

ANVÄNDARHANDLEDNING

**ESCAPE-SEKVENSERNA hos
system ABC1600 och
ABC1600 FÖNSTERHANTERARE**

INNEHÅLL AVSNITT 3.1

3.1.1	Markör upp.....	14
3.1.2	Markör ned.....	15
3.1.3	Markör framåt	
3.1.4	Markör bakåt	
3.1.5	Positionering av markören.....	16
3.1.6	Övre och undre marginal	
3.1.7	Radering på skärmen	
3.1.8	Radera på en rad.....	17
3.1.9	Index	
3.1.10	Nästa rad	
3.1.11	Omvänt index.....	18
3.1.12	Spara markören	
3.1.13	Aterställ markören	
3.1.14	Aterställning till startläge - RESET.....	19
3.1.15	* Baklänges tabulering.....	20
3.1.16	Sätt tabulator-lägen	
3.1.17	Radering av tabulator-lägen	
3.1.18	Teckenattribut	
3.1.19	Rapportera privat status.....	21
3.1.20	Styrning av tangentbordets lysdioder.....	22
3.1.21	(*) Sätt mod	
3.1.22	(*) Aterställ mod.....	23
3.1.23	Välj teckenuppsättning.....	24

INNEHÅLL AVSNITT 3.2

3.2.1	Rita en linje.....	25
3.2.2	Rita en inverterad linje	
3.2.3	Flytta grafikmarkören	
3.2.4	(*) Rita en punkt.....	26
3.2.5	Rita en cirkelbåge.....	27
3.2.6	Rita en inverterad cirkelbåge	
3.2.7	Fyll en yta	
3.2.8	Rita en fylld cirkel.....	28
3.2.9	Måla en yta	
3.2.10	Flytta en yta.....	29
3.2.11	Ändra mönster	
3.2.12	Positionera teckenmarkören.....	30
3.2.13	* Rapportera musens läge.....	31
3.2.14	Rapportera privat status.....	32
3.2.15	* Sätt grafiskt origo.....	33
3.2.16	Aterställ generellt	
3.2.17	* Styrning av tangenternas lysdioder.....	34
3.2.18	(*) Sätt privat mod	
3.2.19	(*) Aterställ privat mod.....	35
3.2.20	* Sprej	
3.2.21	- Ändra släcktid.....	36

* -markerade ESC-sekvenser finns endast i ABC1600 FÖNSTERHANTERARE

(*) -markerade ESC-sekvenser innehåller en eller flera parametrar som endast finns i ABC1600 FÖNSTERHANTERARE

- -markerade ESC-sekvenser finns endast i grundsystemet ABC1600

INNEHÅLL AVSNITT 3.3

3.3.1	Starta och avsluta.....	37
3.3.2	GRAPH-moden.....	38
3.3.3	GIN-moden	
3.3.4	BYPASS-moden.....	39
3.3.5	Radera sida.....	40
3.3.6	Radera linje	

FÖRORD

System ABC1600 representerar en helt ny generation smådatorer. Det standardiserade operativsystemet UNIX vidgar möjligheterna att utnyttja väl beprövade programvaror och tidigare investeringar i maskin- och programvara. I system ABC1600 ingår även relationsdatabashanteraren MIMER.

ABC1600 FÖNSTERHANTERARE är ett hjälpmedel för presentation på bildskärmen. Fönsterhanteraren, som utnyttjar grafiken hos ABC1600, är speciellt användbar är då den tas till hjälp för att konstruera egna meny fönster. Fönsterhanteraren tillför dessutom ett flertal ESC-sekvenser utöver de som ingår i grundsystemet ABC1600.

Fönsterhanteraren kan användas tillsammans med valfri UNIX System III- eller UNIX System V-kompatibel programvara och klarar att arbeta med 16 olika fönster samtidigt.

Vi hoppas att denna användarhandledning skall hjälpa Dig att komma igång med att utnyttja grafiken och konstruera egna menyfönster så snabbt och smidigt som möjligt.

System ABC1600 och ABC1600 FÖNSTERHANTERARE är utvecklade av Luxor Datorer AB i Motala.

Motala i november 1985
LUXOR DATORER AB

1 INLEDNING

1.1 ATT ANVÄNDA HANDLEDNINGEN

Denna handledning beskriver funktionen hos *fönsterhanterarens escape-sekvenser* - de kommandon som styr grafiken. De flesta av dessa finns också i grundsystemet ABC1600.

Om man snabbt vill få en inblick i hur ESC-sekvenserna används i praktiken kan man, efter att ha läst avsnitt 1.3 PRESENTATION, börja med att studera exemplen som finns i avsnitt 3.4.

I kapitel 3 hänvisas till de större exemplen i avsnitt 3.4. Dessa har placerats sist i kapitlet eftersom de används för flera olika beskrivningar.

Handledningen kan också användas som uppslagsverk. Sakregistret innehåller de viktigaste uttrycken och även några fackuttryck - både på engelska och svenska. Några grundläggande begrepp finns förklarade i ett separat kapitel - 4 ORDFÖRKLARINGAR. De ord som finns upptagna i den listan är markerade med *kursiv fetstil* både i sakregistret och första gången de nämns i texten.

1.2 SYNTAX

Följande enkla regler har använts för framställningen i denna handledning.

- Ord och begrepp som skall anges i ett visst sammanhang symboliseras med klamrar runt det aktuella <ordet> som skall specificeras.

- För ett urval översatta engelska fackuttryck visas, första gången det förekommer i texten, även det engelska uttrycket genom (eng: 'ordet'). Ett exempel på detta är ordet skärm som i texten utmärks med: skärm (eng: 'screen').

- Citerade tecken och egennamn anges inom denna typ av 'citattecken'.

- Ett urval av de tecken, till exempel '&', ';' och ':', som citeras förtydligas inom parentes efter tecknet. Ett exempel: ':' (kolon).

- Indrag görs i beteckningar för att understryka nivåförhållanden (i strukturer - nyckelorden presenteras under strukturnamnet, i programexempel - ju lägre nivå på programrutinen desto större indrag för programkoden) och för ESC-sekvenser.

Exempel på detta är strukturen med dess nyckelord:

strukturen heter lås

 första nyckelordet

 andra nyckelordet

 sista nyckelordet

och

for i:=1 to 3 do

 begin strukturen lås

 while l>räknaren do

 begin...

 end

 end

next i

end

samt sekvensen som lyder:

ESC N y c CTRL-k ESC el

för en viss operation.

- Kontrolltecken enligt *ASCII* anges med den standardiserade förkortningen. Ett exempel är tecknet 'Carriage Return', med decimalt värde tretton, som anges med 'CR'. Vid indrag anges dock citerade tecken utan citatmarkering.

- Vissa ASCII-tecken förtydligas genom en decimal värdeangivelse inom parentes:

CR (decimalt 13)

- Tecken som förutsätter samtidig nedtryckning av flera tangenter anges med '-' (bindestreck) emellan:

CTRL-L

Både grundsystemet ABC1600 och ABC1600 FÖNSTERHANTERARE klarar, med undantag för de nedan specificerade, samtliga förekommande escape-sekvenser.

* -markerade ESC-sekvenser finns endast i ABC1600 FÖNSTERHANTERARE

- -markerade ESC-sekvenser finns endast i grundsystemet ABC1600

Då endast någon eller några av parametrarna inte finns i både fönsterhanteraren och grundsystemet markeras detta genom att * eller - placeras direkt efter parametern, till skillnad från de övriga fallen då markeringen gäller gemensamt för samtliga parametrar och placeras direkt före ESC-sekvensen:

PARAMETER - BESKRIVNING

<exempel>=0 ingen markering.

<exempel>=1 * finns endast i fönsterhanteraren.

respektive

* ESC <exempel>

1.3 PRESENTATION

Då resultat av beräkningar som gjorts med hjälp av dator skall presenteras är det ofta fördelaktigt att göra presentationen i grafisk form - dvs med hjälp av diagram, figurer och symboler. Det är då man har nytta av ABC1600 FÖNSTERHANTERARE. Med dess hjälp bygger man enkelt upp egna menyfönster, exempelvis för att sammankoppla olika program.

ESCAPESEKVENSERNA

Handledningen beskriver alla förekommande escapesekvenser. Dessa används för att styra grafiken. De flesta kan utnyttjas utan att fönsterhanteraren startats. Vanligast är att escapesekvenserna används som kommandon i program, men de kan också skrivas in direkt via tangentbordet.

2 PRESENTATION PÅ ABC1600's BILDSKÄRM

Beroende på vilken sorts information som skall ritas på bildskärmen används teckenmarkören eller grafikmarkören. Teckenmarkören används då hela tecken hanteras - bokstäver, siffror och övriga tecken. Då bilder - dvs diagram och figurer - skall byggas upp används grafikmarkören. Eftersom de båda markörerna alltid finns tillgängliga för användning kan både text och grafik blandas valfritt på en och samma bild. För att skärmen skall kunna utnyttjas till fullo för både breda och höga bilder kan bildskärmen vridas 90 grader. Benämningarna landskapmod och porträttmod (eng: 'landscape mode' och 'portrait mode') svarar mot de olika inställningarna.

