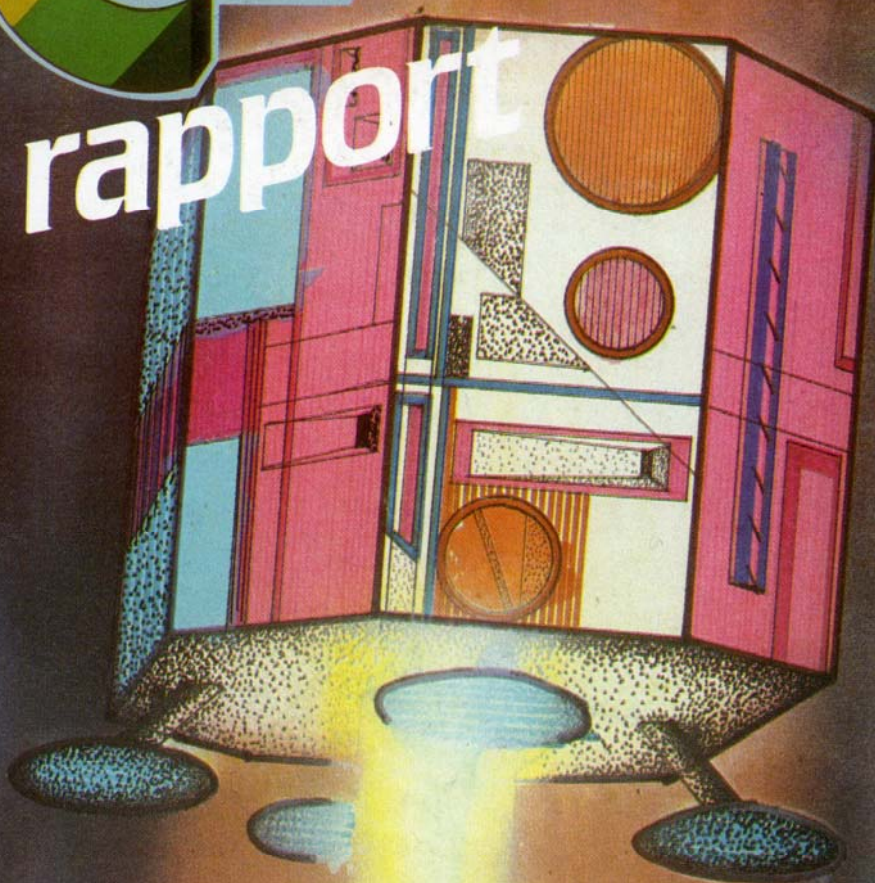


VIC

ÅRGÅNG 2 NR4
PRIS 15:- inkl moms

rapport



**NYBÖRJARSKOLA
SNYGGA SKÄRMTEXTER
DRIV LOKALKLUBBEN RÄTT
SPELTIPS för höga poäng
VIC T-SHIRTS!**

A large, stylized illustration of a person in a red helmet and goggles, wearing a microphone, is the central focus. Below and around this figure are smaller illustrations of athletes: a person on a windsurfer, a person on a bicycle, a person running, and a person in a red wetsuit surfing. All athletes are wearing Handic Micman radios. The background is a mix of blue and white, suggesting water and sky.

handic micman

Sportradion

För folk "in action" på Windsurfer, skidor, rullskridskor, cykel osv. som sportar för skoj skull eller deltar i tufft tävlande. Vilken feeling att kunna snacka med kompisen, tjejen eller hitta nya vänner samtidigt som man sportar. **handic micman fixar detta!**

I många sportgrenar använder elitidrottsmännen radiokommunikation vid träning för att nå bästa resultat vid tävling. Direktkontakt med tränaren, de du!

handic micman är en sådan kvalitetsprodukt som ger dig möjlighet till dessa nya dimensioner av kommunikation, "radiosamband".

handic micman är lätt och liten, bärs antingen i bältesclipset på baksidan apparaten eller i bärsele. Talar och lyssnar gör du med en lättvikts hörlur som är försedd med mikrofonbom. handic micman är som standard utrustad med joggingkanalen och kan försees med ytterligare en kanal t.ex. windsurfing-, cykel-, skid-, hästsportskanal. Det är lätt att byta sportkanal på handic micman. När du använder handic micman där det är vatten, behöver du handic micman våtpaket.

Upplev närheten till den som finns kilometer ifrån dig, vilken känsla, det måste upplevas!

**KOLLA IN ÅRETS
HÄFTIGASTE GREJ,
HANDIC MICMAN, HOS
DIN RADIOHANDLARE!**

handic

electronic ab

Box 1063, 436 00 Askim/Göteborg, Tel. 031/28 97 90
— ett företag i Datatronicgruppen —

Redaktören har ordet:

Jaha, så är vi ute i hetluften igen med ett nytt nummer av VIC-rapport. Förhoppningsvis har vi något färre missar i det här numret, som inte tillkommer under fullt lika stort jäkt som föregående, även om redaktörens förestående Englandresa förrycker produktionen något. De fel vi gjorde i förra numret rättas här i detta nummer och så lär vi väl få fortsätta om vi inte kan göra en absolut felfri upplaga.

Den kända amerikanska tidningen Compute! har en spalt de kallar Capute!, där de gör rättelser som läsarna eller författarna upptäcker. Vi kanske skulle ha en sådan stående spalt även i VIC-rapport? Låt oss utlysa en tävling om bästa namnet på en sådan spalt, vinnaren får en årsprenumeration på VIC-rapport!

Sätt igång din namngenerator bara, kanske du vinner!

Nya program och nya spel strömmar till, både för VIC-20 och 64:an. Några av dem presenteras på nyhetsidan. Ett envist rykte förtäljer att COMAL lär komma någon gång i vinter till 64:an och då! Jag har fått en förhandstitt på handboken till 64:ans COMAL, som har versionsnummer 2,0 och den är kraftfullare än någonsin. I nästa nummer hoppas jag kunna presentera den mjukladdade versionen, förhandlingar om vem som skall distribuera den lär pågå.

Europas största datormässa skall vi besöka i England. Vi, det är PIC-VIC-klubben i Göteborg med nytilkomne medlemmen ÅF. Utförlig berättelse kommer i nästa nummer av VIC-rapport.

Försäljningen av VIC bara ökar och ökar. Särskilt populär är 64:an, som nu också börjar få annat än spelprogram att jobba med. Flera bra affärssystem har kommit fram och fler är på gång. Nämnas speciellt bör kanske den utförliga Calc-Result, som gör 64:an till den verkliga affärsmannadatorn! (Calc-Result finns både i en enk-

Innehållsförteckning

Nybörjarskolan	2
Det kom ett brev (eller hur man håller liv i en datorklubb)	5
Hoppsan!	6
Kapten Storöra	8
Tidsinställd bomb i listig labyrint	11
Bokrecension	12
Vi lär oss skriva strukturerade BASICprogram	13
Video-chippet i VIC 64	14
Assemblerskolan – del 2	24
VICTiga rättelser	26, 43
Reläsladd gör presentationssystem av din VIC-20	28
Textbehandling U P A	30
Markörposition på VIC-20/64	31
Hur man gör snygga skärmtexter	32
Fotografera dina VIC-resultat!	34
Att använda joystick	35
Frågor & svar	36
Spelnytt	38
COMAL-skolan	40
Gratisannonser	42
Opus 2 Software's program till VIC-datorer	43
Fel i användarmanualen till VIC 64	43

<i>Ansvarig utgivare:</i>	Peter Elmlund	Jörgen Stiernborg
<i>Redaktör:</i>	Åke Fredriksson	Peter Tegelberg
<i>Övr. medarbetare:</i>	Lars-Erik Bengtsson	
	Martin Fredriksson	<i>Annonser:</i> Telefon 031-87 70 84
	Åke Hedman	<i>Tryck:</i> Ölandstryckarna AB, Borgholm
		<i>Layout:</i> Ottoson's Reklam

Redaktionens adress är:

VIC-rapport, Gustavsbergsgat. 8, 431 37 MÖLNDAL, Tel. 031-27 62 75

Annonssorder och annonsmaterial (heloriginal eller negativ film) enligt överenskommelse under 1983.

Tryckförfarande: Offset

Upplaga 30.000 ex.

Nästa nummer av VIC-rapport är ett dubbelnummer och utkommer den 7 dec 1983. Manusstopp är den 29/10.

lare och i den utförliga versionen till 64:an.) Vi väntar nu bara på att det aviserade modemmet skall levereras så att 64:an verkligen får visa vad den duger till som terminaldator. Jag har hört talas om att det finns dom som kör en 64 mot en större dator, så här är saker på gång!

Vi hörs igen om en månad!

Åke Fredriksson



Nybörjar skolan

WARNING!

Denna artikelserie är skriven för absoluta nybörjare i datorvärlden och för VIC-datoranvändare enbart. Om du är en van datoranvändare eller programmerare kan läsning av artiklarna orsaka många helt oförutsedda reaktioner, som vi inte kan ta något som helst ansvar för.

WARNING!

Du har alltså skaffat dig en dator. Din VIC har stora möjligheter att ge dig både nöje och hjälp, du kan köra spelprogram på den eller du kan använda den för att hålla kontroll på ditt checkkonto eller dina hushållsutgifter. Du kan använda den för att styra en projektor och en bandspelare för automatisk bildvisning, du kan utnyttja maskinen för att nöta in glosor på ett främmande språk eller träna matematik med. Du kan skapa grafiska bilder på din bildskärm och du kan sätta ljud (musik eller oväsen) till programmen du så småningom skriver.

I denna nybörjarserie skall vi i lugn och ro gå igenom alla (nästan) VICens olika möjligheter – serien är avsedd att kunna läsas av både barn och vuxna. Du jobbar igenom exemplen i din egen takt, här visar vi att det verkligen är du som håller i tyglarna! (det gäller ju all verksamhet med datorer, det är du som styr och datorn som är styrd!).

Sätt dig tillrätta för att börja ditt äventyr tillsammans med VIC. Om du känner för det, ta gärna med dig en kompis för det går ännu lättare om man är två och kan bli trevligare också!

Du har alltså skaffat en VIC. Börja med att packa upp den och koppla in den till din TV så som det står i den medföljande bruksanvisningen. Sätt också i sladden till elnätet och du är klar att börja. **Slå på VICens strömbrytare** som sitter på höger sida av apparaten.

Nu lyser din TV-skärm upp i en klar blåaktig färg med ett vitt mittfält. Är inte färgen klart lysande kanske du har kopplat ihop din VIC med en svartvit TV? Eller behöver du justera färgkontrollen på TVn. Titta nu noga på skärmen en stund. Kontrollera att det som står på skärmen ser ut som nedanstående bild visar.



Det meddelande som syns när du först slår på VICen står med BLÅ bokstäver på en VIT bakgrund. Runt den vita rektangeln finns en färgad kant. Dennas färg kallas CYAN och är en ljusblå, svagt grönaktig färg.

Du kan utan vidare strunta i de första två raderna om du vill. Men, vill du veta vad de betyder så läs vidare. Annars hoppar du till nästa stycke i texten och fortsätter. Först kommer några stjärnor (asterisker, kallas de också) och sen står det CBM BASIC V2, följt av några asterisker. På nästa rad står det "3583 BYTES FREE", men det betyder inte att maskinen har nummer 3583 och den bytes fritt! I stället betyder det att den har ledigt minnesutrymme om 3583 bytes, och bytes är i vårt fall ett minnesutrymme, som rymmer 8 bit-ar eller åtta tecken, som kan vara 0 eller 1. Man

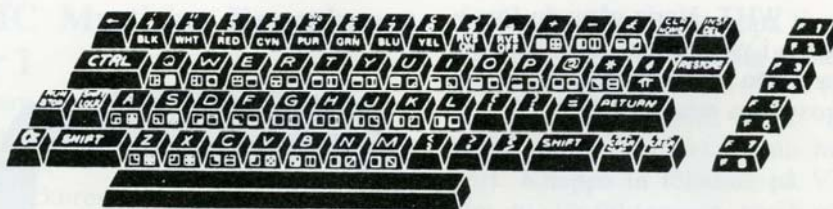
talar om bit, bitar eller bits, som lagts ihop åtta och åtta till något som ibland kallas "ord" och ibland "bytes". I VIC är det fixerat till åtta tecken (0:or eller 1:or). Det är med dessa som man kan återge olika talvärden när man låter placeringen bestämma vad varje etta eller nolla betyder. Men nog om detta nu, du hade alltså ett minnesutrymme på 3583 bytes som du kan använda för att skriva program i.



Ordet "Ready" som står på tredje raden är VICens sätt att tala om för dig att den står beredd att göra något. Du talar om vad den skall göra genom att knappa in det på tangentbordet. Utom ordet READY ser du en blinkande rektangel, det är denna som kallas markören eller cursorn på engelska. Läget för markören visar dig var det första tecken du slår in på tangentbordet kommer att hamna.

Tangentbordet på VIC

Tangentbordet är din styrcentral, från vilken du talar om för VIC vad den skall göra. Vad du än skriver in kommer att, samtidigt som det läggs in i något som kallas för arbetsminnet, också att visas på din TV-skärm. På bildskärmen kommer också att visas



alla de meddelanden, som VIC en vill ge till dig. Eftersom det språk som finns i VIC heter BASIC och är ett engelskt språk, kommer också alla meddelanden från VIC att stå på engelska.

Titta ett tag på tangentbordet. Se så många tangenter det finns med dubbla tecken, ett på översidan och ett på framsidan. Det är de grafiska tecknen som står på kanten på tangenterna. Andra tangenter har två rader med text på översidan, t.ex. RUN/STOP, INST/DEL, CLR/HOME. Två tangenter heter CRSR, den ena med en pil åt höger och en åt vänster, den andra med en pil uppåt och en nedåt. Det är tangenter för styrning av markören (du kommer ihåg "CuRSor" på engelska) och de har liksom de andra tangenterna med två "topptexter" en dubbel funktion. Den ena funktionen är i kraft när man inte gör något särskilt åt tangentbordet, det är den undre textens funktion, den andra når man genom att samtidigt trycka på tangenten SHIFT (det går bra med vilken som helst av de två SHIFT-tangenterna!). En enda tangent har två rader text men bara en funktion, det är SHIFT LOCK, som precis som på en skrivmaskin låser SHIFT-tangenten i nedtryckt läge (det är då man får STORA bokstäver på en skrivmaskin). Vi har några tangenter som har text på framsidan, dels alla siffertangenterna 1-8 och så nr 9 och 0, som har texten RVS/ON respektive RVS/off. Nu kanske du tror att du måste vara en bra maskinskrivare för att skriva på din VIC – nähå, det brukar gå bra med bara pekfingerens, i varje fall i början. Och du har ju alltid möjligheter att göra ett skrivmaskinsträningsprogram, om du vill...

Nu är det dags för första experimentet:

Du har hittat SHIFT-tangenten? Bra. Håll den nere med något finger på vänster hand och tryck samtidigt på CLR/HOME. Vad händer? För-

svinner meddelandena på skärmen? Hur ser bildskärmen ut?



När man trycker SHIFT CLR/HOME så rensas (CLears) bildskärmen. Så varje gång du vill rensa skärmen från text eller bilder kan du trycka på SHIFT+CLR/HOME.

Grattis! Nu har du utfört ditt första kommando till din VIC. Genom att trycka ner de två tangenterna talade du om för VICen att den skulle göra något, dvs att den skulle rensa skärmen. Och VICen svarade glatt och villigt! När skärmen är tom, placerar VIC den blinkande markören längst upp till vänster på skärmen och talar om för dig att VIC nu står beredd och väntar på nya kommandon. Som du slår in på tangentbordet.

Pröva med att slå in följande text (tryck i tur och ordning ner tangenterna som visas här)

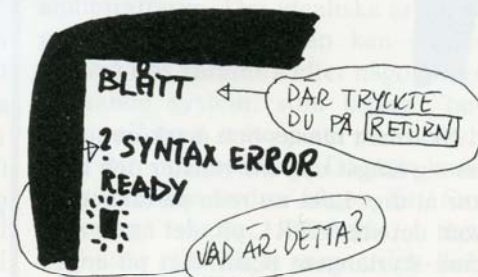
B L Å T T

På din bildskärm bör det se ut så här:



Varje bokstav du skrev kom också upp på skärmen. Markören flyttade sig åt höger ett steg varje gång som du tryckte ner en bokstavs-tangent och stannar vid slutet av ordet du skrev. (I det här fallet spelar det ingen roll vad du skrev in, så om du stavade fel och skrev VLOTT eller BLIFF så fungerar exemplet ändå).

Nu har du ordet BLÅTT (eller något annat ord) i blå färg på bildskärmen. Tryck nu på den stora knappen som det står RETURN på och titta på TV-skärmen.



Det meddelande som du då får upp och som VIC sänt till bildskärmen, kan översättas med "Jag förstår inte vad du menar?" SYNTAX ERROR är just VICs sätt att tala om att den inget begriper. Syntax är egentligen de språkregler som VICs inbyggda språkprogram (BASIC) måste följa till sista bokstaven och tecknet, om VIC skall förstå och kunna göra något med dina kommandon. Error står för fel, det kanske du redan förstått. Detta meddelande kommer varje gång som VIC inte förstår vad du vill att den skall göra.

Bli inte bekymrad när du ser meddelandet "SYNTAX ERROR". Du har inte gjort något som skadar din VIC. (Faktiskt kan du inte göra detta med mindre du ställer dig på tangentbordet, för du kan inte knappa in något till VIC som kommer att ge dig allvarliga problem med VICs funktion). Meddelandet kom upp därför att VIC bara förstår vissa bestämda ord, den har ett begränsat ordförråd. Vi skall tala mer om detta senare. Låt oss gå tillbaka till experiment med tangentbordet.

VIC-färgerna

Genom att skriva ordet BLÅTT fick du ett SYNTAX ERROR. Det får du för varje annat ord också. Men hur

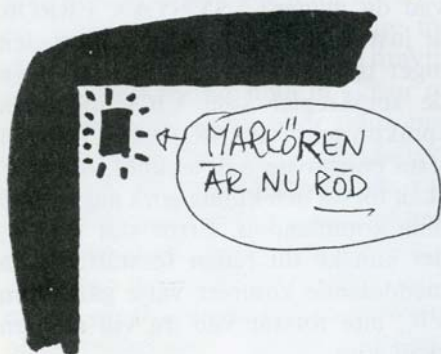
forts. nästa sida

skall du tala om för VIC vilka färger den skall använda? Prova det här experimentet:

Håll nere SHIFT och tryck på CLR/HOME.



Skärmen rensas och markören ställer sig längst upp till vänster och blinkar åt dig. Leta nu reda på tangenten som det står CTRL på, det är en speciell styrtangent (ConTRol på engelska). Håll nu denna CTRL-tangent nere samtidigt som du trycker på tangenten med nr 3 i övre raden. Vad händer?



Ja, nu blev markören röd! Titta omigen på tangenten med 3:an på. Du ser att på framsidan står det RED, vilket är engelska för röd om du inte visste det förut. Du ser på alla siffer-tangenterna från 1 till 8 så har de olika färgbeteckningar på framsidan. Håll bara ner CTRL-tangenten samtidigt, så kommer du åt färgerna: tryck ner t ex CTRL + 8 och du ser att du får gult (YELlow är engelskans gul). Prova nu och byt färg på markören till alla möjliga färger. (Vad händer när du trycker på tangenten som har texten WHT (för WHiTe, vitt)? Prova att skriva in färgnamn i rätt färg. Vad händer när du skriver VITT i vitt? Jo, eftersom bakgrundsfärgen redan är vit, så kommer inte markör eller bokstav att synas när du ger kommandot

CTRL + WHT. Varje gång du försöker skriva i vitt så syns inget på bildskärmen. Du kan ju prova att sända några osynliga meddelanden!

Regnbågsmaskinen

Nu är det tid att upptäcka varför vi kallar VIC för regnbågsmaskinen. Rensa skärmen (nu kommer du väl ihåg hur man gör det: SHIFT + CLR/HOME).

Med rensad bildskärm och markören tillbaka i sitt startläge högst upp till vänster, tryck nu på följande tangenten:

Håll nere CTRL och tryck på BLK (Black för svart), håll sen ner CTRL och tryck på RVS/ON.

Klart? Nu skall du leta rätt på mellanslagstangenten, också kallad SPACE-tangenten. Det är den långa tangenten längst ner (närmast dig) på tangentbordet. När man trycker på den får man normalt ett mellanrum (en SPACE på engelska) på bildskärmen, precis som en skrivmaskin ger ett mellanrum på ett papper när man skriver och trycker mellanslagstangenten. Tryck ner mellanslagstangenten och HÅLL DEN NERE. Vad händer? Rita VICen en svart tjock linje tvärs över bildskärmen?

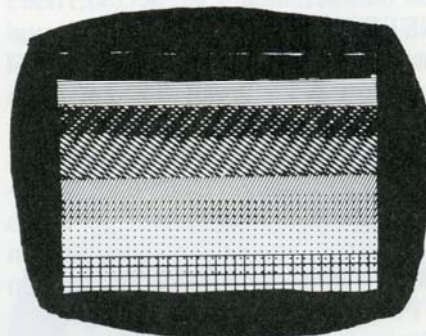


Fortsätt att hålla mellanslagstangenten nere till dess linjen nått högerkanten på bildskärmens ljusa fyrkant. Vad händer då? Jo, VICen fortsätter att rita den svarta linjen på nästa rad. Låt VIC rita till dess du fått två hela rader – sen släpper du upp mellanslagstangenten.



Nu kan du byta till en annan färg, t.ex. till RED (rött var det ju).

Håll nere CTRL, tryck på RED.



Tryck ner mellanslagstangenten än en gång. En RÖD tjock linje kommer fram på bildskärmen. Fortsätt så och rita linjer och byta färger tills bildskärmen är fylld med VIC-REGNBÅGEN!

Du kan använda VIC-regnbågen för att kontrollera din TV:s färginställning: ställ in TVn så att färgerna ser bra ut för dig. När du har slutat göra regnbågar, då kan du göra följande: Håll ner CTRL och tryck på RVS/OFF, dvs sifvertangenten för "9", som på framsidan visar texten RVS/OFF. Detta ställer om VIC mellanslagstangent till dess normala funktion. Genom att trycka på mellanslaget efter att du tryckt på RVS/OFF-tangenten får du nu som vanligt mellanslag, dvs tomrum. Du får flera tillfällen att experimentera med RVS/ON och RVS/OFF längre fram i den här artikelserien.

VIC Mystiska Experiment nr 1

Fram till nu har du tittat på eller experimenterat med följande VIC-egenskaper:

- * Markören
- * Skift-tangenten
- * Return-tangenten
- * Skärmrensning
- * RVS/ON
- * Syntax Error ("VIC förstår inte")
- * Inskrivning av ord
- * Färgtangenterna
- * Mellanslagstangenten
- * CLR/HOME-tangenten
- * RVS/OFF
- * Att rita en regnbåge

Vid slutet av varje artikel i denna serie kommer du att finna en mystisk

övning som du kan knappa in på din VIC och som vill göra dig nyfiken på material i kommande artiklar. Här är Mystiska Experiment nr 1. Rensa VICs skärm (det vet du nu hur du gör). Knappa in följande på VICen. Om du slår fel tangent, tryck på RETURN-tangenten och skriv om raden. Experimentera och se vad du får!

Tips nr 1: "Flyttfåglar"

Tips nr 2: Tryck på RUN/STOP för att "stoppa" flyttningen.

Tips nr 3: Skriv R U N och tryck RETURN för att åter starta flyttningen.

Ha det så trevligt med din VIC, vi ses igen!

Du skriver in: 1 0 ? " MELLANSLAG J O K " ; RETURN

VIC visar: 1 0 ? " (mellanrum) J O K " ;

Du skriver in: 2 0 G O T O 1 0 RETURN

VIC visar: 2 0 G O T O 1 0

Du skriver in: R U N RETURN

VIC visar: Detta är den mystiska delen!

Det kom ett brev...

eller "HUR man HÅLLER liv i en DATOR-KLUBB"

Så här skriver Krister Svärd, Silvermyntsvägen 19, 231 00 Trelleborg, tel. 0410-181 77("Tyvärr måste jag meddela, att Landskrona-klubben nu sjunger på sista versen. Klubben bygger helt på medlemmarnas bidrag och idérikedomen har varit dålig..."

Du tar upp ett viktigt problem, Krister. Problemet hur en liten ideell förening skall kunna aktivera sina medlemmar utan att det kostar all tid i världen för en eller ett par medlemmar och att de andra bara "åker snålskjuts" på deras arbete. Det finns egentligen inget hundra procentigt effektivt recept att bjuda på, men vi kommer här i VIC-rapport att försöka visa på några viktiga sätt att aktivera klubbmedlemmar och bringa liv i en tynande klubbverksamhet.

Ett av de viktigaste medlen är "var informella"! Många klubbar har dött sottdöden därför att man varit alltför

formella och försökt driva föreningsverksamheten som ett storföretag med sammanträden, ordförande, sekreterare, protokoll och hela snurren. Då klubbverksamhet dock kräver en viss stadga (obs vitsen, man bor väl inte i Göteborgstrakten för inte!) så kan man ju lämpligen försöka hålla ungefär följande fördelning av tiden på en "klubbkväll":

30 minuter allmänt samkväm, då senkomna hinner fram och man hinner snacka om litet av varje.

10 minuter som ordföranden, för naturligtvis utses en ordförande i en klubb, (det behövs åtminstone om man skall försöka få kommunalt stöd!), använder för att dra igenom om några speciella frågor dykt upp sedan förra träffen och som behöver diskuteras. Jag talar nu inte om ett årsmöte utan om ett vanligt klubbmöte. För årsmötet gäller speciella regler

om ansvarsfrihet och sånt. Därefter används resten av kvällen till att umgås under fria former, kanske har någon ett eller annat spel eller program att visa - någon behöver kanske hjälp med en strulig programrutin. Det viktiga är att man inte skall styra verksamheten för mycket i denna del utan låta var och en göra vad han/hon känner mest för.

En annan viktig fråga är: "Var skall vi hålla till?" Lämpligaste svaret här beror på klubbens medlemsantal och hur många som brukar komma på klubbträffarna. Det idealiska är ju att man är så få, att man kan träffas hemma hos varandra efter något slags roterande system, som lägger "bekymren" med mötesplats på var och en i tur och ordning. Då kan man också kanske ordna någon lätt förfriskning, kaffe, smörgås, the, VIC-juice eller något annat drick- och ätbart beroende på medlemmarnas önskemål. Är "rotationen" fullt genomförd behöver man ju inte ens ta ut någon mötesavgift eftersom alla drabbas så småningom! Går det inte att hålla till hemma hos varandra kan man försöka tala med någon VIC-återförsäljare, om han har lust att ställa upp en kväll i månaden eller så och låta klubben träffas i firmalokalerna. En fördel med detta är ju att då kanske man kan få låna företagets datorer och slippa släpa på egna prylar för programbytesverksamheten.

