

Hyper-Graphics

Das Programm »Hyper-Graphics« erlaubt über SYS-Befehle vom Basic aus Grafiken mit einer Auflösung von $208 * 256 = 53248$ Bildpunkten zu erstellen. Damit stößt der VC 20 auf seine alten Tage noch in Bereiche des ZX-Spectrum oder Apple II vor.

Immer wenn über die VC-20-Grafik geschrieben wurde, bekam man bisher vorgerechnet, daß die Auflösung nicht 28160 Punkte überschreiten kann. Um es gleich vorwegzunehmen: In der Tat ist die echte Maximalauflösung mit $160 * 192 = 30720$, wie sie durch entsprechendes Umgestalten der Data-Becker-Grafik aus dem Buch »Tips & Tricks für den VC 20« gelang, weit von jenen 53248 Punkten entfernt, doch läßt sich diese Auflösung durch einen Trick erreichen. Doch zuerst sei die Frage beantwortet, warum es überhaupt erstrebenswert ist, eine solche Auflösung mit so »krummen« Werten zu entwerfen: Die 256 Punkte in vertikaler Richtung werden benötigt, um komplette 8-Bit-Werte darstellen zu können. Die 208 Punkte in horizontaler Richtung ergeben sich aus dem Umstand, daß nicht mehr auf den normalen Fernseh Bildschirm paßt. Bei manchen älteren Geräten kann es vorkommen, daß die Bildschirmecken der Grafik abgeschnitten werden oder daß am Bildschirmrand Verzerrungen auftreten. Der Vorteil vertikal 256 Punkte darstellen zu können, wird besonders dadurch wertvoll, daß die meisten mit dem Computer erfaßten Meßwerte 8-Bit-Werte zum Beispiel von Analog/Digital-Wandlern sind. So habe ich mir beispielsweise einen solchen A/D-Wandler mit dem Baustein ZN 427 gebaut und damit in Abhängigkeit von der Zeit die Spannung an einem sich entladenden Kondensator gemessen. Da »Hyper-Graphics« vertikal 256 Punkte setzen kann, brauchten die Daten nicht gerundet werden, was einen gleichmäßigeren und genaueren Graphen ergab, als dies mit Computern, wie zum Beispiel dem C64, dem C128 oder dem Schneider-Computer, möglich ist, da diese zwar der Punktezahl nach eine noch wesentlich bessere Grafik bieten, aber in der Vertikalen nur 200 Punkte darstellen können, was einen deutlichen Genauigkeitsverlust bedeutet, besonders wenn man versucht, 256 Werte auf die 200 möglichen Punkte umzurechnen, da hier bedingt durch das erforderliche Runden die gemessenen Daten arg verzerrt und verfremdet wiedergegeben werden.

Das »Hyper-Graphics«-Verfahren

Das Verfahren ist im Prinzip so alt wie der VC20 selbst, denn der normale Bildschirm arbeitet ähnlich: Er besteht aus $22 * 23$ Zeichen, was bei einer $8 * 8$ -Matrix 32384 Punkte ergibt. Dies ist nur durch Mehrfachbenutzung der einzelnen Buchstaben und Zeichen möglich. Da bei den meisten Darstellungen nur einige Graphen gleichzeitig dargestellt werden müssen, kann man in die übrigen Bildschirmstellen ein und dasselbe punktfreie Zeichen schreiben und erhält so eine Grafik, die einer erheblich höher aufgelösten entspricht, obwohl der Computer genaugenommen nur etwas mehr als den halben Bildschirm »mit Punkten füllen« kann. Es kommt eben nur auf die richtige Verteilung der Punkte an!

Anwendung von Hyper-Graphics

Als erstes ist das Programm abzutippen, und zwar wahlweise als Basic-Lader (Listing 1) oder als Maschinenprogramm (Listing 2). Der Basic-Lader bietet den Vorteil, die eingePOKEten Daten durch eine Checksumme zu überprüfen.