2.1 TECKENMARKÖREN

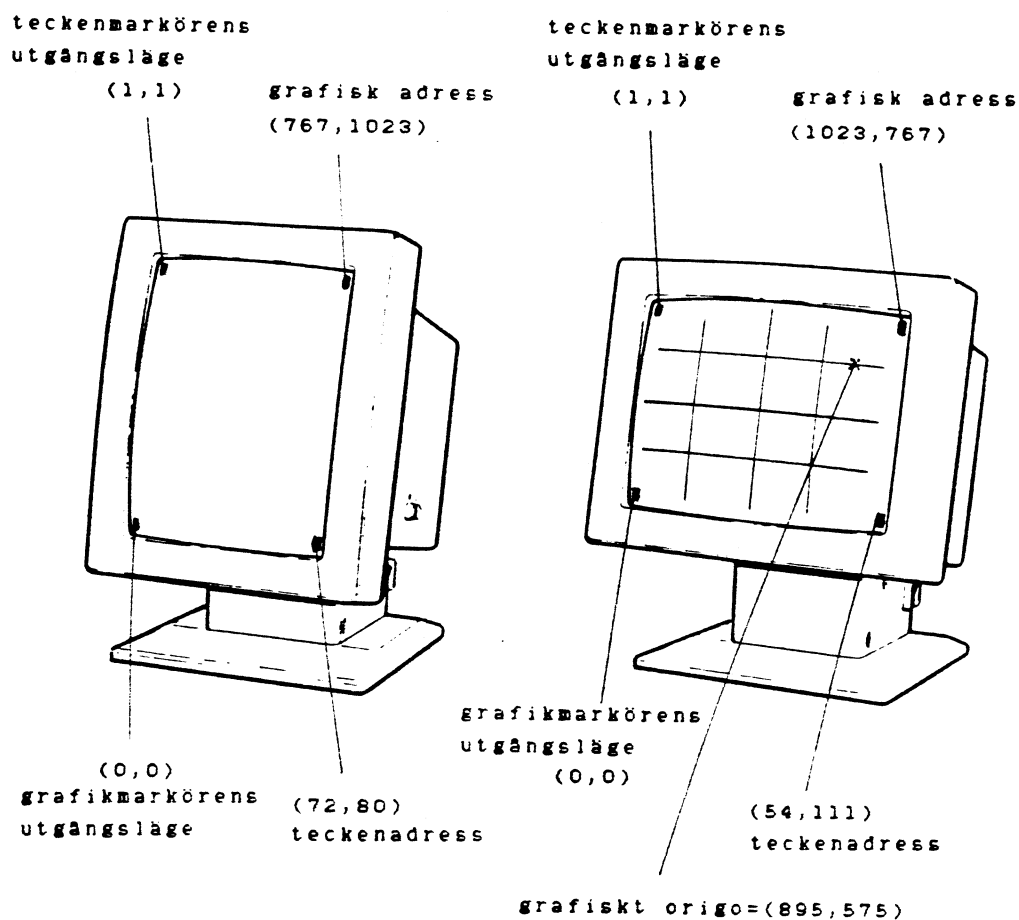
Varje tecken på bildskärmen nås med en angivelse av dess koordinatpar (rad,kolumn). Denna angivelse benämns *adressering*. Adresseringen medför att teckenmarkören flyttas till avsett läge - markören placeras på det aktuella tecknet (eller läget för ett tecken om där inte redan finns något).

Normalt är bildskärmens yta uppdelad i 54 rader * 111 kolumner i landskapmod och 72 rader * 80 kolumner i porträttmod. Läget (rad,kolumn)=(1,1) är det övre vänstra hörnet (eng: 'home'), medan (54,111) eller (72,80) är det nedre högra hörnet.

2.2 GRAFIKMARKÖREN

Bildskärmen består av en punktmatrix om 768 * 1024 punkter (eng: 'pixels'). Både hela tecken och grafik ritas på bildskärmen genom att punkterna tänds eller släcks. Exempelvis är tecken i typsnitt A uppbyggda av 14 * 9 punkter i både landskapmod och i porträttmod. Dock ändras det utrymme varje

tecken upptar på skärmen då ett *typsnitt* med annan tecken-
höjd eller -bredd väljs, vilket medför att även antalet
tecken som får plats på skärmen ökar eller minskar. (Skärmen
kan t ex rymma 48 rader * 55 kolumner av ett större typsnitt).



porträttmod

landskapmod

(standard-teckenstorlek)

Origo är normalt nollpunkten, dvs har koordinaterna $(x,y) = (0,0)$. Dock kan det grafiska origot definieras om till valfritt ställe på bildskärmen. I den högra figuren ovan har det grafiska origot placerats i punkten $(895,575)$. Observera att koordinaterna för ett origo alltid är $(0,0)$ varför punkten $(895,575)$ nås med $(0,0)$ så snart origots placering omddefinieras.

Nysättning av det grafiska origots läge görs för att förenkla adresseringar på bildskärmen. Det grafiska origot är alltid osynligt och måste ges positiva koordinater. Om en adressering begärs till en punkt utanför bildskärmen eller fönstret ignoreras denna.

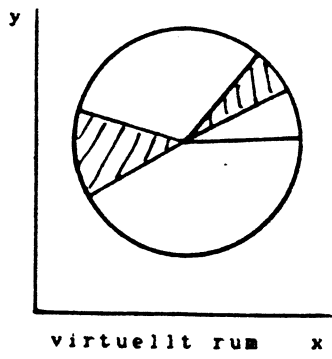
2.3 DET VIRTUELLA RUMMET

Då en stor eller komplex bild skall uppritas och bildskärmens $768 * 1024$ punkter inte räcker till används det *virtuella rummet*. Formatet på den bild som skall konstrueras bestämmer hur stort virtuellt rum som erfordras. Storleken är godtycklig och begränsas i praktiken av datorsystemets minneskapacitet. När det virtuella rummet definierats ritas bilden upp och man använder eller betraktar valfri del av bilden genom att flytta omkring en begränsad yta inom det virtuella rummets ram. Ytans storlek kan väljas valfritt från noll till att omfatta hela bildskärmen - dvs maximalt $768 * 1024$ punkter.

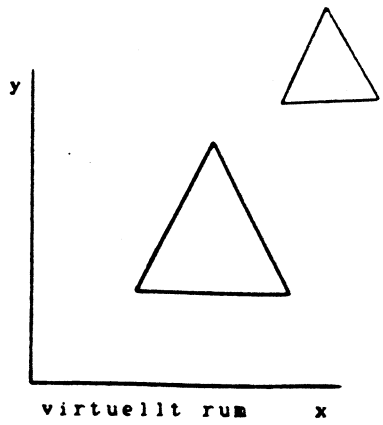
Då en del av det virtuella rummet skall visas på bildskärmen väljs det aktuella området ut genom att ett koordinatpar ges till datorn. Koordinatparet anger var i det virtuella rummet den begränsade ytans nedre vänstra hörn skall placeras.

Om en adressering begärs till en punkt utanför det virtuella rummet ignoreras denna.

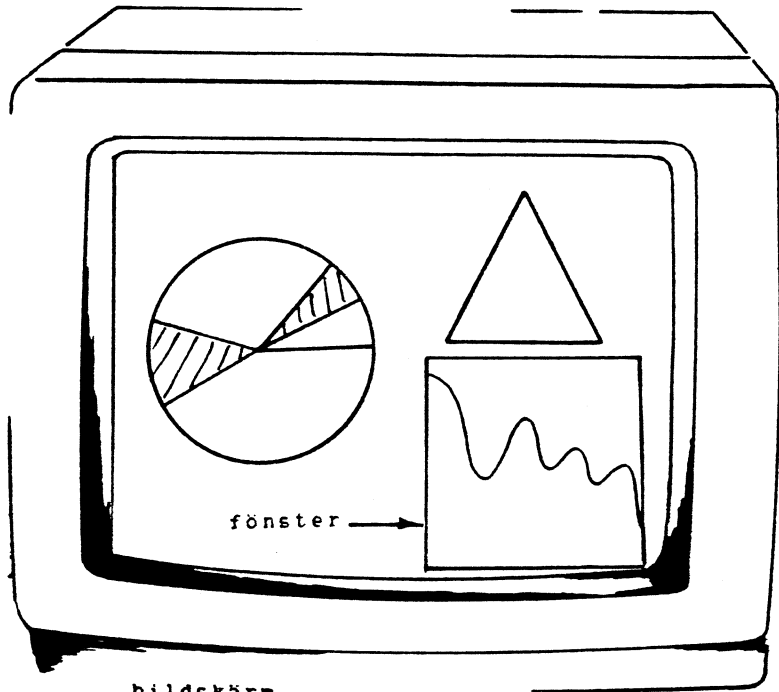
I figuren nedan används information från tre virtuella rum för att skapa den bild som skall visas på bildskärmen. I det nedre virtuella rummet har den begränsade ytan, som anger vilken del av det virtuella rummet som skall användas, markerats.



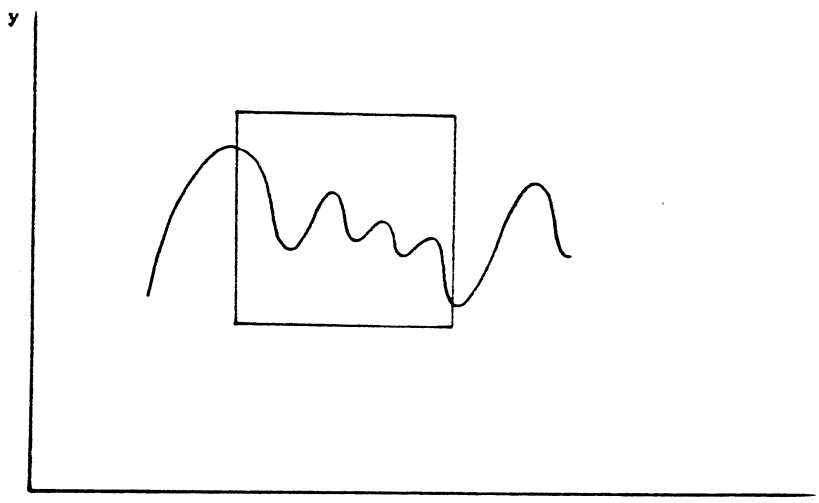
virtuellt rum x



virtuellt rum x



bildskärm



virtuellt rum

x

3 ESCAPESEKVENSERNA HOS ABC1600 FÖNSTERHANTERARE

Det finns tre uppsättningar av escapesekvenser för att styra grafiken. Den första motsvarar DEC VT100 och FACIT TWIST, vars kommandon inleds med 'ESC'. I den andra uppsättningen, som är ABC1600's egen, inleds varje kommando med 'ESC :'. Dessa båda kan blandas med varandra medan den tredje - Tektronix-uppsättningen - endast används då terminalen Tektronix 4014 emuleras.

