

19 tolle Grafik- Befehle für den VC 20

Mit dieser Basic-Erweiterung kommen auch Sie als VC 20-Besitzer mit einer 16-KByte-Erweiterung in den Genuß, hochauflösende Grafiken einfach zu erstellen.

Für die Befehlserweiterung ist mindestens eine Speichererweiterung von 16 KByte erforderlich. Das normale Commodore-Basic V.2 wird um 19 Befehle und 5 Funktionen erweitert. 13 dieser neuen Befehle dienen zur leichteren Handhabung der Grafikfähigkeiten des VC 20.

Mit GLOAD und GSAVE können die Grafiken dann geladen und gespeichert werden.

Es besteht die Möglichkeit, Grafiken anderer Programme zu laden und auf den Bildschirm zu bringen. Da jedoch oft das Bildschirmformat unterschiedlich ist, muß der Bildschirm der jeweiligen Grafik angepaßt werden. Dies wird mit Hilfe des SCREEN-Befehls erreicht. Für das Format von »JOYPAINT« aus 64'er, Ausgabe 2/85 hätte der Befehl folgendes Aussehen:

```
SCREEN 21,9,64
```

Das heißt:

21 Spalten (8*21 = 168 Bildpunkte)

9 Zeilen (16*9 = 144 Bildpunkte)

64 = 1. Zeichen in Bildschirmcode links oben in der Bildschirmcke.

Ist der Grafikmodus eingeschaltet, kann kein Text eingeblendet werden. Einige der Grafikbefehle sind der Turtle-Grafik für den C 64 aus der Ausgabe 11/84 entliehen. Zur einfacheren Handhabung des Textmodus sind die Befehle CURSOR und SCROLL gedacht.

Die Speicherplatzbelegung sieht wie folgt aus:

\$1000 - \$11FF	Bildschirm
\$1200 - \$1FFF	Grafik
\$2000 - \$2AFF	Basic-Erweiterung
\$2B00 - \$XXXX	Basic-Programmspeicher

Als Sprungvektoren werden benutzt:

\$0300 / \$0301	Vektor für Fehlermeldungen
\$0304 / \$0305	Vektor für Interpretercode
\$0306 / \$0307	Vektor für LIST

\$0308 / \$0309	Vektor für Basic-Befehl
\$030A / \$030B	Vektor für Ausdruck auswerten

Der Speicherbereich \$02A6 bis \$02FF wird vom Programm als Zwischenspeicher genutzt.

Die x-Position des Grafik-Cursors ist in \$02A6 gespeichert, die y-Position in \$02A7.

Eingabe des Programms

Vor der Programmeingabe muß folgende Zeile eingegeben werden

```
1 SYS 8192
```

Dann wird im Direktmodus eingegeben:

```
POKE 44,43:POKE 43*256,0:NEW
```

Hiermit wird der Basic-Anfang auf \$2B00 (dez=43*256) hochgesetzt. Jetzt kann der Basic-Lader eingegeben werden. Lief er ohne Fehlermeldung durch, wird POKE 44,16 eingetippt. Läßt man das Programm jetzt listen, so darf nur 1 SYS 8192 auf dem Bildschirm erscheinen.

Jetzt kann das Programm wie jedes Basic-Programm gespeichert werden.

Alle Zusatzbefehle und Funktionen lassen sich wie die Commodore-Befehle abkürzen.

Befehls Erläuterungen

CURSOR xp, yp, x\$:

xp = x-Koordinate;

yp = y-Koordinate;

x\$ = Ausgabestring, er kann den gleichen Aufbau haben wie der PRINT-Befehl.

Kann nur im Textmodus benutzt werden

SCROLL:

Der Bildschirminhalt wird nach oben gescrollt

SOUND tg, th, la:

tg = Tongenerator 0 bis 3

th = Tonhöhe 128 bis 255

la = Lautstärke 0 bis 15

RENEW:

Nach einem Reset oder NEW kann mit RENEW das Programm wieder in den Speicher geholt werden. Es darf jedoch in der Zwischenzeit keine Variable definiert worden sein.

OFF:

Die Fehlermeldungen werden unterdrückt.

NORM:

Die Fehlermeldungen werden wieder eingeschaltet.

HIRES, mo, hi, ra:

mo = Text- oder Grafikmodus

(Text : mo = 0; Grafik: mo = 1)

hi = Hintergrundfarbe 0 bis 15

ra = Rahmenfarbe 0 bis 7

bei mo = 1 wird die Zeichenfarbe = 1, der Grafik-Cursor an die Position 80,80 gesetzt.

COLOR fa, hi, ra:

- fa =Farbe, in der gezeichnet wird
- hi =Hintergrundfarbe
- ra =Rahmenfarbe

Die Hintergrund- und Rahmenfarbe kann auch weggelassen werden. Das gleiche gilt für den Befehl HIRES. Die Befehle sähen dann wie folgt aus:
 »HIRES mo« oder »COLOR fa«.

SCREEN sp, zl, ze:

- sp = Bildschirmbreite;
 - zl = Bildschirmhöhe;
 - ze = erstes Zeichen auf dem Grafikschild
- Im normalen Grafikmodus ist sp = 20; zl = 10; ze = 32. Das heißt, es gibt 8*20=160 Punkte in x-Richtung und 16*10=160 Punkte in y-Richtung.

