av sitt utseende nästla sig in i Megalomaniacs satellit.

Ditt uppdrag börjar när Megalone just kommit in i satelliten. Det är på Dig det hänger om Megalone ska kunna rädda världen och undgå sitt ohyggliga öde.

Programmet kostar endast

# 139.-

inkl. fraktkostnader.  
Sätt in pengar på  
PG 485 60 20-5 så får  
du programmet omgående.  
Eller ring och beställ  
på postförskott (+6kr).

Vi vet hur det känns  
att vänta. Därför  
skickas prog. 24t:m  
efter beställningen.



# SCREENSAVE PÅ MSX

Någonting som har visat sig svårt för medlemmarna i SSMK när de har gått över från Spectravideo till någon av de olika Msx-datorerna har varit det faktum att det inte längre går att spara ner skärmbilder till band. I den här artikeln tar vi upp två olika sätt att göra detta. Dessa programexempel går att skriva på flera olika sätt och om du inte har sett något liknande programexempel förut så kommer du säkert att få se det i framtiden. Det går självklart bra att ändra rutinerna. Eftersom rutinerna i sig är helt utan originalitet och skulle kunna skrivas av vem som helst som är lite Basic-kunnig så är rutinerna helt fria att användas i vilka sammanhang som helst, t.ex om ni har gjort ett eget ritprogram så kan ni ta någon av rutinerna här och antingen skriva om dem så det passar era applikationer eller också bara "sno" dem rakt av.

Den här rutinen sparar ner de delarna av bildminnet som används för screen 0. Det kan vara halvsmart att tänka på att kalla på rutinen när den bild man ska spara finns på skärmen och inte massa annat krafs.

```
10 A%=BASE(0)
20 B%=BASE(2)
30 OPEN "BILD" FOR OUTPUT AS#1
40 FOR I%=A% TO A%+959
50 C%=VPEEK(I%)
60 PRINT#1,CHR$(C%)
70 NEXT I%
80 FOR I%=B% TO B%+2047
90 C%=VPEEK(I%)
100 PRINT #1,CHR$(C%)
110 NEXT I%
120 CLOSE#1
130 RETURN
```

De två första exemplen visar de två alternativen när man ska spara en skärm i Screen 0.

För att ladda in den tjugiga (?) bilden i minnet igen krävs en liten laddningsrutin som ser ut på följande vis:

```
10 A%=BASE(0)
20 B%=BASE(2)
30 OPEN "BILD" FOR INPUT AS#1
40 FOR I%=A% TO A%+959
50 A%=INPUT$(1,#1)
60 VPOKE I%,ASC(A%)
70 NEXT I%
80 FOR I%=B% TO B%+2047
90 A%=INPUT$(1,#1)
100 VPOKE I%,ASC(A%)
110 NEXT I%
120 CLOSE #1
130 RETURN
```

Som alla säkert förstår så kan det vara lämpligt att ändra radnumren på programmen eftersom dom annars har en tendens att radera ut varandra...

Det kan också vara halvsmart att lägga rutinerna som subrutiner någonstans långt bak i programmet eftersom dessa faktiskt är gjorda som subrutiner (observera RETURN).

Den andra metoden för att spara ner en bild är att föra över en del av minnet i vram till vanligt ram och sedan använda bsave för att spara ner det på band. Den här rutinen är också halvsmart att lägga som subrutin men det kan då också vara ganska bra att av förståliga skäl lägga rad tio som rad tio först i programmet.

```
10 CLEAR 200,54999
20 A=55000
30 FOR I%=BASE(0) TO BASE(0)+959
40 POKE A,VPEEK(I%)
50 A=A+1
60 NEXT I%
70 A=56000
80 FOR I%=BASE(2) TO BASE(2)+2047
90 POKE A,VPEEK(I%)
90 A=A+1
100 NEXT I%
110 BSAVE "BILD",55000,58100
120 RETURN
```

Efter att ha slösat band på en bild finns det alltid en risk att man vill ladda in bilden i minnet igen bara för att titta på den och se hur fin (?) den var. Om man vill göra det så kan man använda en rutin som ser ut ungefär så här: (Även här kan man tänka på att rad 10 ska först i programmet och att man nog även här ska ändra radnumren på programmen om man hade tänkt ha dem i samma program eftersom rader med samma radnummer har en viss tendens att radera ut varandra).

```
10 CLEAR 200,54999
20 SCREEN 0
30 BLOAD "BILD"
40 A=55000
60 FOR I%=BASE(0) TO BASE(0)+959
70 VPOKE I%,PEEK(J)
80 NEXT I%
90 A=56000
100 FOR I% = BASE(2) TO BASE(2)
+ 2047
110 VPOKE I%,PEEK(A)
120 NEXT I%
130 RETURN
```