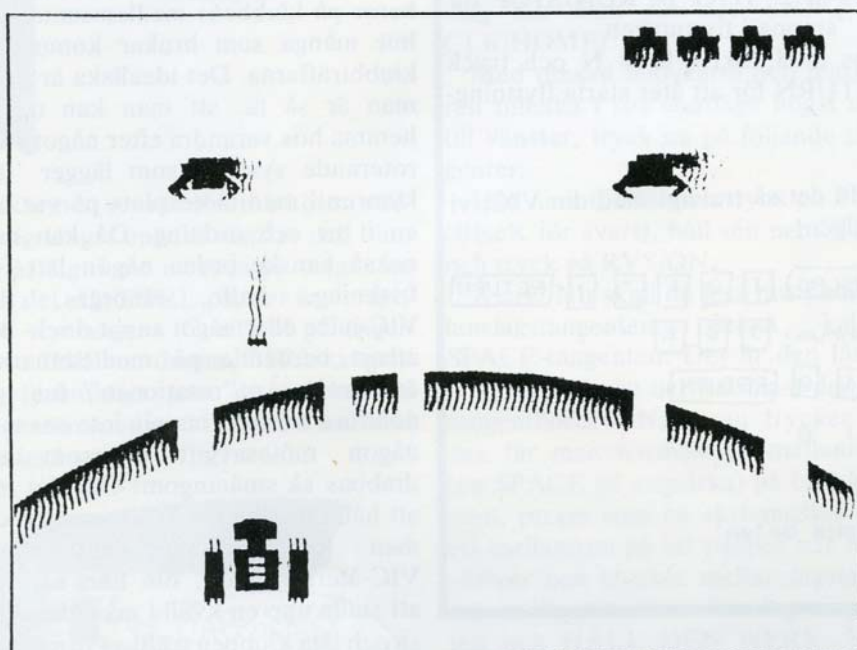
Ta kontakt med fritidskonsumenten i kommunen!

Fritids- eller föreningskonsumenten i kommunen brukar också kunna ge ett visst stöd åt nybildade ideella föreningar. Man har kanske lämpliga lokaler, man ger bidrag per möte per deltagare, man kan hjälpa till att skaffa föredragshållare och man kanske t.o.m. har möjlighet att lämna bidrag till en "klubb dator"? Vilka möjligheter som finns, får man bara reda på om man frågar! Se i telefonkatalogen under "Fritidsförvaltningen" i din kommun.

En sak, som brukar bidra till att föreningen växer sig starkare, är ett eget medlemsblad. Försök bara ordna så, att det inte alltid är endast en enda medlem som får ansvaret att producera.

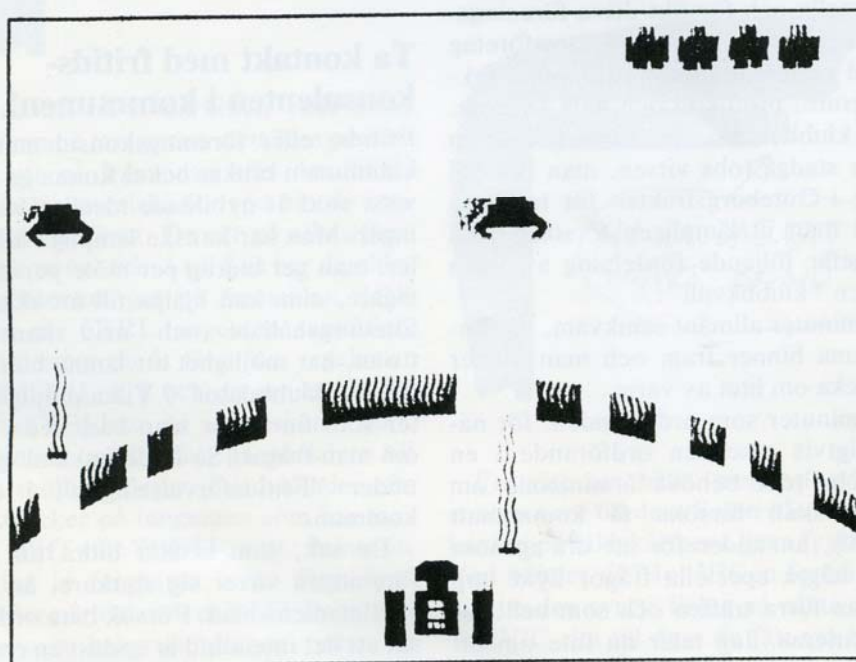
forts. sid. 11

HOPPSAN!



Figur A

**eller HUR du
kan få högre
poäng genom
fiffighet när du
spelar VIC-20
Omega Race
eller Gorf**



Figur B

Even progomamerare jör nåra fehl dåodå. Medan man lägger ner allt intresse på att få spelet att vara lättspelat och passa in i ett begränsat minnesutrymme, händer det att man missar något i programmet i alla fall. Och ibland kan dessa missar utnyttjas av spelaren. Här skall berättas om några sådana i Commodore-spelen Omega Race och Gorf, som kan hjälpa dig att få ännu högre poäng än du annars brukar få.

Förmodligen finns de mest användbara missarna i VIC-20-versionen av Omega Race. Den första miss som skall påvisas är inte ett egentligt fel, utan låt oss kalla det för en "icke avsedd egenskap" i stället. Normalt får du tre skepp i detta spel när du startar. Om du emellertid håller nere SHIFT-tangenten och samtidigt trycker antingen på F1 (för joystick) eller F3 (för paddle) i början av spelet kommer du att få – räkna dom bara! –

hela fem skepp. Prova och se om du inte får bättre poäng!

En annan intressant grej med Omega Race är att du förmodligen inte känner till (även om några läsare kanske redan råkat ut för det av en händelse) att man kan återställa förlorade skepp om man kommit ner till ett eller två. Håll reda på det antal skärmar som du har rensat. När du rensat den fjärde skärmen och då musiken

börjar spela, håll ner funktionstangenten du använder för att starta spelet. Fortsätt hålla nere tangenten tills budskapet "droid force eliminated" har dykt upp i skärmen (ibland kommer bara en del av budskapet upp på skärmen, men det spelar ingen roll). När sedan femte skärmen dyker upp bör du nu ha tre skepp. Denna procedur fungerar ibland när man använder skiftade funktionstangenter, men re-

sultatet är i detta fall inte helt förutsägbart.

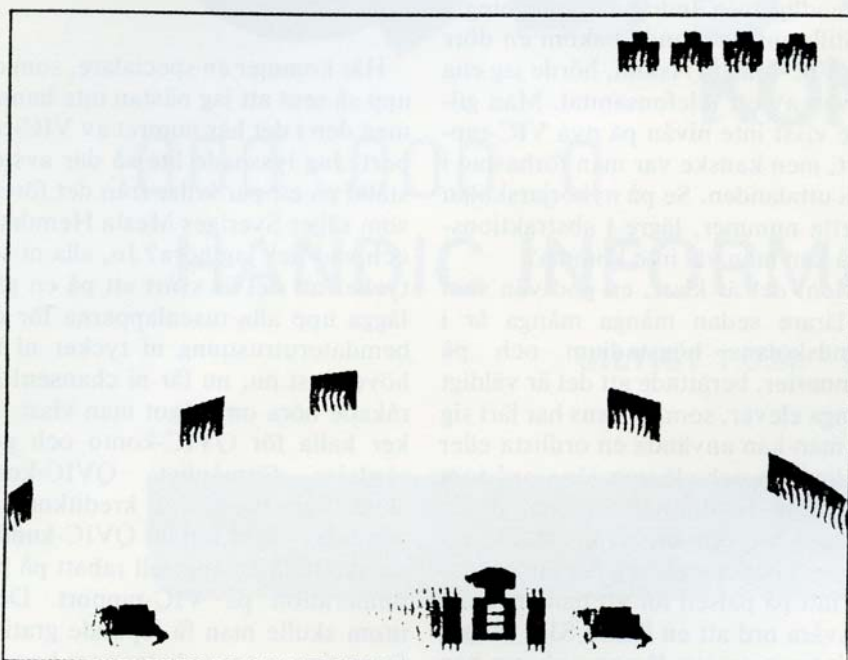
Håll i minnet att om du inte har förlorat några skepp fram till detta stadium och har fått ett bonus-skepp, då vore det inte bra för dig att använda denna metod, därför att antalet skepp sätts till tre oavsett hur många du har innan. Du kan fortsätta och göra detta varje fjärde skärm, alldeles före budskapet "droid force eliminated". Prova det! Det borde också förbättra dina poängsiffror.

Nästa "nyttiga" miss finns i Gorf för VIC-20. Detta tillåter att du spelar i "invaders"-omgången så länge du vill, den låter dig också hacka fram poäng nästan till oändligheten. Proceduren är ganska invecklad, då här måste vi visa några bilder för att förklara tydligt nog.

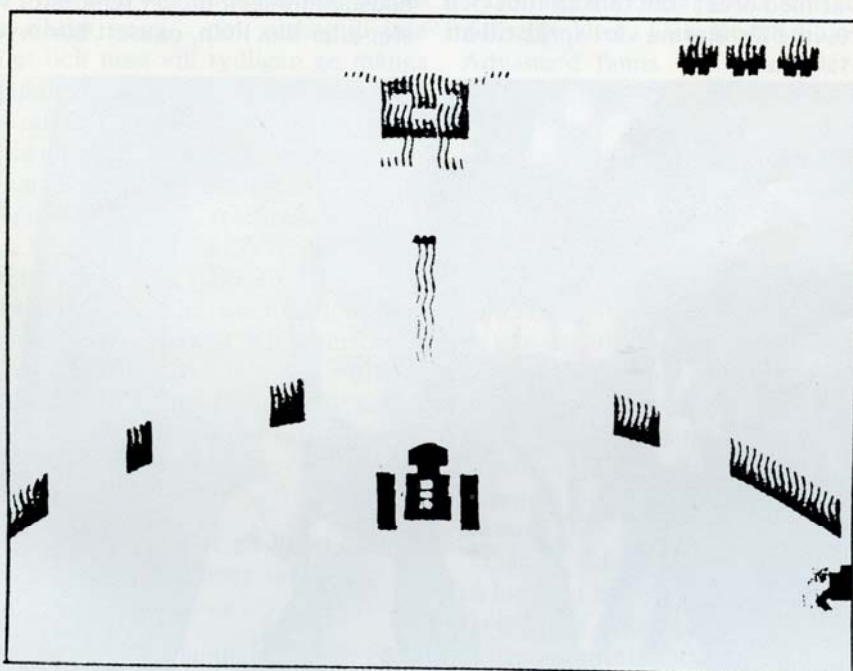
Så här går det till: Skjut alla invaders utom den som finns i övre högra hörnet (se fig A). Stanna sen i mitten nedtill på skärmen för att undvika invaderarna och deras projektiler (se fig B). När invaderarna har nått nedersta raden på skärmen, flytta med dom så att ditt skepp kommer bort från mitten på skärmen (se fig C).

När sedan en av invaderarna faller ur skärmen kommer ditt skepp att förstöras. Då skall du omedelbart skjuta fram joysticken och flytta nästa skepp till mitten av skärmen. En invaderare kommer aldrig att träffa dig om du placerar dig så högt i skärmen som du kan. Dessutom kommer denne invaderare aldrig att falla ur neråt i skärmen. Nu kan du bara sitta lugnt och skjuta Gorf och två olika flygande tefat för att få bonuspoäng (se fig D). När du tröttnar kan du gå över till nästa omgång genom att krascha ditt skepp mot den kvarvarande invaderaren.

Detta är de hittills enda användbara missarna som vi känner till just nu. Förhoppningsvis kommer de att ge dig möjligheter till högre poängtal med Omega Race och Gorf. Men om du finner på något annat som du tycker verkar mystiskt när du spelar några av dessa spel, varför inte skriva till oss och berätta om det? Då tar vi upp det i VIC-rapport och fler får reda på det och du får en present som tack för din medverkan! Redaktionen adress står på ledarsidan.



Figur C



Figur D

Kapten Storöra



Hej igen!

Det är jag som är kapten Storöra och som lyssnar så bra även när man inte talar direkt till mig utan helst kanske såg att jag inte fanns ens i närheten...

Den här gången kan jag berätta lite om ett rykte, som dock enligt uppgift från allra högsta möjliga ort är helt utan all grund. Han, som trodde att VIC-20 skulle slås ut helt av VIC-64:an måste vara lite lång i ansiktet, när nu Commodore USA aviserar full produktion av 20:an. Man har ibland t.o.m. haft leveransproblem, som dock verkar lösas genom den nya satsningen.

Den länge aviserade VIP-64:an, jo du läste rätt, den skall heta VIP för det kunde ju stå för Very Interesting Portable, för det är den. Med färgbildskärm och allt, dubbelfloppy och hela surven och inte väger den för mycket heller. Trots ivrigt lyssnande och gömmande bakom draperierna fick jag dock inte fram vikten vid mitt besök på huvudkontoret. Jo, denna trevliga lilla bärbara dator, som är helt tänkt för den jäktade affärsmanen, för konsulten på resor och för hemmapularen som vill ha något som inte de flesta har, en dator han kan gömma undan och bara ha alldeles för sig själv. Just denna lilla trevna apparat sägs komma till vårt land strax kring jul – kanske blir det årets julklapp? Speciellt som den mer kända portabla datorn som börjar på "O" kanske kommer att vara obefintlig ganska snart eftersom tillverkaren inte finns mer...

Något som jag fick reda på var att man nu på HK bestämt sig för att göra ett billigt interface för PET och 64:an (Centronics) och det skall komma som en kabel, färdig att stoppa i både dator och skrivare. Så det finns gott hopp om att vi snart skall kunna använda en billig skönskrivare med V24-snitt till våra mest älskade datorer, VIC och PET. Priset lär bli ynkligt litet, bara några hundralappar och kabeln kommer att kunna passas till

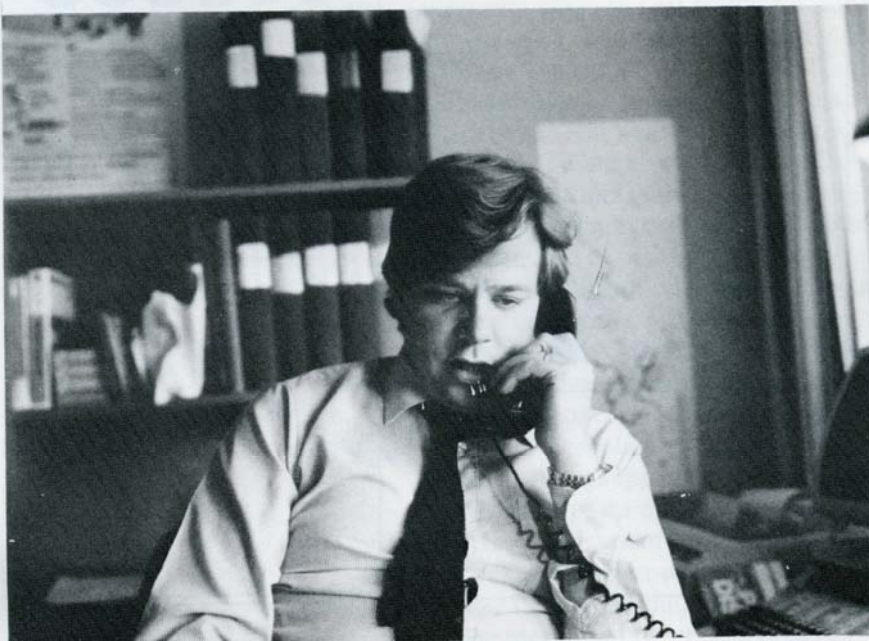
praktiskt taget alla slag av periferenheter, säger man. (Fråga mig bara inte vem, för om jag skvallrar får jag förmodligen en lindrigare kölhalning!)

Stillsamt lyssnande bakom en dörr hos VIC-folket i Askim, hörde jag ena halvan av ett telefonsamtal. Man gillade visst inte nivån på nya VIC-rapport, men kanske var man förhastad i sina uttalanden. Se på nybörjarskolan i detta nummer, lägre i abstraktionsnivå kan man väl inte komma?

Men, det är klart, en god vän som är lärare sedan många många år i grundskolans högstadium och på gymnasier, berättade att det är väldigt många elever, som inte ens har lärt sig att man kan använda en ordlista eller ett lexikon och slå upp såna ord som man inte förstår. En annan av de många Storöra-vännerna stack ett papper i handen på mig där en journalist fått på pälsen för att han använde så svåra ord att en läsare SJU gånger varit tvungen att slå upp ord som han inte förstod och tyckte det var självaste hmhm att sånt skulle få skrivas. Journalisten försvarade sig med att det är med ordet som tanken föds och att vi inte får utarma vårt språk till att

enbart vara enstaviga datorord med en och endast en betydelse. Vi behöver nyanser, gråtoner, schatteringar, synonymet eller vad du vill kalla dom ord som har olika betydelse alltefter sammanhanget de står i och där den precisa nyansen kanske kräver ett "svårt" ord. Såväl jag Storöra, som vår redaktör ÅF väddar därför till de öppna sinnenas positiva kritik: Skriv nåt bättre själv då! Du får ju till och med honorar för införda bidrag.

Här kommer en specialare, som dök upp så sent att jag nästan inte hann få med den i det här numret av VIC-rapport. Jag lyssnade lite så där avsides ställd på ett par killar från det företag som säljer Sveriges Mesta Hemdator, och vad fick jag höra? Jo, alla ni som tycker att det är svårt att på en gång lägga upp alla tusenlapparna för den hemdatorutrustning ni tycker ni behöver just nu, nu får ni chansen! Jag råkade höra om något man visst tänker kalla för QVIC-konto och göra särdeles förmånligt. QVIC-kortet skulle vara mer än ett kreditkort, det nämndes något om att QVIC-kunderna skulle få en speciell rabatt på prenumeration på VIC-rapport. Dessutom skulle man få löpande gratisinformationer om nyheter som kommer inom VIC-familjen på ett tidigt stadium. En, som det sägs, absolut nyhet är att man tänker tydligen hålla månadsbetalningen till QVIC-kontot lika stor eller lika liten, oavsett hur myck-



QVIC

KONTO

000 000 0

HANDIC INFORMATION

GILTIGT T.O.M. 00 00

et kontot är utnyttjat inom de olika beloppsgränserna! En siffra som 125:-/månad nämndes när jag lyssnade riktigt noga... det verkar ju inte högt och man vill tydligen ge många människor chansen att skaffa sig datakraft utan att för den skull bli ruinerade eller behöva avstå från semesterresan man drömt om och sparat till. Jag råkade höra ett telefonsamtal mellan Per Hagklint på QVIC-konto och en handlare som tydligen var mycket förtjust, det verkar som om idén slagit mycket väl ut bland VIC-återförsäljarna. (Jag smög mig t.o.m. till att knäppa en bild med min ljudlösa kamera, just på Per, som satt i telefon hela tiden jag såg honom. "Telefonen gick varm").

Nä, nu lägger Per på luren, det är bäst att jag försvinner innan jag får höra för mycket...

Innan jag helt lämnar min väl gömda plats på VIC-kontoret så får jag tag

i en ny prislista med mina fingrar, som är lika långa som mina öron...

Det var verkligen godis som dukades upp: Utom Calc Result Easy och Advanced fanns VIC Kalender, ett Text-program och ett Tool 64. Där fanns Simons basic, den unge skolgrabben som gjorde så många fina tillägg till Basic. Och där fanns, hör och häpna! något som kallades Tele Data! Det tyder på att man nu snart kan koppla upp sig på Teledatanäten i vårt land. Och om ett annat rykte jag hörde häromnatten talar sant, så är det så att Televerket har bestämt hur alla hemdatorer skall kunna tala med varandra över Teledata – ja Televerket bestämde ju inte allt själva utan konfererade med hemdatorföretagen först.

Går vi vidare i prislistan för 64:an så hittar vi mer om Teletext, man har ett kablage också och en massa annat nyttigt som interface till RS232 både av terminaltyp och av current loop

typ, men fråga mig inte vad det betyder, då börjar jag humma och ööäaaa. Man har också ett modem 300/300, vilket troligen betyder att det fungerar med 300 baud ut och in plus ett s.k. Split Speed modem med 75/1200 i sifferbeteckning, vilket förmodligen betyder 75 baud in och 1200 ut eller tvärtom(?). I katalogen finns också en massa litteratur och hör här, man har t.o.m. tagit med VIC NYTT, katalogen måtte varit tryckt innan vi bytte namn till VIC-rapport!

Före alla spelprogrammen presenteras två intressanta instruktionsprogram för utbildare: Redovisning 64 och Register 64. Båda två ingår i det s.k. ADB-paketet, som också omfattar fyra böcker: ADB på VIC 64, Ord & Text, Redovisning, Registerhantering.

Men nu är det definitivt slut från kapten Storöra!


```

2 POKE56,24:POKE55,103:GOSUB29
3 D=37154:P1=D-3:P2=D-2:DF=30720:V=36878:S=V-4:M1=30:X=50:GOTO19
4 FORT=240T0208STEP-4:POKE5,T:FORTT=0T030:POKEV,TT/2:NEXT:NEXTT:POKE5,0:ME=7932
5 POKEOM,32:POKEOM+DF,10:POKEME,M1:POKEME+DF,7:IFFTHEN40
6 K=K+1:ON-(K/2)INT(K/2)GOTO8:IFK>600THEN37
7 FORT=1T02:POKEV,T#4:POKE5+1,128+K/5:NEXT:POKE5+1,0
8 POKED,127:P=PEEK(P2)AND128:J0=-(P=0)
9 POKED,255:P=PEEK(P1):J1=-(PAND8=0):J2=-(PAND16=0):J3=-(PAND4=0)
10 IFJ0THENC=1:M1=62:GOTO14
11 IFJ1THENC=22:M1=22:GOTO14
12 IFJ2THENC=-1:M1=60:GOTO14
13 IFJ3THENC=-22:M1=30
14 OM=ME:ME=ME+C:C=0
15 IFPEEK(ME)<>32ANDPEEK(ME)<>42THENME=OM
16 IFPEEK(ME)=42THENF=1:GOTO5
17 ON-(ME>7921)GOTO18:SYS887:ME=ME+22:GOTO5
18 ON-(ME>7944)GOTO5:SYS905:ME=ME-22:GOTO5
19 DIMA(3):A(0)=2:A(1)=-44:A(2)=-2:A(3)=44:WL=209:HL=32:SC=6228:A9=6943
20 SYS861:PRINT"LABYRINTEN TILLVERKAS"
21 FORT=SC+21T07679STEP22:POKET,32:NEXT:FORT=SCTOSC+21:POKET,32:NEXT
22 J=INT(RND(1)*4):X3=J
23 B=A9+A(J)
24 IFPEEK(B)=WLTHENPOKEB,J:POKEA9+A(J)/2,HL:A9=B:GOTO22
25 J=(J+1)*-(J<3):IFJ<>X3THEN23
26 J=PEEK(A9):POKEA9,HL:IFJ<4THENA9=A9-A(J):GOTO22
27 TB=SC+INT(RND(0)*20)+220:ON-(PEEK(TB)<>32)GOTO27:POKETB,42
28 SYS830:POKE828,204:POKE829,28:SYS923:GOTO4
29 FORI=830T0974:READA:POKEI,A:NEXT:RETURN
30 DATA 169,238,141,15,144,169,0,133,251,169,150,133,252,160,0,169,10,145,251,20
0,208
31 DATA 251,230,252,165,252,201,152,208,241,96,169,84,133,251,169,24,133,252,160
,0,169
32 DATA 209,145,251,200,208,251,230,252,165,252,201,30,208
33 DATA 241,96,173,60,3,56,233,22,176,3,206,61,3,141,60,3,56,176,19,234,173,60,3
,24,105
34 DATA 22,144,3,238,61,3,141,60,3,24,144,1,234,169,0,133,0,169,30,133,1,173,60,
3,133
35 DATA 254,173,61,3,133,255,169,0,133,253,160,0,177,254,164,253,145,0,132,253,2
30,253
36 DATA 234,208,2,230,1,230,254,208,2,230,255,169,32,197,1,208,227,96
37 POKEV,15:FORT=255T0127STEP-2:POKE5,T:POKEV-9,255:FORG=1T010:NEXT
38 POKEV-9,242:FORG=1T010:NEXT:POKEV-9,240:NEXT:POKEV-1,220:FORG=15T008STEP-.05
39 POKEV,0:POKEV+1,0*10:NEXT:POKEV-1,0:POKEV+1,238:GOSUB42:RUN
40 POKETB,32:POKEV-1,253:FORG=30T008STEP-.15:POKEV,0/2:NEXT:X=X+50:IFX>449THENX=4
50
41 POKEV-1,0:F=0:K=X:R=R+1:GOSUB42:GOTO27
42 PRINT"NIVA"R"II":PRINT"TRYCK F7 ":A$="":GETA$:ON-(A$<>"II")GOTO42:RETURN

```

READY.

Tidsinställd bomb i listig labyrint

Programmet till detta spel är förrädisk kort. Man kan enkelt säga att det är det kortaste och bästa spelprogram vi sett till VIC-20. Du befinner dig i en labyrint, större än din bildskärm. Du måste flytta dig genom labyrinten för att försöka desarmera en tickande bomb, som gömts någonstans högt upp i labyrinten och som du inte ser i början av spelet. Skärmen flyttar sig (bildlikt talat) det är egentligen labyrinten som flyttar sig på skärmen allteftersom du själv rör dig genom den. Du måste lära av dina misstag för att inte bomben skall ticka för fort och till slut explodera och allt är förlorat... (För en oexpanderad VIC-20).

Du spelar "Tidsinställd" mot klockan. Du börjar i botten av en labyrint, som är ungefär tre gånger större än bildskärmen. Högst uppe i labyrinten finns en tidsinställd bomb som tickar och tickar. Ju närmare explosionen man kommer i tiden desto snabbare tickar bomben. Om du når bomben i tid måste du styra din pil in i den för att desarmera den. Om du lyckas, får du ett nytt försök med samma labyrint, men med bomben på en annan plats och med kortare stubin. Detta fortsätter tills tiden tar slut för dig. Om du missar att desarmera bomben får du en ny labyrint och en ny bomb med längre stubin.

Mina vänner som jag spelat med, anser inte detta spel något speciellt bra tävlingsspel. Istället blir de utomordentliga baksätteschaufförer och talat om för mig (eller den som spelar) vart han skall gå och retas om det går illa.

Tidsinställd är ganska utmanande mot spelarens minne för rymdmässiga förhållanden. Människor, som först retar sig på att de bara ser en del av labyrinten, brukar snabbt vänja sig vid att tänka framåt och komma ihåg återvändsgränder och fria vägar genom labyrinten. En förmåga att minnas de bra och de dåliga rörelserna är verkligen viktig när man kommer till de senare delarna av spelet.