Parametrarna i de båda första sekvenserna skiljs alltid åt med ett ';' (semikolon). Observera att inget semikolon förekommer mellan det inledande 'ESC' eller 'ESC :' och den första parametern, och inte heller mellan den sista parametern och den avslutande operatoren (A för 3.1.1, d för 3.2.1 osv).

De båda första uppsättningarna sekvenser motsvarar, i de fall de är jämförbara, de som används i ABC1600's terminalemulator.

I beskrivningen används olika termer för att förtydliga användandet av skilda typer av parametrar. <Pn> betecknar en parameter som skall anges i form av ett numeriskt värde, medan <Ps> betecknar en parameter som ges i form av godtyckliga tecken. <op> betecknar operation och <sel> betecknar val. På motsvarande sätt samtliga parametrar separerats, exempelvis <len>, <pno> och <cno>.

I syfte att förenkla läsningen har mellanslag placerats mellan tecknen i escapesekvenserna. Beskrivningens 'ESC Å ? Pn' skall alltså ges till ABC1600 som 'ESCÅ?Pn'.

3.1 ESCAPESEKVENSER MOTSVARANDE DEC VT100 OCH FACIT TWIST

3.1.1 MARKÖR UP

ESC Å <Pn> A

flyttar teckenmarkören <Pn> rader uppåt. Markören stannar vid övre marginalen (eng: 'top of margin'). Om <Pn> är '0' eller icke angivet kommer markören att flyttas en rad uppåt.

3.1.2 MARKÖR NED

ESC Å <Pn> B

flyttar teckenmarkören <Pn> rader nedåt. Markören stannar vid undre marginalen (eng: 'bottom margin'). Om <Pn> är '0' eller icke angivet kommer markören att flyttas en rad nedåt.

3.1.3 MARKÖR FRAMÅT

ESC Å <Pn> C

flyttar teckenmarkören <Pn> steg åt höger. Markören stannar vid högra marginalen (eng: 'right margin'). Om <Pn> är '0' eller icke angivet kommer markören att flyttas ett steg åt höger.

3.1.4 MARKÖR BAKÅT

ESC Å <Pn> D

flyttar teckenmarkören <Pn> steg åt vänster. Markören stannar vid vänstra marginalen (eng: 'left margin'). Om <Pn> är '0' eller icke angivet kommer markören att flyttas ett steg åt vänster.

3.1.5 POSITIONERING AV MARKÖREN

ESC Å <Pn> ; <Pn> H eller ESC Å <Pn> ; <Pn> f

flyttar teckenmarkören till det av parametrarna angivna läget. Den första parametern anger raden och den andra anger kolumnen. Om en parameter är '0' eller inte angiven kommer markören att flyttas till första raden eller kolumnen.

3.1.6 ÖVRE OCH UNDER MARGINAL

ESC Å <Pn> ; <Pn> r

sätter övre och undre marginalerna för den *rullande* delen av skärmen. Den första parametern är radnumret för den första rullande raden och den andra är radnumret för den sista raden. Om inga parametrar anges kommer hela det virtuella rummet att definieras som rullande. Som minst kan den rullande delen bestämmas till två rader. Markören placeras i normalläge.

3.1.7 RADERING PÅ SKÄRMEN

ESC Å <Ps> J

raderar det virtuella rummet enligt parametern - helt eller delvis. Teckenmarkörens aktuella läge ändras ej. Om ingen parameter anges tolkas detta som om '0' angetts.

PARAMETER - BESKRIVNING

- 0 Raderar från och med teckenmarkörens läge till slutet av den rullande delen.
- 1 Raderar från den rullande delens början till och med teckenmarkörens läge.

2 Raderar hela den rullande delen.

3.1.8 RADERA PÅ EN RAD

ESC Å <Ps> K

raderar raden enligt parametern - helt eller delvis. Teckenmarkörens aktuella läge ändras ej. Om ingen parameter anges tolkas detta som om '0' angetts.

PARAMETER - BESKRIVNING

- 0 Raderar från och med teckenmarkörens läge till radens slut.
- 1 Raderar från radens början till och med teckenmarkörens läge.
- 2 Raderar hela raden.

3.1.9 INDEX

ESC D

flyttar teckenmarkören en rad nedåt inom samma kolumn. Om markören befinner sig på sista raden av den rullande delen kommer en rullning uppåt att ske.

3.1.10 NÄSTA RAD

ESC E

flyttar teckenmarkören till början på nästa rad. Om markören

befinner sig på sista raden av den rullande delen kommer en rullning uppåt att ske.

3.1.11 OMVÄNT INDEX

ESC M

flyttar teckenmarkören en rad uppåt i samma kolumn. Om markören befinner sig på översta raden av den rullande delen kommer en rullning nedåt att ske.

3.1.12 SPARA MARKÖREN

ESC 7

sparar den aktuella teckenmarkörens läge. Även det grafiska origot, grafikmarkörens läge, teckenattributen och typsnitt sparas.

3.1.13 ÅTERSTÄLL MARKÖREN

ESC 8

återställer markören enligt de data som sparades av 'SPARA MARKÖREN' senaste gången. Om ingen 'SPARA MARKÖREN'-sekvens har skickats till fönstret kommer teckenmarkören, den grafiska markören och det grafiska origot att inställas i normalläge.

3.1.14 ÅTERSTÄLLNING TILL STARTLÄGE - RESET

ESC c

På en DEC VT100-terminal återställer denna sekvens till terminalens startläge. För att simulera detta i ett fönster kommer följande att utföras då sekvensen exekveras:

- Teckenmarkören ställs i normalläget.
- Den grafiska markören ställs i normalläget.
- Teckenmarkören sätts till normalutseende.
- Flaggorna som ändras med escapesekvenserna för att sätta och återställa moder sätts till sina utgångslägen.
- Teckenattributen ställs i sina utgångslägen.
- Hela det virtuella rummet definieras som rullande.
- Det grafiska origot ställs i det virtuella rummets nedre vänstra hörn.
- Tab-stoppar sätts i normallägena.
- Tabellen med grafiska mönster normalinställs.
- Det aktuella typsnittet ställs in som standardtypsnitt.
- Hela det virtuella rummet rensas.
- Alla lysdioder på tangentbordet släcks.
- Avslutar musrapportering som begärts med 'ESC 7 M' eller 'ESC 8 M' (se 3.2.13 sid 31-32).

3.1.15 BAKLÄNGES TABULERING

* ESC Å Z

flyttar teckenmarkören bakåt till nästa tabulatormarkering.
Markören stannar vid vänster marginal.

3.1.16 SÄTT TABULATOR-LÄGEN

ESC H

sätter en tabulatormarkering på den plats där markören
befinner sig.

3.1.17 RADERING AV TABULATOR-LÄGEN

ESC Å <Ps> g

om <Ps> är '0' kommer tabulatorläget på den plats där markö-
ren befinner sig att raderas. Om ingen parameter anges tolkas
detta som om '0' angetts. Om <Ps> är tre kommer alla tabula-
torlägen att raderas.

3.1.18 TECKENATTRIBUT

ESC Å <Ps> ; <Ps> ;... ...; <Ps> m

ställer eller återställer attributen i enlighet med parametrarna.

PARAMETER - BESKRIVNING

0 Samtliga attributen återställs.

- 1 Större intensitet. På ABC1600 har detta ~~samma~~ effekt som attributet 'reverse character', dvs inverterar tecknen.
- 4 Understrykning.
- 5 Blinkande tecken. På ABC1600 har detta ~~samma~~ effekt som attributet 'reverse character'.
- 7 Inverterar tecknen (eng: 'reverse character').

3.1.19 RAPPORTERA PRIVAT STATUS

ESC Å <Ps> ; <Ps> ;... ..; <Ps> n

Förfrågan för att erhålla statusrapport. Följande rapporter kan efterfrågas

PARAMETER - BESKRIVNING

- 6 Rapporterar teckenmarkörens läge. Rapportsekvensen lyder

ESC Å <Pn> ; <Pn> R

där den första parametern anger raden och den andra anger kolumnen.

- ?1 Rapportera porträtt-/landskap-mod. Detta motsvarar FACIT TWIST-terminalens kommando. Rapportsekvensen lyder

ESC Å ? Pn resp ESC Å ? Ln

för porträtt- resp landskap-mod.

3.1.20 STYRNING AV TANGENTBORDETS LYSDIODER

ESC Å <Ps> ; <Ps> ;... ...; <Ps> q

tänder och släcker de åtta programmerbara lysdioderna på tangentbordet enligt parametrarna. Om ingen parameter anges tolkas detta som om '0' angetts. Observera att tangentbordets lysdioder alltid visar lysdiodinställningen hos det översta fönstret.