Trotzdem kann es aber zu unerkannten Fehlern kommen; deshalb sollte das Programm unbedingt vorher gespeichert werden, zumal der Lader durch Starten des Maschinenprogramms auf jeden Fall zerstört wird. Ist das Programm einmal korrekt eingegeben und erfolgreich gestartet, meldet sich der VC 20 mit einer Bildschirmzeile mehr, die den Benutzer in roter, reverser Schrift daran erinnert, daß er »Hyper-Graphics« eingeschaltet hat. Diese Zeile läßt sich mit dem normalen Editor nicht ändern und bleibt auch nach STOP/RESTORE erhalten. Wem die Farbe aber nicht gefällt, der kann in den Basic-Lader vor den SYS-Befehl ein POKE 9487, Punktfarbe setzen und damit den Schriftzug seinen Vorstellungen anpassen.

Folgende Befehle stellt das Programm zur Verfügung (siehe auch Listing 3):

SYS IN: Schaltet Grafik-Modus ein

SYS CL: Löscht den Grafik-Bildschirm

SYS OF: Schaltet den Grafik-Modus aus

SYS SE, X, Y, Punktfarbe: Setzt einen Punkt

SYS RE, X, Y: Löscht einen Punkt

SYS GS, »Filename«, Devicenumber: Speichert Grafik

SYS GL, »Filename«, Devicenumber: Lädt Grafik

Die Variablen haben folgende Werte beziehungsweise Bedeutungen:

IN = 9330: CL = IN+3: SE = CL+3: RE = SE+3: GS = RE+3: GL = GS+3: OF = GL+3:

$0 \leq X \leq 207$ und bedeutet die X-Koordinate

$0 \leq Y \leq 255$ und bedeutet die Y-Koordinate

Die Punktfarbe entspricht der sonst üblichen Buchstabenfarbe, »Devicenumber« gibt das Ein-/Ausgabegerät an.

Da der Grafik nur maximal $255 - 26 = 229$ Zeichen zur Verfügung stehen (der Videochip läßt nicht mehr als 256 Zeichen zu, die Zeichen 0 bis 25 überschneiden sich mit dem Video-RAM, und Zeichen 26 ist das Grafik-Leerzeichen), kann es vorkommen, daß die Grafikkapazität vor Fertigstellen der Grafik erschöpft ist; der Computer meldet dann einen »TOO MANY CHARACTERS ERROR«, was dem Benutzer signalisieren soll, daß zur Darstellung der Grafik mehr als die vorhandenen 229 Zeichen benötigt werden (wie das Demo-Programm aber beweist, muß man den Bildschirm schon sehr vollschreiben, um diese Fehlermeldung zu erreichen). Soll diese Fehlermeldung ignoriert werden, um wenigstens den bisher fertiggestellten Teil der Grafik weiter anzuzeigen, so läßt sich dies durch POKE 9609,27 erreichen, indem man nach jedem Punktsetzen die Speicherstelle 680 ausliest: Solange ihr Wert größer als 26 ist, können beruhigt weitere Punkte gesetzt werden, ist ihr Wert aber 26, so können nur dort weitere Punkte gesetzt werden, wo bereits Zeichen auf dem Bildschirm sind. Um festzustellen, ob an der betreffenden Stelle bereits ein Zeichen im Video RAM steht, gibt man

SYS 9663,X,Y : SYS 9707 : A = PEEK (256*PEEK(252)+PEEK(251))

ein. Ist A=26, dann kann an die betreffende Stelle kein Punkt mehr gesetzt werden. Die Punktlöschroutine prüft bei jedem Punktlöschen, ob das aktuell bearbeitete Zeichen leer geworden ist und so durch das Leerzeichen ersetzt werden kann. Ist dies der Fall, dann wird das freigewordene Zeichen