Hemligheten med att kunna göra labyrinten så mycket större än bildskärmen ligger i en kort maskinkods-rutin, som anropas för att datera upp den visade delen varje gång spelaren flyttar sig uppåt eller neråt i labyrinten. Den gör detta så snabbt, att jag använder samma BASIC-rutin för joysticken som finns i Compute:s First Book of VIC (den finns ännu inte i Sverige annat än direktimporterad, men får vi många förfrågningar till redaktionen kanske vi kan ta hem en bunt). I programmet finns faktiskt tre olika maskinspråksrutiner som representeras av DATA-satserna. En fyller labyrinten med rätt tecken, en annan fyller skärmens färgminne med rätt färger och den tredje låter labyrinten "skrolla" dvs rulla. Man hade kunnat använda BASIC POKE-kommandon för att göra sådana saker, men det skulle tagit för lång tid vid programkörningen. Speciellt rullningen hade blivit så långsam att det inte ens hade varit roligt att titta på!

När du knappar in programmet, glöm inte att SAVEa programmet innan du kör det med RUN. Ett skrivfel i en DATA-sats kan göra att du förlorar hela programmet! Var därför extra noga när du skriver in DATA-satserna! Om du hittar ett fel i programmet är det troligen just i DATA-satserna, så titta där först.

När du kör det med RUN, kommer först en kort paus, medan maskinspråksdelarna POKEas in i kassetbufferten. Därefter kommer skärmen att rensas och orden "labyrinten tillverkas" att komma fram. Eftersom labyrinten är så stor tar det VICen ungefär en minut att rita upp den, så ta det med ro. När den är färdig kommer en liten musikstump och väcker dig till spel.

Ge inte upp om du elimineras första gången du spelar, det tar ett tag att vänja sig vid att se utanför skärmen och planera sin väg genom labyrinten.

Detta spel kräver joystick!

Martin Fredriksson

...det kom ett brev...

forts. från sid. 5

ra tidningen eller bladet. Dela upp arbetet, gör några smågrupper inom föreningen, som ägnar sig åt en bestämd fråga eller ett visst program, som man arbetar igenom och presenterar för sina klubbkompisar i medlemsbladet. Någon annan kanske prenumererar på någon utländsk tidning som inte alla har, skriv och berättar om vad som står i den, var dock försiktig med copyright-rätten så inga tråkigheter inträffar med "stulet material". I de flesta fall är det tillåtet att referera artiklar, bara man anger källan!

För den klubb, som vill gå ut och göra reklam för sig finns det många sätt: Man kan lägga informationsblad om klubben hos de lokala VIC-återförsäljarna, de är säkert bara glada åt att hjälpa till, bara någon klubbmedlem någon gång i veckan eller så tittar till bunt med informationsblad och håller den i ordning...

Man kan också gå till lokaltidning och be dem göra ett reportage från en klubbkväll – naturligtvis ser man då till att man har ett roligt spelprogram att visa eller en bra föredragshållare eller gör något annat mindre vanligt, kanske lär ut programmering till ungdomar från en ungdomsgård... Här finns 100 idéer att plocka fram ur fantasin!

Var dock så mycket realist, ni som tänker starta egna klubbar ute i landet, att ni börjar försiktigt och skapar en "kärna" av ideellt inställda kamrater i kring er innan ni går ut med någon större kampanj för medlemsvärvning. Låt klubben växa i lugn takt, försök inte tvinga fram snabba resultat inom något område. Hackers, datioter, hobbyprogrammerare, datafnattar, VICtigpettrar eller vad man nu vill kalla de individer, som sysslar med datorer som hobby – de mår bäst av att man inte jagar dem (liksom alla människor egentligen gör). Och framför allt: deppa inte om en klubbträff blir mindre lyckad, det kanske var ett alltför bra TV-program den kvällen? Eller kanske man hade förberett alltför mycket och väntade sig alltför mycket av mötet. Ta klubblivet för vad det är – en trevlig avkoppling tillsammans med likasinnade!

Red.



HEMELEKTRONIK 83

CONSUMER ELECTRONICS SHOW

Visa upp "dig" på
**Skandinavians största
Hemelektronikmässa**
i Göteborg 10 – 13 november 1983.

Fack- och publikmäs-
sa för HiFi, video och
hemdatorbranschen.

Missa inte framtids-
mässan. Boka din
monterplats idag!

Ring Tom Larsen eller Anitha Asplind som
ger dig detaljinformation. Tel. 031-20 00 00.

Arrangör:

Svenska Mässan  Stiftelse

BOX 5222, 402 24 GÖTEBORG: TEL. 031/20 00 00. TELEX 20 600.

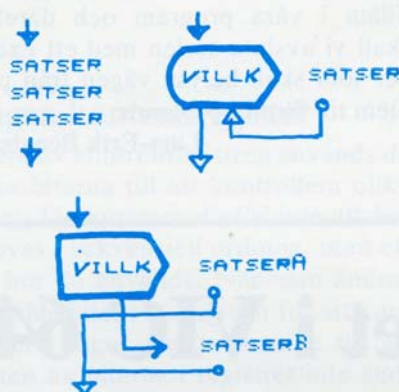
i samarbete med:

elektronik VÄRLDEN
I AUDIO & TELEVISION

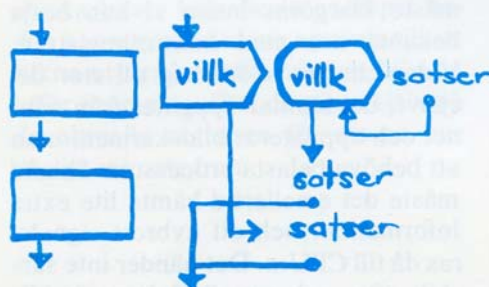
Vi lär oss skriva strukturerade BASICprogram

Satser i sekvens, upprepade satser och alternativa vägar, är den strukturerade programmeringens grundformer. Dessa former skall beskrivas med en väg in och en väg ut.

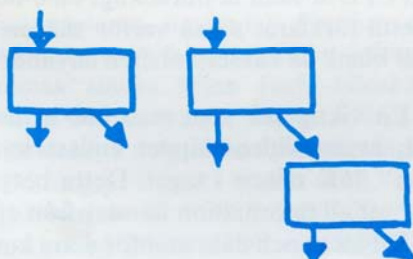
En godtycklig dellösning skall inte tillåtas ha mer än en ingång och en utgång.



Sammanfogas dessa dellösningar uppstår nya konstruktioner, men även om dessa blir stora och komplexa, invecklade, så består grundformen "en väg in en väg ut".



Tänker vi oss däremot en dellösning med en väg in och två ut, så får vi ett annat förlopp. Två dellösningar som sätts samman ger plötsligt en väg in och TRE ut!



Fortsätter vi och bygger en lösning med 100-tals dellösningar så växer snabbt en okontrollerbar struktur fram. Det som ser oskyldigt ut när man bara ser på enkla dellösningar, får förödande döljder i sammansatta större beskrivningar. En väg in och en ut ger inte denna utveckling.

Satser som hör samman av en eller annan orsak, skrivs lämpligen på en rad, eller om denna inte räcker, som ett par rader efter varandra.

Satser som inte har något gemensamt bör man skilja från varandra och kanske till och med förtydliga skillnaden med en kommentarrad eller en "satslös rad".

Ex.:
100 A=0:B=0:C=0
110:
120 INPUT "ANGE MAX ", MAX
130:

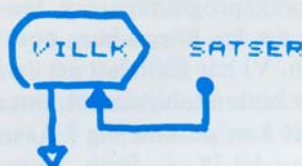
En grupp samhörande satser kan också ofta med fördel placeras i ett subrutinblock:

Ex.:
100 GOSUB 1000:REM INITIERA
110 INPUT "ANGE MAX ",MAX

1000 REM INITIERA
1010 DIM TAL(100)
1020 A=0:B=0:C=0:FOR I=1
TO 100:TAL(I)=0:NEXT
1030 RETURN

Även om BASIC inte uppfattar mer än 2 tecken som signifikanta (dvs som särskiljande, som viktiga) i variabelnamn, kan man använda fler och därmed göra programtexten tydligare.

Upprepade satser



"Upprepa utförandet av satser så länge villkoret är uppfyllt. Fortsätt sedan nedåt."

Med BASICs IF-sats och en programrad får vi nedanstående lösning:

```
xxxx IF villkor THEN
satser: GOTO xxxx
```

Den avslutande GOTO-satsen skall ALLTID vara riktad tillbaka till samma rad.

Om antalet satser i upprepningen är stort, räcker inte raden. Satserna kan då flyttas till ett subrutinblock och anropas från upprepningen med en GOSUN-sats:

Ex.:
1000 IF A<0 THEN GOSUB
2000: A=A+1:GOTO 1000

2000 REM
2010:RETURN

En annan möjlighet som ibland kan vara lämplig är att "vända villkoret" och skapa utrymme för många rader i anslutning till IF-satsen.

Ex.:
xxxx IF NOT villkor THEN yyyy
---- satser
----
----:GOTO xxxx
yyyy REM SLUT UPPREPNING

Observera den avslutande kommentarraden, som ger upprepningen en väldefinierad utgång.

FOR-satsen är ett specialfall av ovanstående generella modell. Den ger vi en egen grafisk beskrivning, där den stegvisa uppräknings tydligt framgår, och där likheten med den vanliga upprepningen finns kvar:



```
xxxx FOR var=st TO sl STEP steg
yyyy satser
---- .....
---- .....:NEXT
```

FOR-satsen är enkel och utan komplikationer eftersom den genom sin utformning har en ingång och en

forts. nästa sida

utgång. Vi kan använda en eller flera rader allt efter behov:

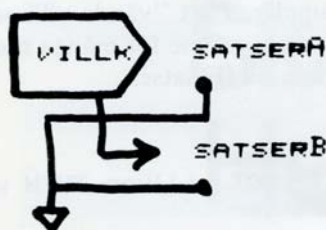
Ex.:

```
100 FOR I=1 TO 10:SUM=
SUM+TAL(I):NEXT
-----
200 FOR I=1 TO 1000
210 .....
220 .....
230 .....:NEXT
```

Väg-val

Denna struktur kräver stor disciplin. IF-satsen har den olyckliga formen en väg in och två ut. Låt oss studera två vanliga fall:

A) Grundformen



```
xxxx IF villkor THEN
satserA:GOTO zzzz
yyyy REM ELSE:satserB
-----
zzzz REM ENDIF
```

Bara en rad kan användas för IF,satserA och GOTO zzzz. Detta är ofta otillräckligt även om subrutinblock och GOSUB kan utnyttjas på samma sätt som för upprepningar.

För "ELSE"-delen och satserB finns inte denna begränsning. Konstruktionen avslutas med en REM ENDIF-sats till vilken THEN-radens GOTO är riktad.

Problemet med den ensamma "satserA-raden" kan lösas med en "vänd villkoret"-konstruktion:

```
xxxx IF NOT villkor THEN zzzz
yyyy satserA
-----:GOTO www
zzzz REM ELSE:satserB
-----
www REM ENDIF
```

Här kan både satserA och SatserB bestå av godtyckligt många rader.

B) "ELSE-delen" saknas



```
xxxx IF villkor THEN satser
```

Denna form blir enkel om en rad räcker. Är satserna många tvingas vi åter använda antingen ett särskilt subrutinblock eller ett "omvänt villkor":

```
xxxx IF NOT villkor THEN zzzz
yyyy satser
-----
zzzz REM ENDIF
```

SAMMANFATTNING: För upprepningen och vägvalsstrukturen har vi studerat dels enkla enradslösningar

dels något mer komplicerade flerradskonstruktioner med ett omvänt villkor. Det är viktigt att vi försöker begränsa oss till dessa former och inte för in varianter med två eller flera utgångar. Dessutom bör vi beskriva dessa former så att vi lätt kan skilja dem i vår programtext. Flerradskonstruktionen med vänt villkor inleds t ex med en IF-sats som efter THEN har ett radnummer. Den enkla enradslösningen har programsatser efter THEN.

I nästa avsnitt skall vi föra in ytterligare ett par strukturer som vi bör tillåta i våra program och därefter skall vi avsluta serien med ett exempel som skall belysa vägen från problem till färdigt program.

Lars-Erik Bengtsson

Video-chippet i VIC 64

Hur video-chippet arbetar

Någon detaljerad beskrivning av hur video-chippet utför sina uppgifter kommer inte här, men några saker måste klargöras innan vi kan börja bekanta oss med kontrollregistren. Video-chippet sköter sig till stor del självt, det hämtar uppgifter från minnet och uppdaterar bildskärmen, utan att behöva belasta processorn. Ibland måste det emellertid hämta lite extra information, och ett avbrott signaleras då till CPU:n. Det händer inte särskilt ofta, och tar alltså inte särskilt mycket tid. Men när maskinen kommer i tidsnöd, som t ex när kassettbandspelaren skall användas och signalerna måste time-as, då måste video-chippet stängas av tillfälligt för att CPU:n skall få tillräckligt med tid. Detta förklarar alltså varför skärmen blir blank då kassettenheten används.

En viktig sak som man bör känna till, är att video-chippet endast kan "se" 16K minne i taget. Detta betyder att all information hämtas från ett 16K-block, och data utanför detta kan inte nås. De två lägsta bitarna i adress

Som du säkert känner till, har VIC 64 stora möjligheter då det gäller att använda grafik på olika sätt. Men hur får man då tillgång till denna grafik? Det finns ju inga kommandon för att rita linjer, osv i BASIC. Jo, det är så att i 64:an finns ett video-chip (Video Interface Chip 6566) som man kommunicerar med genom ett antal kontrollregister som ligger på adresserna 53248-53294 (\$D000-\$D02E, hexadecimalt). I den här artikeln skall vi beskriva en del av dem, och ge exempel på hur de kan användas. Tabell 1 beskriver de aktuella registren och visar deras normalvärden (deras innehåll då datorn sätts på). De flesta registren går bra att kontrollera från BASIC, men en del tekniker (såsom raster-avbrott) kräver att man behärskar maskinspråksprogrammering. Om du inte gör det än, hoppa bara över de avsnitten. Vi har förutsatt att du känner till det binära talsystemet, om du inte gör det kan du lära dig i Assembler-skolan del 1 i förra numret av VIC-rapport. På det stora hela har vi försökt hålla det så enkelt som möjligt.

56576 (\$DD00) är två kontroll-linjer (VA14 och VA15) som bestämmer vilket block som skall användas. Följaktligen kan man styra detta genom att POKEa in olika värden i detta register. Tabell 2 visar vad de olika värdena ger för placering av 16K-blocket. En annan adress som kan vara bra att känna till är 648, som talar om för BASICen på vilken "sida" i minnet bildskärmsminnet börjar. (En sida är lika med 256 bytes.) Normalt innehåller denna adress 4, vilket betyder att bildskärmsminnet ligger på adress 1024 (4×256) och framåt.

Några kontrollregister

I flera av kontrollregistren används de olika bitarna till att kontrollera olika saker. De kommer därför inte att beskrivas i sekvensiell ordning, utan efter hur de används. När man ändrar enskilda biter i ett register för att kontrollera något, måste man se till att resten av bitarna i registret inte ändras. I tabell 1 listas normalvärdena för de enskilda bitarna i registren. Om man tar hänsyn till dessa värden när man räknar ut vilket värde som skall POKEas in blir det inga problem. Man kan också använda sig av OR och AND för att tända respektive släcka bitar. I de flesta exempel nedan har normalvärdena använts till att räkna ut det korrekta värdet. När du utför exemplen, titta då gärna själv på de aktuella registren för att se hur värdet räknats fram.

Display enable

Bit 4 i adress 53265 bestämmer om en bild skall visas på monitorn eller ej. Då biten är tänd (=1) visas bilden och då den är släckt (=0) visas inte bilden (ger samma resultat som när kassettbandspelaren används). Genom att subtrahera 16 från innehållet i denna adress släcks biten (och bildskärmen). Om du skriver POKE 53265,11 släcks bit 4 och du har plötsligt ingen bild längre. För att bilden skall visas igen skriver du POKE 53265,27 eller trycker på RUN/STOP och RESTORE.

Y-skroll, X-skroll, rader och kolumner

Bitarna 0-3 i 53265 och 53270 används till att kontrollera antalet rader respektive kolumner, samt skrollning i y- respektive x-led.

Bit 3 i 53265 bestämmer om skärmen skall bestå av 24 eller 25 rader. Om vi släcker biten genom att skriva POKE 53265,19 kommer hälften av den understa och översta raden att "klippas bort" och skärmen består nu av endast 24 rader. För att slippa ha en halv rad överst och underst på skärmen, kan bitarna 0-2 (y-skroll) justeras till att flytta upp eller ned bilden. POKE 53265,16 medför (förutom att bit 3 släcks- att y-skroll sätts till 0 och att bilden alltså flyttas upp så långt som möjligt. Resultatet blir då att vi nu har en bildskärm bestående av 24 "hela" rader. Y-skroll kan alltså användas till att flytta bilden vertikalt med små steg. Normalvärde för y-skroll är 3. Om man minskar det flyttas skärmen uppåt, och om man ökar det flyttas den nedåt. Program 1 (efter artikeln) är ett exempel på hur skärmen kan flyttas. Först kommer skärmen att flyttas snabbt nedåt, genom att y-skroll sätts till 7, sedan kommer den långsamt att flyttas uppåt genom att y-skroll minskas.

Bit 3 i 53270 bestämmer hur många kolumner som skärmen skall bestå av. Biten är normalt tänd, och skärmen innehåller då 40 kolumner. Om vi släcker biten genom t.ex. POKE 53270,0 kommer kolumnerna längst till vänster och längst till höger att försvinna, och skärmen består nu av 38 kolumner. Om bitarna 0-2 (x-skroll) ändras kommer skärmen att flyttas åt höger. Ett exempel på horisontell förflyttning visas i program 2. Programmet flyttar skärmen åt höger och sedan tillbaks igen. Det sker relativt snabbt så det kommer att se ut som skärmen studsade fram och tillbaka. Man kan lägga in en fördröjningsloop (t.ex. FOR J=1 TO 200: NEXT) på raderna 35 och 65 för att lugna ner det hela lite.

Genom att kombinera antalet rader och y-skroll samt antalet kolumner och x-skroll, kan man få fram många intressanta effekter. En rutin som skrollar skärmen mjukt uppåt med små steg skulle t.ex. kunna göras.

Chip reset

Bit 5 i 53270 används till att göra en reset på video-chippet. Detta görs då datorn slås på, och man behöver själv göra det själv. För att utföra en reset skall bit 5 först tändas och sedan släckas. Detta kan man göra med POKE 53265, 32: POKE 53265,8.

Skärm- och teckengenerator-placering

I adress 53272 finns två pekare som pekar till adresserna för bildskärmsminnet och teckengenerator inuti det 16K-block som video-chippet ser. Bitarna 4-7 respektive bitarna 0-3 multipliceras med 1024 och läggs till startadressen på blocket för att avgöra placeringen. För att förstå det lite bättre kan vi studera hur det ser ut när datorn slagits på. Det 16K-block som video-chippet då ser börjar på adress 0 i minnet. 53272 innehåller 21 (0001 0101 binärt). Bit 0 i denna adress används inte och kan visa lite olika. Den räknas emellertid alltid som 0. Adressen till bildminnet blir då $0 + 1 \times 1024$ vilket är lika med 1024 (samma adress som adress 648 pekar till för BASICen). Adressen till teckengeneratoren blir $0 + 4 \times 1024$ vilket är lika med 4096. Detta betyder alltså att video-chippet ser teckengeneratoren på den adressen. (För processorn ligger den fortfarande på adress \$D000. Krångligt? Det tycker vi också.)

För att visa hur man kan flytta på bildskärmsminnet kan vi som exempel flytta det till adress 32768 (\$8000) där det ligger i en PET-dator. Först måste vi då tala om för video-chippet att hämta data från ett annat 16K-block än det som startar på adress 0. Enligt tabell 2 skall bitarna 0-1 i adress 56576 då sättas till 1. Detta kan göras med POKE 56576,5 (vi måste ta hänsyn till bit 3 som skall vara tänd). Sedan måste vi tala om för BASICen var bildminnet finns. POKE 648,128 gör detta ($128 \times 256 = 32768$). Slutligen vill också video-chippet veta var bildskärmen finns i minnet någonstans. POKE 53272,4 betyder att bildskärmsminnet ligger på $32768 + 0 \times 1024$ (bitarna 4-7 är 0) vilket naturligtvis är lika med 32768. Teckengeneratoren finns nu på adress $32768 + 4 \times 1024$ vilket är lika

forts. nästa sida

med 36864, precis där den skall vara enligt tabell 2. När vi har gjort detta kommer skärmen antagligen att innehålla lite skräp. Vi kan rensa den med SHIFT CLR/HOME och se vårt verk fullbordat.

Bildskärmsminnet börjar nu på 32768, vi kan kontrollera detta t.ex. genom att POKEa in något direkt i skärmen (en färg måste då också POKEas in i färgminnet som ligger kvar på adress 55296).

Genom att ändra pekaren till teckengeneratoren i adress 53272 (bitarna 0-4) kan man byta mellan stora bokstäver/grafiska tecken och små/stora bokstäver. Om denna pekare ökas med 2, betyder det att startadressen för teckengeneratoren ökats med 2*1024 och nu pekar på den alternativa teckenuppsättningen istället för den normala. Följande rad belyser detta: 10 POKE 53272,23 : POKE 53272,21 : GOTO 10. När du skrivit in raden och skrivit RUN, kommer texten på skärmen att alternera mellan stora och små bokstäver.

Extended color

Genom att tända bit 5 i 53265 kopplar man på extended color. Vanligtvis har alla tecken på skärmen en gemensam bakgrundsfärg, som lagras i 53281 (\$D021). Genom att använda Extended color kan vi emellertid samtidigt använda upp till fyra olika bakgrundsfärger på skärmen. Dessa lagras i registren 53281 (\$D021) - 53284 (\$D024). Vi kan t.ex. ha ett tecken med vit färg på svart bakgrund medan övriga delar av skärmen har röd bakgrundsfärg.

Man får inget gratis här i världen, utan för att få tillgång till dessa färger måste vi släppa ifrån oss 75% av teckengeneratoren. Vi får behålla skärmkoderna 0-63 i fyra upplagor med var sin bakgrundsfärg. M.a.o. kommer bitarna 0-5 i skärmkoden att bestämma tecknet och bitarna 6-7 bakgrundsfärgen.

Bakgrundsfärgen bestäms enligt följande:

Bit 6 & 7:	Färgregister:
00	53281 (\$D021)
01	53282 (\$D022)
10	53283 (\$D023)
11	53284 (\$D024)

Det kan vara lite problematiskt att utnyttja alla möjligheter i PRINT-satser. Ett tips är att tecken i reverse ger bit-mönstret 10, vilket motsvarar bakgrundsfärg enligt 53282 (\$D023).

Program 3 är en demonstration av några möjligheter med Extended color. Rad 10 bestämmer ramens färg samt de fyra bakgrundsfärgerna. Rad 20 kopplar på Extended color och tömmer skärmen. Rad 30-60 "pokar" in i bildskärmsminnet och färgminnet fyra rader text med olika bakgrundsfärger och teckenfärger. Prova och se möjligheterna!

Multicolor

Varje teckenposition på skärmen kan innehålla två färger, nämligen teckenfärgen och bakgrundsfärgen. I en 8x8 matris, motsvarande ett tecken, motsvarar en etta teckenfärgen och en nolla bakgrundsfärgen. Det finns inga möjligheter att med de givna resurserna definiera fler färger per tecken och samtidigt behålla samma upplösning. Om vi minskade upplösningen i x-led, så att två punkter (bitar) bredvid varandra resulterar i en bred punkt med enhetlig färg, skall vi genom att variera denna två-bitars kombination kunna definiera fyra färger i stället för två.

Eftersom vi inte har tre-dubbelt färgminne till förfogande kan vi inte definiera olika extra-färger för varje tecken, utan vi får använda gemensamma extra-färger för hela skärmen, liksom bakgrundsfärgen är gemensam för hela skärmen (undantag finns i samband med extended color eller raster-interrupt). Detta sätt att få fler färger kallas multicolor och sätts på genom att sätta bit 4 i 53270 (\$D016). I Basic skulle detta kunna åstadkommas på följande vis: POKE 53270, PEEK(53270) OR 16.

Även när vi har multicolor kan vi använda tecken med normal upplösning. Vi väljer för varje tecken om det skall vara i multicolor eller ej, genom att sätta bit 3 i färgminnet eller ej. Rent praktiskt innebär detta färgkoderna 0-7, som fås tillsammans med CTRL-tangenten, blir vanliga tecken och färgkoderna 8-15, som fås tillsammans med LOGO-tangenten, blir i multicolor, men färgerna blir ej 8-15 utan 0-7. Vi får alltså två uppsättningar av teckenfärgerna 0-7, där den ena

blir i multicolor. Färgkodsbitarna 0-2 styr alltså färgen och bit 3 multicolor eller ej. Denna restriktion i färganvändning gäller endast för teckenfärgerna.

Kombinationen av de två bitarna kommer att bestämma den gemensamma färgen enligt följande:

Kombination:	Färgregister:
00	53281 (\$D021)
01	53282 (\$D022)
10	53283 (\$D023)
11	teckenfärg

Användning av multicolor på vanlig textskärm är ej så användbart, såvida inte omdefinierade tecken används. Tecknen bör vara utformade för användning i multicolor, för att någon större effekt skall uppnås.

Multicolor i högupplösningsgrafik (bit-map) fungerar på likartat sätt med upplösningsbegränsning, men här kan extrafärgerna definieras för varje enskild 8x8 matris. Hela skärmen är i multicolor (utan undantag), så alla färger kan varieras i sexton kombinationer.

Färgerna till de olika bitkombinationerna i högupplösningsgrafik kan härledas enligt följande:

Kombination:	Färginformation:
00	53281 (\$D021)
01	grafikfärgminne bit 4-7
10	grafikfärgminne bit 0-3
11	textskärmsfärgminne (bit 0-3)

Även sprites kan ha multicolor, men det kommer vi att behandla i ett senare nummer.

Högupplösningsgrafik

Varje tecken på skärmen består av 8*8 punkter. Om vi t.ex. skriver ut ett reverse mellanslag (som fyller ut ett helt tecken), kan vi räkna antalet rader och kolumner till 8. Detta betyder att skärmen består av totalt 8*8*1000=64000 punkter. Genom att använda de olika grafiska tecken som finns, kan man i begränsad utsträckning bestämma vilka av alla dessa punkter som skall vara tända, och vilken färg de skall ha. Det är dock inte alltid som de grafiska tecknen räcker till för att skapa de bilder man har

tänkt sig. Det finns då ett annat sätt. Genom att använda högupplösningsgrafik, s.k. bit-map, kan man tända och släcka alla 64000 punkterna var för sig, och följaktligen konstruera mycket detaljerade bilder.