Im Textmodus entspricht sp der Spaltenzahl und zl der Zeilenzahl. ze kann weggelassen werden, wenn nur die Bildgröße verändert werden soll. Mit Hilfe dieses Befehls ist es möglich, Grafiken anderer Grafikprogramme auf den Bildschirm zu bringen.

GLOAD x\$, nr:

- x\$ = Name der zu ladenden Grafik;
 - nr = Device-Nummer
(Kassette nr = 1; Diskette nr = 8)
- Es können auch Grafiken anderer Programme geladen werden.

GSAVE x\$, nr:

- x\$ = Name der zu speichernden Grafik
- nr = Gerätenummer

CLEAR:

Grafikbildschirm löschen

MODE mo:

- mo = Zeichenmodus
(Zeichen mo = 0; Löschen mo = 1; Invertieren mo = 2)
- mo > 2 hebt den Grafik-Cursor, es wird weder gezeichnet noch gelöscht.

PLOT xp, yp:

- xp = x-Koordinate;
 - yp = y-Koordinate
- Der Grafik-Cursor wird an die Position xp, yp gesetzt.

DRAW xp, yp:

- xp =x-Koordinate;
 - yp =y-Koordinate
- Es wird eine Linie von der alten x, y-Position zu den Koordinaten xP, yP gezogen.

DEG ri:

- ri = Richtung
- | | |
|---|-----|
| 0 | |
| 7 | 1 |
| 6 | * 2 |
| 5 | 3 |
| 4 | |

RETURN ri:

Die Richtung wird um »ri«-Einheiten nach rechts gedreht.

LTURN ri

Die Richtung wird um »ri«-Einheiten nach links gedreht.

MOVE vo:

Es wird eine Linie mit »vo«-Punkten in die gesetzte Richtung gezogen.

Funktionserläuterungen

JOY:

Ist (JOY AND 23)=16, dann wird der Joystick in keine Richtung bewegt.
 Ist (JOY AND 23) < 16, dann wird der Joystick in die angegebene Richtung gedrückt.
 Bei (JOY AND 8)=8 wird der Feuerknopf betätigt.

MULT\$(x\$, xx)

x\$ = String;
 xx = Wert, mit dem x\$ verlängert wird.
 Beispiel:
 MULT\$(" +. " ,3) = " +.+.+. "

HEX\$(xx):

xx wird in einen 4-stelligen Hexadezimalstring umgewandelt.

BIN\$(xx):

xx wird in einen 8-stelligen Binärstring umgewandelt.

DEC(x\$):

x\$ wird in eine Dezimalzahl umgerechnet.
 Ist LEN(x\$)= 4 oder =2, dann wird vom Hexadezimalsystem ausgegangen.
 Ist LEN(x4)=8, wird vom Binärsystem ausgegangen.

\$XXXX

\$XXXX ist eine 4-stellige Hexadezimalzahl.
 Beispiel:
 ?\$1E00
 7680

Lassen Sie sich also nicht durch die »endlose« DATA-Wüste abschrecken.

Denn hat man diese Einöde bewältigt, so verfügt man über eine nützliche Befehlsenerweiterung.

(Th. Schaper/ah)

```

20 REM THORSTEN SCHAPER <237>
25 REM KURZE KAMPSTR.21 <010>
30 REM 3300 BRAUNSCHWEIG <060>
35 REM <097>
50 PRINT "(CLR)**** TH.BASIC BR *****" <001>
55 PRINT "(DOWN)(C) BY" <020>
60 PRINT "(DOWN,3SPACE)THORSTEN SCHAPER" <102>
65 PRINT "(DOWN,3SPACE)KURZE KAMPSTR.21" <200>
70 PRINT "(DOWN,3SPACE)3300 BRAUNSCHWEIG" <191>
75 PRINT "(2DOWN,2SPACE)BITTE WARTEN" <155>
100 FOR N=0192 TO 10912 STEP 16 <100>
105 CS=0 <065>
110 FOR M=0 TO 15 <005>
115 READ A <155>
120 CS=CS+A <020>
125 POKE N+M,A <075>
130 NEXT M <246>
135 READ CC <243>
140 IF CC<>CS THEN PRINT"DATAERROR IN"1000 <212>
150 NEXT N <013>
155 SYS 0192 <151>
1000 DATA 120,234,234,234, 32,141,253,169, <118>
43,141,130, 2, 32, 82,253, 32,2132
1005 DATA 249,253, 32, 24,229,162, 11,189, <221>
59, 32,157, 0, 3,202, 16,247,1865
1010 DATA 80,234,234,234, 32,164,227,165, <045>
43,164, 44, 32, 8,196,169, 71,2105
1015 DATA 160, 32, 32, 30,203, 32, 18,228, <150>
76,129,227, 50,196,131,196,229,1977
1020 DATA 32,152, 33,226, 33, 17, 41, 42, <036>
42, 42, 42, 32, 84, 72, 46, 32, 968
1025 DATA 66, 65, 83, 73, 67, 32, 71, 82, <052>
32, 42, 42, 42, 42, 13, 29, 29, 010
    
```