

Tabellarische Übersicht

Freie Speicherbereiche:

Zeropage: \$D8 bis \$E8 (Anwendersoftware)
 \$61 bis \$70 (Fließkommaakkumulatoren)

Sonstige Bereiche:

\$0200 bis \$02AC (Basic-Eingabepuffer/DOS-Arbeitsbereich)
 \$025D bis \$02AC (aktuelle DOS-Parameter, zum Beispiel Filenamen)
 \$0332 bis \$03F6 (Kassettenpuffer)
 \$03F7 bis \$0436 (RS232-Puffer)
 \$055F bis \$05E6 (Funktionstastenbelegung)

Speicherbereiche/Zeiger:

Farbspeicher :\$0800 bis \$0BFF
 Bildschirmspeicher :\$0C00 bis \$0FFF
 Zeiger Basic-Anfang (Standard \$1000) :\$2B/\$2C
 Zeiger Basic-Ende (Standard \$3FFF) :\$33/\$34

Betriebssystemroutinen:

Adresse	Label	Funktion	Übergabeparameter
\$FFE4	GETIN	Zeichen holen	keine; Zeichenübergabe im Akku
\$FFCF	BASIN	Zeichen holen bis RETURN	keine; Zeichenübergabe im Akku
\$FFD2	BSOUT	Zeichen ausgeben	Zeichen im Akku übergeben
\$FFF0	PLOT	Cursor setzen/ Position holen	Zeile(X)/Spalte(Y)/Carry Clear Carry Set/Positionsübergabe in X/Y
\$FFBA	SETLFS	Filepar. setzen	Filenr.(A)/Gerät(X)/Sek.Adresse(Y)
\$FFBD	SETNAM	Filename setzen	Länge Name(A)/Adresse Name(X/Y)
\$FFD5	LOAD	Programm laden/ Ladeadresse Verify durchführen (X/Y) bei relativem Laden, absolutes Laden bei Sek.-Adresse 1 (SETLFS)	Load(0)-Verify(1)- Flag(A)/ Ladeadresse
\$FFD8	SAVE	Programm speichern	Zeropage-Adresse des Zeigers auf die Anfangsadresse(A)/ Endadresse+1(X/Y)

Reaktionstest

Mit diesem Einzeiler können Sie Ihre Reaktionszeit messen. Um die Länge einer Basic-Zeile nicht zu überschreiten, müssen einige Befehle in abgekürzter Schreibweise eingegeben werden. Nach dem Start des Einzeilers erscheint nach einiger Zeit das Signal »T=«. Nun müssen Sie so schnell wie möglich eine beliebige Taste drücken. Nachdem die Reaktionszeit (in Sekunden) auf dem Bildschirm erschienen ist, beginnt auch gleich der nächste Test. Etwaigem Schummeln wurde vorgebeugt: Beim frühzeitigen Drücken einer Taste wird der Test abgebrochen.

```
2 FOR=0TORND (1) *2500+900:NEXT:GETA$:IFA$=
""THENPRINT "T=";:T=TI:GETKEYA$:PRINT (TI-T) /60:RUN
(Hannes Kaltenbach/tr)
```

RESET ohne Reset-Schalter

Nach langem Herumprobieren wurde im Betriebssystem die Routine gefunden, die einen Reset-Schalter überflüssig macht. Die Routine wird vom Basic aus mit dem SYS-Befehl wie folgt angesprungen:

```
SYS 65529
```

Bei diesem Befehl werden alle Zeiger für das Basic und für das Betriebssystem in den Ausgangszustand zurückgesetzt (Basic-Kaltstart). Alle Maschinenprogramme bleiben jedoch erhalten und können wie gewohnt aufgerufen werden.

(J. Kunz/ah)

Grafik und Sound mit dem C 16

Der beste Computer nützt einem wenig, wenn man seine Fähigkeiten nicht einzusetzen weiß. Wir zeigen, wie man mit dem C16 Grafik und Sound programmiert.