PARAMETER	BESKRIVNING
0	Släcker lysdioderna ett till åtta.
1	Tänder lysdiod ett.
2	två.
3	tre.
4	fyra.
5	fem.
6	sex.
7	sju.
8	åtta.

3.1.21 SÄTT MOD

ESC Å <Ps> ; <Ps> ;... ...; <Ps> h

sätter de moder som angetts enligt parametrarna. Moderna ?32, ?33, ?34 och ?35 motsvarar moder hos terminalen FACIT TWIST.

PARAMETER - BESKRIVNING

20 * Manuell mod: Radframmatningtangenter (eng: LF) flyttar markören till början på nästa rad. RETURN-tangenten innebär både vagnretur och radframmatning (CR & LF).

- ?5 * Skärmmod: Fönstret inverteras.
- ?6 Normalmod: Teckenmarkörens normalläge sätts till den rullande delens övre vänstra hörn.
- ?7 Automatisk mod: Teckenmarkören kommer automatiskt att flyttas till nästa rad då den når högermarginalen.
- ?32 * Sidmod: Fönstret rullar ej alls.
- ?33 * Understruken teckenmarkör.
- ?34 Blinkande teckenmarkör.
- ?35 Osynlig teckenmarkör.
- ?36 - Släck skärmen omgående.

3.1.22 ÅTERSTÄLL MOD

ESC Å <Ps> ; <Ps> ;... ...; <Ps> l

återställer moderna som angetts av parametrarna. Moderna ?32, ?33, ?34 och ?35 motsvarar moder hos terminalen FACIT TWIST.

PARAMETER - BESKRIVNING

- 20 * Manuell mod: Radframmatningtangenter ger endast förflyttning av markören i y-led och RETURN-tangenten ger endast vagnretur (eng: CR).
- ?5 * Skärmmod: Det inverterade fönstret återgår till normalläge.
- ?6 Normalmod: Teckenmarkörens normalläge sätts till det virtuella rummets övre vänstra hörn.
- ?7 Automatisk mod: Teckenmarkören kommer inte längre automatiskt att flyttas till nästa rad då den når högermarginalen.

?32 * Rullande mod: Fönstret rullar kontinuerligt.

?33 * Inverterad teckenmarkör.

?34 Icke blinkande teckenmarkör.

?35 Synlig teckenmarkör.

?36 - Tänd skärmen omgående.

3.1.23 VÄLJ TECKENUPPSÄTTNING

ESC (A	eller	ESC) A
ESC (B	eller	ESC) B
.	-"-	.
.	-"-	.
.	-"-	.
ESC (Z	-"-	ESC) Z

väljer önskat typsnitt.

Då olika typsnitt blandas kommer tecknen att placeras så att *baslinjerna* för de olika typsnitten sammanfaller. Med baslinje för ett typsnitt menas den linje som kan dras direkt nedanför versalerna i texten. Baslinjen skärs endast av de gemenas underslängar.

Observera att då ett fönsters typsnitt ändras kommer den eventuella lagringen av fönstrets innehåll att raderas.
(se EXEMPEL 4, sid 45).

3.2 ABC1600's PRIVATA ESCAPESEKVENSER

I fem av de privata escapesekvenserna förekommer den färgbestämmande parametern <cn> utan närmare beskrivning. Normalt ges <cn> värdet '1', vilket medför att en linje ritas med mönstret <pn>. Om <cn> istället ges värdet '0' ritas en linje med inversen till mönstret <pn>.

Då det är fråga om fyllnad eller målning av en yta görs denna med av <pn> angivet mönster om <cn> ges värdet '1' och med det motsatta mönstret om <cn> ges värdet '0'.

3.2.1 RITA EN LINJE

ESC : <x> ; <y> ; <pn> ; <cn> d

ritar en linje från grafikmarkörens aktuella läge till <x>,<y>, med mönstret som angetts av <pn>. Om <pn> ej anges kommer ett heldraget streck att ritas. Grafikmarkörens läge uppdateras till <x>,<y>. (se EXEMPEL 1, 3 och 4, sid 40, 43 och 45).

Beträffande <cn> - se inledningen till avsnitt 3.2 på sid 25.

3.2.2 RITA EN INVERTERAD LINJE

ESC : <x> ; <y> i

ritar en linje från grafikmarkörens aktuella läge till <x>,<y> genom att invertera samtliga i linjen ingående skärmpunkter. Detta kommando är även användbart då ingen information ritats på skärmen. För t ex en helt mörk bakgrund ritas ett helt vitt streck. Linjen kan avlägsnas genom att den inverterade linjen ritas en andra gång. Grafikmarkörens läge uppdateras till <x>,<y>.

3.2.3 FLYTTA GRAFIKMARKÖREN

ESC : <x> ; <y> ■

flyttar grafikmarkören till <x>, <y>. (se EXEMPEL 1, 2, 3 och 4, sid 40-46).

3.2.4 RITA EN PUNKT

ESC : <x> ; <y> ; <op> ; <cno> p

ändrar eller rapporterar färgen på punkten med läge <x>, <y>. (<cno> är reserverad för framtida bruk och kan utelämnas).

PARAMETER - BETYDELSE

<op>=0 punkten tänds.
 <op>=1 punkten släcks.
 <op>=2 punkten inverteras.
 <op>=10 * den angivna punktens färg rapporteras:

Om <op> inte är angiven tolkas detta som om '0' angetts.

Grafikmarkörens läge uppdateras till <x>, <y> om <op> är '0', '1' eller '2'.

Rapportsekvensen lyder

ESC : <x> ; <y> ; ll ; <cno> p

där <cno> är '1' om punkten är tänd, '0' i annat fall.

Om <cno> utelämnas i rapportsekvensen lyder denna

ESC : <x> ; <y> ; ll p

och betyder att den angivna punkten befinner sig utanför det virtuella rummet.

3.2.5 RITA EN CIRKELBÅGE

ESC : <x> ; <y> ; <len> ; <pno> ; <cno> a

ritar en cirkelbåge moturs med längden <len>, och med origo i <x>, <y> från grafikmarkörens aktuella läge. Bågen ritas med ett streck av mönstertypen <pno>, och är <len> lång. Om <pno> inte anges ritas cirkelbågen med ett heldraget streck. Längden <len> är antalet punktvisa steg i x- och y-led. Detta innebär att en hel cirkel drages då <len> anges till åtta gånger cirkelns radie. Grafikmarkörens läge uppdateras till cirkelbågens senast ritade punkt.

Beträffande <cno> - se inledningen till avsnitt 3.2 på sid 25.

3.2.6 RITA EN INVERTERAD CIRKELBÅGE

ESC : <x> ; <y> ; <len> I

ritar en cirkelbåge som kommandot 'RITA EN CIRKELBÅGE' men genom att invertera samtliga i cirkelbågen ingående punkter. Detta kommando är även användbart då ingen information ritats på skärmen. För t ex en helt mörk bakgrund ritas en helt vit cirkelbåge. Längden <len> är antalet punktvisa steg i x- och y-led. Detta innebär att en hel cirkel drages då <len> anges till åtta gånger cirkelns radie. Grafikmarkörens läge uppdateras till cirkelbågens senast ritade punkt.

3.2.7 FYLL EN YTA

ESC : <x> ; <y> ; <pno> ; <cno> f

yller en rektangel med mönstret <pno>. Om <pno> är noll eller icke angivet tänds alla punkter i rektangeln. Rektangelns ena hörn sätts till <x>, <y> och det motsatta hörnet till

grafikmarkörens aktuella läge. Grafikmarkörens läge uppdateras till $\langle x \rangle, \langle y \rangle$ (se EXEMPEL 2 och 3 sid 42 och 43).

Beträffande $\langle cno \rangle$ - se inledningen till avsnitt 3.2 på sid 25.

3.2.8 RITA EN FYLLD CIRKEL

ESC : $\langle x \rangle$; $\langle y \rangle$: $\langle rad \rangle$; $\langle pno \rangle$; $\langle cno \rangle$ c

ritar en fylld cirkel med origo i $\langle x \rangle, \langle y \rangle$, med radien $\langle rad \rangle$ och med mönstret $\langle pno \rangle$. Om $\langle pno \rangle$ ej anges kommer alla punkter i cirkeln att tändas. Grafikmarkörens läge uppdateras till $\langle x \rangle, \langle y \rangle$.

Beträffande $\langle cno \rangle$ - se inledningen till avsnitt 3.2 på sid 25.

3.2.9 MÅLA EN YTA

ESC : $\langle x \rangle$; $\langle y \rangle$; $\langle pno \rangle$; $\langle cno \rangle$ F

målar en yta med mönstret $\langle pno \rangle$. Den målade ytan begränsas av heldragna linjer (eller kurvor) som ritats med någon av funktionerna 'RITA EN LINJE', 'FYLL EN YTA' eller liknande.

$\langle x \rangle, \langle y \rangle$ anger den punkt målandet utgår ifrån. Punkten måste befinna sig inom de heldragna linjerna. Om punkten $\langle x \rangle, \langle y \rangle$ har värdet '0' antages ytans begränsande linjer ha värdet '1' och vice versa. Om $\langle pno \rangle$ är '0' eller icke angivet täcks ytan helt. Om $\langle pno \rangle$ inte är '0' kommer ytan att täckas med mönstret $\langle pno \rangle$. Grafikmarkörens läge uppdateras till $\langle x \rangle, \langle y \rangle$.

Om en komplicerad yta skall målas och $\langle pno \rangle$ inte är '0' kan en eller flera av escapesekvenserna behöva upprepas för att ytan skall bli korrekt målade. Innan en escapesekvens upprepas skall sekvensens koordinatangivelse justeras så att en position där ytan behöver målas ytterligare anges.