Man kopplar på bit-map genom att tända bit 5 i adress 53265. När man kopplar på bit-map händer emellertid en del speciella saker. Pekaren till bildskärmen och teckengeneratoren (adress 53272, som vi tidigare gått igenom) fungerar nu inte på samma sätt som förut. Bitarna 4-7, som tidigare pekade till bildskärmen, pekar nu istället till bit-map:s färgminne (mer om det senare), och bitarna 0-3 pekar på den minnesarea i vilken man tänder och släcker punkter (bit 0 räknas fortfarande alltid som 0), bit-map-arean. För att få fram adresserna till respektive areor gör vi som förut, multiplicerar pekarna med 1024 och lägger till resultatet till startadressen av det 16K-block som video-chippet ser.

Bit-map-arean består av 8000 bytes, och eftersom varje byte består av 8 bitar består arean av totalt 64000 bitar. Genom att lägga in olika värden i dessa bytes, och därigenom tända och släcka de olika bitarna i dem, tänder och släcker man de olika punkterna på skärmen (därför kallas det bit-map). De 8000 byte:n ligger lagrade som i teckengeneratoren, dvs 8 byte/tecken (skärmen består som bekant av 1000 tecken). De 8 första byte:n ligger i första tecknet på översta raden, nästa 8 byte ligger i andra tecknet, osv. Att de ligger lagrade på detta sätt medför naturligtvis att det kan vara krångligt att t.ex. dra diagonala streck över skärmen. Det kan man dock lösa genom att skriva en rutin som från en x- och y-koordinat räknar fram vilken bit i vilken byte som skall tändas och tänder denna.

Färgminnet som vi hänvisade till ovan, består av 1000 bytes som bestämmer färger för motsvarande tecken på skärmen. Bitarna 0-3 anger bakgrundsfärg för tecknet, och bitarna 4-7 anger punkternas färg. Alla punkter inom ett och samma tecken har alltså alltid samma färg.

Om vi nu skall använda bit-map räcker det inte att tända bit 5 i 53265. Vi måste först bestämma var bit-map-arean skall placeras. Efter-

som denna area tillsammans med färgminnet nu upptar 9K måste vi flytta upp den i minnet. Video-chippets 16K-block måste då först flyttas. Om vi bestämmer oss för att lägga blocket med start på adress 16384, gör vi detta med POKE 53265,6 (se tabell 2). Bit-map-arean måste ligga i antingen nedre eller övre delen av blocket, så bitarna 0-3 i 53272 skall vara 8 eller 0. POKE 53272,128 lägger bit-map-arean med början på 16384, och färgminnet på 24576. Program 4 är ett exempel på hur man kan använda bit-map. En beskrivning av hur programmet fungerar följer.

Programbeskrivning program 4

100-120

Det 16K-block video-chippet ser sätts till att starta på 16384, bit-map på (bit 5 i 53265 tänds), och pekare till bit-map-minne samt färgminne sätts.

170-220

Bit-map-minnet rensas genom att fylla det med nollor (alla bitarna sätts till 0), och färger i färgminnet sätts (lika för alla tecken). Bakgrundsfärg sätts till blå och punkternas färg till ljusblå.

250-310

En cirkel ritas med hjälp av SINus och COSinus. Eftersom resultaten av dessa funktioner varierar mellan -1 och 1, måste X- och Y-värdena anpassas till skärmen.

340-390

Ensinus-kurva plottas. X divideras med 25 för att få med lämplig del av kurvan.

420-470

Först nollas koordinaterna för förra satta punkten (X0 och Y0). Sedan väljs slumpmässigt X och Y ut, och en linje ritas från förra punkten och dit. Detta håller på tills man trycker på STOP.

1000-1040

Subrutin för att plotta en punkt. X och Y antas variera från 0 till 319 respektive 0 till 199. (0,0) Är punkten i nedre vänstra hörnet av skärmen. Med dessa värden räknas ut vilken adress som skall POKEas i och vilken kod som skall ORas in. X0 och Y0 sätts till X respektive Y för att användas i nästa subrutin.

2000-2070

Subrutin för att dra en linje från den senast plottade punkten (koordinaterna X0 och Y0) till X, Y. Längden räknas ut med hjälp av Pythagoras sats, och stegvärdena i X- och Y-riktningen räknas ut. Den förra subrutinen anropas för att plotta punkterna.

Kommentarer:

Beskrivningen ovan beskriver inte i detalj hur alla delar av programmet fungerar. Om du vill förstå det bättre kan du "torrköra" det, dvs gå igenom det med papper och penna och utföra beräkningarna för hand.

Tomrader (rader med enbart ett radnummer) fås genom att man på raden efter radnummret skriver ett grafiskt tecken, ett mellanslag och ytterligare ett grafiskt tecken, och sedan slår return. De grafiska tecknen kommer att ignoreras, och raden innehåller då endast ett mellanslag som inte kommer att påverka något när programmet körs.

Efter GOSUB xxxx kan man skriva en kommentar utan att behöva använda REM. BASICen bryr sig nämligen inte om vad som står på resten av raden om inte xxxx följs av ett kolon.

Högupplösningsgrafik i maskinspråk

Program 5 är en demonstration av hur bit-mapp kan användas i maskinspråk. Programmet kommer att koppla på högupplösningsgrafik och rita 7 diagonala streck på skärmen. Syftet med programmet är främst att visa hur man i maskinspråk kan utnyttja bit-map, och att fungera som inspirationskälla till dina egna program.

Programbeskrivning:

Programmet består av fyra delar (GRAFIK, RENZA, FÄRG och RITA) samt en uppstartningsrutin (START), som anropar de övriga.

GRAFIK: Rutinen kopplar på högupplösningsgrafik. Först bestäms var grafikminnet skall läggas. Bitarna 0 & 1 i \$DD00 bestämmer lokaliseringen. Vi sätter dem till 10, vilket motsvarar \$4000. Sedan bestäms var färgminnet skall läggas. Vi lägger det på offset (\$4000) + \$2000, genom att

forts. nästa sida

lägga \$80 i \$D018. Slutligen sätter vi bit 5 i \$D011, vilket kopplar på bit-map.

RENSA: Rutinen tömmer grafikminnet. Minnesarean \$4000-\$5FFF fylls med \$00.

FÄRG: Färgminnet på adress \$6000-\$63FF fylls med ackumulatorns värde vid anropet. Värdet motsvarar bakgrundsfärg + punktfärg x \$10.

RITA: Rutinen drar 7 st diagonala streck, i 45 graders vinkel, tvärs över skärmen. Först beräknas startoffset för linjen. Detta förskjuts 8 bytes för varje linje. Offsetvärdet lagras i \$C1 & \$C2. Efter detta fylls första 8x8 matrisen med ett diagonalt streck. Detta sker genom att bit 7 i ackumulatorn sätts och sedan shiftas ackumulatorn ett steg för varje rad på skärmen tills bit 0 sätts.

Offsetvärdet ökas med \$0148 (\$08 bitrader x \$29 positioner). Med andra ord förflyttar vi oss en 8x8 matris nedåt och en 8x8 matris åt höger. Sedan görs kontroll om nedersta raden är nådd. Om inte så är fallet fortsätter vi att rita på strecket. Sist räknas antal streck ned och om sista strecket är färdigt avslutas rutinen.

Raster Interrupt

Till vår TV eller monitor genererar video-chipet en videosignal, som innehåller informationen till bilden. Hela bilden överförs inte på en gång, utan bilden överförs punkt för punkt till monitorn. Till varje punkt skall också dess färg överföras, vilket sker tillsammans med den övriga informationen. Punkterna överförs från vänster till höger, rad för rad, från översta till nedersta raden. När hela bilden överförs börjar förloppet om på nytt. 50 ggr i sekunden (50 Hz) uppdateras bilden till monitorn. Video-chipet låter oss ta del av vilken rad på skärmen som för tillfället uppdateras, genom att efter varje avslutad rad inkrementera (öka med ett) ett register. Detta register kallas rasterregister och har adressen \$D012 samt bit 7 i \$D011. Genom att studera detta register kan vi ta reda på vilken del av skärmen som uppdateras vid en viss tidpunkt.

Skärmen innehåller 200 rader (bit-rader, ett tecken är åtta bit-rader högt) och ramen runt skärmen har c:a

50 rader både över och under skärmen. Totalt överför då c:a 300 "synliga" rader till monitorn. Ett register om åtta bitar skulle alltså inte räcka till för 300 rader ($2^8=256$). En extra bit blir behövlig och därför används bit 7 i \$D011 som mest signifikant bit.

Raster-registret kommer att ändra värde väldigt ofta, så en förutsättning för att ha någon användning av det är att programmera i maskinspråk. Basic är på tok för långsamt för sådana här ändamål.

Vad har jag då för nytta utav raster-registret? Att hela tiden läsa av raster-registret för att se var "elektronstrålen" befinner sig är ju väldigt tidsödande. Är man intresserad av när en exakt rad uppdateras kommer processorn inte att hinna göra mycket annat än att läsa av raster-registret, trots att vi skriver i maskinspråk.

Nu finns det i och för sig en smartare lösning till problemet. I stället för att vi skall förbruka vår tid till att läsa av raster-registret och jämföra med vår önskade rad, så kan vi säga till video-chipet vilken rad vi är intresserade av och sedan låta video-chipet säga till oss när tiden är inne. På så vis kan processorn arbeta ostört ända tills video-chipet säger till. Processorn får ett s.k. IRQ (interrupt request), dvs. avbrottsförfrågan, av video-chipet. Det är detta vi kallar för raster-interrupt.

När processorn får ett IRQ avbryter den det den för tillfället är upptagen av och lägger undan alla interna register på stacken. Sedan fortsätter den att exekvera från den adress, som står i \$FFFF. Denna rutin gör i sin tur ett hopp till den adress, som lagras i IRQ-vektorn (\$0314). Senare förklaras hur denna vektor kan ändras för att anpassas till bl.a. raster-interrupt (return from interrupt), och fortsätter sedan med det den gjorde innan avbrottet.

Avbrottsbehandling används redan av operativsystemet till bl.a. tangentbordsavkodning, klock-uppdatering och cursorblinkning. En timer i CIA 1 genererar ett avbrott till processorn 60 ggr i sekunden.

Program 6 visar exempel på hur raster-interrupt kan användas. Programmet listas som en assemblerlistning, och dels som ett BASIC-program, som laddar in maskinkodspro-

grammet från DATA-satser. Programmet kommer att dela av skärmen vid två specifika rader så att bakgrundsfärgen ändras på ett parti i mitten. Programmet används som en avbrottsrutin, så att datorn blir ledig för övrigt till att exekvera andra program etc. Andra användningsområden än att byta färg skulle t.ex. vara att mixa högupplösningsgrafik (bit-map) med vanlig textskärm eller att byta sprite-register, så att fler än åtta sprites kan synas samtidigt på skärmen. Kommande nummer av VIC-rapport kommer att behandla mer om sprites och dess användning.

Programbeskrivning:

Den första delen av programmet är en inkopplingsrutin för avbrottsrutinen, vilken först ändrar IRQ-vektorn (\$0314) till att peka på vår nya avbrottsrutin. För säkerhet skall stänger vi av IRQ med SEI (set interrupt disable status) ifall ett sådant skulle genereras när vi ändrar IRQ-vektorn. Nästa steg är att sätta på raster-interrupt med att sätta bit 0: \$S01A, Interrupt Enable. Sedan stänger vi av det vanliga IRQ:et genom att lagra \$7F i CIA 1's register \$DC0D.

Nu är vår nya avbrottsrutin inkopplad och den vanliga urkopplad, så vid nästa avbrott vet vi att det är frågan om raster-avbrott. Det kan tyckas farligt att stänga av det vanliga IRQ:et, men det kommer vi att anropa från vår nya avbrottsrutin. Eftersom vi både måste sätta på och stänga av den nya bakgrundsfärgen 50 ggr i sekunden kommer vår avbrottsrutin att exekveras 100 ggr i sekunden. Den vanliga avbrottsrutinen skall exekveras 60 ggr i sekunden för att klockan skall gå rätt och eftersom timern som beräknar tidsintervallen fortfarande går, kan vi efter vår avbrottsrutin se efter om det är tid för vanligt IRQ, och i så fall exekvera det. En annan variant hade varit att låta båda IRQ:na användas parallellt, men då blir våra brytlinjer något instabila, emedan raster-IRQ kan få vänta på vanligt IRQ, om det senare exekveras när det är tid för raster-IRQ.

Åter till programmet. Det första den nya avbrottsrutinen gör är att ladda in ett register med IRQ-flaggor, \$D019. Detta register talar om vad för sorts IRQ det är frågan om, men eftersom endast raster-IRQ är på-

kopplat behöver vi ej verifiera detta. Sedan lägger vi tillbaka samma värde för att stänga av flaggorna. Observera att en flagga släcks i detta register genom att försöka sätta den!

Register X och Y laddas med nästa avbrottsrad respektive bakgrunds-färg. Kontroll görs om bit 7; \$D012 är satt, vilket skulle innebära att nuvarande rad är lägre än rad 127 och att nästa avbrottsrad blir den "högre" av de båda. Den nya avbrottsraden läggs in i raster-registret (\$D011 & \$D012). Det värde vi lägger i \$D011 kommer alltid att bli samma, emedan raster-registrets åttonde bit aldrig blir 1. Vi lägger därför in det värde, som vid normala betingelser är \$1B, i \$D011. Observera att vi måste skriva i detta register, ehuru det aldrig ändras, för att video-chippet skall acceptera det nya raster-värdet.

Nu kontrollerar vi om det är dags för vanligt IRQ. Bit 0 i CIA 1's register \$DC0D sätts när tiden är inne och då hoppar vi vidare till den vanliga avbrottsrutinen på adress SEA31. I annat fall avslutar vi avbrottet med att hoppa till \$FEBC, där processorn hämtar tillbaka sina interna register från stacken och utför ett RTI (return from interrupt), så att den kan fortsätta exekveringen där den var innan avbrottet.

Per Andersson
Martin Fredriksson

Program 1

```
10 FOR I=0 TO 7
20 : POKE 53255, 24 OR I
30 : FOR J=1 TO 500: NEXT
40 NEXT
```

Program 2

```
10 P=53270
20 FOR I=0 TO 7
30 : POKE P, 200 OR I
40 NEXT
50 FOR I=7 TO 0 STEP -1
60 : POKE P, 200 OR I
70 NEXT
80 GOTO 20
```

Program 3

```
10 POKE 53280,0: POKE 53281,0
20 POKE 53282,6: POKE 53283,2: POKE 53284,11
30 FOR A=0 TO 3
40 : FOR B=0 TO 39
50 : POKE 1424+A*40+B,A*64+B
60 : POKE 55696+A*40+B,(B AND 15)
70 : NEXT
80 NEXT
```

Program 4

```
100 POKE 56576, PEEK(56576) AND 252 OR 2
110 POKE 53265, PEEK(53265) OR 32
120 POKE 53272, 128
130
140 BM = 16384: REM START BIT MAP MINNE
150 FM = 24576: REM START FARG-MINNE
160
170 FOR I=BM TO BM+8000: REM RENSA BIT MAP MINNET
180 : POKE I,0
190 NEXT
200 FOR I=FM TO FM+1000: REM SATT FARGER
210 : POKE I,14*16+6
220 NEXT
230
240
250 REM RITA CIRKEL
260
270 FOR A=0 TO 2*PI STEP .04
280 : X=INT(COS(A)*80+159)
290 : Y=INT(SIN(A)*80+99)
300 : GOSUB 1000 PLOT X,Y
310 NEXT
320
330
340 REM PLOTTA SINUS-KURVA
350
360 FOR X=0 TO 319
370 : Y=99+INT(SIN(X/25)*90)
380 : GOSUB 1000 PLOT X,Y
390 NEXT
400
410
420 REM DRA LINJER SLUMPMASSIGT
430
440 X0=0: Y0=0
450 X=INT(RND(1)*320): Y=INT(RND(1)*200)
460 GOSUB 2000 DRA LINJE
470 GOTO 450
480
490 END
980
990
1000 REM --- PLOT X,Y ---
1010 ADR=BM+8*INT(X/8)+320*INT((199-Y)/8)+((199-Y) AND 7)
1020 POKE ADR, PEEK(ADR) OR 2+(7-(X AND 7))
1030 X0=X: Y0=Y: REM SATT STARTPUNKT FÖR ATT DRA LINJER
1040 RETURN
1050
1060
1070 REM --- DRA LINJE TILL X,Y ---
2010 DX=X-X0: DY=Y-Y0: L=SQR(DX*DX+DY*DY) IF L=0 THEN L=1
2020 XS=DX/L: YS=DY/L: X=X0: Y=Y0
2030 FOR I=1 TO L
2040 : GOSUB 1000
2050 : X=X+XS: Y=Y+YS
2060 NEXT
2070 RETURN
```

forts. nästa sida

Program 5

```

0000          BEGIN = $3000          ;PROGRAMSTART
0000          BITMAP = $4000
0000          COLOR = $E000

0000          * = BEGIN

3000 20 0F 30  START JSR GRAFIK      ;KOPPLA PA HOGGRAFIK
3003 20 27 30  JSR RENSA          ;TOM GRAFIKMINNET
3006 A9 05      LDA #5            ;GRÖN BAKGRUND, SVARTA PUNKTER
3008 20 3D 30  JSR FARG          ;SATT FARGMINNET
300B 20 54 30  JSR RITA          ;RITA I HOGGRAFIK
300E 60        RTS

300F AD 00 DD  GRAFIK LDA #DD00
3012 29 FC      AND #FFC          ;GRAFIKMINNET LAGGS
3014 09 02      ORA #02          ;PA $4000 - $5FFF
3016 8D 00 DD  STA #DD00
3019 A9 80      LDA #80          ;FARGMINNE VID $6000
301B 8D 18 D0  STA #D018
301E AD 11 D0  LDA #D011
3021 09 20      ORA #X00100000   ;KOPPLA PA BITMAP
3023 8D 11 D0  STA #D011
3026 60        RTS

3027 A9 40      RENSA LDA #D011
3029 85 C2      STA #C2
302B A9 00      LDA #C01
302D 85 C1      STA #C1
302F A8        TAY
3030 A2 20      LDX #32
3032 91 C1      RENSA1 STA (#C1),Y
3034 C8        INY
3035 D0 FB      BNE RENSA1
3037 E6 C2      INC #C2
3039 CA        DEX
303A D0 F6      BNE RENSA1
303C 60        RTS          ;OKA HOGBYTE
                              ;SISTA SIDAN?
                              ;NEJ -> FORTSATT

303D A2 60      FARG LDX #D0COLOR
303F 86 C2      STX #C2
3041 A2 00      LDX #C0COLOR
3043 86 C1      STX #C1
3045 A0 00      LDY #0
3047 A2 04      LDX #4
3049 91 C1      FARG1 STA (#C1),Y
304B C8        INY
304C D0 FB      BNE FARG1
304E E6 C2      INC #C2
3050 CA        DEX
3051 D0 F6      BNE FARG1
3053 60        RTS          ;OKA HOGBYTE
                              ;SISTA SIDAN?
                              ;NEJ -> FORTSATT

3054 A2 07      RITA LDX #7
3056 8A        NASTA TXA
3057 0A        ASL A
3058 0A        ASL A
3059 0A        ASL A
305A 85 C1      STA #C1
305C A9 40      LDA #D011
305E 85 C2      STA #C2
3060 A0 00      RITA1 LDY #0
3062 A9 80      LDA #X10000000
3064 91 C1      RITA2 STA (#C1),Y
3066 C8        INY
3067 4A        LSR A
3068 90 FA      BCC RITA2
306A 18        CLC
306B A9 48      LDA #C2
306D 65 C1      ADC #C1
306F 85 C1      STA #C1
3071 A9 01      LDA #1
3073 65 C2      ADC #C2
3075 85 C2      STA #C2
3077 C9 5F      CMP #5F
3079 90 E5      BCC RITA1
307B CA        DEX
307C D0 D8      BNE NASTA
307E 60        RTS          ;7 STRECK
                              ;OFFSET FOR STRECK

                              ;NOLLSTALL RAKNARE
                              ;SATT BIT 7

                              ;SHIFTA BITEN AT HÖGER
                              ;EJ FÄRDIG!

                              ;OKA OFFSET TILL
                              ;NASTA 8-BITARS RAD

                              ;NEDERSTA RADEN?
                              ;NEJ -> FORTSATT
                              ;SISTA STRECKET?
                              ;NEJ -> NASTA STRECK

```


Tabell 1

	7	6	5	4	3	2	1	0
53265 D011		Extended Color	Bit Map	Display Enable	Row Select	Y-scroll		
Normalt	0	0	0	1	1	0	1	1
53266 D012	Raster Register							
53270 D016			Reset	Multi Color	Column Select	X-scroll		
Normalt	1	1	0	0	1	0	0	0
53272 D018	Screen				Character Base			
53273 D019	IRQ	Interrupt Sense			Light Pen	SSC	SBC	Raster
53274 D01A	Interrupt Enable				Light Pen	SSC	SBC	Raster

Tabell 2

Bit 0-1	RAM-area	Teckengenerator
0 (00)	49152-65535	-----
1 (01)	32768-36863, 40960-49151	36864-40959
2 (10)	16384-32767	-----
3 (11)	0-4095, 8192-16383	4096-8191

```

Program 6a 0000          BEGIN = #0030          ;KASSETT-BUFFERTEN
            0000          CINV = #0314          ;IRQ-VEKTOR
            0000          ICR = #DC0D          ;IRQ-KONTROLL-REGISTER
            0000          IRQDEF = #EA31        ;DEFAULT ENTRY FOR IRQ
            0000          IRQEXT = #FEBC        ;EXIT FROM IRQ

            0000          * = BEGIN

            0030 78          SEI                  ;LAGRA NY IRQ-VEKTOR
            003D A9 53          LDA #START
            003F 8D 14 03          STA CINV
            0042 A9 03          LDA #START
            0044 8D 15 03          STA CINV+1
            0047 58          CLI
            0048 A9 01          LDA #01          ;KOPPLA PA RASTER-IRQ
            004A 8D 1A D0          STA #D01A
            004D A9 7F          LDA #7F          ;KOPPLA UR VANLIG IRQ
            004F 8D 0D DC          STA ICR
            0052 60          RTS
            0053 AD 19 D0          START LDA #D019          ;LADDA IRQ-FLAGGOR
            0056 8D 19 D0          STA #D019          ;SLACK IRQ-FLAGGOR
            0059 A2 96          LDX #150          ;RADNR FOR MITTEN AV SKARMEN
            005B A0 05          LDY #5          ;BAKGRUNDSFARG = GRON
            005D AD 12 D0          LDA #D012          ;AR VI PA MITTEN REDAN?
            0060 10 04          BPL STOVAL          ;NEJ, HOPPA OVER
            0062 A2 00          LDX #0          ;RADNR FOR BORJAN AV SKARMEN
            0064 A0 01          LDY #1          ;BAKGRUNDSFARG = VIT
            0066 A9 1B          STOVAL LDA #00011011 ;NORMALVARDE FOR #D011
            0068 8D 11 D0          STA #D011          ;SATT RASTERVARDE BIT 8
            006B 8E 12 D0          STX #D012          ;SATT RASTERVARDE BIT 0-7
            006E 8C 21 D0          STY #D021          ;SATT BAKGRUNDSFARG
            0071 AD 0D DC          LDA ICR          ;LAS DEN VANLIGA IRQ-FLAGGAN
            0074 29 01          AND #01          ;MASKA UR FLAGGAN
            0076 F0 03          BEQ EXIT          ;EJ SATT -> AVSLUTA
            0078 4C 31 EA          JMP IRQDEF          ;SATT -> UTFOR VANLIGT IRQ
            007B 4C BC FE          EXIT JMP IRQEXT          ;AVSLUTA

```

```

Program 6b 10 FOR I=828 TO 893
            20 : READ A: POKE I,A
            30 NEXT
            40
            50 SYS 828
            60
            200 DATA 120,169,83,141,20,3,169,3,141,21,3,88,169,1,141,26,208
            210 DATA 169,127,141,13,220,96,173,25,208,141,25,208,162,150,160,5
            220 DATA 173,18,208,16,4,162,0,160,1,169,27,141,17,208,142,18
            230 DATA 208,140,33,208,173,13,220,41,1,240,3,76,49,234,76,188,254