Als Commodore 1984 die beiden Computer C 16/C 116 und Plus 4 auf den Markt brachte, ahnte wohl noch niemand, daß sich der C 16 gegen seinen größeren Bruder durchsetzen würde. Doch es geschah noch mehr: Der Plus 4 verschwand schon nach kurzer Zeit wieder von den Ladentischen. Er war für seine mäßige Leistung einfach zu teuer. Der C 16 hingegen konnte sich bis heute auf dem Markt halten. Dies liegt wohl an seinem geringen Preis: er ist im Moment schon für weit unter 300 Mark erhältlich. Für die Ausführung mit Gummitastatur des C 116 sind sogar zirka 50 Mark weniger zu berappen. Ob sich diese Einsparung jedoch lohnt, ist Ansichtssache. Wer zahlt nicht gerne 50 Mark mehr, um auf einer sehr guten und altbewährten Schreibmaschinentastatur ermüdungsfrei arbeiten zu können?

Wenn im folgenden Artikel vom C 16 die Rede ist, so ist damit natürlich ebenso der C 116 gemeint.

Der C 16 eignet sich besonders für Einsteiger, die mal in die Computerei hineinschnüffeln wollen. Der geringe Preis des Gerätes setzt die Hemmschwelle für Anfänger weit herunter.

Woran liegt es nun, daß der C 16 trotz seines geringen Preises nicht ein Renner wie beispielsweise der C 64 wurde?

Das einzige große Manko des C 16 ist sein geringer Speicherplatz. Er stellt dem Benutzer nur ganze 16 KByte zur Verfügung. Schaltet man in den Grafikmodus, dann bleiben sogar nur noch 2 KByte für das Programm. Abhilfen in Form von Speichererweiterungen sind vorhanden, aber nicht unbedingt billig. Von einigen Herstellern (außer Commodore) werden sie inzwischen angeboten.

Ansonsten läßt der C 16 kaum Wünsche offen (zumindest in seiner Preisklasse): Erstmals rüstet Commodore die Computer C 16 und Plus 4 mit dem verbesserten Basic 3.5 aus. Endlose Programme mit PEEK und POKE-Sequenzen gehören der Vergangenheit an. Sound und Grafik können bequem über Basic-Befehle programmiert werden (in diesem Punkt ist er sogar dem C 64 überlegen). In diesem Artikel wollen wir uns vor allem der Grafik- und Sound-Programmierung widmen.

Zur Grafik gehören natürlich auch die Farben. Der C 16 bietet im Gegensatz zum VC 20 oder C 64 erstmals die Möglichkeit, 121 verschiedene Farbstufen darzustellen. Diese setzen sich aus 16 Farben zusammen, von denen 15 in 8 Helligkeitsstufen dargestellt werden können. Für die sechzehnte Farbe (Schwarz) gibt es keine Helligkeitsunterschiede.

Das Beispielprogramm »FARBDEMO« (siehe Listing 1) zeigt alle 121 Farben gleichzeitig am Bildschirm. Zur Farbprogrammierung findet man im Befehlssatz den »COLOR«-Befehl. Ihm sollten drei Parameter folgen:

COLOR Bereich, Farbe, Helligkeit

Die Bereich- und Farb-Zuordnung können Sie den Tabellen 1 und 2 entnehmen. Die Helligkeit kann zwischen 0 (sehr dunkel) und 7 (sehr hell) variiert werden. Ebenso, wie man bei den älteren Commodore-Geräten den jeweiligen Farbwert mittels »PEEK« wieder abfragen konnte, gibt es im neuen Basic 3.5 die Funktionen »RCLR(X)«, um die Farbe des Bildschirmpunktes X auszugeben, und »RLUM(X)«, um die Helligkeit der Farbe zu bestimmen.

Die verschiedenen Grafikmodi

Soviel zu den Farben. Mit den Grafiksymbolen des C16 lassen sich zwar viele einfache Grafiken (Blockgrafiken) erstellen, aber man stößt sehr schnell an die Grenzen des Machbaren. Um auch komplexere Grafiken zu erstellen, muß in den Grafik-Modus umgeschaltet werden. Man unterscheidet zwei Grafik-Modi: zum einen hochauflösende Grafik, zum

anderen Multicolor-Grafik. Im HiRes-Modus können 64000 Punkte einzeln angesprochen werden (siehe Bild 1). Hier können jedoch in jeder Cursorposition (8 x 8 Einzelpunkte) nur zwei Farben dargestellt werden (Zeichen- und Hintergrundfarbe). Bis zu vier Farben lassen sich im Multicolor-Modus darstellen, allerdings auf Kosten der Auflösung: Pro Zeichen können nur noch 4 x 8 Punkte dargestellt werden. Daraus ergibt sich eine Gesamtauflösung von 32000 Grafikpunkten (siehe Bild 2).