Observera att 'MÅLA EN YTA' arbetar direkt med hjälp av grafikminnet varför resultatet av en målning med fönstret delvis täckt av ett annat fönster kan skilja sig från en målning där fönstret ej är täckt.

Beträffande <no> - se inledningen till avsnitt 3.2 på sid 25.

3.2.10 FLYTTA EN YTA

```
ESC : <xsrc> ; <ysrc> ; <xdest> ; <ydest> ; <width> ;  
      <height> ; <op> r
```

flyttar (kopierar egentligen) den rektangulära ytan med nedre vänstra hörnet i <xsrc>, <ysrc> till <xdest>, <ydest>. Ytan har bredden <width> och höjden <height>. Om operatorn <op> är '0' eller inte angiven kommer ytan att flyttas (kopieras) som den är och om den är '1' kommer ytan att inverteras medan den flyttas. Observera att grafikmarkörens läge inte uppdateras och att endast ytor där både ursprunglig placering (<xsrc>; <ysrc>) och slutlig placering är synliga kommer att flyttas.

3.2.11 ÄNDRA MÖNSTER

```
ESC : <pno> ; <hmask> ; <vmask> ; <shift> ; <op> R
```

bestämmer utseendet hos mönstret <pno> enligt angivna parametrar. Mönstret definieras för porträtt-moden och kommer att vridas 90 grader då det används i landskap-moden.

<hmask> definierar en 16-bitars horisontalmask för mönsteruppritning på skärmen. Om mönstret består av flera rader används masken upprepat rad för rad så att mönstret bildas.

<vmask> definierar en 16-bitars vertikalmask. Om en bit är

satt används <hmask> för att rita raden. Om den ej är satt bestämmer <op> vad som skall göras. (se EXEMPEL 3, sid 43).

PARAMETER - BESKRIVNING

<op>=0 Återställer raden och roterar <hmask> det antal bitar som angetts med <shift>.

<op>=1 Ettställer raden och roterar <hmask> det antal bitar som angetts med <shift>.

<op>=2 Använder <hmask> inverterad och roterar <hmask> det antal bitar som angetts med <shift>.

<op>=3 Lämnar raden i befintligt skick och roterar <hmask> det antal bitar som angetts med <shift>.

<op>=4 Återställer raden utan att rotera.

<op>=5 Ettställer raden utan att rotera.

<op>=6 Använder <hmask> inverterad utan att rotera.

<op>=7 Lämnar raden i befintligt skick utan att rotera.

<pno> kan anges inom området 1 - 15. Mönsternummer noll kan ej ändras. <shift> kan anges inom området 0 - 15. Endast <hmask> används av escapesevenserna 'RITA EN LINJE' och 'RITA EN CIRKELBÅGE'.

3.2.12 POSITIONERA TECKENMARKÖREN

ESC : <sel> H

flyttar teckenmarkören till samma läge som grafikmarkören, enligt den angivna parametern. Observera att då teckenmarkören

positioneras kommer en eventuell lagring av fönstrets tecken-
innehåll att raderas (se EXEMPEL 4, sid 45).

PARAMETER - BESKRIVNING

- <sel>=0 Typsnittrutans övre vänstra hörn placeras på grafikmarkören.
- <sel>=1 Typsnittrutans nedre vänstra hörn placeras på grafikmarkören.
- <sel>=2 Vänstra sidan av typsnittets baslinje placeras vid grafikmarkören.

3.2.13 RAPPORTERA MUSENS LÄGE

* ESC : <sel> M

rapporterar musmarkörens aktuella läge. Beroende på parametern <sel> kommer rapportering att ske endast då musens läge och knappar eller enbart knappar har ändrats. Observera att rapportering endast sker till det översta fönstret.

PARAMETER - BESKRIVNING

- <sel>=0 Avslutar rapportering som begärts med nedanstående två parametrar. 'ESC 7 M' eller 'ESC 8 M' fortsätter annars sedan man avslutat en programkörning om ingen tangent nedtryckts eller musen ej förflyttats dessförinnan.
- <sel>=7 Om förändring har skett sedan den senaste rapporteringen sänds en ny rapport omedelbart. I annat fall kommer rapporten att sändas så snart förändring sker. Som förändring räknas här en musförflyttning eller en lägesändring av någon musknapp. Rapport-

sekvensen lyder

ESC : <x> ; <y> ; <buttons> P

där <x> och <y> är musmarkörens läge. Om musmarkören befinner sig utanför det virtuella rummet kommer det rapporterade läget att vara det virtuella rummets gräns. <buttons> är '1' om den vänstra knappen trycks ned, '2' om den mellersta knappen trycks ned, '3' om både den vänstra och högra trycks ned samt '0' om ingen knapp trycks ned.

<sel>=8 är identisk med föregående med undantag för att rapport endast sänds då den vänstra eller mellersta knappens läge ändras.

3.2.14 RAPPORTERA PRIVAT STATUS

ESC : <sel> n

Följande rapporter kan efterfrågas

PARAMETER - BESKRIVNING

<sel>=1 Rapporterar grafikmarkörens läge. Rapportsekvensen lyder

ESC : <x> ; <y> R

där <x>, <y> anger grafikmarkörens aktuella läge.

<sel>=2 Rapporterar musmarkörens läge och knappstatus. Detta är identiskt med 'RAPPORTERA MUSENS LÄGE', ESC : 7 M med undantag för att rapportering sker omedelbart. Observera att rapportering endast görs till toppnivåfönstret.

<sel>=3 Rapporterar det virtuella rummets storlek och dess aktuella typsnitt. Rapportsekvensen lyder

ESC : <vsx> ; <vsy> ; <fsx> ; <fsy> ; <bl> ; <fno> W

där <vsx> och <vsy> är det virtuella rummets storlek (antal punkter), <fsx> och <fsy> är typsnitt-rutans storlek, <bl> är typsnittrutans baslinje, och <fno> är ASCII för namnet på det aktuella typsnittet.

3.2.15 SÄTT GRAFISKT ORIGO

* ESC : <x> ; <y> 0

sätter det grafiska origot till <x>,<y>. Grafikmarkörens läge sätts till 0,0. Alla koordinater som ges av de grafiska escape-sekvenserna gäller relativt det grafiska origot. Observera att musmarkörens läge alltid rapporteras relativt det virtuella rummets nedre vänstra hörn.

3.2.16 ÅTERSTÄLL GENERELLT

ESC : J

återställer fönstret och dess markörer enligt

- Teckenmarkören ställs i läge 1,1.
- Grafikmarkören ställs i läge 0,0.
- Det grafiska origot sätts till läge 0,0.
- Den rullande delen av skärmen sätts lika med hela det virtuella rummet.

- Om det aktuella typsnittet är samma som fönstrets standard-typsnitt börjar fönstrets teckeninnehåll att sparas kontinuerligt igen.
- Hela det virtuella rummet raderas.

3.2.17 STYRNING AV TANGENTERNAS LYSDIODER

* ESC : <sel> ; <sel> ;... ...; <sel> q

tänder och släcker INS- och ALT-tangenternas lysdioder enligt parametrarna. Om ingen parameter anges kommer lysdioderna att släckas. Observera att tangenternas lysdioder alltid visar toppnivå-fönstrets status.

PARAMETER	BESKRIVNING
<sel>=0	Släcker båda lysdioderna.
<sel>=1	Tänder INS-knappens lysdiod.
<sel>=2	Tänder ALT-knappens lysdiod.

3.2.18 SÄTT PRIVAT MOD

ESC : <sel> ; <sel> ;... ...; <sel> h

sätter ABC1600's privata mod enligt parametern

<sel>=1 - Välj landskap-mod.

<sel>=2 * Mod för fasat mönster. Funktionen kan användas av escapesequenser som 'FYLL EN YTA', 'RITA EN FYLLD CIRKEL' och liknande för att erhålla fasade mönster.

3.2.19 ÅTERSTÄLL PRIVAT MOD

ESC : <sel> ; <sel> ;... ...; <sel> 1

återställer ABC1600's privata mod enligt parametern

<sel>=1 - Välj porträtt-mod.

<sel>=2 * Mod för icke-fasat mönster.(se 3.2.18).

3.2.20 SPREJ

* ESC : <x> ; <y> ; <pno> ; <op> s

är en escapesekvens som beroende på operatören <op> bestämmer huruvida de punkter som är tända både i sprejmasken och i det <pno>-specificerade mönstret skall tändas eller släckas.

<x>,<y> är det nedre vänstra hörnet där den 32 gånger 32-punkters mönstret skall placeras. Både om <pno> eller <op> inte angetts kommer detta att tolkas som om '0' angetts. Observera att sekvensen endast gäller de punkter som är tända både i sprejmasken och i det <pno>-specificerade mönstret.

Följande operationer kan utföras:

OPERATOR - BESKRIVNING

<op>=0 Alla berörda punkter tänds, de återstående släcks.

<op>=1 Alla berörda punkter tänds, de återstående påverkas ej.

<op>=2 Alla berörda punkter släcks, de återstående påverkas ej.

<op>=3 Alla berörda punkter inverteras, de återstående påverkas ej.

För varje fönster som skapas finns redan en slumpmässigt genererad sprejmask definierad. Denna kan man själv ändra genom att definiera om den.

Sprejmasken för ett fönster byts ut genom att en begäran sänds till fönsterhanteraren. För att denna begäran skall ge ett meningsfullt resultat måste fönstret vara inställt i ett läge med fasat mönster. Grafikmarkörens läge uppdateras till <x>,<y> (se EXEMPEL 5, sid 47).