```


Det är vi som

Alingsås: Alingsås HiFi-Consult AB 0322 366 66, Jannes Radio-TV 0322 107 58 • **Alvesta:** Görans Radio-TV AB 0472 127 88 • **Arboga:** Arboga Radio-TV AB 0589 104 15 • **Arvidsjaur:** AB Brännströms Radio TV 0960 12073 • **Arvika:** Magnusson Radio TV AB 0570 109 85
• **Askersund:** Åhlins Radio & TV AB 0583 113 00-01 • **Bandhagen:** Bild & Ljud 08 86 68 10 • **Bjuv:** Werners Radio TV AB 042 709 77 • **Boden:** Bodens Ljus & Kraft AB 0921 160 00 • **Boden:** Oves Radio & TV 0921 190 46 • **Bollnäs:** Oves Radio TV AB 0278 164 10 • **Borlänge:** Karl Larsson Musikhandel AB 0243 102 73 • **Borås:** Televisionen Andersson & Berndtsson AB 033 13 72 80 • **Braås:** Åseda Radio TV AB 0474 302 92 • **Bromölla:** Franssons Radio TV AB 0456 274 51 • **Båstad:** Wennströms Radio TV AB 0431 705 53 • **Danderyd:** FOKUS Ljud & Bild 08 753 15 05 • **Degerfors:** (Greeks) Elvaruhus 0586 421 25 • **Eksjö:** Gunders HiFi & TV AB 0381 151 80 • **Enköping:** Enköpings Sound 0171 391 20 • **Eskilstuna:** Bonus Hämtköp 016 14 14 10 • **Eskilstuna:** Studio Sound AB 016 11 56 11 • **Eslöv:** Billinge Radio & TV AB 0413 188 00 • **Falkenberg:** Radio & TV Service 0346 112 34 • **Farsta:** Sköndals TV-Foto AB 08 94 15 10 • **Finspång:** Stig Perssons Radio & TV AB 0122 149 60 • **Gislaved:** Bild & Ljud AB 0371 140 30 • **Grums:** Grums TV-Service 0555 118 18 • **Gällivare:** Elvaruhuset Erikssons Eltjänst AB 0970 140 90 • **Göteborg:** Durehed Radio TV 031 15 12 77, Karrins 031 14 64 36, L.C. Radio & TV-Salong AB 031 46 08 12 • **Ljudkällarn Musiclovers** AB 031 17 84 45, **REKORD TV & Radio** AB 031 12 71 80, **Ågrens HiFi** AB 031 81 01 95, **Änäs Radio** TV AB 031 15 53 33 • **Habo:** Mullsjö El AB 0392 408 80 • **Hallsberg:** Allanssons Radio TV AB 0582 107 08 • **Halmstad:** Cronwalls Radio & TV 035 11 05 57 • **Hedemora:** Alfs Radio & TV AB 0225 135 81 • **Helsingborg:** Uno Hallbäck AB 042 11 90 01 • **Musik & Radiocity i Hbg** AB 042 12 75 20, **Sjunnes Radio-TV** AB 042 14 44 55 • **Herrljunga:** Radioservice AB 0513 104 04 • **Hisings Backa:** Trimex Radio TV Fabricenta AB 031 23 00 00 • **Hudiksvall:** Hälsinge Radio 0650 103 88 • **Hultsfred:** Edvinssons Cykel Radio-TV 0495 102 25 • **Hyltebruk:** Hylte Radio TV 0345 109 48 • **Hägersten:** Norrmalms Radio AB 08 45 13 00 • **Hässleholm:** A-Butiken AB 0451 821 51 • **Höganäs:** Höganäs Radio & TV Service 042 406 26 • **Hörby:** Wilsson Radio TV AB 0415 123 67 • **Järfälla:** Kallhälls El. AB 0758 511 74 • **Jönköping:** TV-Centrum 036 16 23 75, **US Radio** TV 036 16 40 85, 16 70 90 • **Kalix:** Elektro-Steinwall AB 0923 122 10 • **Karlskoga:** Elbolaget 0586 509 70 • **Karlskrona:** BL Radio TV HB 0455 107 95 • **Karlstad:** Ströms Radio-TV 054 11 83 67, 11 15 89 • **Kinna:** Pehrssons Radio-TV AB 0320 101 20 • **Klippan:** Görans Radio & TV Klippan AB 0435 131 67 • **Kristianstad:** Centralservice Radio TV 044 12 00 71 • **Kungsbacka:** EL-BE HiFi 0300 124 69 • **Kävlinge:** Nilsson & Son Radio TV AB 046 73 04 55 • **Köping:** Epiteone HiFi AB 0221 215 10, 162 60 • **Laholm:** AB Laholm Radioaffär 0430 105 55 • **Lerum:** Lerums Radio & TV-Service 0302 123 25 • **Lidköping:** Lidköpings Radio & TV 0510 609 50 • **Linköping:** B Gustavssons Radio TV AB 013 13 89 20, **Bra Ljud & Bild** AB 013 12 25 75, **Linköpings Radioaffär** 013 12 49 41 • **Ljungby:** AB Bergs Eftr 0372 101 13 • **Ludvika:** Ludvika Radio & Motor AB 0240 100 51, 142 05 • **Luleå:** Jakobs Radio & TV AB 0920 121 16 • **Lund:** Record Radio TV AB 046 11 16 36 • **Lundevärv:** Ådalens TV-Service 0612 326 20 • **Lycksele:** Ulf Larsson Radio-TV AB 0950 111 40 • **Malmö:** Grip Radio AB 040 11 17 17, **Stereohörnan** 040 91 10 30, **TE-VE Radio i Malmö** AB 040 630 92 • **Markaryd:** Rubin Johansson & Co AB 0433 110 25 • **Mellerud:** Herbert Anderssons Eftr. 0530 100 12 • **Mjölby:** AB Mjölby Radioservice 0142 111 78 • **Motala:** Radiocentralen i Motala AB 0141 109 27 • **Mullsjö:** Mullsjö El AB 0392 111 31 • **Mölnlycke:** Mölnlycke Radio TV 031 88 37 54 • **Mörbylånga:** TV-Lasses i Mörbylånga AB 0485 407 06 • **Nordmaling:** Tonex Radio 0930 112 60 • **Norrköping:** Mandins i Hageby Centrum 011 14 73 60 • **Nybro:** Cedmerts Radio TV AB 0481 107 51 • **Nyköping:** TV-Matz AB 0155 182 30 • **Nässjö:** JM-Radio AB 0380 192 80 • **Osby:** A-Butiken AB 0479 104 14 • **Perstorp:** Schölds

200 butiker i sam

m säljer VIC

Radio TV 0435 301 07 • Piteå: Wiksténs Radio TV AB 0911 177 80 • Pålsboda: Allanssons Radio TV AB 0582 402 40 • Ronneby: B Johannesson Radio TV AB 0457 220 53 • Sala: Sala Sound AB 0224 115 15 • Saltsjöbaden: Saltsjö Radio AB 08 717 02 55 • Saltsjö-Boo: Saltsjö Radio AB 08 715 80 53 • Sandviken: Firma TV-Teknik 026 27 16 82 • Sigtuna: Peter's Ljud & Bild 0760 503 22 • Simrishamn: Eldhs Radio&TV AB 0414 140 55 • Sjöbo: Sjöbo Teleservice AB 0416 188 40 • Skellefteå: TV Stereoexperten 0910 782 22 • Skurup: Knut Pers, AB 0411 402 82 • Skärhamn: Sörängs Radio&TV AB 0304 705 80 • Skövde: Rapport Radio AB 0500 805 50 • Sollefteå: Hagbergs Radio TV AB 0620 101 12 • Staffanstorps: Herrlins Radio TV 046 25 25 20 • Stockholm: A G Anderssons Radio & TV AB 08 52 24 24, Bild & Ljud R Sylvendahl AB 08 84 84 75, Norrmalms Video-Bild-Ljud 08 20 21 20, Radiohörnan AB 08 60 38 92, Skanstulls TV Radio 08 43 54 00, Stevemans HiFi-TV AB 08 20 69 20 • Storuman: Orrkvist El-Radio-TV 0951 108 39 • Strömstad: Strömstads Musikcenter AB 0526 127 52 • Sundsvall: Pekka Nordins Kapitalvaror AB 060 11 47 73 • Svedala: Svedala Radio & TV AB 040 40 01 20 • Säffle: Radiocentralen i Säffle AB 0533 101 02 • Säters: Sätters Radio TV AB 0225 515 20 • Sävsjö: Ljudprodukter i Sävsjö 0382 124 38 • Söderhamn: Fa Göransons 0270 106 80 • Södertälje: AB Attes Radio 0755 175 85 • Söderåkra: Hemtema AB 0486 213 53 • Sölvesborg: Franssons Radio TV AB 0456 134 44 • Tidaholm: TV-Centralen Wallgren & Co 0502 115 00-20 • Tingsryd: BJ:s Radio & TV AB 0477 109 01 • Tranås: Axfors Radio & TV AB 0140 123 21 • Trollhättan: Radio-TV-Center Blomgren & Co AB 0520 387 90 • Trångsund: Trångsunds Bild & Ljud AB 08 771 20 50 • Tyringe: Tyrs Radio TV Stereo 0451 511 52 • Täby: FOKUS Ljud & Bild 08 758 60 95 • Umeå: Gruffman Umeå Radiobyrå AB 090 11 18 80, Ulf Radio AB 090 12 37 17 • Upplands Väsby: Sigfrid Hanssons El AB 0760 300 27 • Uppsala: Radiohörnan i Uppsala AB 018 13 15 17 • Vallentuna: M Thorins Ljud & Bild AB 0762 746 00 • Vara: AB Göran Janssons Radio-TV 0512 111 00 • Varberg: BARRES (Varbergs El Affär AB) 0340 105 25 • Vaxholm: Vaxholms Radio-TV AB 0764 324 60 • Vellinge: LP-HiFi-TV 040 42 44 00 • Vimmerby: Edvinssons Cykel Radio 0492 110 31 • Visby: Radio TV-Hörnan 0498 195 00 • Vänersborg: Herbert Anderssons Eftr 0521 100 39 • Vännäs: Bygdéns Radio & TV AB 0935 110 80 • Värnamo: Radio-Peterson AB 0370 111 25 • Västervik: TV-Larzon 0490 343 00 • Västerås: Boom Ljud & Bild, Sala Sound AB 021 12 42 80-81, Ramquist TV & Radio 021 14 27 20 • Västerås Sound AB 021 11 37 00 • Växjö: Ginos Radio TV 0470 152 00, Görans HiFi Center AB 0470 469 10 • Ystad: Tufvessons Radio TV AB 0411 100 07 • Åmål: Elbyrån Allan Samuelsson 0532 103 65 • Åseda: Åseda Radio TV AB 0474 106 22 • Älmhult: Musik & Ljudcenter 0476 132 30 • Ängelholm: Thulins Radio & TV Co 0431 304 00 • Öckerö: HB Mats Anderssons Elektronik 031 78 20 21 • Örebro: Borgs Radio, Radio-Special 019 11 19 27, 11 49 69, Åkes Radio-TV AB 019 13 77 70 • Örnsköldsvik: Gidlund Radio TV AB 0660 192 80, Ljudbutiken Arac 0660 187 60 • Östersund: Andersson & Fridell 063 12 15 65 • Övertorneå: HB Nybergs Radio, TV & Bokhandel 0927 102 71, 104 70



nverkan



ASSEMBLERSKOLAN – del 2

Artikelserie av: Peter Tegelberg och Åke Hedman

I detta avsnitt skall vi behandla dels hexadecimala tal (hextal) och dels de register som finns i processorn.

Efter denna genomgång har vi skaffat oss en god grund att stå på och vi kan sedan (äntligen) ge oss i kast med själva programmeringen. Det får tyvärr anstå till nästa avsnitt då vi 'assembler freaks' inte skall pressa redaktör'n på alltför många sidor.

I slutet av artikeln hittar du ett BASIC-program, som ger dig lösningarna på uppgifterna från det första avsnittet. Programmet omvandlar tal mellan olika talbaser. Använd det som ett facit, när du sitter och övar.

Innan vi på nytt byter talsystem, skall vi enas om en konvention för att beteckna tal i olika talbaser.

1. Tal som föregås av % (procent) är binära och har basen 2. Ett exempel är %10001001.
2. Tal som inte föregås av något tecken är decimala, exvis 255. Basen är ... Just precis ... 10!
3. Tal som föregås av \$ (dollar) är hexadecimala och har basen 16, exvis. \$FF.
4. Tal som föregås av (kanelbulle, snabel-a) är oktala. Detta talsystem har basen 8.

Detta är internationella beteckningar. Du lär hitta dem både i assemblers och i litteraturen på området.

Punkt 4 ges endast av upplysande skäl, då detta talsystem knappast används längre, men risken är fortfarande stor att du stöter på det någon gång.

Nu borde ett uttryck på formen %1010=10 inte vara något problem för dig längre.

Hexital har alltså basen 16. Detta betyder, som vi lärde oss i förra avsnittet, att vi behöver 16 siffror (vi kallar dem siffror) för att representera ett tal i detta talsystem. Det decimala har bara 10 siffror (0-9). Vi måste därför hitta andra symboler för att representera 10, 11, 12, 13, 14 och 15.

Någon (i det förgångna) kom på idén att använda de sex första bokstäverna i alfabetet, dvs A-F. Detta är både logiskt och bra, eftersom vi redan i alfabetet placerar bokstäverna i ordning och denna ordning överensstämmer internationellt. Räknar vi till 15 i hex, får vi serien 01 2 3 4 5 6 7 8 9 A B C D E F. Det här med att blanda siffror och bokstäver, brukar kännas litet ovant i början, men precis som i algebran ($X=X+Y/3$ etc), vänjer man sig snabbt. En tankebild kan ju vara att se A-F, som konstanter. I BASIC (Commodore-varianten) skulle vi då t.ex. ha skrivit $A = 10 : B = 11$ osv och ALDRIG ändrat på detta.

Den främsta fördelen med att använda hexital är att vi får färre siffror. Det behövs bara två stycken för att representera det största värdet en byte kan antaga, dvs \$FF (255). Vi kommer senare att se andra fördelar, men ovanstående enkla exempel bör övertyga dig om att hexital är värda att lära sig behärska.

Som för alla binära och decimala tal (naturligtvis för ALLA talsystem) gäller att en siffras värde bestäms av dess position. Vi kan återigen skriva upp vårt välbekanta uttryck:

$$X \times 16^1 + Y \times 16^0 = \text{decimalt värde}$$

(X och Y är siffrornas värden)

För t ex \$FF får vi då:

$$15 \times 16^1 + 15 \times 16^0$$

$$240 + 15 = 255$$

Omvandla själv följande tal till decimaltal (gärna också binära).

\$D0, \$10, \$1C, \$AD, \$16, \$80

Facit har du i programmet.

De värden vi hittills har sett, har varit 1-bytes värden, dvs mindre än eller lika med 255. För att behandla större tal, måste vi använda två eller flera bytes. 2 bytes ger max 65535 och är vanligt när vi vill peka på en adress till en minnesposition.

Ännu en fördel med hexital är, att om vi omvandlar vart och ett av dessa

2 bytes för sig till hexital (gäller även binära), är det sedan bara att sätta ihop dem för att få våra 2-bytes värden. Antag t.ex. att två byte efter varandra innehåller \$D2 och \$FF. Sätter vi ihop dem, får vi 16;bits adressen \$FFD2. I det decimala fallet är det inte lika självklart. Eller Hur!? Men, säger nu den uppmärksamme, varför får vi \$FFD2 och inte \$D2FF? Detta är en specifik sak för VIC (6502 processorn). En 16-bits adress (256-65535) lagras på detta sätt. Lågbyten, LSB (Least Significant Byte) Lagras först och högbyten, MSB (Most Significant Byte) sedan. Hur denna lagring sker, dvs om LSB kommer före eller efter MSB skiljer sig mellan olika datorer. När du studerar datorer med andra processorer, bör du vara uppmärksam på detta. OBS! Man ser ofta MSB i betydelsen Most Significant BIT. Detsamma gäller för LSB. Ta nu och omvandla \$2000, \$4D3E, \$FD00, \$E43F till decimaltal. Beräkna också de enskilda bytens värden och tänk efter hur de lagras i minnet.

Omvandlingen från decimal till hex är inte heller särskilt svår. Vi beskriver här en omvandling från ett 16-bitars tal, men metoden är densamma för ett 8-bitars. Det allmänna uttrycket för ett 16-bitars tal, ser ut så här:

$$X \times 16^3 + Y \times 16^2 + Z \times 16^1 + P \times 16^0$$

där X, Y, Z, P innehåller respektive siffror i ett uttryck på formen \$XYZP.

Låt oss beräkna de ingående potenserna av 16.

$$16^3 = 4096, 16^2 = 256, 16^1 = 16, 16^0 = 1.$$

Vid omvandling från decimal till hex kan vi inte bara subtrahera, som är fallet med binära tal, utan vi måste också göra en division.

Som exempel tar vi 32763. Först dividerar vi detta tal med den högsta potensen (4096). Är resultatet mindre

än 1, skall en 0:a sättas i position längst till vänster. Ett värde mellan 1 och 15 ger denna siffra längst till vänster. I vårt fall blir det 7. Vi har nu \$7xxx. Subtrahera ursprungstaler med det erhållna värdet (7) gånger 16 upphöjt till positionens plats.

$$32763 - 7 \times 16^3 (4096) = 4091$$

Dividera resultatet med nästa siffras värde (256) och fortsätt som ovan. dvs $4091/256=15$ (F).

Vårt hextal ser nu ut så här

\$7Fxx

Sedan gör vi den välbekanta subtraktionen igen

$$4091 - 15 \times 16^2 (256) = 251$$

Dela det erhållna talet med nästa potens ($16^1 = 16$), dvs $251/16=15$. Detta ger oss \$tFFx. Subtrahera igen. Resultatet bör bli mellan 0-15. Detta ger oss siffran längst till höger.

$$251 - 15 \times 16^1 (16) = 11 \text{ (B)}$$

\$7FFB

Ett exempel till är på sin plats, den här gången okommenterat. Låt oss omvandla 6438 till hex.

$$6438 / 4096 = 1 \rightarrow \$1xxx$$

$$6438 - 1 \times 4096 = 2342 \rightarrow \$19xx$$

$$2342 / 256 = 9$$

$$2342 - 9 \times 256 = 38$$

$$38 / 16 = 2$$

$$38 - 2 \times 16 = 6 \rightarrow \$1926$$

Ansträng dig för att verkligen förstå det här med omvandling mellan olika talbaser. Du får igen det många gånger om sedan, när du skall programmera. Det är bättre att göra felen nu, än när du sitter och gör ett program. 'Buggar' (fel) i ett assemblerprogram, är inte det lättaste att rätta till. Några omvandlingar från dec till hex som du kan öva på: 11437, 7700, 3654, 58112, 8192. Nu kan vi det här med talbaser. Eller hur? (Jaaaa)

PROCESSOR kallas den krets, som är själva hjärnan i en dator. Den kontrollerar genom en adressbuss och en databuss samt en del kontrollsignaler, de övriga kretsarna i systemet, som främst består av minnen i olika former, samt kretsar för att kommunicera med omvärlden. Det finns många olika varianter av processorer och kan ni tänka, alla programmeras de i olika typer av maskinkod. Detta innebär att den som lär sig maskinspråksprogrammering för en processor, inte direkt kan sätta sig och programmera en annan. För att göra program till den andra processorn, måste han/hon lära sig ett helt nytt maskinspråk. Lugnande är dock, att många saker överensstämmer mellan olika processorer. Detta gör det oftast lättare att lära sig sitt andra maskinspråk (puh).

Här skall vi helt och hållet inrikta oss på en processor, som 'heter' 6502 (lustigt namn, eller hur?). Den är till-

sammans med Z80, som bl.a. finns i Sinclairdatorerna, den absolut populäraste processorn idag. 6502:an finns t.ex. i VIC, PET och Apple. Dess styrka ligger främst i snabbhet och lättillgänglighet. För att slippa gå in alltför mycket på den interna strukturen, skall vi i fortsättningen låtsas att man i processorn murat in en energisk japan, som sköter om vår lilla dator och ränner runt som en oljad blix. Vi kallar honom Sven.

Att Sven har det jobbigt det kan man förstå, när man får reda på att hans klocka inte tickar en gång per sekund, utan c:a en miljon gånger per sekund. Varje gång den tickar, måste Sven utföra något. Den tid det tar, kallar vi en CYKEL. Antalet cykler per sekund kallar vi för KLOCK-FREKVENS.

Till sin hjälp i processorn har Sven en miniräknare, som kallas ALU (Arithmetic Logic Unit). Med denna kan han addera, subtrahera, jämföra tal och utföra t.ex. logiska funktioner. Dessutom har han ett par flaggor, som han kan hissa om något speciellt inträffar, så att han inte glömmer bort det.

Han har dessutom ett par svarta tavlor, som han kallar för A, X, Y, P, S och PC. De frem första har plats för 8 bitar medan den sjätte (PC) kan hålla 16 bitar. Dessa tavlor brukar kallas REGISTER, men det behöver vi inte berätta för Sven! *forts. nästa sida*

BYTEFLIGHT AB

Ett företag helt specialiserat på mindre datorer och dess användning i hemmet eller företaget.

RINGVÄGEN 99, STOCKHOLM

Butiken du använder när du behöver något till din dator. Innan du bestämmer dig får du en god överblick över hela sortimentet av datorer, tillbehör, kunskap och service.

08-40 69 00

Ordertelefonen du använder när du behöver något snabbt till din dator. Vi levererar direkt från lager med garanti mot postförskott. Obs! Inga expeditiionskostnader.

Aktuella rutan till VIC

- FÖRETAGAR-PAKET VIC 64
- VIXPLODE VIC 64 strategispel
- PARKERINGSTERROR VIC 20 spel
- INTERFACE
- LJUSPENNOR
- DAMMSKYDD

Öppettider:

Mån-fre 10.00-18.00
Lör 9.30-14.00

BYTE FLIGHT AB

Box 20054 · Ringv. 99 · 104 60 Stockholm

Sven har också en lucka där han får instruktioner om vad som skall göras. Här kan han även hämta och skicka ut data till alla minnesfack, som finns utanför Svens lilla värld.

ROM facken kan han bara läsa i. I RAM facken däremot, kan han både läsa och skriva information. Sven har dessutom en del I/O (Input/Output) fack, där han kan ta in och skicka ut information till omvärlden. Sven är väl inte riktigt säker på om det här är RAM eller inte, eftersom I/O facken ser exakt ut som RAM för honom.

Vad använder då Sven sina svarta taylor till? Vi tar dem i tur och ordning.

PC (Program Counter) är programräknaren. Detta register kan som sagt innehålla 16 bitar och det pekar alltid på nästa instruktion, dvs till en instruktion, som talar om vad Sven skall göra härnäst. PC pekar hela tiden framåt eller gör ett skutt, men räknar alltid upp ett antal steg varje gång en instruktion utförs. PC kan jämföras med ett register som håller reda på det radnummer, som skall exekveras (utföras) i ett BASIC program. Skutten följer av att man även i maskinkod har instruktioner för att hoppa till ett annat ställe i programmet (jämför BASIC GOTO) eller för att utföra en subrutin (jfr GOSUB).

Nästa register vi skall titta på är A. Det är det absolut viktigaste registret och används av nästan alla instruktioner. A kan jämföras med displayen på en miniräknare, då det oftast håller det senaste resultatet av en operation. Registret kallas normalt ACKUMULATOR. Man ser även förkortningarna ACC, A eller ACCUM, här använder vi A.

De två registren X och Y, kallas INDEXREGISTER. Namnet kommer av att de ofta tillsammans med en adress pekar på någon minnesposition. Man kan även se dem som tillfälliga register där information kan lagras för kortare stunder analogt med de minnen som brukar finnas i en miniräknare, där man kan shifta information mellan displayen och minnet.

P eller STATUSREGISTERET kräver en noggrannare genomgång, därför får vi ta det litet senare, när vi gått igenom en del kommandon. Vi kan redan här berätta att registret innehåller

en avskrift av de flaggor, som Sven hissar.

S eller STACKPEKAREN är ett annat mycket viktigt register. Det pekar på en area i minnet mellan 256-511 eller \$0100-\$01FF. Denna area kallas PAGE 1 (sida ett) och används av

processorn för tillfällig lagring av information. Mer om stackpekaren senare.

Vi återkommer i nästa nummer.

ASSEMBLERSKOLAN ® Copyright
1983 Peter Tegelberg och Åke Hedman

Här är programmet som omvandlar tal mellan olika talbaser.

READY.

```

100 REM *** K O N V E R T E R I N G ***
120 REM OMVANDLAR MELLAN TALBASERNA - HEXADECIMAL (BAS 16),
130 REM DECIMAL (10), OKTAL (8) OCH BINÄR (2)
150 PRINT " : REM CLEAR SCREEN <SHIFT OCH CLR/HOME> MELLAN CITATIONSTECKNEN
160 HX$="0123456789ABCDEF"
170 PRINT "TAL ATT KONVERTERA"
180 PRINT "(TAL, INBAS, UTBAS)"
190 INPUT B1, B2: PRINT
195 PRINT B1: PRINT "I BASEN"; B1: PRINT "BLIR ..."
200 T=B1: GOSUB 250
210 T=B2: GOSUB 300
220 PRINT B2: PRINT "I BASEN"; T
230 PRINT
240 GOTO 170
250 X=0
260 FOR I=1 TO LEN(HX$):
270 FOR K=1 TO T
280 IF MID$(HX$, I, 1)=MID$(HX$, K, 1) THEN X=X+T+K-1: K=K+1
290 NEXT K, I: RETURN
300 H$=""
310 K=INT(X/T): J=X-T*K: H$=MID$(HX$, J+1, 1)+H$+CHR$(0): IF K>0 THEN X=K: GOTO 310
320 RETURN
READY.
```

VICtiga rättelser

Rättelser till VIC NEWS nr 1 -83

- A. En sol förekommer på ett flertal platser, ska givetvis vara \$
B. Normala värdet på adress 36864 ska vara 12
Normala värdet på adress 36865 ska vara 38
C. På ett antal ställen finns citations-tecken men inget innanför, ska vara @

Bilder till 'Egna tecken' sid. 34-35.

ADRESS	76543210		ADRESS	76543210	
32768	***	28	7168	* *	18
32769	* *	34	7169		9
32770	* * *	74	7170	**	12
32771	* * **	86	7171	* *	18
32772	* **	76	7172	* *	33
32773	*	32	7173	* *	33
32774	***	30	7174	* *	18
32775		0	7175	**	12

Rättelse till program sidan 37

```

100 PRINT "X": XPOT=PEEK(36872): YPOT=PEEK(36873)
110 XPOT=XPOT*.085: YPOT=YPOT*.085: REM POTENTIOMETERKONSTANT
115 X=INT(XPOT): Y=INT(YPOT): GETA$: IF A$<>" " THEN PRINT "J"
120 X$="": Y$=""
130 Y$=" "
140 PRINT "X"
150 PRINT "X="; X, "Y="; Y
160 PRINT "X" LEFT$(X$, XPOT)+LEFT$(Y$, YPOT); " "
170 GOTO 100
```


VIC-20 VIC-64



GRANA SOFTWARE

**SPEL och NYTTOPROGRAM i MÄNGDER
plus hela HANDIC-sortimentet.**

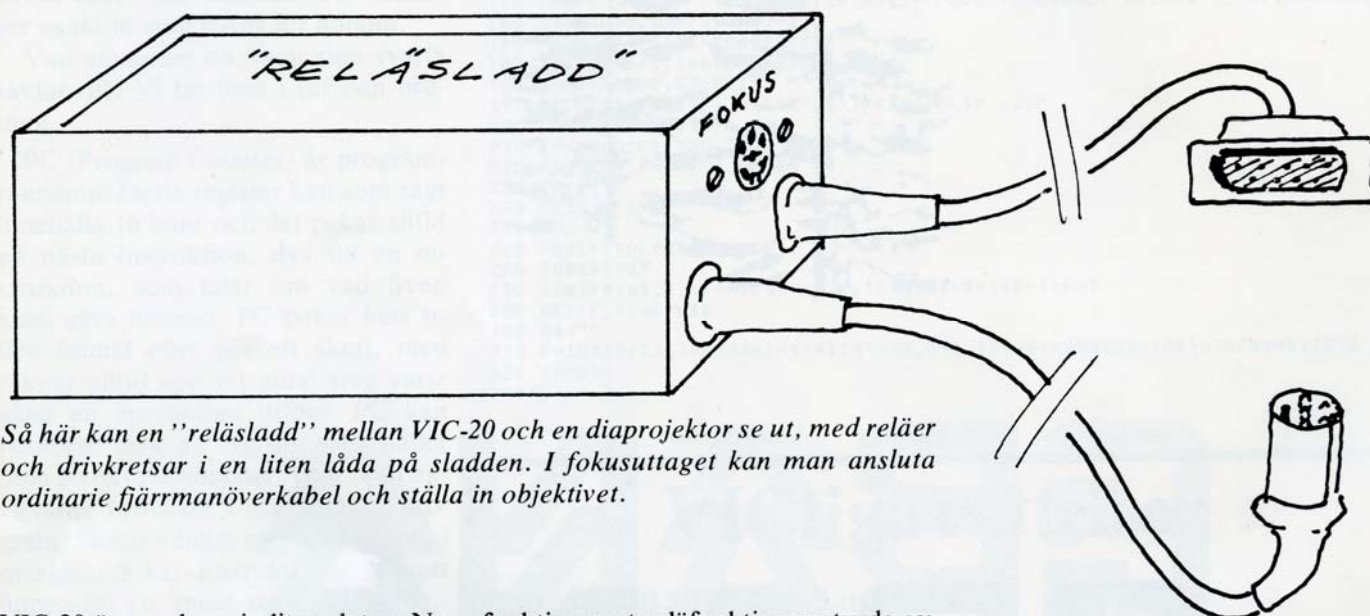
Gratis katalog!