Nu har säkert alla förstätt principen i det hela och försöker nu tänka ut hur man gör för att spara en bild från screen 2, så att får en fin och färggrann inladdningsbild eller vad man nu ska använda tekniken till. Vad man då ska tänka på är att cleara tillräckligt mycket minne så att man inte får med bara halva bilden eller något i den stilen. Det ska tilläggas att eftersom de tekniker som tillämpas i den här artikeln är skrivna i det gudomliga (ja, jag vet att det är fult att ljuga) språket Basic, så kan det gå minst sagt ganska långsamt att spara en bild. För att inte tala om vilken tid det skulle ta om man tillämpade den här tekniken när man skulle spara en högupplösningbild. Därför kan det finnas stora fördelar med att skriva ett sådant program i assembler som går betydligt snabbare än Basic. Hur man läser och skriver i bildminnet tar vi för övrigt upp i maskinkodsskolan i det här numret.

#### Angående minneskapslar till SV-318

Efter att ha läst Lars Hellvigs artikel om expanderings av SV-318 (nummer 2) skulle jag vilja göra ett tillägg. Jag har själv byggt om flera SV-318 och det har visat sig att det inte bara är kapslar från Texas Instrument som har ett annorlunda refreshsystem. Nedan följer en lista på några andra typer som är direkt olämpliga att använda:

MCM6664 MOTOROLA  
EP6665 THOMSON  
IMS2600 INMOS

HYB4164 SIEMENS  
F4164 FAIRCHILD  
KM4164A

Dessa typer fungerar bra:

2164 INTEL  
MK4564 MOSTEK  
HM4864 HITACHI  
MB8264 FUJITSU

MSM3664 OKI  
mPD4164 NEC  
TMM4164P TOSHIBA  
MSK4164ANP MITSUBISHI

Andreas Björnstad tel 0158-60530

VINNARNA I

**MEGALONE**

TÄVLINGEN:

ANDERS HERMANSSON

NÄSSJÖ

EDWIN JOHANSSON

HÖGANÄS

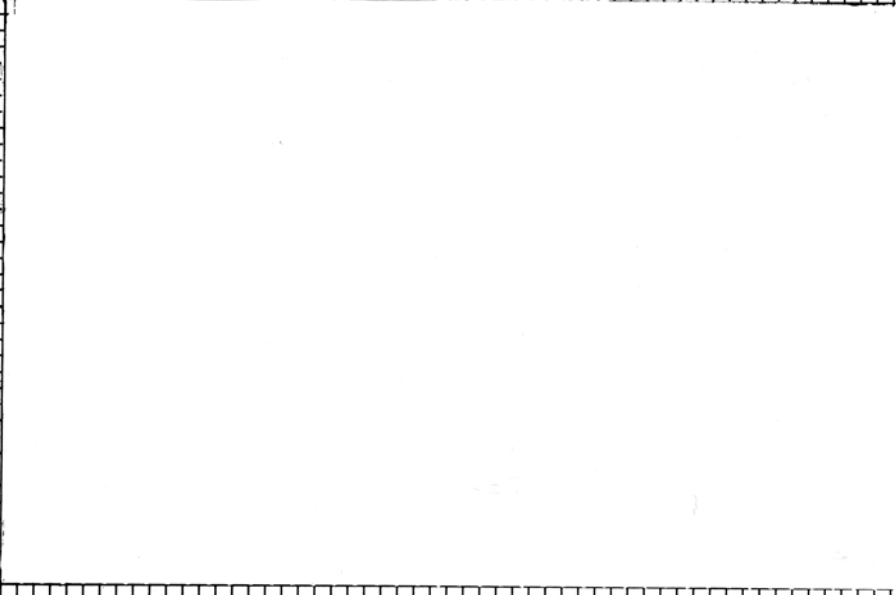
DESSA VINNARE HAR MOTTAGIT

500 KR VAR

557K

BOX 5150

162 05 VÄLLINGBY



FRÅSSBREV