3.2.21 ÄNDRA SLÄCKTID

Den tid som terminalen skall vänta innan den automatiska nedsläckningsfunktionen aktiveras bestäms med sekvensen

- ESC : <time> S

Om <time>=0 släcks skärmen inte alls. <time> har ursprungligen värdet 3000, vilket medför en tid på ca 3.5 minuter.

3.3 TEKTRONIX-MODEN

ABC1600 FÖNSTERHANTERARE kan utföra alla de operationer som finns hos den grafiska bildskärmsterminalen Tektronix 4014. För de tillfällen man behöver använda exakt samma koder och sekvenser som Tektronix-terminalen finns det i system ABC1600 en inbyggd Tektronix 4014 -emulator. Tektronix-moden kan ej användas från fönster.

I Tektronix-moden används hela skärmen för att visa både punktvis grafik samt bokstäver, siffror och skiljetecken. Automatisk rullning/*panorer*ing används ej. Då markören flyttas nedanför skärmens understa rad fortsätter den från det övre vänstra hörnet.

Vid Tektronix-emulering skiljer man på uppritning på skärmen - GRAPH-moden - och rapportering från skärmen - GIN-moden.

Detta avsnitt ger en viss överblick över Tektronix-emulators arbetsområden. Här beskrivs endast de fundamentala operationerna - t ex starta och avsluta Tektronix-emuleringen. För en

komplett beskrivning av Tektronix 4014 -terminalens funktioner hänvisas till Tektronix' originaldokumentation.

3.3.1 STARTA OCH AVSLUTA

Tektronix-emuleringen startas med sekvensen

ESC : 4014 ; <strap> T

Tecken som sänds efter en lägesrapport i GIN-moden bestäms av parameterns <strap> värde.

Funktionen hos <strap> motsvaras på en Tektronix-terminal av omkopplare som ställs i önskat läge. I ABC1600's Tektronix-emulator ersätts omkopplarna helt av escape-sekvensen med efterföljande parameter. Då emulatorens har sändt en lägesrapport befinner den sig i en speciell mod - BYPASS-moden. Detta innebär att alla tecken som sänds från värddatorn ignoreras till dess att BYPASS-moden har avslutats.

PARAMETER - BETYDELSE

<strap>=0	Lägesrapporten består av fyra ord. BYPASS-moden måste ha avslutats först.
<strap>=1	Lägesrapporten består av fyra ord plus CR. Då CR <i>ekas</i> tillbaka avslutas BYPASS-moden.
<strap>=2	Lägesrapporten består av fyra ord plus CR och ROT

Tektronix-emuleringen avslutas med

ESC Å

varefter övergång automatiskt sker till emulering av DEC VT100.

3.3.2 GRAPH-MODEN

GRAPH-moden startas med

GS (decimalt 29)

I GRAPH-moden sänds koordinaterna i form av tecken större än eller lika med 'mellanslag' (eng: 'space'), vars decimala värde är 32. Tio bitar krävs för adressering av koordinater från noll till 1023.

För att avkoda x- och y-koordinaterna används fyra stycken fembitars ord (eng: 'bytes') med koordinatinformation. Orden sänds i ordningen

high y	med offset	' '	(decimalt 32)
low y		'é'	96
high x		' '	32
low x		'g'	64

Den första vektorn som begärs utritad då man startat GRAPH-moden ritas ej ut (den ritas egentligen, men har samma färg som bakgrunden).

3.3.3 GIN-MODEN

I GIN-moden aktiveras hårkors-markören, vars läge på skärmen ändras med hjälp av musförflyttningar eller nedtryckning av tangentbordets markörplacerare.

ESC SUB (decimalt 27,26)

startar GIN-moden.

GIN-moden avslutas så fort någon tangent trycks ned. Den

sekvens som sänds vid en tangentnedtryckning består av tangentens tecken följt av fyra ord med koordinaterna för hårkorsmarkörens läge. Beroende på vilket värde parametern <strap> getts då Tektronix-emulatorn startades följer sedan CR eller CR EOT.

Den fyra ord långa koordinatangivelsen består i tur och ordning av:

de fem mest signifikanta x-bitarna,
de fem minst signifikanta x-bitarna,
de fem mest signifikanta y-bitarna,
de fem minst signifikanta y-bitarna.

I samtliga fall föregås koordinatinformationen i angivelsen av '01' som bitarna sju och sex, (med offset ' ' - decimalt 32).

Då GIN-moden avslutas ger de tre mus-tangenterna samma resultat som tangenterna '1', '2' och '3'.

GIN-moden kan även avslutas genom att

ESC ENQ

sänds från värddatorn.

3.3.4 BYPASS-moden

Då GIN-moden avslutas sker automatiskt en övergång till BYPASS-moden. Ingen information kommer att sändas till dess att 'BYPASS' har avslutats genom att värddatorn sänder ett CR (decimalt 13) till emulatorn.

Om emulatorn startats med <strap>=1 eller <strap>=2 kommer 'BYPASS' att avslutas genom att värddatorn skickar det sända CR tillbaka till bildskärmen.

3.3.5 RADERA SIDA

Bilskärmen raderas helt med

ESC CTRL-L

(decimalt 27,12)

3.3.6 RADERA LINJE

Eftersom terminalen Tektronix 4014 är försedd med minnesbildskärm måste hela skärmsidan raderas varefter den aktuella ändringen görs och resten av bilden ritas upp på nytt.

I ABC1600's Tektronix-emulator har därför de ursprungliga kommando-sekvenserna utökats med

ESC x

(decimalt 27, 120)

Kommandot ställer in vektorgeneratorns linjefärg till samma färg som bildskärmens bakgrund. I själva verket ritas alltså en linje med bakgrundsfärg över den linje som skall raderas. (se även TILLÄGG: VALJ LINJETYP, sid 59).

3.4 EXEMPEL

Nedan följer fem exempel som ger prov på användningen av escapesekvenserna och olika kommandon. Alla exempel är skrivna i programspråken C, FORTRAN och BASIC.

EXEMPEL 1

Följande exempel är skrivet i C, FORTRAN samt BASIC och visar hur en heldragen linje kan ritas mellan punkterna (x,y)= (125,250) och (700,875). Observera att parametrarna kan ges både i form av tecken i en sträng eller som ett numeriskt värde.

C

```

main()
ä
    printf("033:%d;%dm", 125, 250);
    printf("033:%d;%d;0;ld", 700, 875);

```

å

```

main()
ä
    printf("033:125;250dm");
    printf("033:700;875;0;ld");

```

å

FORTRAN

```

        WRITE (6,10) CHAR(27)
10     FORMAT (A,'A2J',α)
        WRITE (6,20) CHAR(27),125,250
20     FORMAT (A,':',I3,',';',';I3,'m',α)
        WRITE (6,30) CHAR(27),700,875
30     FORMAT (A,':',I3,',';',';I3,';0;ld',α)
        END

        WRITE (6,10) CHAR(27),'A2J'
10     FORMAT (2A,α)
        WRITE (6,10) CHAR(27),':125;250m'
        WRITE (6,10) CHAR(27),':700;875;0;ld'
        END

```

BASIC

```

10     PRINT CHRα(27);"A2J";
20     PRINT CHRα(27);": ";125;";";250;"m";
30     PRINT CHRα(27);": ";700;";";875;";0;ld";
40     END

```



```

10 PRINT CHR$(27);"A2J";
20 PRINT CHR$(27);":125;250m";
30 PRINT CHR$(27);":700;875;0;1d";
40 END

```

EXEMPEL 2

Följande exempel är skrivet i C, FORTRAN samt BASIC och visar rektanglar med prov på samtliga 16 mönster och dessas inverser.

C

```

main()
ä
    int i, j;

    printf("033A2J");
    for (i=0; i <= 15; i++)
        ä
            for (j=1; j >= 0; j--)
                ä
                    printf("033:%d;%dm", 400-300*j, 60*i);
                    printf("033:%d;%d;%d;%df", 600-300*j, 60*i+50, i, j);
                å
            å
        å
    å

```

FORTRAN

```

WRITE (6,10) CHAR(27)
10 FORMAT (A,'A2J',)
DO 100 I=0,15
    DO 200 J=1,0,-1
        WRITE (6,20) CHAR(27),400-300*J,60*I

```

```

20      FORMAT (A,':',I3,',';','I3,'m',□)
        WRITE (6,30) CHAR(27),600-300*J,60*I+50,I,J
30      FORMAT (A,':',I3,',';','I3,',';','I2,',';','I1,'f',□)
200     CONTINUE
100     CONTINUE
        END

```

BASIC

```

10     PRINT CHR(27);"A2J";
20     FOR I%=0 TO 15
30         FOR J%=1 TO 0 STEP -1
40             PRINT CHR(27);":":400-300*J%";":60*I%";"m";
50             PRINT CHR(27);":":600-300*J%";":60*I%+50;
60             PRINT ":":I%";":J%";"f";
70         NEXT J%
80     NEXT I%
90     END

```

EXEMPEL 3

Följande exempel är skrivet i C, FORTRAN samt BASIC och visar hur egna mönster kan definieras. Programexemplet presenterar mönstren i form av en rektangel och en linje.