Återförsäljare sökes!



Reläsladd gör presentationssystem av din VIC-20

Frenne Söderberg



Så här kan en "reläsladd" mellan VIC-20 och en diaprojektor se ut, med reläer och drivkretsar i en liten låda på sladden. I fokusuttaget kan man ansluta ordinarie fjärrmanöverkabel och ställa in objektivet.

VIC-20 är en charmig liten dator. Nu när den har fått en storebror kan det vara på plats att dra en lans för "tjugan" också och peka på en alldeles särskild fördel och hur den kan utnyttjas. Det har oftast setts som en brgränsning och ingenting annat att VIC-20 har bara 23 rader med 22 tecken. Det var väl ursprungligen en försiktighetsåtgärd för att man även på dåligt trimmade tv-apparater skulle få tydliga, stora bokstäver. Den satsningen på användarvänlighet gör det möjligt att "texta bilder", 12 rader med mellanslag, som blir alldeles utmärkta föredragsillustrationer eller fristående informationstexter. Du har säkert märkt hur bra det går att köra dataspel för en hel grupp intresserade kring en stor färg-tv.

Men man behöver inte begränsa sig till att titta på text eller spel som man genererar på datorn. De flesta dia-projektorer, för att ta ett exempel, har idag fjärrstyrning med en manöverkabel, som innehåller några slutande eller växlande kontakter. Det är i princip mycket lätt att byta ut deras

funktion mot reläfunktioner styrda av datorn. Jag skall beskriva en lösning som har fördelen att lämna "USER PORT" ledig för andra periferiapparater.

Om du vill lägga ett system för textad diabilddisning i VIC-en verkar det rimligt att använda joystickkontakten för projektorstyrningen. Visningen skall ändå skötas genom ett program som tar emot rätt så detaljerade kommandon.

Joystick-kontakten hanteras av de två 6522-chipen i VIC-20. 6522 är en s.k. via, en "versatile interface adapter" eller programmerbar in/ut-krets. Vi kommer att klara oss med att använda via nr 1, som utom mycket annat också hanterar trådarna joy nr 0, 1 och 2, dvs de som går till stift nr 1, 2 och 3 i den 9-poliga D-kontakten. Ur denna hämtar vi också +5V på stift 7 och jord på stift 8. Med dessa kan vi manövrera upp till 3 reläer om de är känsliga nog att inte belasta 5V-spänningen med mer än 100 ma tillsammans. Styrningen får ske med en liten tranistor för varje relä, så belastas vi

an inte med mer än någon mA. Relä-kontakterna kan kopplas var för sig till de slutningar projektor vill ha eller kombineras i serie om man t.ex. vill förhindra att framåt- och bakåtväxlingskommando ges samtidigt.

Hur kan datorn nu få reläerna att arbeta? Jo, det finns ett data direction register (DDR) och ett databitregister (DBR), som vi kan styra med POKE-kommandon när vi bara vet deras adresser. Då är det så här: DDR för via nr 1 är 37139. Joy nr 0 och 1 är bit nr 2 och 3 respektive. Om man sätter dem till 1 så blir motsvarande stift en utgång istället för en ingång som den brukar vara. För att sätta bit 2 och 3 ger vi 37139 värdet 12, eller hur? Själva signalen ligger på 37137, samma bitar. Om vi ger 37137 värdet 12 kommer alltså båda reläerna att dra, förutsatt vi definierat utgångarna som ovan. Om vi tar med ett tredje relä får vi tänka på bit nr 4 också, dvs lägga till 16.

I ett program styr du alltså reläerna och därmed projektorn genom att poka till 37129 en gång i början och till 37137 för varje bildväxling, hur länge och till vilket värde beror på projektortypen och relälogiken. Exempelvärdena ovan fungerar tillsammans med en Kodak Carousel-S om man kopplar reläerna enligt figuren. Mellan växlingarna sätter man 37137 lämpligen till 240, som är normalvärdet. Om du tittar med PEEK (37137)

och får värdet 176, har du säkert "PLAY" nedtryckt på kassettspelaren.

Det här var mycket om pokar och vior. Ändå finns det massvis kvar att göra i den vägen eftersom vi inte har använt "USER PORT" alls. Den kan t.ex. användas för att ansluta en pratapparat som läser upp bildtexterna eller särskilda kommentarer, men det får bli nästa historia.

Computer Books

Böcker till VIC-20/64

Getting acquainted with your VIC 20	(priser exkl. porto)
VIC Innovative Computing	116:—
ZAP! POW! BOOM! Games for VIC 20	145:—
Symphony for VIC 20	116:—
VIC 20 Exposed	135:—
The Working Commodore 64	115:—
Commodore 64 Games Book	115:—
Commodore 64 Exposed	135:—
Commodore Machine	
Code Master	135:—
Commodore 64 Adventures	135:—

Beställ katalog!

Studieförlaget

Box 386, 751 06 Uppsala 1, Telefon 018-15 53 90

Senaste Nytt

Nyss återkommen från den stora datormässan i England, Personal Computer World Show, är sinnesintrycken något kaotiska, men jag skall försöka ge litet av vad som händer i England just nu.

Framförallt märks att man börjar få fram program för affärsmässiga användningar av både VIC-20 och 64:an. Dessa program importeras ännu ej eller endast i begränsas omfattning och vi återkommer med detaljer när vi sett de senaste planerna från Handic.

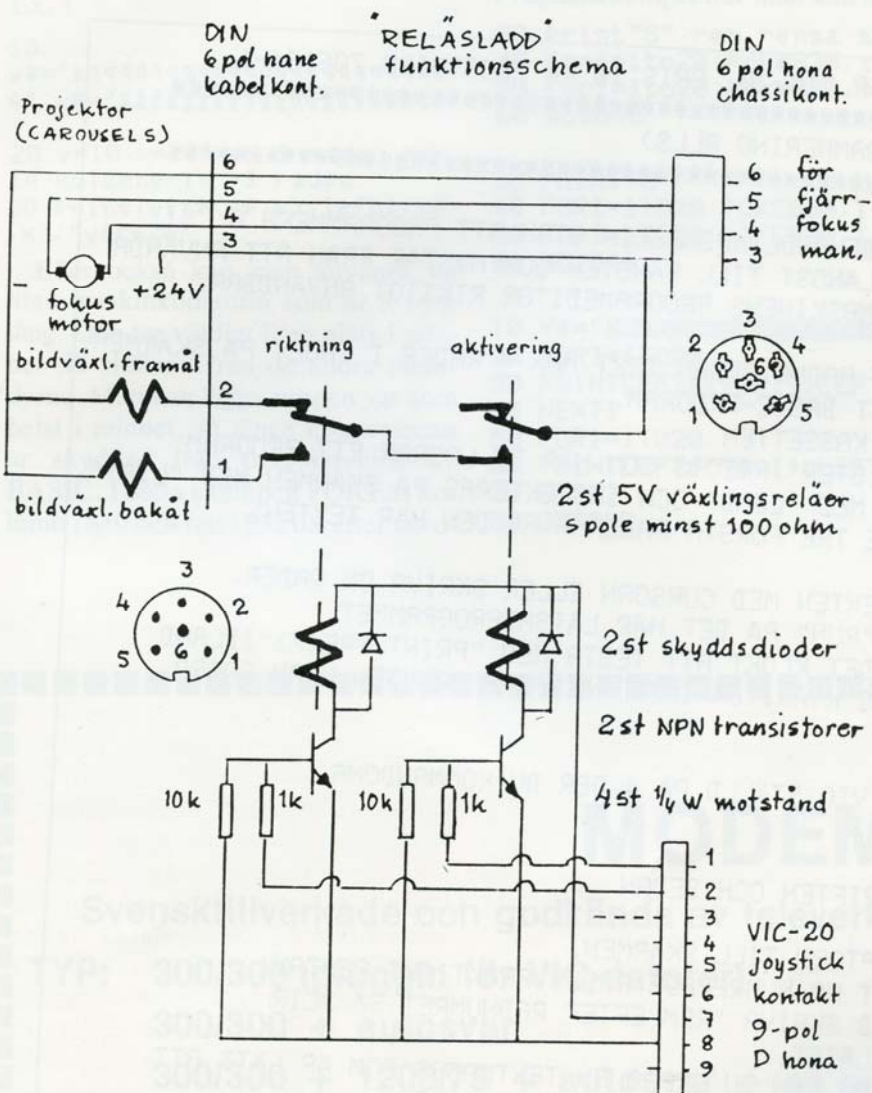
Red.

FEL I VIC SÖKARE

I programmet VIC-sökare nr 3/83 sid. 32 finns ett fel som rättas här.

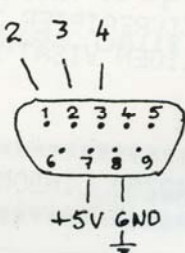
Det gäller rad 170 som skall lyda, "DATA 240, 10, 136, 230, 1, 208, 2, 230, 2, 24, 144" Felet ligger i näst sista siffergruppen som blivit 200 för stort.

Vi beklagar.



adr. 37137 bit

Joystick - kontakten sedd mot VIC-20 utifrån



OBS! kontakterna 4, 5, 6, 9 har andra funktioner, gör inga anslutningar till dem!

RELÄSLADD

1983-09-12 rev. A

Tierne Söderberg

Textbehandling U P A (Utan Programmering Alls)

Frenne Söderberg

Kan man göra ordbehandling på VIC-20? Och utan programmering? Javisst! Det vill säga, ett litet "om" får man acceptera. Men det är ju så här att folkdatorägaren har julaften en

gång om året bara och vill ändå pröva hur det känns att utnyttja den datakraft han har så långt det är möjligt. Maximum funktion ur minimum hårdvara alltså. Alltnog, det blev

mycket roligare när skrivaren tillkom i utrustningen, för då kunde man dokumentera sina program till att börja med. Men det allra roligaste var att inse hur man med en liten omväg faktiskt kan ha just den nyttan av VIC som ett ordbehandlingssystem erbjuder: man kan skriva rätt på maskin och lagra dokumentet för framtida uppdatering. Annars är det, för mig, en plåga att försöka få fram ett helt igenom rättstavat papper på min gamla skrivmaskin. Jag låter metoden demonstrera sig själv, se nedan!

```

10 GRANDET DATA PROGRAMTIPS FÖR VIC-20 PRIS 10 KR POSTGIRO 706164-1
20 *****
30 TEXTREGISTER UPA (UTAN PROGRAMMERING ALLS)
40 *****
45 /
50 DU KAN ANVÄNDA VIC SOM TEXTBEHANDLINGSMASKIN UTAN ATT PROGRAMMERA ALLS!
60 OM DU ACCEPTERAR RADNUMMER LÄNGST TILL VÄNSTER OCH AVSTAR FRÅN ATT ANVÄNDA
70 FRÅGETECKEN (=PRINT) SÅ ÄR MASKINENS PROGRAMEDITOR RIKTIGT ANVÄNDBAR.
75 /
80 DU SKRIVER ALLTSA FÖRST ETT RADNR, SEDAN TEXT, MAX 4 RADER I TAGET PÅ SKÄRMEN.
90 DATORN TROR ATT DETTA ÄR ETT BASIC-PROGRAM.
95 FÖR ATT LAGRA REGISTRET PÅ KASSETTEN
100 SKRIVER DU: SAVE"MITT REGISTER",PRECIS SOM NÄR DU LAGRAR ETT PROGRAM.
110 REGISTRET HÄMTAS TILLBAKA MED "LOAD" OCH INSPEKTERAS PÅ SKÄRMEN MED "LIST".
120 T.EX."LIST 10-30" VISAR DE TRE FÖRSTA RADERNA AV DEN HÄR TEXTEN.
121 /
122 DU KAN ÄNDRA HEJDLÖST I TEXTEN MED CURSORN ELLER SKRIVA OM RADER.
123 MEN GÖR FÖR ALL DEL INTE "RUN" PÅ DET HÄR LATSASPROGRAMMET.
124 OM DU HAR MYCKET TEXT ÄR DET KLOKT ATT TESTA MED "PRINT FRE(X)"IBLAND
125 HUR MYCKET MINNE SOM FINNS KVAR. GÅ INTE FÖR NÄRA GRÄNSEN UTAN SPARA
126 OCH BÖRJA OM IGEN.
127 /
130 OM DU HAR SKRIVAREN TILL VIC,STÄLLD PÅ 4,GER DU KOMMANDONA:
140 OPEN4,4
150 CMD4:LIST
160 VÄNTAR MEDAN DU FÅR UTSKRIFTEN OCH SEDAN
170 PRINT#4
180 SÅ ATT DU FÅR TILLBAKA DATORN TILL SKÄRMEN.
190 DU VÄNJR DIG NOG VID ATT HA NUMRERADE RADER.PRAKTISKT RENTAV!
200 ÄR DU PETIG KAN DU ALLTID SKRIVA "REM"EFTER RADNUMRET SÅ BLIR
210 REMFRÅGETECKNET (=?) TILLÄTET.
220 REM MED NÅGRA MELLANSLAG KAN DU SKAPA EN TEXTMASSA SOM ÄR LÄTT ATT
230 REM KLIPPA LOSS OCH SÄTTA UPP PÅ ETT VANLIGT PAPPER.
235 REM
240 REM HA SÅ ROLIGT MED TELEFON-,ADRESS- OCH SKIVREGISTER T. EX.
245 REM OBSERVERA OCKSÅ ATT JUST NU HAR DU VERKLIGEN VISAT DATORN
250 REM ORDENTLIGT VEM DET ÄR SOM BESTÄMMER!
260 REM
265 *****
270 REM HÄLSNINGAR FRÅN GRANDET DATA BOX 123 43700 LINDOME © 1983
275 *****

```


Markörposition på VIC-20/64

Det finns ett antal sätt att förflytta markören dit man vill på skärmen. Dels genom att i PRINT-kommandot flytta ner markören, genom att använda markör-kontrollen. Eller definiera X\$ och Y\$ för x och y riktning, och sedan med hjälp av strängmanipulering komma på rätt plats på skärmen (se ex 1.).

Ex. 1

```
10
y$="sqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq"
q":x$="1111111111111111111111111111111111"
"
20 y=10:x=14:rem 9 rader ner
14 kolumnerin på raden
30 printleft$(y$,y);left$(x$,
,x);"vic-20"
```

Eller också kan man använda en liten maskinkodsrutin som är 9 byte lång. Den tar väldigt liten plats i minnet till skillnad från de andra metoderna. Man kan lägga rutinen var som helst i minnet, så länge minnesarean är skyddad från överskrivning av BASIC. I detta exempel POKEas kolumn i 830 och rad i 832 därefter anro-

pas rutinen med SYS 828. Sedan PRINTar man sitt tecken eller sin sträng.

Basic loadern:

```
10 FOR I = 828 TO 836:READ A:POKE I,A:NEXT
20 DATA 24,160,0,162,0,32,240,255,96
```

Programexempel hur man kan använda rutinen:

```
30 Print"S":rem rensa skärmen
40 fori=1to20:POKE830,i:POKE832,i:SYS828:Print"Q":next
50 fori=1to20:POKE830,i:POKE832,i:SYS828:Print" ":next
60 goto40
```

```
30 PRINT"Q"
40 FORI=1TO20:POKE830,I:POKE832,I:SYS828:PRINT"Q":NEXT
50 FORI=1TO20:POKE830,I:POKE832,I:SYS828:PRINT" ":NEXT
60 GOTO40
```

```
10 Y$="sqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq":X$="1111111111111111111111111111111111"
20 FORI=1TO20
30 PRINTLEFT$(Y$,I);LEFT$(X$,I);"Q"
40 NEXTI
50 FORI=1TO20
60 PRINTLEFT$(Y$,I);LEFT$(X$,I);" "
70 NEXTI
80 GOTO20
```

Kjell Segerbom

MODEM

Svensktillverkade och godkända av televerket. Nu till introduktionspris.

TYP: 300/300 (pluggin för VIC-datorer)

300/300 + autosvar

300/300 + 1200/75 + autosvar + en programmerbar kanal

Dessutom har vi splitspeedkonvertrar för terminal och datorände

Återförsäljare sökes. Skriv eller ring för information.

SELIC AB

Box 44002, 400 76 Göteborg, Tel: 031/22 33 10

Hur man gör snygga skärmtexter

Det är alltid trevligt att kunna starta ett BASIC-program med en snygg text på bildskärmen. Det är t o m möjligt att låta användaren välja sin egen text, och dramatiskt låta den komma upp på skärmen när programmet börjar.

En sak att komma ihåg när man väljer en skärmtext är att man bör försöka hålla radlängden på texten kortare än 22 tecken inklusive mellanrummen, därför att VIC-20 bara visar 22 tecken på en rad och texten annars kommer att "rinna över" till nästa rad om den är längre.

på skärmen genom att CLEARa skärmen och därefter PRINTa texten tre mellanrum från vänsterkanten. (För att få fram varje mellanrum i ditt program skall du trycka på den långa mellanslagstangenten en gång, snabbt. Varje mellanrum som du tar med innanför citationstecknen kommer att märkas som ett mellanrum på skärmen när texten PRINTas ut. Lagg märke till att du inte behöver bekymra dig om mellanrummen på höger sida om texten därför att de lämnas automatiskt teckenfria (blanka). Slå in följande rad och tryck <RETURN>:

```
10 PRINT" shift clr/home space
space space VIC-TROLLKARLEN"
```

VICen använder en sorts "steno-grafi" när den visar CLEAR-kommandot innanför citationstecken, så när du håller ner shift och trycker på clr/home kommer VIC att visa ett hjärta i reverse, vilket betyder CLEAR. Mellanrummen du slog in kommer att synas som tomma mellanrum, så här:

```
10 PRINT" ♥ VIC-TROLLKARLEN"
```

För att du skall se din text slår du in RUN och trycker på <RETURN>. Får du problem eller slår fel, skriv LIST och slå <RETURN> och skriv om raden.

TRYCK PÅ "9"-TANGENTEN

```
10 PRINT" shift clr/home space space space CTRL RVS ON VIC-TROLLKARLEN"
```

Användning av SPC-kommandot

Det finns ännu ett sätt att få in mellanrum i ditt BASIC-program... Det går att använda SPC-kommandot. Om du slår in PRINT SPC(5) kommer ditt program att sätta in fem mellanrum. Kom bara ihåg att SPC-kommandot aldrig sätts mellan citationstecken... och att du alltid måste använda PRINT-kommandot framför SPC-kommandot. Så här kan du använda SPC-kommandot för att skriva den rad som du nyss slog in:

```
10 PRINT" shift clr/home"SPC(3)
"VIC-TROLLKARLEN"
```

Här är ett annat exempel på SPC-kommandot:

```
10 PRINTSPC(5) "FLYTTAR 5 MELLAN
RUM"SPC(3)"FLYTTAR 3 MELLANRUM"
```

Hur man visar sin text i omvänd video (reverse video)

När du nu har skapat en rubrik. Låt oss "trixa till" den litet genom att vända om rubrikfärgen. För att göra detta måste du skriva om rad 10. Den här gången, alldeles innan du slår in rubriken, håll nere CTRL-tangenten och tryck på RVS ON-tangenten. Detta får texten att visas i omvänd video när du RUNar programmet. Märk att när du håller nere CTRL och trycker på RVS ON så visar VIC ett omvänt R. Skriv följande rad och tryck ner <RETURN>:

Hur man centrerar (mittställer) sin text

Det finns många sätt att göra tilltalande texter, men en av de lättaste är att man sätter texten mitt på och högst upp på bildskärmen när programmet startar.

För att centrera texten mitt på en rad måste man först räkna antalet tecken i texten. Låt oss anta att din rubriktext är "VIC-TROLLKARLEN".

VIC-TROLLKARLEN

Denna text har 15 tecken totalt. Dra nu dessa 15 från radlängden som var 22 och du får 7 kvar. För att sätta texten på mitten måste du dela 7 med 2 och får då 3+4, vilket betyder att du får sätta 3 mellanrum före och 4 efter din text. Det är så mycket på mitten som du kan komma med ett ojämnt antal tecken i din titel.

Följande enradiga program centrerar texten "VIC-TROLLKARLEN"

Raden som du nu skrivit in kommer att se ut så här på skärmen:

```
10 PRINT" VIC-TROLLKARLEN"
```

Skriv nu RUN och tryck på <RETURN> för att se den nya rubriken. Hur skall du göra då om du vill vända på hela raden, inklusive mellanrummen? Jo, då slår du in följande rad exakt som den visas och trycker på <RETURN>. Sen skriver du RUN och <RETURN> så får du se:

```
10 PRINT" shift [clr/home] [ctrl]
[rvs on] [3 mellanslag]VIC-TROLLKA
RLEN [4 mellanslag]"
```

Den här gången satte vi REVERSE ON-kommandot vid början av det första tomma mellanrummet och lade till fyra mellanrum på höger sida om vår rubrik. När du vill skriva omvända mellanrum (dvs heltäckande ytor) på skärmen, måste du ta med alla mellanrum som du vill ha synliga som massiva ytor. Detta är orsaken till att vi tog med högersidans mellanrum i denna programrad.

Att låta användaren välja rubrik

Någon av de prydligaste "kosmetiska" egenskaperna som du kan ge en rubrik är att låta användaren välja sin egen rubrik. Följande program ber användaren skriva in en rubrik, därefter centreras den högst upp på bildskärmen. Du kan använda dessa två rader i början av varje BASIC-program.

```
10 PRINT" shift [clr/home] RUBRIK"
;:INPUTT$
20 PRINT" shift [clr/home]";:FORX =
1TO(22-LEN(T$))/2:PRINT" [space]"
:;:NEXT:PRINTT$
```

När du knappat in detta program, skriv RUN och <RETURN>. Skärmen kommer att rensas och säga "Rubrik"? Skriv in en rubrik och tryck på <RETURN>. Rubriken syns omedelbart centrerad högst upp på din bildskärm.

Nyckeln ligger här i att man använder INPUT-satsen på rad 10 för att definiera den rubrik som skrivs in av användaren som T\$. Sedan, på rad 20, använder vi ett speciellt BASIC-kommando som heter LEN för att bestämma längden (LENgth) på T\$. LEN(T\$) räknar antalet tecken i rubriken, som INPUTtades på rad 10 som T\$. Kom ihåg centreringsformeln som vi använde i början av artikeln: hela denna formel ingår i rad 20 som $(22-LEN(T$))/2$. Om vi återigen använder VIC-TROLLKARLEN som vår rubrik, är LEN fortfarande 15 tecken och $(22-15)/2=3,5$ dvs FOR-snurran går till 3 och lägger ut tre mellanrum före rubriken.

OBSERVANDUM FÖR VIC-64-ägare: ändra mätsatsen till $((40-LEN(T$))/2)$ så kommer även er rubrik upp i mitten på en 40-teckens bildskärm!

Formeln ger alltså antalet mellanrum på ena sidan om vår titel. (Om det är så att du undrar varför vi har dubbla parentestecken efter T\$ så är det därför att alla BASIC-formler måste ha samma antal höger och vänsterparenteser och de två vänsterparenteserna balanserar de två vänsterparenteserna som står tidigare på programraden. Detta är viktigt att komma ihåg, därför att det är ett av de vanligaste misstagen när man gör formler i BASIC för beräkningar, just att glömma en ynka parentes...

Efter beräkning av antalet mellanrum använder vi PRINT-kommandot i FOR-satsen för att lägga ut mellanrummen och när det är avslutat ytterligare en PRINT-sats för att skriva vår rubrik. Semikolon (;) som används efter PRINT-satserna innebär att allting skall skrivas omedelbart efter varandra och NEXT-kommandot innebär att man håller på tills övre gränsen, dvs här 3,5 eller avkortat till 3 har nåtts. När X har kommit till 4 är det ju större än 3,5 och då lämnas FOR-snurran och man skriver ut rubriktexten. Dvs T\$

den högst upp i bildskärmen ställd helt åt vänster:

```
10 PRINT" [shift] [clr/home] RUBR
IK";:INPUTT$
20 PRINT" [shift] [clr/home]";:
PRINT" [ctrl] [rvs on] "T$::FORX
=LEN(T$) T021:
PRINT" [ctrl][rvs on] [space]";:
NEXT
```

Att lägga till färg till rubriker

En ytterligare dimension du kan lägga till dina skärmtexter är färg. Färgkommandot fungerar precis som rvs on-kommandot. Håll nere ctrl-tangenten och tryck ner färg-tangenten för önskad färg. Följande program PRINTar en rubrik i rött, ändrar färgen sedan till blått så övriga delar av programmet fortsätter i blått. "ctrl red" betyder att man håller nere ctrl-tangenten när man trycker på den tangent som märkts red (dvs tangenten "3", blått fås genom tangent "7"):

```
10 PRINT" [shift] [clr/home] RUBRIK
";:INPUTT$
20 PRINT" [shift] [clr/home]";:PR
INT" [ctrl] [rvs on] [ctrl red] "
T$::FORX=LEN(T$)T021:PRINT"
[ctrl] [rvs on] [space]";:NEXT:
PRINT" [ctrl blue] "
```

Experimentera!

Det finns massor med andra sätt att skapa intressanta skärmtexter. Försök t ex att centrera titeln på mitten av skärmen (tips: använd crsr down-tangenten i en PRINT-sats och samma centreringsformel som vi använde för att centrera titeln i sidled). Du kan också skapa ramar runt dina texter, på båda sidor eller över och under, genom att använda ett grafiskt tecken.

Ett enkelt sätt att vända en användarbestämd rubrik

Följande program vänder om den användar-bestämda rubriken och visar

forts. nästa sida

Försök och sätt in grafiska rader eller linjer över eller under din rubrik, ungefär så här:

```
5 PRINT" [shift] [clr/home]";:FORG = 1TO 22:PRINT" [ ]";:NEXT
10 PRINT"VIC-TROLLKARLEN"
20 FORG = 1TO22:PRINT" [ ]";:NEXT
```

←(går över rubriken)
←(rubrikraden)
←(går under rubriken)

Du får denna grafiska symbol genom att hålla nere shift-tangenten och slå "C"

Här är ytterligare en variant som du kan prova på. Detta program centrerar en enradig rubrik på skärmen. Försök att ändra numret 10 på rad 20 till ett tal som flyttar titeln högre eller lägre på skärmen.

"crsr down" kommer att synas som ett omvänt Q på skärmen i programinskrivningen

```
10 PRINT" [clr/home] RUBRIK";:
INPUTT$
20 PRINT" [clr/home]":FORX =
1TO10: PRINT" [crsr down]";:NEXT
30 FORX = 1TO(22-LEN(T$))/2:PRI
NT" [space]";:NEXTX:PRINTT$
```

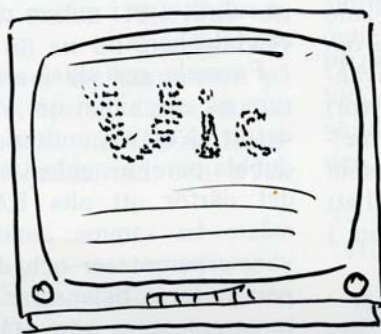
Lägg märke till att på rad 20 säger vi till VICen att PRINTa tio "markör-ner" (crsr down) och flyttar alltså läget på markören 10 mellanrum eller rader neråt. Därefter använder vi vår centreringsformel i rad 30 för att centrera texten. Du kan ta med mellanrum, markör upp- eller markör ner-flyttningar, ctrl-reverse on eller off, ctrl-färg-kommandon. Bara PRINTa dem som vilken bokstav som helst, eller siffra eller grafisk symbol och de kommer att synas i ditt program!