Der Grund für diese Reduzierung der Auflösung läßt sich so erklären:

Im HiRes-Modus genügt ein Bit, um einen Punkt zu setzen, da die Zeichenfarbe hier entweder ein- oder ausgeschaltet wird. Im Multicolor-Modus muß jedem Bildpunkt aber auch noch eine von vier Farben zugeteilt werden. Das heißt, man muß jedem Punkt einen Wert zwischen Null und Drei zuordnen. Hierfür genügt aber nicht mehr ein Bit. Um vier unterschiedliche Zustände darstellen zu können, benötigt man also 2 Bit ($2^2 = 4$). Da aber im Multicolor-Modus nicht doppelt so viel Speicherplatz aufgewendet werden kann, muß man für Farbe die halbe Auflösung in Kauf nehmen.

Die beiden Grafik-Modi lassen sich mit dem »GRAPHIC«-Befehl ein- und ausschalten. Dem Befehl sollten zwei Parameter folgen, von denen der erste den Modus angibt (Tabelle 3). Ist der zweite Parameter eine Eins, so wird der Grafik-Bildschirm gelöscht. Folgt kein zweiter Parameter oder ist dieser Null, so bleibt die Grafik erhalten. Wollen Sie im Programm erfahren, in welchem Modus Sie sich befinden, so gibt die Funktion »RGR(X)« den gegenwärtigen Modus aus (Tabelle 3). Befindet man sich schon im Grafikmodus und soll der Bildschirm gelöscht werden, so benützt man den »SCNCLR«-Befehl. Verläßt man den Grafikmodus mit »GRAPHIC 0«, so stehen dem Benutzer trotzdem nur 2 KByte Speicher zur Verfügung, da das Grafikbild bis zum erneuten Einschalten erhalten bleibt. Erst nach dem Befehl »GRAPHIC CLR« werden die restlichen 14 KByte freigegeben. In diesem Fall bleibt auch die Grafik nicht mehr erhalten.

In beiden Grafik-Modi befindet sich der Koordinatensprung (0/0) links oben. Damit zum Beispiel beim Konstruieren eines Graphen nicht jedesmal überlegt werden muß, in welchem Modus man sich gerade befindet, gibt es den »SCALE«-Befehl. Schaltet man die Skalierung mit »SCALE 1« ein, so stehen, unabhängig vom Modus, auf den X- und Y-Achsen jeweils die Werte zwischen 0 und 1023 zur Verfügung. Das heißt aber nicht, daß sich auch die Auflösung auf 1048576 (1024 x 1024) Punkte erhöht hat. Es wird vielmehr mit mehreren benachbarten Adressen jeweils ein- und derselbe Punkt angesprochen. Abgeschaltet wird die Skalierung mit »SCALE 0«.

WERT	BEREICH
0	Bildschirm-Hintergrund
1	Vordergrund (Zeichen, Cursor)
2	Zusatzfarbe 1 (für Multicolor)
3	Zusatzfarbe 2 (für Multicolor)
4	Bildschirmrand

Tabelle 1. Die Zuordnung der Farbparameter im Multicolor-Modus

WERT	Farbe	WERT	Farbe
1	Schwarz	9	Orange
2	Weiß	10	Braun
3	Rot	11	Gelbgrün
4	Cyan	12	Rosa
5	Purpur	13	Blaugrün
6	Grün	14	Hellblau
7	Blau	15	Dunkelblau
8	Gelb	16	Hellgrün