C

```

main()
ä
    int hmask, vmask, shift, op;

a:printf("Ange 'hmask' ?Ön");
    scanf("%d", &hmask);

```

```

printf("Ange 'vmask' ?Ön");
scanf("%d", &vmask);
printf("Ange 'shift' ?Ön");
scanf("%d", &shift);
printf("Ange 'op' ?Ön");
scanf("%d", &op);
printf("Ö33Ä2J");
printf("Ö33:l;%d;%d;%d;%dR", hmask, vmask, shift, op);
printf("Ö33:%d;%dm", 100, 700);
printf("Ö33:%d;%d;l;ld", 600, 700);
printf("Ö33:%d;%dm", 100, 100);
printf("Ö33:%d;%d;l;lf", 600, 600);
goto a;

```

ä

FORTRAN

```

5  WRITE (6,*) 'Ange ''hmask'' ?'
    READ (5,*) IHMASK
    WRITE (6,*) 'Ange ''vmask'' ?'
    READ (5,*) IVMASK
    WRITE (6,*) 'Ange ''shift'' ?'
    READ (5,*) ISHIFT
    WRITE (6,*) 'Ange ''op'' ?'
    READ (5,*) IOP
    WRITE (6,10) CHAR(27)
10  FORMAT (A,'Ä2J',)
    WRITE (6,20) CHAR(27), IHMASK, IVMASK, ISHIFT, IOP
20  FORMAT (A,':l;', I5, ';', I5, ';', I2, ';', I1, 'R',)
    WRITE (6,30) CHAR(27), 100, 700
30  FORMAT (A,':', I3, ';', I3, 'm',)
    WRITE (6,40) CHAR(27), 600, 700
40  FORMAT (A,':', I3, ';', I3, ';l;ld',)
    WRITE (6,30) CHAR(27), 100, 100
    WRITE (6,50) CHAR(27), 600, 600
50  FORMAT (A,':', I3, ';', I3, ';l;lf',)
    GOTO 5
    END

```

BASIC

```

10 PRINT "Ange 'hmask' ?"
20 INPUT H%
30 PRINT "Ange 'vmask' ?"
40 INPUT V%
50 PRINT "Ange 'shift' ?"
60 INPUT S%
70 PRINT "Ange 'op' ?"
80 INPUT O%
90 PRINT CHR$(27);"A2J";
100 PRINT CHR$(27);":l";H%;";";V%;";";S%;";";O%;"R";
110 PRINT CHR$(27);":":100;";":700;"m";
120 PRINT CHR$(27);":":600;";":700;":l;ld";
130 PRINT CHR$(27);":":100;";":100;"m";
140 PRINT CHR$(27);":":600;";":600;":l;lf";
150 GOTO 10
160 END

```

EXEMPEL 4

Följande exempel är skrivet i C, FORTRAN samt BASIC och visar olika sätt att placera teckenmarkören i förhållande till den grafiska markören. Programmet börjar med att välja typsnittet 'O' och skriver ut texten 'CYKELPUMP' för att visa placeringen.

C

```

main()
ä
    int i;

    printf("Ö33A2J");
    printf("Ö33(%c", 'O');
    for (i = 100; i <= 300; i += 100)

```

```

ä
printf("033:%d;%dm", 100, i);
printf("033:%d;%d;0;ld", 400, i);
printf("033:%dH", i/100-1);
printf("CYKELPUMP");
å

```

å

FORTRAN

```

WRITE (6,10) CHAR(27)
10  FORMAT (A,'Å2J',)
WRITE (6,20) CHAR(27),'O'
20  FORMAT (A,'(',A,)
DO 100 I=100,300,100
    WRITE (6,30) CHAR(27),100,I
30  FORMAT (A,':',I3,',',I3,'m',)
    WRITE (6,40) CHAR(27),400,I
40  FORMAT (A,':',I3,',',I3,';0;ld',)
    WRITE (6,50) CHAR(27),I/100-1
50  FORMAT (A,':',I1,'H',)
    WRITE (6,60) 'CYKELPUMP'
60  FORMAT (A,)
100 CONTINUE
END

```

BASIC

```

10  PRINT CHR(27);"Å2J";
20  PRINT CHR(27);"("; "O";
30  FOR I%=100 TO 300 STEP 100
40    PRINT CHR(27);":":100;":":I%;"m";
50    PRINT CHR(27);":":400;":":I%";"0;ld";
60    PRINT CHR(27);":":I%/100-1;"H";
70    PRINT "CYKELPUMP";
80  NEXT I%
90  END

```

EXEMPEL 5

Följande exempel är skrivet i C, FORTRAN samt BASIC och visar hur SPREJ-funktionen kan användas. I detta fall sprejas ett rutemönster. Observera att programmet måste användas från ett fönster.

C

```

main()
ä
    int i, j;

    printf("033A2J");
    printf("033:2h");
    for (i = 1; i <= 40; i += 3)
        ä
            for (j = 1; j <= 40; j += 3)
                printf("033:%d;%d;2;ls", 100+i, 100+j);
        å
    å

```

FORTRAN

```

        WRITE (6,10) CHAR(27)
10     FORMAT (A,'A2J',x)
        WRITE (6,20) CHAR(27)
20     FORMAT (A,':2h',x)
        DO 100 I=1,40,3
            DO 200 J=1,40,3
                WRITE (6,30) CHAR(27),100+I,100+J
30         FORMAT (A,':',I3,':',I3,':2;ls',x)
200     CONTINUE
100    CONTINUE
        END

```

BASIC

```

10  PRINT CHR$(27);"A2J";
20  PRINT CHR$(27);":2h";
30  FOR I%=1 TO 40 STEP 3
40    FOR J%=1 TO 40 TO STEP 3
50      PRINT CHR$(27);":":100+I%";":100+J%";":2;1s";
60    NEXT J%
70  NEXT I%
80  END

```

EXEMPEL 6

Följande exempel är skrivet i C och visar hur information kan hämtas om musmarkörens läge och huruvida musens vänstra knapp är nedtryckt eller ej. Observera att kanalen 'fd' måste vara definierad till ett fönster. Genom att bestämma värdet på kanalen 'fd' knyter man den till det fönster man önskar arbeta med. 'fd' är alltid ett heltal.

```

/* detta är huvudprogrammet */

#include <win/w_const.h>
#include <win/w_types.h>
#include <win/w_structs.h>
#include <win/w_macros.h>

/* Strukturen 'winstruc' används för att skapa fönster */

struct winstruc win;

main()
ä
    int fd;
    int x, y, b;;

/* Fyll strukturen */

```

```
win.wp_xorig = 100;
win.wp_yorig = 100;
win.wp_xsize = 400;
.
.
.

/* Öppna en kanal för fönstret */
```

```
fd = open("/win", 2);
```

```
/* Skapa fönstret */
```

```
Wincreat(fd, &win);
```

```
draw(fd);
```

```
muspos(fd, &x, &y, &b);
```

```
.
.
.
```

å

```
muspos(fd, x, y, b) /* rapporterar muspekarens läge*/
```

```
int fd, *x, *y, *b;
```

ä

```
int i = 1, j = 0;
```

```
char sx[4], sy[5], but[2];
```

```
char buf[14];
```

```
strcpy(sx, " ");
```

```
strcpy(sy, " ");
```

```
strcpy(buf, " ");
```



```
write(fd, "033:2n", 4);
read(fd, buf, 13);
```

```
while(bufÄ++iÄ != ';' )
    sxÄi-2Ä = bufÄiÄ;
while(bufÄ++iÄ != ';' )
    syÄj++Ä = bufÄiÄ;
butÄ0Ä = bufÄ++iÄ;
```

```
*x= atoi(sx);
*y= atoi(sy);
*b= atoi(but);
```

```
ä
```

```
atoi(s) /* Omvandla en sträng till ett heltal */
char sÄÄ;
```

```
ä
```

```
int i, n;
```

```
n = 0;
```

```
for (i = 0; sÄiÄ >= '0' && sÄiÄ <= '9'; i++)
```

```
    n = 10 * n + sÄiÄ - '0';
```

```
return(n);
```

```
ä
```

```
#include <sgtty.h>
```

```
struct sgtyb tty;
```

```
crw(fd) /* Ändra kanalen till råmod */
```

```
int fd;
```

```
ä
```

```
ioctl(fd, TIOCGTTP, &tty);
```

```
tty.sg_flags = tty.sg_flags ö RAW;
```

```
ioctl(fd, TIOCSTTP, &tty);
```

```
ä
```

4 ORDFÖRKLARINIGAR

Nedanstående begrepp är förklarade speciellt med tanke på användning i datorsammanhang och de tillämpningar denna handledning omfattar. Orden är markerade med *kursiv fetstil* i sakregistret och första gången de uppträder i texten.

ASCII American Standard Code of Information Interchange. En standardisering av de tecken som används i informations-sammanhang. Standarden innebär att varje tecken har en given plats i en lista, med löpande numrering av tecknen. Här används sju-bitars ASCII vilket ger en lista med tecken från 0 till 127 decimalt. Exempel: 'J'=74 decimalt, '_'=95 decimalt.

typsnitt Teckenuppsättning av visst utseende. Stil.

baslinje Den linje som för ett typsnitt kan dras direkt nedanför versalerna och endast skärs av de gemenas underslängar. (Se figur sid 24).

**tecken-
sekvens** Teckenmängd ordnad i linjär följd.

**virtuellt
rum** Rum definierat av koordinatsystem som är oberoende av bildskärmen. (Se figur sid 13).

fönster Del av virtuellt rum som avgränsats för visning i form av en ruta på bildskärmen. (Se figur sid 13). Vanligen förekommer två typer av fönster.

I ett menyfönster ges information och valmöjligheter - t ex öppna eller stänga fönster, fortsätta till andra menyer och starta program.

I ett terminalfönster emuleras en bildskärmsterminal. Normalt förs dialogen med operativsystem och program från ett terminalfönster.