Som en sista variation kanske du vill ha något som säger till att det är en rubrik, en ljudeffekt t ex. Prova med att sätta nedanstående enradiga program på raden omedelbart före den rad som PRINTar din rubrik:

```
25 POKE36878,15:FORM=200TO250:
POKE36876,M:NEXTM:POKE36876,0
```

Nu får du klara dig själv. Prova olika färger, ljudeffekter, skärmlinjer och andra variationer. Lycka till!

Fotografera dina VIC-resultat!



Stillbilder och långsamma spel går bra att fotografera av från TV-skärmen om man bara följer några enkla regler.

1. Använd inte blix (det lär man ha provat på bio förr och det gick inte bra där heller).

2. Välj ett tillfälle på kvällen. Sätt igång VIC och TV-n en stund innan, det kan ta en halvtimme eller så innan TV-modulatorens och TV-n går riktigt stabilt ihop. Under tiden, kör dina program och kolla att allt är lika snyggt som du tänkt dig.

3. Helt bör du ha en systemkamera av enögd spegelreflex typ. Om du kan, så välj ett objektiv med 75-100 mm brännvidd.

4. Ställ kameran på ett stadigt stativ och vinkla in den noggrant. Skärmen bör vara vinkelrät mot syftlinjen.

5. Släck alla lampor i närheten som kan spegla sig i skärmen. Kolla detta noga!

6. Ställ gärna färginställningen på TV-n lite gladare än vanligt.

7. Välj en rätt lång slutartid, åtminstone 1/4 sek. Det räcker för ca 6 bildrutor och det är liten risk att någon bildskarv syns. Med kortare tider råkar man lätt ut för att klippa av en halv bild och med mindre än 1/50 får man aldrig en hel bild belyst ens med halva linjetätheten. TV-n blinkar ju i 50 Hz takt med 312 eller 313 linjer i taget som tillsammans ger en 625 linjers bild på 1/25 sek.

8. Bländare 11-16 kan komma igång med 100 Asa-film. Det brukar gå bra att följa exponeringsmätaren som vanligt.

9. Ha överseende med om din hemma-TV skulle kröka till bilden något. Du kanske kan prova en annan vinkel nästa gång.

Frenne Söderberg

Att använda joystick

Joystick till VIC 20 och Commodore 64 finns i många utföranden, men en sak har de alla gemensamt: tillvägagångssättet vid avläsning. Detta är naturligtvis en oklanderlig fördel, men tyvärr finns det nackdelar med detta avläsningsförfarande på VIC 20. Att läsa av joysticken vållar nämligen lite problem även för den lagom erfarne basicprogrammeraren. Efter lite tänkande och framför allt testande kan ett par basicrader åstadkommas för att klara av nämnda syssla. Inte nog med att de olika riktningarna samt fire läses av från olika register – de ligger dessutom i ordning.

Att skriva en basicrutin som läser av joysticken och lägger upp riktningarna samt fire i lätthanterlig ordning (t.ex. som de ligger på Commodore 64) är som redan nämnt något omständigt och resultatet kommer att ta onödigt lång tid. Tid är som bekant, för basicprogrammerare av spel etc., en stor bristvara man gärna är sparsam med.

I maskinspråksspel spelar de här problemen knappast någon roll och det kan vi även utnyttja i basicspel med en maskinspråksrutin för just avläsningen.

Här är ett exempel på en sådan rutin som även den mest inbitne basicprogrammerare kan använda utan problem. Rutinen är helt reallokerbar, dvs att den kan läggas var som helst i minnet. Jag lägger den för enkelhetens skull på adress \$02A1 (dec. 673) och framåt. \$02A1 – \$02FF är ledigt RAM som ej används av basictolken eller operativsystemet.

Följande program lägger in rutinen på ovan nämnda plats:

```
10 REM JOYSTICK-RUTIN
20 FOR A=673 TO 714 : READ B : POKE A,B : NEXT
30 DATA 162,127,142,34,145,173,32,145,162,255,142,34,145
40 DATA 41,128,74,74,74,133,87,173,31,145,72,41,32,5,87
50 DATA 133,87,104,74,41,14,5,87,74,73,31,133,87,96
```

Efter att rutinen lagts in i minnet kan den anropas med SYS 673. Ifall den läggs annorstädes anropas den med SYS och rutinens begynnelseadress. När rutinen anropas läser den av joysticken och lägger resultatet i byte \$57 (dec. 87). Uppläggningsen är som följer:

Bit nr:	Funktion:
0	upp
1	ned
2	vänster
3	höger
4	fire

Bitarna är satta då de "aktiveras". Även kombinationer bitarna emellan förekommer, t.ex. upp + höger eller vänster + fire.

Exempel på användning:

```
700 SYS 673 : REM ANROPA JOYSTICK-RUTINEN
710 A=PEEK(87) : REM LÄS AV RESULTATET
720 REM KONTROLLERA NU DE ENSKILDA BITARNA
730 IF A AND 1 THEN ... : REM UPP
740 IF A AND 2 THEN ... : REM NED
750 IF A AND 4 THEN ... : REM VÄNSTER
760 IF A AND 8 THEN ... : REM HÖGER
770 IF A AND 16 THEN ... : REM FIRE
```

Avläsning av Commodore 64:ans joystickar bör ej vålla några problem emedan uppläggningsen redan är logisk. Bitarna överensstämmer med resultatet av rutinen här beskriven, men bitarna är inverterade, dvs de släcks då de "aktiveras". För att kunna använda samma programexempel till Commodore 64 får man byta ut rad 700 och 710 mot följande: för joystick 1:

```
700 A=255-PEEK(56321)
```

för joystick 2:

```
700 POKE 56322,0 : A=255-PEEK(56320) : POKE 56322,255
```

När vi använder joystick 2 får vi först ställa om ett datariktningssystem innan vi kan läsa av den. Vi måste återställa det efter avläsningen så att tangentbordsavkodningen fungerar normalt.

Per Andersson

Bengt Holmqvist, Hagalundsgatan 8, 736 00 Kungsör, tel. 0227-13455, skriver och frågar om vi vet om det finns något intresse för en VIC-klubb i KAK-området, uttytt Köping-Arbo-ga-Kungsör. Vi låter frågan gå vidare till läsekretsen, ring direkt till Bengt om du bor i området och vill vara med och starta en klubb!

Andras Mezaros, Folkegatan 17, 2:8, 613 00 Oxelösund frågar samma sak för sin ort. Bor du i hans närhet, var-för inte skriva en rad direkt till ho-nom?

Jono Ragnarsson, Backagatan 5, 270 52 Borrbj vill veta lite om spel-program, jag hoppas han/hon(?) fick sitt lystmäte i förra numret av VIC-rapport, där nästan allt i spelväg som finns presenterades i programka-talogen.

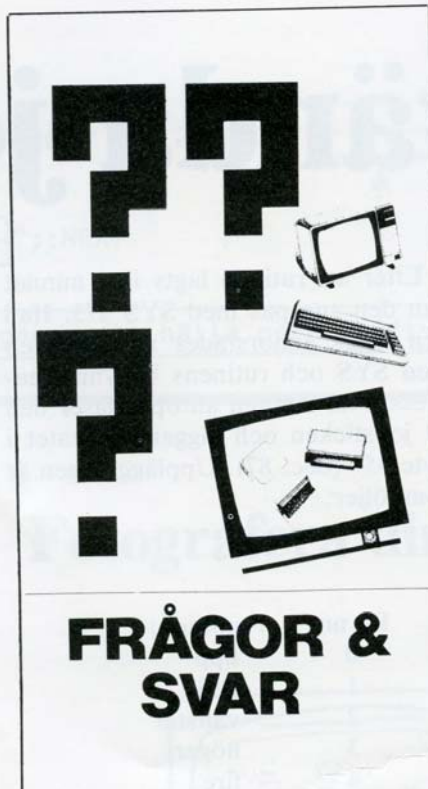
Anders Carlsson, Violvägen 8, 593 00 Västervik vill veta vad som står i bo-ken "Från Spel till Basic". Mitt för-slag är att du försöker hitta boken hos en återförsäljare och tittar i den där, då jag inte vet vad du kan om spelpro-gram kan jag heller inte gärna råda dig vad du behöver för att kunna göra spel!

Programlistningar till spel finns i många olika böcker, följ med i annon-serna i VIC-rapport!

Beträffande frågan om spelpro-grammet i "Lär dig VIC-20" så har jag överlämnat den till våra spelspeci-alister, som just nu tyvärr inte hinner svara, men lovade återkomma i nästa nummer av VIC-rapport.

Christian Jönsson, Axel Danielssons väg 53, 214 74 Malmö, vill veta om PET och VIC-64 kan kommunicera med varandra. Detta kan någon av våra återförsäljare i Malmö svara på i detalj med priser och annat, men visst är det möjligt, de har ju båda samma processorchip, 6502 och det är inte så svårt att få program för PET att fun-gera på en 64:a eller tvärtom, bortsett från att alla minnesadresser ju får ändras, då 64:an har ett annat sätt att placera program och data än PET och dessutom har både färg- och musik-funktioner, som inte finns på PET-en.

Jens Tjernström, Kungsvägen 49, 930 10 Lövnånger, frågar om det går att koppla upp VIC-64 mot QZ-centralen och här får du kanske vänta tills



VIC-MODEMET är ute i handeln el-ler använda den nya kabeln som går till Televerkets modem. Fakta om printrar kan du få från närmaste VIC-återförsäljare. Modemet till VIC kommer att presenteras i ett kom-mande nummer, när alla detaljer och tillstånd är klart.

Jimmy Engelbrekt, Mellanstigen 3, 578 00 Aneby, undrar om vi har några klistermärken för VIC-20 och frågar också efter närmaste VIC-klubb. Aneby ligger i Jönköpings län, om nu någon inte visste det! Tyvärr, ännu finns inga speciella dekaler eller klis-termärken till VIC, men vi har disku-terat det. Klubbfrågan får vi lämna till läsarna, bor ni i närheten, hör av er till Jimmy!

Lars Benjaminsson, Växelmyntsgatan 54, 414 83 Göteborg, undrar varför han inte fått VIC-Nytt nr 2-83. För-modligen för att vi haft mycket svårt att hinna ifatt alla utskick på grund av att vi haft mycket stor personalfrån-varo under sommaren och att detta nummer kom mycket sent. Nu hop-pas vi att Lars fått både nr 2 och den till VIC-rapport omdöpta nr 3-83. Be-träffande frågan om programbytar-servicen så hänger den samman med att verksamheten i att från Göteborg

styra en rikstäckande VIC-klubb vi-sade sig överstiga våra möjligheter och att därför beslutats att medlem-mar som betalt avgiften under 1983 i stället får en helt fri prenumeration på tidningen under ett år. Från VIC-rap-ports sida kommer vi i stället att på alla möjliga sätt stödja bildandet av lokala VIC-klubbar genom att lämna utrymme för klubbnyheter, ge tips om hur man sköter en klubb, aktiverar medlemmarna och överhuvudtaget får en lokalklubb att fungera samt hjälper till med råd och sådant som vi kan göra, t ex ge tips om kontaktper-soner osv.

Jan Karlsson, Ranstorpsgratan 19, 582 45 Linköping är dataingenjör och vill ha så mycket tekniska data som möjligt om 64:an med kringutrust-ning. Tyvärr Jan, det som finns på VIC-rapports sidor är vad vi har möj-lighet att bidra med, resten får du hö-ra med närmaste VIC-representant om. Han skall ha alla nödvändiga uppgifter. I nästa nummer av VIC-rapport kommer en utförlig pre-sentation av en mängd nya böcker vi fann vid besök i England på Personal Computer World Show i slutet på sep-tember.

Assistant General Manager Ulf Sved-berg vid NCC Yanbu i Saudi-Arabien vill att vi skall göra program åt honom och sända det per telex. Tyvärr måste vi göra honom besviken, redaktionen har ingen telex ännu... vi får därför inskränka oss till ett svar här i VIC-rapport, som vi hoppas når ho-nom där nere i värmen: Ditt önskemål om ett program för en matris med fyra kolumner och 10 rader är inte så lätt tillgodosett om du inte ger mer infor-mationer. Jag förslår att du hör med t ex någon av pojkarna som annonse-rar i VIC-rapport om någon har tid att hjälpa dig, kanske får du betala en slant för att det skall gå lättare...?

VARNING! En läsare skriver från Skåne och berättar att han köpt en VIC-64 på småannons i en dagstid-ning. Han fick en splitter ny maskin med lika ny bandspelare. Medföljan-de garantikort var icke ifyllt och sälja-ren berättade att han medvetet under-låtit att sända in garantikort för att på så vis förlänga garantitiden! I själva verket kan det gott hända att maski-nen var piratimporterad, och då gäller

inga garantier utan Handic tar fullt pris för alla reparationer. Läsaren vill att vi skulle varna alla unga spekulanter som kanske inte uppmärksammar eller som "går på" mer eller mindre sanna förklaringar från säljare som inte är helt snövita i själen.

Fredrik Börjesson, Solängsvägen 49, 191 54 Sollentuna vill veta allt om vad som finns till VIC-64. Vi kan bara upprepa: följ med i VIC-rapport så skall allt så småningom vara uppenbart för dig! En bra Daisy-wheel-printer kostar många tusenlappar med ett interface, jag hörde att Olivetti 35 och 45 lär gå att få med ett interface som fungerar för något under 10.000 kronor inklusive moms.

A. Ståhl-Mousa, Kornettgatan 167, 654 71 Karlstad ville veta en hel del om datorer och om programmering och om VIC-ar. Vi föreslår att han hör med en auktoriserad VIC-återförsäljare och att han börjar med att läsa någon introduktionsbok om datorer, t ex Jan Älmeby "Mikrodatorn, en introduktion", som utkommit på Studentlitteraturs förlag. Den finns i bokhandeln.

Per Larsson, Hanåsvägen 3B, 280 10 Sösdala vill ha hjälp med att göra program som inte går att lista utom med en särskild kod. Ledsen Per, egentligen finns inget bra listskydd i kodform, men om du följer med i våra spalter kommer du då och då att hitta programförslag som visar hur du skall bära dig åt för att listskydda program mot alla som inte läser VIC-rapport! (Eller är kluriga hackers ändå.)

Anders Granström ringde och frågade hur man länkar ett vanligt BASIC-program (t ex en subrutin) till ett annat vanligt BASIC-program. Med risk för att bli tjatig måste jag återigen hänvisa till VIC-rapports spalter. Förr eller senare dyker just den biten upp som DU letar efter. Fortsätt att läsa bara!

TEKNIKTIPS En medlem i VIC-klubben skriver och berättar att om man spelar VIC 1907 Superlander, så bör man försöka hålla nålen så långt upp som möjligt. Medlemmen sade att han därigenom nått upp till 5620 poäng!

TÄVLING Den föregående notisen gav idén till en tävling mellan läsarna

om vem som nått högsta poängresultatet i de olika spelen som finns till VIC-20 och 64:an. **SKRIV OCH BERÄTTA SÅ PUBLICERAR VI NAMNEN PÅ DE SOM HAR HÖGA POÄNGTAL.** OBS! Poängtalet bör vara bevisat av minst en myndig person, dvs någon över 18 år. **VÄLKOMMEN MED DITT SVAR!**

Hans Hjertqvist, Dalbogatan 3A, 431 38 Mölndal, vill skriva program i Pascal på sin 64:a. Han undrar när det kommer någon plug-in och vad den kommer att kosta. Sen vill han veta vilka matematikfunktioner som finns till 64:an.

Pascal lär väl knappast komma till 64:an nu, när COMAL är så nära förestående och detta språk kan sägas ha vissa fördelar framför Pascal förutom att det kommer att bli svenskt skolspråk framöver. Vi väntar att plug-in-nen med COMAL kommer till jul och att den kommer att kosta under tusenlappen. Men dessförinnan väntas en enklare mjukladdad variant av COMAL med version nummer 0.12, dvs den rena enkla COMAL-kärnan enligt specifikationen COMAL-80. Denna mjukladdade ryktas bli fri för allmänheten ("in the public domain") och det kan förväntas att någon får i uppdrag att mot kostnadstäckning distribuera den till den som så önskar. Datatronic distribuerar ju redan i dag genom PET-klubben den mjukladdade versionen av COMAL som passar till PET-datorerna och något liknande kommer säkert att göras även för 64:an, hoppas vi. Beträffande matematikfunktionerna så måste jag nog hänvisa till de speciella handböcker, t.ex. VIC-64 Användarmanual.

något han kallar "System Extension Diskett". Hans återförsäljare har den inte. Jag ber om förlåtelse att jag är okunnig, men ivrigt letande i senaste prislistan från Handic om 64:an eller VIC-20 ger ingen ledtråd. Vill Du vara snäll och ge mig lite mera "kött på informationsbenet", Pål?

Lennart Bringfeldt, Myrstadsvägen 20, 560 27 Tenhult, frågar efter redovisnings- och lagerprogram m.m. på VIC-20, och där måste jag tyvärr säga att det verkar litet magert här i Sverige. I England finns ett flertal program som vänder sig till småföretagares behov av enkla redovisnings- och faktureringsprogram, men såvitt bekant importeras de inte av något företag. Du kan dock själv skriva till någon engelsk firma, adresserna hittar du i datortidningarna t.ex. VIC Computing, Practical Computing eller kanske t.o.m. i Educational Computing. Fråga efter dem hos din tidningsförsäljare eller bokhandel.

Till 64:an finns åtminstone från Handic ett redovisningsprogram och ett registerprogram utom det omskrivna Calc Result, som finns både i en enkel och i en fullständig version som har 3-dimensionell funktion.

Jakob Lassen, Ö Fäladsvägen 23, 223 60 Lund, vill också ha uppgifter om kringutrustning och tillbehör. VIC-handlarna har en alldeles utomordentligt utförlig prislista, som de fått nyligen. Fråga efter den hos din handlare.

Han frågar också om man kan använda en enkel skivenhet. Jodå, det går utmärkt, men man skall vara medveten om att man då får långa "back-up"-tider om skivorna är fulla när man gör sin skyddskopiering – för det gör man väl...? 64:an går inte att bygga ut mera i RAM.

Jakob frågar om VIC-klubb i Lund, men det vet vi inte om någon. Tyvärr, för det kanske finns ett eller flera gäng med entusiaster, som träffas i största tysthet, kända bara av sig själva. Hör av er i så fall till VIC-rapport, ni får gratis spaltutrymme för er klubbansons!

Frågan om plug-in för att få fler BASIC-kommandon gäller säkert den plug-in, som i katalogen kallas Simons basic 64. Tala med din handlare.



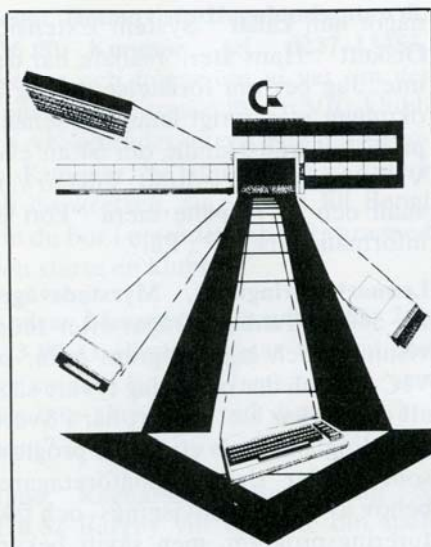
TEXT!

Ölands tryckarna ab

0485-100 40

KOMMUNICERA MED OSS VIA
MODEM OCH LÅT DIN VIC 64 BLI
EN FOTOSÄTTARE.

Pål Börje, Kristinebergsvägen 21, 572 00 Oskarshamn, är intresserad av



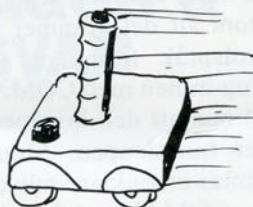
SPELNYTT

NIGHT DRIVER

Ett bilspel av arcadtyp. Du skall köra en bil mitt i natten och vid sidan av vägen står det stolpar. Det gäller att inte köra på dessa stolpar. Till din hjälp har du instrument såsom hastighetsmätare, vägmätare och klocka. Spelet är tidsbegränsat till 99 tidsenheter men om du under den tiden lyckas köra 6 km får du bonustid.

Spelet kräver joystick eller paddles.

Speltips: När du har kommit upp i ungefär 170 km/h så blir det ganska svårt att hinna med att svänga. Detta kan korrigeras genom att släppa och trycka ner skjutknappen med jämna mellanrum.



Styra "gubben" kan man göra med tangenter eller med joystick. Joysticken rekommenderas. När man styr med tangenterna använder man 3 för upp, x för ned, < för vänster, > för höger och mellanslagstangenten för att släppa bomber med. Styr du med joysticken måste den vara i port ett och man släpper bomber med avfyringsknappen på joysticken.

Jag hoppas ni tycker att Panic är lika roligt som jag tycker det är. Det finns säkert fler knep i Panic som jag inte har upptäckt. Jag önskar alla Panic-spelare lycka till!

Tvilling

Så här ser spelet ut:

1. Din "gubbe"
2. Bomb
3. Monsterna
4. Här startar "Gubben"
5. Här startar monsterna på 1:a till 4:e graden
6. Här startar monsterna på 5:e graden
7. Poäng
8. Vilken spelares tur det är
9. Här anges den högsta poängen
10. Hur många "gubbar" du har kvar
11. Bomber kvar
12. Vilken level (svårighetsgrad) du är på.

PANIC!

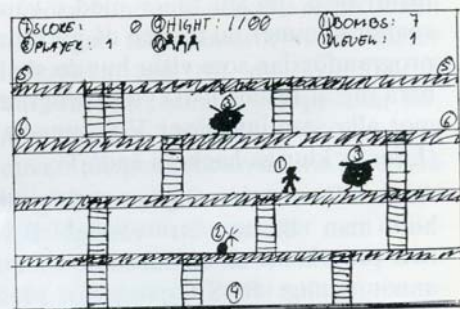
Panic är för mig ett nytt sorts spel. I spelet är man en liten "gubbe" som springer runt på fyra våningar, ett bottenplan och på stegar som finns mellan dom olika våningarna. Din uppgift är att göra hål i våningarna så att två monster faller ner i hålen och dör. Akta dig för monsterna. Rör dom dig, dör du. Man gör hål i våningarna med bomber som man placerar ut. Men här får du akta dig, bomberna har kort stubin. Men bomberna har inte så kort stubin att du kan lägga en bomb mindre än 4-5 cm ifrån monstret när det jagar dig. Du får också se upp med att spränga sönder vägen framför dig, så att du inte kan ta dig upp på nästa våning. Kom också ihåg att det inte går att spränga sönder stegarna eller bottenplan. Monsterna dör inte heller av själva explosionen. Ha alltid en utväg om du inte är säker på att bomben kommer att explodera i tid. Slösa inte heller med bomberna, du har bara ett visst antal.

Panic är uppbyggt i olika svårighetsgrader. För varje svårighetsgrad måste monsterna falla en våning för att dö. Men femte graden och högre

grader är ett undantag. Då måste monsterna bara falla fyra våningar, då börjar monsterna på tredje våningen (femte svårighetsgraden) och "gubben" på bottenplan (som vanligt). På sjätte svårighetsgraden börjar monsterna på andra våningen och på sjunde graden börjar monsterna på första våningen. Då måste man ha tur med stegarnas placering. Det hade inte jag, så jag kom inte längre. En taktik, när man börjar på femte graden och högre grader, är att försöka ta sig upp på översta våningen så fort som möjligt, så man får lite springrum. "Gubben" rör sig fortare än monsterna. I fall du skall ta dig mellan flera våningar, utan uppehåll, så välj dom långa stegarna så du slipper ödsla tid på stegbyte.

Man får 50 poäng per våning som monsterna faller. Är man t ex på andra svårighetsgraden måste monsterna falla två våningar för att dö.

Det ger för båda monsterna sammanlagt 200 poäng (50+50+50+50). Men om man låter monsterna falla mer än vad dom behöver, t ex tre våningar (50 poäng extra per monster), får man mer poäng. Högsta möjliga poäng på andra våningen är alltså 400 poäng. Då måste monsterna falla fyra våningar.



GRIDRUNNER

Ett höghastighetsspel som utspelar sig ute i rymden.

Året är 2190 och Jorden har skjutit ut enorma kraftstationer i omlopp runt jorden. Dessa har som uppgift att försörja Jorden med den enorma energi som den förbrukar. De kallas GRIDDEN p.g.a. sitt rutmönstrade utseende.

Plötsligt slutar Gridden att leverera sin energi. En undersökningspatrull skickas upp och upptäcker att Grid-den har invaderats av rymdvarelser, droider. Dessa droider använder energin i Gridden för att föröka sig och sedan invadera Jorden.

Paniken är nära. Skickliga ingenjörer, professorer och vetenskapsmän lyckas dock konstruera ett skepp för att bekämpa droiderna. Detta skepp kallas GRIDRUNNER. Detta är ett snabbt och kraftfullt skepp som kan tillgodogöra sig griddens energi. Energin används bl.a. till att bestycka plasmakanonen som den är försedd med.

Din uppgift som pilot är att tillintetgöra så många droider som möjligt och därigenom avskräcka dem från att anfälla Jorden. Uppdraget verkar kanske lätt, men låt dig inte luras. Gridden bevakas av griddsökande patruller som korsar Gridden fram och tillbaka, Pods som växer och till slut faller mot dig och till sist X/Y Zappers som går längs två kanter och avfyrar sina dödliga strålar.

Allteftersom du klarar av en attackvåg kommer en ny, lite svårare.

Spelet har bra grafik och mysiga ljudeffekter. Det är inte ett spel som man lär sig på en dag, utan det tar lång tid och mycket övning att behärska.

Spelet kräver joystick.

Speltips: När du spelar så försök att hålla till på höger sida av spelplanen, för dit skjuter inte Y/Zappersen så ofta. Försök också lära dig X/Zappers skjutsystem för varje ny attackvåg. För att effektivt skjuta ner patrullerna som korsar spelplanen, kan du ställa dig näst längst till höger och skjuta effektivt och sedan flytta ett steg åt höger. Då bildas först en ränna där patrullerna söker sig ner. Genom att sedan flytta längst till höger förgör du dom effektivt.