Tabelle 2. 16 Grundfarben lassen sich mit dem C16 darstellen

```

10 REM *****
20 REM *
30 REM * (C 16) FARBDEMO (116) *
40 REM * ----- *
50 REM * CHRISTIAN QUIRIN SPITZNER *
60 REM * GRUBERSTR. 53, 8011 POING *
70 REM * TELEFON: 08121/81100 *
80 REM *
90 REM *****
100 COLOR0,4
110 COLOR4,4
120 PRINT "CLR,DOWN,BLACK,10SPACE)F A R
B D E M O(CTRL-@)F(CTRL-@,DOWN,FLASHON,CTRL-@,L
IG.GREEN)"
140 PRINT "H(2SPACE)S W R Z P G B G O B
G R B H D H(CTRL-@,HOME,RVSON,LIG.RED,CT
RL-@,LIG.GREEN)" E C E O Y U R L E R R
E O L E U E
160 PRINT "L(2SPACE)H I T A R U A L A A
L S A L N L(CTRL-@)G(RVSON)W(CTRL-@,LIG.
GREEN)" L W E . N P E U B N U B A U L K
L
180 PRINT "I(2SPACE)A S . . U N . . G .
G . G B E G(CTRL-@)F(RVSON)V(CTRL-@,LIG.
GREEN)" G R S . . R . . . E . R . R L L
R
200 PRINT "K(2SPACE)Z . . . . .
U . U A B U(CTRL-@,CTRL-O,HOME)J(CTRL-@,
LIG.GREEN)" E . . . . . E . E
U L E
220 PRINT "I(2SPACE). . . . .
N . N . A N(CTRL-@)J(HOME)F(CTRL-@,LIG.G
REEN)" T . . . . . U
."(CTRL-@)J(HOME)W(CTRL-@,LIG.GREEN,CTRL
-@)J(HOME)G(CTRL-@,ORANGE)J(CTRL-@,F6,HOME,CTRL-D,CTRL-A):{2SPACE,LIG.
GREEN}"STOP"J:{CTRL-@,LEFT,HOME,CTRL-N
,CTRL-A}:{2SPACE,ORANGE}I(CTRL-@,F7,HOME,CTRL-X,CTRL-A):{2SPACE}J 1,I,J{C
TRL-@}J(HOME)": PRINT "{SPACE,RVSON,SPAC
E,RVDF}";
300 : NEXT I
310 : PRINT
320 NEXT J
330 GETKEYA$
664'er

```

Listing. Dieses Demo-Programm bringt alle 121 Farben des C16 auf den Bildschirm (siehe auch Seite 76)

Genau wie im Textmodus, steht dem Benutzer im Grafik-Modus ein Cursor zur Verfügung. Dieser Pixel-Cursor bleibt, solange nicht gezeichnet wird, unsichtbar. Um ihn an eine beliebige Stelle des Bildschirms zu bewegen, benützt man den »LOCATE«-Befehl:

LOCATE X-Koordinate, Y-Koordinate.

Die X- und Y-Werte sind natürlich abhängig vom jeweiligen Modus beziehungsweise Skalierung. Der Pixel-Cursor bleibt jeweils auf dem zuletzt bearbeiteten Punkt stehen und kann mit der Funktion »RDOT(X)« abgefragt werden. Setzt man für

X Null ein, so wird die X-Koordinate angegeben. Die Y-Koordinate kann für X gleich 1 in Erfahrung gebracht werden. Setzt man in die »RDOT«-Funktion die Zahl 2 ein, so wird die aktuelle Farbe des Pixel-Cursors ausgegeben. Ein Beispiel LOCATE 100,150:PRINT RDOT (1) ergibt 150.