- rulla** Kontinuerligt förskjuta fönster i höjded eller sidled så att nya delar av bilden framträder i rutan - vanligen tecken.
- panorera** Som rulla, men används vanligen då bildskärmen visar grafer och diagram, till skillnad från text.
- markör** Flyttbart märke som för användaren visar aktuellt läge på bildskärmen.
- eko** Representation på bildskärmen av just inmatad information.
- eka** Åstadkomma ett *eko*.
- mus** Till utseendet musliknande positionerare som med handen förs över en yta.
- musmarkör** Med hjälp av musen flyttbart märke som för användaren visar aktuellt läge på bildskärmen.
- markör-
placerare** Den stora vita kvadratiska tangenten på tangentbordet.
- adressera** Nå element genom angivelse av dess koordinater eller annan lägesbestämmande information.
- bitmappa** Bestäm utseende på visst element genom att adressera enskilda punkter. Bitmönster.
- standard-
tolk** Kallas även SH (eng: 'SHell') eller skal. Läser av inmatningar från tangentbordet. Tolken ger en prompt, t ex 'n', varefter det inmatade kommandot exekveras - t ex starta program.
- kommando-
tolk** Kallas även WSH (eng: 'Window SHell') eller fönstertolk. Läser av inmatningar från fönster. Tolken ritar upp ett fönster varefter den specificerade uppgiften utförs - t ex starta ett program direkt från fönstret. Att låta WSH ersätta SH kan innebära avsevärda arbetsbesparingar.

5 SAKREGISTER

De ord som markerats med *kursiv fetstil* i förteckningen nedan återfinnes i kapitel 4 ORDFÖRKLARINGAR.

ABC1600 terminalemulator.....	14, 36-40
<i>adressera</i>	10, 12, 38, 52
ALT-tangenten.....	34
<i>ASCII</i>	33, 51
automatisk mod - se moder	
avslutande operator.....	14
<i>baslinje</i>	24, 31, 33, 51
bitmapdata.....	35, 36
<i>bitmappa</i>	35, 36, 52
bitmönster - se bitmapdata, bitmappa	
'bottom margin'.....	15
'bytes'.....	38
CR - se kontrolltecken	
DEC VT100.....	14, 19, 37
<i>eka</i>	37, 39, 52
<i>eko</i>	52
emulera - se ABC1600 terminalemulator	
ESC-sekvenser	
motsvarande DEC VT100 och FACIT TWIST	
Å<Pn>A.....	14
Å<Pn>B.....	15
Å<Pn>C.....	15
Å<Pn>D.....	15
Å<Pn><Pn>H.....	16
Å<Pn><Pn>f.....	16
Å<Pn><Pn>r.....	16
Å<Ps>J.....	16
Å<Ps>K.....	17
D.....	17
E.....	17
M.....	18
7.....	18

B.....	18
c.....	19
ÄZ.....	20
H.....	20
Ä<Ps>g.....	20
Ä<Ps>;<Ps>;...;<Ps>m.....	20
Ä<Ps>;<Ps>;...;<Ps>n.....	21
Ä<Pn>;<Pn>R.....	21
Ä?Pn.....	21
Ä?Ln.....	21
Ä<Ps>;<Ps>;...;<Ps>q.....	22
Ä<Ps>;<Ps>;...;<Ps>h.....	22
Ä<Ps>;<Ps>;...;<Ps>l.....	23
(A eller)A.....	24
t o m	
(Z eller)Z.....	24

ABC1600's privata

:<x>;<y>;<pno>;<cno>d.....	25
:<x>;<y>i.....	25
:<x>;<y>m.....	25
:<x>;<y>;<op>;<cno>p.....	26
:<x>;<y>;ll;<cno>p.....	26
:<x>;<y>;llp.....	26
:<x>;<y>;<len>;<pno>;<cno>a.....	27
:<x>;<y>;<len>I.....	27
:<x>;<y>;<pno>;<cno>f.....	27
:<x>;<y>;<rad>;<pno>;<cno>c.....	28
:<x>;<y>;<pno>;<cno>F.....	28
:<xsrc>;<ysrc>;<xdest>;<ydest>;<width>; <height>;<op>r.....	29
:<pno>;<hmask>;<vmask>;<shift>;<op>R.....	29
:<sel>M.....	31
:<x>;<y>;<buttons>P.....	32
:<sel>n.....	32
:<x>;<y>R.....	32
:<vsx>;<vsy>;<fsx>;<fsy>;<bl>;<fno>W.....	33
:<x>;<y>O.....	33
:J.....	33

:<sel>;<sel>;...;<sel>q.....	34
:<sel>;<sel>;...;<sel>h.....	34
:<sel>;<sel>;...;<sel>l.....	35
:<x>;<y>;<pno>;<op>s.....	35
:<time>S.....	36
 Tektronixmodens	
:4014;<strap>T.....	36
Ä.....	37
SUB.....	38
ENQ.....	39
CTRL-L.....	40
x.....	40
 escapesekvens - se ESC-sekvenser	
FACIT TWIST.....	14, 21, 22, 23
FACIT TWIST mod ?NN.....	21-24
flagga.....	19
<i>fönster</i> - se även toppnivåfönster.....	51
förklaringar av ord.....	51
grafikmarkör.....	10, 18, 19, 25-33, 35, 36
grafiskt origo.....	12, 18, 19, 33
GS - se kontrolltecken	
handledningen.....	6
'home'.....	10
horisontalmask - se parametrar <hmask>	
hårkorsmarkör.....	39
INS-tangenten.....	34
kanalen 'fd'.....	48-50
kommando (= ESC-sekvens) - se ESC-sekvens	
kontrollkoder - se kontrolltecken	
kontrolltecken	
CR.....	22, 23, 37-39
GS.....	38
LF.....	22
kopiera en yta.....	29
'landscape mode' - se moder	
landskapmod - se moder	
'left margin'.....	15
LF - se kontrolltecken	

linjefärg.....	25, 40
lysdiod.....	19, 22, 34
manuell mod - se moder	
markör - se även grafikM, musM, teckenM.....	10, 52
markörplacerare.....	38, 52
mask - se även sprejmask.....	29, 30, 35, 36
menyfönster.....	53
mod ?NN - se FACIT TWIST mod ?NN	
moder	
automatisk mod.....	23
landskap-mod.....	10, 21, 29
manuell mod.....	22, 23
normalmod.....	23
porträtt-mod.....	10, 21, 29
privata moder.....	34, 35
rullande mod.....	24
sidmod.....	23
skärmmod.....	23
Tektronixmod.....	36
BYPASS.....	37, 39
GIN.....	36-39
GRAPH.....	36-38
teckenmarkör - se teckenmarkör	
- se även FACIT TWIST mod ?NN	
mus.....	52
musmarkör.....	31-33, 48, 52
normalmod - se moder	
operation - se parametrar <op>	
operator - se även avslutande operator.....	14, 29, 35
ordförklaringar.....	51
origo.....	28
osynlig.....	12, 23
panorera.....	36, 52
parametrar.....	14, 36, 37
<bl>.....	33
<buttons>.....	32
<cno>.....	25-29
<fno>.....	33
<fsx>.....	33

<fsy>.....	33
<height>.....	29
<hmask>.....	29, 30
<len>.....	27
<op>.....	26, 29, 35
<Pn>.....	14-16, 21
<pno>.....	25, 27-30, 35
<Ps>.....	16, 17, 20-23
<sel>.....	30-35
<shift>.....	29, 30
<strap>.....	36, 37, 39
<time>.....	36
<vmask>.....	29
<vsx>.....	33
<vsy>.....	33
<width>.....	29
<xdest>.....	29
<xsrc>.....	29
<ydest>.....	29
<ysrc>.....	29
'pixels'.....	10
'portrait mode' - se moder	
porträttmod - se moder	
privat mod - se moder	
'reverse character'.....	21
'right margin'.....	15
rotera.....	30
rulla.....	16-19, 23, 24, 33, 36, 52
rullande mod - se moder	
rullning - se rulla	
råmod.....	50
sidmod - se moder	
skärmmod - se moder	
släcktid.....	36
'space'.....	38
sprejmask.....	35, 36
standardtypsnitt.....	19, 34
statusrapport.....	21, 22
syntax.....	6

tangentbordets lysdioder - se lysdiod	
tecken	
' '.....	38, 39
';'.....	14
'É'.....	38
'é'.....	38
teckenmarkör.....	10, 15-24, 30, 33, 45
Tektronix-emulator.....	36
Tektronix-mod - se moder	
terminalen DEC VT100 - se DEC VT100	
terminalen FACIT TWIST - se FACIT TWIST	
terminalfönster.....	53
'top of margin'.....	15
toppnivå.....	32, 34
toppnivåfönster - se toppnivå	
<i>typsnitt</i>	11, 18, 19, 24, 31, 33, 34, 51
- se även standardT	
typsnittruta.....	31, 33
val - se parametrar <sel>	
vektor.....	38
vektorgenerator.....	40
vertikalmask - se parametrar <vmask>	
<i>virtuellt rum</i>	16, 19, 23, 26, 32-34 51

Vi förbehåller oss rätten att justera produkten utan avisering samt full ansvarsfrihet för följderna av tryckfel eller felaktig användning.

Luxor Datorer AB.

produktion: IFS DATA AB Linköping och Luxor Datorer AB Motala

TILLÄGG

VÄLJ LINJETYP (Tektronix-moden, se sid 40)

Följande escape-sekvenser väljer vektorgeneratorns linjefärg.

ESC é	Heldragen linje.
ESC a	Prickad linje.
ESC b	Streck-prickad linje.
ESC c	Korta streck.
ESC d	Långa streck.
ESC x	Radera linje.