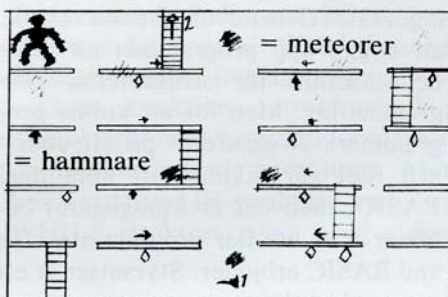
CRAZY KONG

"Donkey-kong" och "Donkey-kong-junior" spelades först i spelhallarna. Sedan kom "Game and Watch"-spelen och "Donkey-kong" introducerades även där.

64:ans modell heter "Crazy-kong". Din uppgift i spelet är att befria din flicka som Kong fångat. Du måste klara av 3 kartor för att befria henne. För att kontrollera din "gubbe" är joy-stick bäst, "fire" är då "hopp". För att "hopp" ska fungera måste du hålla joy-sticken åt något håll (höger/vänster, upp/ner).

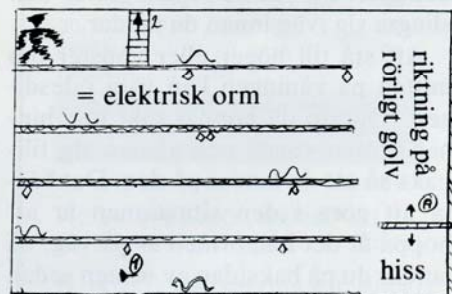


På den första kartan skall du klättra upp för 4 våningar och plocka ner alla diamanterna, som hänger i taket på varje våning. Du måste plocka ner alla diamanterna annars kan du inte klättra upp för den sista stegen. Du tar ner diamanterna genom att hoppa upp på dem. För att hindra dig att komma upp till översta våningens steg, kastar Kong ner tunnor på dig. Tunnorna rullar ner samma väg som du skall upp, det ger dig en "friplats" bredvid alla stegarna. Till din hjälp mot tunnorna har du "hamrarna". "Hamrarna" är placerade i taket på 1:a, 2:a och 3:e våningen. Hamrarna tar du ner på samma sätt som diamanterna. När du plockat ner en hammare kan du krossa tunnorna, här får du se upp. Man vet inte när hamrarna slutar verka. En hammare kan sluta verka precis före eller efter en tunna. På översta våningen slutar hamrarna att verka. Det finns inga diamanter på översta våningen. Det enda du skall göra här är att hoppa över ett par tunnor och klättra upp för den sista stegen. När du är längst upp på stegen hoppar du uppåt och har i och med det klarat



av kartan. För varje diamant du plockar ner får du 50 poäng på den här kartan och varje krossad tunna är värd 10 poäng.

På nästa karta finns det både diamanter och hammare att plocka ner. Diamanternas värde är 100 p per styck. Hamrarna används till att krossa de meteorer som Kong den här gången använder sig av för att hindra dig. En krossad meteor är värd 20 p. Meteorerna kommer nedfärande från översta delen av mitten på skärmen. Plötsligt kan en meteor svänga och åka längs golvet på vilken våning som helst. Mest svänger meteorerna på de översta våningarna. De största delarna av våningarnas golv rör sig hela tiden mot mitten av skärmen, det måste man kompensera genom att hålla joy-sticken åt motsatt håll. Det är inte hälsosamt att trilla ner från en våning. Det finns inga friplatser på den här kartan, men säkrast står man på någon av våningarnas ytterdelar, där golvet inte rör sig. Ett knep för att komma upp för en steg där golvet rör sig är att placera sig så att om man släppte joy-sticken skulle man glida mot stegen, sedan håller du joy-sticken uppåt och glider mot stegen. När du har kommit fram till stegen klättrar du automatiskt upp för stegen. Väl uppe måste du akta dig så att du inte glider över någon kant. När du har kommit upp till de översta våningarna börjar meteorerna falla rätt igenom våningarna. Det är inte lätt att klara av den sista stegen. När du har gjort det har du klarat den andra kartan.



Den sista kartan är den lurigaste. Det finns inga hammare, bara diamanter som är värda 150 p per styck. På den här kartan använder sig Kong av elektriska ormar som slingrar sig fram och tillbaks på alla våningar utom den översta. För att ta sig mellan våningarna skall du använda hissen som

forts. nästa sida



"Alfalopp" är ett färgstarkt, spännande spel, som hjälper dina tre- till sexåringar att lära sig alfabetet. Programmet skrevs ursprungligen för grundmodellen av VIC-20 (5K) och det finns också en version för 64:an.

När jag försökte göra min VIC-20 till något mer än bara ett tangentbord för mina ungar att knacka på, skrev jag detta lilla program för att få apparaten att göra något samtidigt som den ger en möjlighet att träna in alfabetet. Det betyder att "Alfalopp" är avsett för relativt unga människor, från tre till sex år gamla. För ett barn, som håller på och lär sig alfabetet, är uppgiften att trycka på rätt knapp för att starta en kapplöpning mellan loppor ganska spännande.

Meningen med spelet är att man skall trycka på tangenten med samma bokstav i alfabetet som den, som vi-

forts. från sid. 39

åker upp och ner till höger på skärmen. Det är lättast att hoppa av och på när hissen är på väg upp. Eftersom ormarna slingrar sig fram och tillbaks kan du ställa dig vid kanten av våningen och hoppa rakt upp när ormen kommer. Då hinner ormen vända och slingra sig iväg innan du landar.

Att stå till höger eller vänster om mitten på våningen kan vara ödesdigert. Om du då hoppar rakt upp hinner ormen vända och slingra sig tillbaks så att du landar på den. Det bästa att göra i den situationen är att hoppa åt det håll ormen är på väg, då landar du på baksidan av ormen sedan den vänt. På översta våningen finns det ingen orm. När du har klättrat upp för den sista stegen faller Kong ner från den översta våningen och krossas mot golvet. Du har räddat din flicka.

Spelet börjar om igen men med en diamant extra på varje våning...

LYCKA TILL!!

Jesper Fredriksson

sas på bildskärmen. När rätt knapp har tryckts ned börjar kapplöpningen mellan fem olikfärgade loppor på bildskärmen. Om fel bokstavstangent trycks ner tutar datorn och väntar på att rätt tangent skall tryckas ner. Loppornas turordning i mål markeras vid slutet av varje rad, så spelaren/spelarna också kan tävla om poäng genom att gissa vinnaren. I slutet av varje kapplöpning frågas spelaren om han/hon vill spela ett spel till. Här svarar man J för ja och N för nej. Även här tutar datorn till, om fel svar matas in.

I det program som följer väljer datorn automatiskt bokstäver med början vid A och återgång till A när man nått Z. Om på rad 76 REM tas bort, kommer programmet att välja bok-

stäver slumpmässigt, hur som helst. Programmet kan också ändras så, att det frågar spelaren vilken metod han/hon vill ha för bokstavsval. Dock har jag funnit, att ju mindre hjälp barnet får, desto lättare tycker det att det är! Kom ihåg, detta spel är ju avsett för barn som just håller på att lära sig alfabetet eller som nyss börjat lära sig läsa. Som exempel kan jag nämna, att med urvalsmetoden bestämd i programmet har min femåriga dotter kunnat ladda och köra programmet utan någon som helst hjälp.

Detta enkla program kan ge barnen många timmars nöje när det hjälper dem lära sig alfabetet. Men se upp! Du själv kanske inte får tillgång till din dator förrän ungarna somnat...

Programlistningen finns på sidan 44. Den finns också att få på kassett, sätt in 25:- på pg. 720300-3, Åke Fredriksson, och skriv om Du vill ha för VIC 20 eller 64:an. Kassetten sänds portofritt.

COMAL-skolan

I förra numret av VIC-rapport introducerades COMAL, och några av språkets fördelar gentemot andra språk (som t.ex. BASIC) nämndes. I den här artikeln skall jag gå igenom några grundläggande program-satser, och hur man kan använda dem för att skriva ett program.

När man skriver in ett COMAL-program använder man sig av radnummer och av samma bekväma skärmeditor som i BASIC. Att hålla ordning på programraderna, samt att användas av några direkt-kommandon (exempelvis DEL och AUTO), är radnumrens enda funktion. De hänvisas aldrig till av ett program. VAD-DÅ? Kan man inte använda GOTO? Då går det ju inte att programmera!! Jo, det går det visst! Med hjälp av styrsatser och procedurer bygger man upp programmet av små enheter. Om man gör detta på ett riktigt sätt skall inget GOTO behövas. Denna teknik, att bygga upp programmet av olika delar, kallas för strukturerad programmering. Men för att kunna programmera strukturerat på ett enkelt sätt (det går faktiskt att göra med BASIC, men det är krångligare) behöver man lite fler programsatser än vad BASIC erbjuder. Styrsatser är ett exempel på detta.

Jämförelse eller tilldelning

I BASIC används samma tecken för tilldelning som för jämförelse. I COMAL, liksom i Pascal och en del andra språk, är så inte fallet. Om man skall jämföra två värden eller strängar används '=' (lika med), men om man skall tilldela en variabel ett värde används ':=' (kolon, likamed). En anledning till att inte ha ett enkelt '=' också för tilldelningar, är att man ibland då skriver uttryck som matematiskt sett inte kan stämma. $X=X+1$ (tilldela värdet av uttrycket $X+1$ till variabeln X) är ett exempel. Vi som sysslar med datorer vet att det ger som resultat att variabeln X ökas med 1, medan det säkert ser knepigt och omöjligt ut för många människor. Man behöver inga större kunskaper i matematik för att förstå att X inte kan vara lika med $X+1$. Vad som sker är alltså en tilldelning och inte en jämförelse eller en matematisk likhet, och detta har som sagt sin egen symbol i COMAL. Exemplet lyder då: $X:=X+1$.

Styrsatser

COMAL erbjuder flera olika satser för att styra programexekvering. Några av dem skall vi gå igenom här.

IF-satsen används när ett villkor skall bestämma om vissa programrad-er skall utföras eller ej. Exempel:

IF X=1 THEN

(här kommer de rader som skall utföras om X är lika med 1)

ENDIF

Ett godtyckligt antal rader kan finnas mellan IF och ENDIF, och de kommer att utföras endast om uttrycket är sant (i exemplet ovan om X=1. När raderna har exekverats, eller om uttrycket efter IF var falskt, fortsätter programmet på raden efter ENDIF. Detta var väl rätt enkelt? Det är dock inte allt som IF-satsen har att erbjuda. I ett program kan det hända att man vill utföra vissa rader om ett villkor är sant, vissa andra om det är falskt. Man använder då nyckelordet ELSE (annars) i IF-strukturen. Exempel:

IF BETYG = 5 THEN

(Här kommer rader som utförs om BETYG = 5)

ELSE

(här kommer rader som utförs om inte BETYG = 5)

ENDIF

Raderna mellan ELSE och ENDIF utförs alltså om villkoret är falskt. IF-strukturen har ytterligare en möjlighet, ELIF-satsen. ELIF är en förkortning av ELSE IF (annars om) och följs av ett uttryck. Detta betyder att om inte första uttrycket var sant, kollas uttrycket efter ELIF. En IF-struktur kan innehålla flera ELIF-satser och dessa kan följas av en ELSE-sats. Låter det krångligt? Exemplet nedan gör det nog lite klarare.

IF NUMMER = 1 THEN

(om NUMMER = 1 kommer vi hit)

ELIF NUMMER = 2 THEN

(hit kommer vi om NUMMER = 2)

ELIF NUMMER <= 10 THEN

(raderna här utförs om NUMMER är mellan 3 och 10)

(eftersom NUMMER = 1 och 2 redan kollats)

ELSE

(här kommer rader som utförs om NUMMER är mindre än 1 eller större än 10)

ENDIF

Med hjälp av IF-satsen kan man alltså utföra vissa programrader beroende på bestämda villkor. Genom att kombinera IF med ELIF och ELSE skapas en struktur där det är lätt att se vad som händer för olika villkor. Man behöver alltså inte hoppa kors och tvärs i programmet.

Det räcker emellertid inte att kunna välja olika möjligheter på detta sätt. Ofta vill man också upprepa ett antal rader. Man kan vilja upprepa dem ett bestämt antal gånger, eller tills att ett villkor blivit sant eller falskt. I det första fallet använder man, liksom i BASIC, en FOR-sats. Den ser ut på följande sätt.

FOR I := 1 TO 50 DO

(här kommer rader som skall utföras 50 gånger)

ENDFOR

Skillnader från BASIC är att nyckelorden DO och ENDFOR används. Det förra avslutar raden med IF (DO betyder "gör!"), och det senare avslutar strukturen. Det heter ENDFOR för att det skall vara enhetligt med andra styrsatser i COMAL. Observera också att tilldelnings-symbolen ':=' används istället för '=' (I tilldelas ju värden från 1 till 50). Förutom att upprepa exekvering av rader ett bestämt antal gånger, kan man också upprepa så länge ett bestämt villkor är sant. Det finns tre strukturer som man kan använda till detta i COMAL (de är lite olika men fungerar på liknande sätt), men jag skall bara beskriva ett av dem nu, WHILE-satsen. Den används på följande sätt.

WHILE TAL > 0 DO

(rader som kommer här utförs "medan" TAL är större än 0)

ENDWHILE

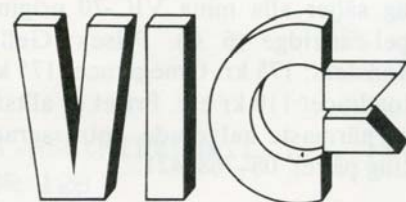
Raderna mellan WHILE och ENDWHILE kommer att utföras tills dess att TAL blir 0 eller mindre. Om TAL redan är 0 eller mindre när programmet kommer till WHILE-satsen kommer raderna inte att utföras.

Ovan beskrivna strukturer kan kombineras med varandra för att skapa mer invecklade strukturer. Man kan t.ex. ha en IF-struktur "inuti" en WHILE-struktur, osv. Med dessa

strukturer kan man lösa de flesta problem som kan uppkomma, och man behöver inte använda GOTO. Observera också att även om villkoren i exemplen ovan är enkla, kan man som i BASIC kombinera olika villkor med t.ex. AND och OR.

Mycket nytt

Mycket nytt har introducerats nu, och det är kanske svårt att förstå (och förklara!) allt på en gång. I nästa nummer kommer jag därför att visa ett program-exempel som använder de strukturer som blivit beskrivna ovan. Genom att gå igenom det kommer det säkert att klarna. Då kommer också flera detaljer som skiljer sig från BASIC att behandlas.



SPEL TILL VIC20 - 64
RING
0413-21030

KRAZY KONG / VIC 20 16K	125:-
KRAZY KONG / VIC 64	125:-
SKRAMBLE 64 / VIC 64	125:-
SLAP DAB / VIC 20 oexp	99:-
MINI KONG / VIC 20 oexp	99:-
3D TIME TREK / VIC 20 16K	99:-
3D TIME TREK / VIC 64	99:-
FROGRUN / VIC 20 oexp	99:-
FROGRUN / VIC 64	99:-

DUNGEONS, nytt textadventure med 100 rum att utforska. Möjlighet att spara dina färder på band!!!

För VIC 20 16K	99:-
VIC 64 GAMES BOOK, spelbok	135:-
VIC 64 EXPOSED, ny bok	155:-

Vi skickar vår katalog mot dubbelt svarsporto.

DIGICOM
BOX 194 S-243 00 HÖÖR

Du kan även köpa våra produkter hos välsorterade fackhandlare t.ex. DISKETT i Malmö * ABS MIKRO i Eslöv

Säljes

4 bra spel + ett gissningsspel på oexp. VIC-20. Sätt in 40:- på pg 25 77 31-0, så skickar jag kassett.

Säljes eller bytes

Program till VIC-20. Om du vill byta så skicka först lista över dina program till:

Dean Tosic, Lokvägen 37, 260 33 PÅARP

VIC-20 spel säljes!

Spel till VIC-20 säljes för endast 20-30 kr/spel. Ex: Subhunt, Anaconda, Crazy Kong, Frogrun m.m. Ring eller skriv till: Patrik Hansen, Grängsgat. 74, 235 00 Vellinge, Tel. 040/422164 och begär GRATIS katalog.

VIC-20 plug-in VIC-20

Jag säljer alla mina VIC-20 original spel-cartridge (6 st). Pris: Gelly-Monsters: 175 kr. Omegarace: 175 kr, Roadrace: 110 kr etc. Priset är alltså i det närmaste halverade. Intresserad? Ring på tel. 08-7684421.

VIC 64

Mastermind+ Fantasy + Skatteberäkning -83+ET (skit-snygg med 8 sprites) + Joystickskontroll. Sänd 100 kr kont. eller check eller postgiro till J. Sandstedt, Peppargatan 1, 424 39 ANGERED. Märk kuvertet med ett "A".

VIC 64 spel säljes

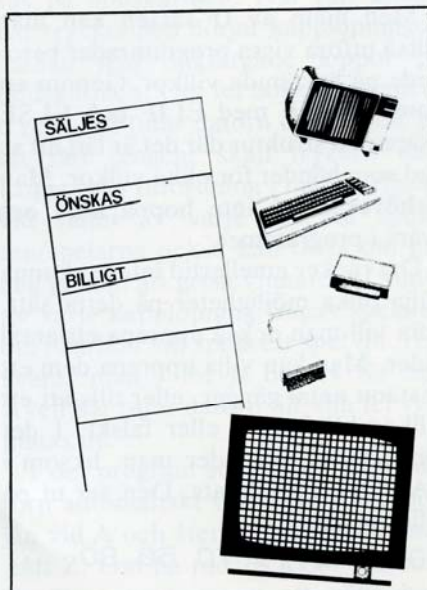
Utan listskydd. Skrivna i Basic och maskinkod. Billiga! Roliga! Lärarika! Skicka mig 15 kr och jag skickar dig spelet Alfa Bandit + spelförteckning. Slå till! Markus Jakobsson, Sportvägen 33, 240 21 LÖDDEKÖPINGE. Tel. 0418/30126.

Prisbomb på VIC-20

VIC-20 med kassetstation och litteratur plus massor av program säljes förmånligt. Värde nypris cirka 6.000:- säljes för 3.990:- eller till högstbjudande. Tel. 031-411842.

Köpes

Temp. mätare till VIC-20 + 50° - 50° C. Att kopplas till paddelängden även förslag eller ritning till dito. Arne Nilsson, Storgatan 70, 520 50 STENSTORP. Tel. 0500/50986.



GRATIS ANNONSER

OBS! Här får endast privatpersoner annonsera. Tele.nr. eller adress måste anges. Endast en annons per nummer av VIC-rapport.

Bytes: Advenger, Slot, Super-Lander.
Köpes: VIC-printer till rätt pris! Jag säljer också program.
Tel: 08/637717. Fråga efter Peter.

Text och demospel på VIC-20, demoprogram 25 kr på kassett 100 kr. "Reläsladden" kan beställas per pg 622521-3, Frenne Söderberg, Box 123, 437 00 LINDOME.

VIC-64

Bra M-kodspel, Adventurespel och Nyttoprogram. Säljes eller bytes mot likvärdiga program. Tel. 031-524847 (Lars-Inge).

VIC tillbehör säljes

VIC 1010 exp. enh. 950:-, Super Expander 400:-, Pris kan diskuteras. Tel. 031-230144, Robert.

VIC-20 säljes

Dator med bandspelare, böcker, superexpander, 2 st plug-in spel samt ca 200 spel på kassettband. Pris 3.000:- eller högstbjudande. Tel: 0940/123 67.

VIC-20 Spel

Säljes ex. VIC-man, Masken, Poker m.m. Beställ lista. Christer Carlsson, Lingonstigen 101, 951 55 LULEÅ.

VIC-20

Ljuspenne med spel 350:-, Bugbyte-spel till VIC-20 40:- per styck. Programmeringshjälpmedel i eprom. Vickit 1, 2, 3 250:- per styck. VIC-20 med spel 1800:-. Tel. 08/509001.

BYTES

Star Battle, Sargon schack, Woodo Castle bytes mot Road Race, Mission impossible, Jelly Monster eller Superexpander + Mission impossible eller förslag. Skriv (p.g.a. resa) till: Lars Benjaminsson, Växelmyntsgatan 54, 414 83 GÖTEBORG.

Säljes

Program till VIC-20 säljes eller bytes. STORT SORTIMENT. Ring efter kl 17 till Robert 031-230144 eller Peter 031-877087.

Säljes

Olika spel av bra kvalitet. Mycket billigt! Skriv till Dean Tosic, Lokvägen 37, 260 33 PÅARP.

Säljes

VIC-20 spel säljes, 3 st för 50 kr, Ormen, Othello, Frogrun, Logger, Ladders, Kuben, Missile Command, Arrow, Ta stickor. Vilka vill du ha pg 4592640-9, Tore Gullstrand.

Tillbehör VIC-20

Tillbehör inhandlades i juni -83. 1 st Användarmanual 50 kr. 1 st Programmeringshandbok 100kr. 1 st Spel Super Lander 200 kr. (Pluginkasset). Ring 0451-60426.

VIC-20 skrivarex

Med 43 tecken 1.200 kr med interface 16K RAM och 4 exp.-portar med strömmar 995 kr. Tel. 0418-25565 efter kl. 18.

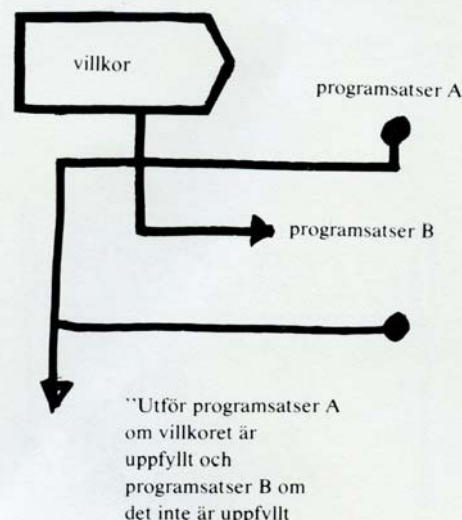
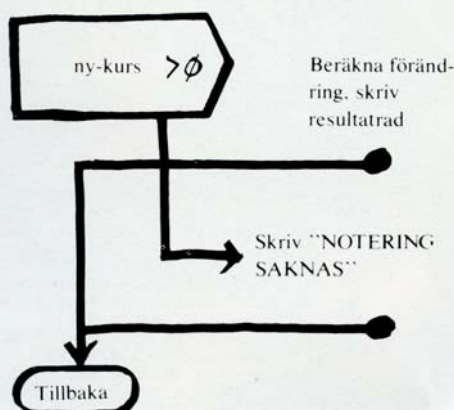
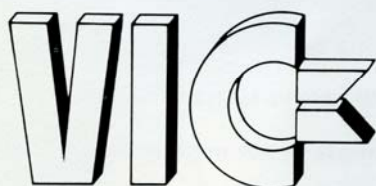
Spel till VIC-20

4 st spel säljes för 300 kr. Amok, Galna Robotar i en rymdstation. Bonzo, panikspel med Hans och monster. Roadtoad, ta fem grodor till hamnen. Millipede, krig i trädgården. Telefon 0591-13676.

VICtiga rättelser

Tyvärr smög sig några fel in i första numret av VIC-rapport, som vi vill korrigera här:

Figurerna på sidan 8, den övre och undre fick ett extra streck som inte var meningen att de skulle ha. Så här skall de egentligen se ut



Sidan 30, programlistningen över texten saknade det avslutande strängtecknet (") på raden 150, men kanske har den listige VIC-rapportläsaren redan lurat ut detta med hjälp av "bild-texten".

Ett värre fel smög sig in i artikeln om minnesorganisationen i VIC-64 på sidan 37, där programlistningarna högst upp måste vara totalt obegripliga för vem som helst som inte sett originalet. Dessutom fattas en programsnutt längst ned i tredje spalten på sidan 36. Förlåt oss våra missar! Här kommer i alla fall de rätta programraderna:

Sidan 36, efter texten "Följande BASIC-program utför kopieringen:"

```
10 FOR I=40960 TO 49151
20 : POKE I,PEEK(I)
30 NEXT
```

Sidan 37, efter texten "Följande BASIC-program gör detta:"

```
10 A$ = "SKRIV"
20 FOR I=1 TO 5: POKE 41219+I,ASC(MID$(A$,I,1)): NEXT
30 POKE 41224, PEEK(41224) OR 128
```

I artikeln med COMALprogrammet hittade vi också ett litet fel: Överst på sidan 14 skall rad 130 naturligtvis vara:

```
130 F:= N*FAKULTET (N-1)
```

RÄTTELSE till sidan 16, i VIC-rapport 3 årg 2, där tryckfelsnisse hade klippt bort några ord och gjort en mening hart när obegriplig.

I första spalten, alldeles ovanför

programlistningen på mitten, står "Det kan kanske tyckas felaktigt att krångla till en ut-" och sen kommer inte mer.

Fortsättningen skall vara: "räkning –

på det här" och hela meningen lyda: "Det kan kanske tyckas felaktigt att krångla till en uträkning på det här sättet bara för att spara en BASIC-rad." Vi beklagar oklarheten.

FEL i användarmanualen till VIC 64:

Programmet på sidan 146 som spelar piano är ett ovanligt tyst piano. Det beror på att man i programmet glömt ett POKE-kommando, som lämpligen placeras som radnummer 205 POKE 54296,15.

Detta ger full volym åt pianot, om man sätter ett annat tal i stället för "15" så får man motsvarande lägre volym.

Morgondagens konstruktörer & industriledare



Läser VIC-rapport varje månad.

OPUS 2 SOFTWARE'S program till VIC-datorer nu i Sverige

Programdistributören, Box 3009, 580 03 Linköping, Tel. 013/12 12 40 kommer att sälja OPUS 2 Software's program till VIC 20 och VIC 64. Programmen kommer att finnas hos återförsäljare av Commodore's VIC datorer.

Programdistributören har funnits sedan 1981 och säljer program till smådatorer.

Folktröjor för VIC-ägare



49:-

Finns i alla storlekar.

handic
electronic ab

Folktröjorna finns att köpa hos din VIC-handlare.

Skoldatorn

Sveriges mest köpta skoldator heter VIC-20. Det är ganska naturligt. För det finns inte lika mycket svenskt utbildningsmateriel till någon annan hemdator. Det gör det enkelt att lära sig.

Och när du har kunskapen behöver du inte bara använda färdiga program. Du kan skriva egna. Förändra och förbättra andra. Byta med kompisen. Och fortfarande hålla på att leka och spela.

Det finns inga gränser. Det är bara gränslöst roligt.

Nu skall vi vara lite förnuftiga mitt i allt det här roliga. Allt det utbildningsmateriel som finns till VIC-20 ger dig ett kunnande som gör att du kan styra datorn.

Det kommer du att få nytta av direkt. För datasamhället är faktiskt redan här.

Och att förena nytta med nöje är ju inte så dumt.

Köp en VIC-20 direkt. Så att du får tillgång till kunskapen som ger dig framgång.



VIC-20



Sveriges
mest köpta dator!

1.995:-
Ca pris inkl moms

handic
electronic ab