Grafikbefehle

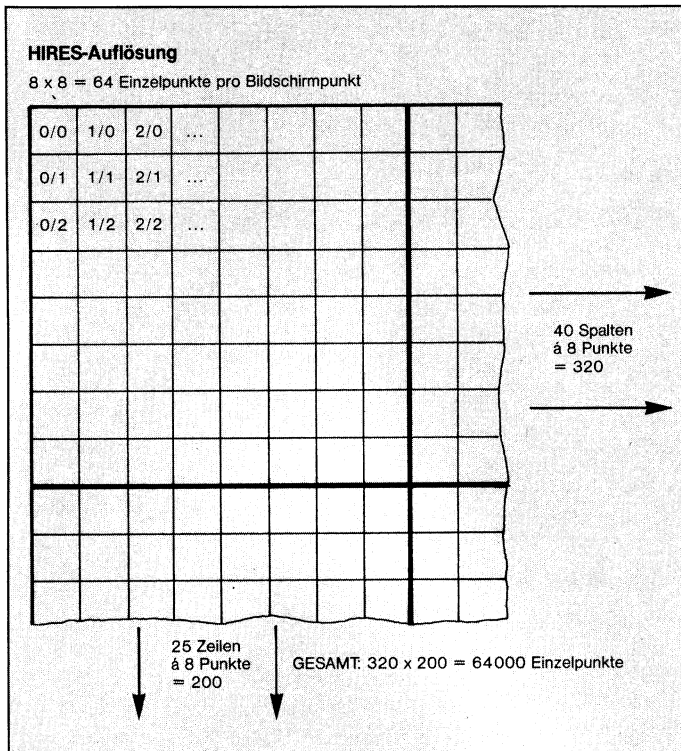


Bild 1. Im HiRes-Modus lassen sich 64000 Punkte darstellen

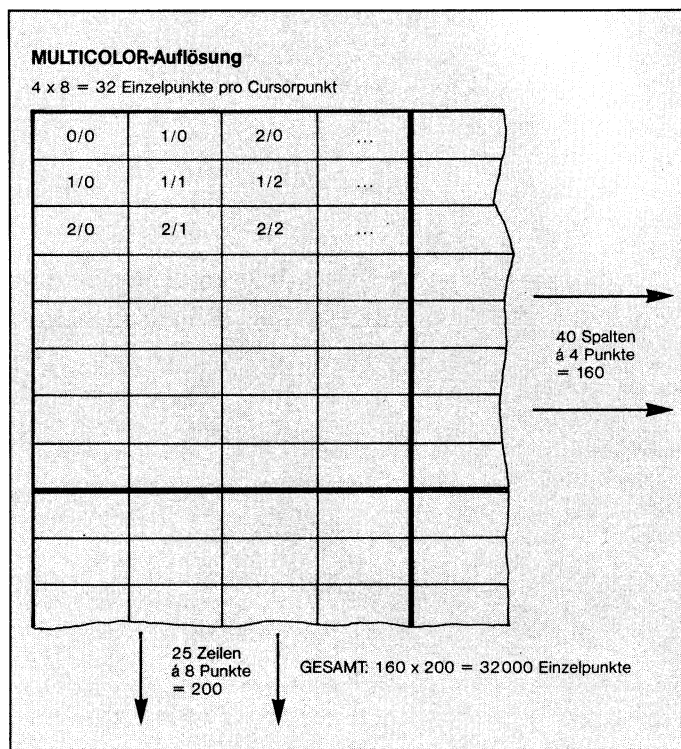


Bild 2. Im Multicolor-Modus reduziert sich die Auflösung auf 32000 Einzelpunkte

Gezeichnet wird mit Hilfe des »DRAW«-Befehls. Mit ihm lassen sich Punkte, Linien und ganze Linienzüge darstellen. Für einen Punkt sind zusätzlich drei Parameter notwendig: Die erste Zahl gibt die Farbe an. Sie kann im HiRes-Modus maximal zwei Werte annehmen:

0 = Hintergrundfarbe (=Punkt löschen)

1 = Zeichenfarbe (=Punkt setzen)

Im Multicolor-Modus stehen zwei zusätzliche Werte zur Auswahl:

2 = Zusatzfarbe 1 (siehe Color-Befehl)

3 = Zusatzfarbe 2 (siehe Color-Befehl)

Der zweite und dritte Parameter des »DRAW«-Befehls geben die X- und Y-Koordinaten des Punktes an. Der Befehl »DRAW 1,160,100« setzt im HiRes-Modus einen Punkt in der Bildschirmmitte. Soll eine Linie gezeichnet werden, so sind weiter zwei Parameter nötig, die die Koordinaten des Endpunktes angeben. Diese beiden Parameter werden mit »TO« abgetrennt:

DRAW 1,0,0 TO 319,199

Um eine Linienkette zu zeichnen, können weitere Punkte mit »TO« abgetrennt werden.

DRAW 1,0,0 TO 319,0 TO 319,199 TO 0,199 TO 0,0

Definiert man im »DRAW«-Befehl keinen Anfangspunkt, so wird eine Linie von der letzten Cursorposition zum Endpunkt gezogen.

DRAW 1 TO 100,100

Mit der »DRAW«-Anweisung lassen sich also auch Rechtecke und Quadrate zeichnen. Allerdings benötigt man die Koordinaten aller Eckpunkte. Rechtwinklige Vierecke lassen sich mit dem »BOX«-Befehl jedoch viel einfacher zeichnen. Hier müssen nur noch Farbe (0 – 3) und die Koordinaten der gegenüberliegenden Eckpunkte angegeben werden. Diese Rechtecke können natürlich auch gedreht dargestellt werden. Zu diesem Zweck gibt man als sechsten Parameter noch den Drehwinkel in Grad ein. Soll das Rechteck zudem noch ausgefüllt werden, muß als siebter Parameter die Zahl 1 folgen. Mit folgendem Beispiel wird in der Mitte des HiRes-Grafikbildschirms ein auf der Spitze stehendes, ausgefülltes Quadrat dargestellt:

BOX 1,110,50,210,150,45,1

Ein weiterer nützlicher Befehl des Basic 3.5 ist die »CIRCLE«-Anweisung. Mit ihr lassen sich Kreise, Ovale, Kreissegmente und Vielecke zeichnen. Für einen einfachen Kreis genügen schon vier Parameter:

»CIRCLE (Farbzone), (X-Koordinate des Mittelpunkts), (Y-Koordinate des Mittelpunkts), (Radius)«

zum Beispiel: CIRCLE 1,160,100,100

WERT	Modus
0	Textmodus
1	HiRes-Grafik
2	HiRes-Grafik (Zeile 1 – 20) & Text (Zeile 21 – 25)
3	Multicolor-Grafik
4	Multicolor-Grafik (Zeile 1 – 20) & Text (Zeile 21 – 25)

Tabelle 3. Fünf verschiedene Grafikmodi stehen dem Programmierer zur Verfügung

An einem Fernsehgerät sieht dieser Kreis eventuell nicht rund, sondern oval aus. Um auch hier einen Kreis zu zeichnen, kann zusätzlich noch eine Zahl für die Höhe eingegeben werden.

CIRCLE 1,160,100,100,84

Kreissegmente lassen sich mit zwei weiteren Parametern darstellen, indem man noch Anfangs- und Endwinkel (in Grad) angibt. Soll das Ganze auch noch gedreht werden, so kann als achter Parameter noch der Drehwinkel angegeben werden. Aber der »CIRCLE«-Befehl zeichnet nicht nur Kreise, Ovale oder deren Segmente. Es lassen sich ebenso Dreiecke, Vierecke und Vielecke darstellen. Als Zusatzangabe muß hier als neunter Parameter der Winkel in den Ecken des n-Ecks angegeben werden. Dieser Wert errechnet sich für ein n-Eck folgendermaßen:

Wert = $360/n$

Mit der »PAINT«-Anweisung lassen sich geschlossene Flächen farbig ausfüllen. Dem »PAINT«-Befehl folgen maximal vier Parameter: »PAINT (Farbzone), (X-Koordinate), (Y-Koordinate), (Modus)«.

Farbig gefüllt wird die Fläche, in der die angegebenen Koordinaten liegen. Die Grenzen müssen entweder dieselbe Farbe oder (falls Modus 1 gewählt wurde) eine der beiden Zusatzfarben besitzen. Hier ist jedoch jedoch Vorsicht geboten, denn befindet sich irgendwo eine Lücke in der Begrenzung, so wird der gesamte Bildschirm gefüllt.

Schriftzüge

Die schönste Grafik ist ohne Beschriftung aber nur halb so schön. Um Grafiken mit dem nötigen Text zu unterlegen, steht die »CHAR«-Anweisung zur Verfügung. Sie funktioniert ähnlich wie der PRINT-Befehl. Als zusätzliche Parameter sind Farbzone, Zeile, Spalte und eventuell Reverse-Flag gefragt: »CHAR (Farbzone), (Spalte (0-39)), (Zeile (0-24)), (STRING), (REVERS-Flag 0 = AUS; 1 = EIN)«

Folgendes Beispiel schreibt in der fünften Spalte und zehnten Zeile in reverser Schrift »C16/116«:

CHAR 1,4,9,"C 16/116",1

Die Fülle dieser Grafik-Befehle ist bestimmt eine Grundlage für kleine oder größere Experimente. Sie werden mit ein wenig Übung und Phantasie feststellen, daß mit geringem Aufwand erstaunliche Effekte erzielt werden können. Im nächsten Absatz werden wir uns mit den Shapes befassen.

SHAPES

Da der C16 im Gegensatz zum C64 über keine Sprites verfügt, versucht man sich beim C16 mit sogenannten »Shapes« zu behelfen. Shapes sind rechtwinklige Ausschnitte aus dem Grafikbildschirm, die einer Stringvariablen übergeben wurden.

Diese Ausschnitte lassen sich dann beliebig an jede Stelle des Bildschirms setzen. Das Shape kann nach Belieben in fünf verschiedenen Modi wiedergegeben werden. Das Shape kann entweder revers oder wie die Originalfläche dargestellt werden. Es läßt sich ebenso mit dem neuen Hintergrund »logisch« verknüpfen. Die entsprechenden Modi können Sie der Tabelle 4 entnehmen.

WERT	Wiedergabe-Modus
0	Wie Original
1	Revers
2	Oder-Verknüpfung mit Grafikfläche
3	UND-Verknüpfung mit Grafikfläche
4	Exklusiv-Oder-Verknüpfung mit Grafikfläche

Tabelle 4. Shapes lassen sich mit Grafiken logisch verknüpfen

Zur Shapehandhabung stehen dem Benutzer zwei Befehle zur Verfügung. Der »SSHAPE«-Befehl übernimmt eine Rechteckfläche aus dem Grafikbild in eine Basic-Stringvariable. Der Befehl hat folgende Syntax:

»SSHAPE Stringvariable, X1,Y1,X2,Y2«

Hierbei geben die vier Parameter die Eckkoordinaten des Shapes an. X2 und Y2 liegen diagonal gegenüber den Koordinaten X1 und Y1. Auch hier sind »skalierte« Werte möglich. Da aber eine Stringvariable nur maximal 255 Zeichen lang sein darf, können die Shapes nicht beliebig groß sein. Die Größe eines Sprites (wie beim C64) kann jedoch spielend erreicht werden. Um ein Shape an einer anderen Stelle des Grafik-Bildschirms wiederzugeben, steht der »GSHAPE«-Befehl zur Verfügung. Dieser Befehl hat folgendes Format: »GSHAPE (Stringvariable), X1, Y1 (Modus)

Weil die Größe des Shapes zu diesem Zeitpunkt schon feststeht, sind hier nur die Koordinaten des linken, oberen Eckpunktes des Shapes anzugeben. Der letzte Parameter bestimmt den Wiedergabemodus (Tabelle 4).

Durch die verschiedenartigen Verknüpfungsmöglichkeiten stehen dem Programmierer erstaunliche Fähigkeiten zur Verfügung. Shapes sind zwar kein Ersatz für Sprites, können aber auch zu reizvollen Ergebnissen und Effekten führen.

Sound

Warum Commodore bei diesem Computer auf die altbewährten Sprites verzichtet hat, ist ebenso wenig zu verstehen wie der Verzicht auf den SID, den Sound-Chip des C64.

Während man beim C64 noch von »Sound«-Programmierung sprechen konnte, begnügt sich der C16 mit »Tongeneratoren«. Tongenerator 1 erzeugt nur Töne, während Tongenerator 2 auch Geräusche von sich geben kann. Generator 1 wird über Stimme 1 gesteuert. Stimme 2 verwaltet den Tongenerator 2, während die dritte Stimme den Rauschgenerator des zweiten Tongenerators übernimmt.

Da die Tongeneratoren weder Filter, Hüllkurven noch sonstige Effekte beherrschen, ist die Tonerzeugung auf dem C16 recht einfach: Die Lautstärke kann mit dem »VOL«-Befehl zwischen 0 (Lautstärke aus) und 8 (maximale Lautstärke) geregelt werden. Der Befehl setzt sich aus dem Befehlswort einer Zahl zusammen:

»VOL x (zwischen 0 und 8)«

Die drei Stimmen werden mit dem »SOUND«-Befehl angesprochen. Hier müssen drei Parameter folgen. Der erste gibt Auskunft über die Stimme. Parameter Nummer 2 bestimmt den Notenwert, während der dritte Parameter die Länge des Tones angibt:

»SOUND (Stimme), (Note), (Länge)«

Der Wert für die Tonhöhe kann zwischen 0 und 1015 variiert werden. Die Länge des Tones wird in 1/50stel Sekunden angegeben und kann zwischen 0 und 65535 liegen. Es kann somit eine maximale Tonlänge von über 20 Minuten erreicht werden.

Obwohl die Tongeneratoren mit dem SID des C64 nicht mithalten können, bieten sich genug Möglichkeiten, sie auszunutzen. Vor allem Stimme 3 eignet sich hervorragend zur Erzeugung schöner Effekte.

Der C16 zeichnet sich dadurch aus, daß alle Möglichkeiten des Computers ohne die lästigen »PEEKs« und »POKEs« voll ausgeschöpft werden können. Das Basic 3.5 läßt kaum Wünsche offen und ist für Einsteiger bestens geeignet.

Zur Programmierung schneller Spiele wird man aber auch hier kaum die Maschinensprache umgehen können. Es sei hier auf den Artikel »Maschinensprache auf dem C16« in diesem Sonderheft verwiesen.

(C. Q. Spitzner/tr)