Fortsetzung Text auf Seite 62

10 REM *****	<064>
20 REM * HYPER-GRAPHICS *	<200>
30 REM * (C) 1985 BY JOACHIM BROCKE *	<026>
40 REM * DIETRICH-BONHOEFFERSTR.5 *	<087>
50 REM * 4790 PADERBORN *	<104>
60 REM *****	<114>
70 PRINT"(CLR,RVSON)>>> HYPER-GRAPHICS <<<"	<168>
80 PRINT TAB(7)"(C) 1985"SPC(31)"BY JOACHIM BROCKE	<118>
90 PRINT"(DOWN,RVSON,4SPACE)BITTE WARTEN !(4SPACE)"	<080>
100 REM DATEN EINPOKEN	<065>
110 FOR I=0 TO 831	<201>
120 READ X:A=A+X	<201>
130 POKE I+8192+1024,X	<111>
140 NEXT	<150>
150 REM TEST AUF KORREKTE DATENEINGABE	<201>
160 IF A<97379 THEN PRINT"(RVSON)FEHLER IN DATAZEILEN !":END	<142>
170 REM PROGRAMMSTART	<071>
180 SYS 9980	<157>
190 REM MASCHINENPROGRAMM IN DATA-ZEILEN	<102>
200 DATA 190,190,190,160,136,153,144,133,146,173,135,146,129,144,136,137,131	<142>
210 DATA 147,160,188,188,188,63,84,79,79,32,77,65,78,89,32,67,72,65,82,65,67	<180>
220 DATA 84,69,82,83,0,26,52,78,104,130,156,182,208,234,4,30,56,82,108,134,16	<086>
230 DATA 16,16,16,16,16,16,16,16,17,17,17,17,17,17,128,64,32,16,8,4,2,1,9	<130>
240 DATA 19,26,33,255,204,40,67,41,32,49,57,56,53,32,66,89,32,74,79,65,67,72	<172>
250 DATA 73,77,32,66,82,79,67,75,69,76,135,36,76,53,37,76,112,37,76,27,38,76	<068>
260 DATA 129,38,76,162,38,76,220,36,173,5,144,201,204,240,51,160,6,185,81,36	<167>
270 DATA 153,255,143,136,208,247,162,208,189,255,31,157,255,15,189,207,32,157	<080>
280 DATA 207,16,189,159,33,157,255,147,189,111,34,157,207,148,202,208,229,162	<239>
290 DATA 112,189,63,35,157,159,17,202,208,247,96,40,3,41,32,49,57,56,53,32,2	<210>
300 DATA 25,32,10,15,1,3,8,9,13,32,2,18,15,3,11,5,173,5,144,201,192,240,43,162	<089>
310 DATA 208,189,255,15,157,255,31,189,207,16,157,207,32,189,255,147,157,159	<167>
320 DATA 33,189,207,148,157,111,34,202,208,229,162,112,189,159,17,157,63,35,202	<196>
330 DATA 208,247,32,95,229,160,2,162,22,189,255,35,157,249,17,152,157,249,149	<017>
340 DATA 202,208,243,160,6,185,227,237,153,255,143,136,208,247,169,34,141,1,144	<113>
350 DATA 169,48,141,3,144,96,169,26,162,208,157,255,15,157,207,16,157,255,31	<233>
360 DATA 157,207,32,202,208,241,162,96,169,0,157,159,17,157,159,33,202,208,247	<114>
370 DATA 169,18,133,252,169,0,133,251,162,14,168,145,251,136,208,251,230,252	<241>
380 DATA 202,208,246,169,255,141,168,2,96,32,191,37,32,253,206,32,158,215,142	<095>
390 DATA 173,2,32,235,37,201,26,208,12,173,168,2,201,27,144,32,206,168,2,145	<205>
400 DATA 251,32,254,37,17,253,145,253,165,252,24,105,132,133,252,172,169,2,173	<181>
410 DATA 173,2,145,251,96,169,0,133,62,32,220,36,169,22,160,36,32,30,203,76,98	<030>
420 DATA 196,32,220,36,162,14,76,55,196,32,253,206,32,158,215,224,208,176,238	<249>
430 DATA 138,74,74,74,141,169,2,138,41,7,141,170,2,32,253,206,32,158,215,138	<252>
440 DATA 74,74,74,74,141,171,2,138,41,15,141,172,2,96,172,171,2,185,42,36,133	<112>
450 DATA 251,185,58,36,133,252,172,169,2,177,251,96,168,74,74,74,74,24,105,16	<053>
460 DATA 133,254,152,10,10,10,10,24,133,253,173,170,2,170,189,74,36,172,172,2	<135>
470 DATA 96,32,191,37,32,235,37,201,26,240,94,141,174,2,32,254,37,73,255,49,253	<056>
480 DATA 145,253,160,0,177,253,208,73,200,192,16,208,247,172,169,2,169,26,145	<013>
490 DATA 251,238,168,2,162,208,189,207,16,205,168,2,240,17,189,255,15,240,3,202	<086>
500 DATA 208,240,173,174,2,157,255,15,76,103,38,173,174,2,157,207,16,165,253	<223>
510 DATA 133,251,165,254,133,252,173,168,2,32,254,37,160,0,177,253,145,251,200	<040>
520 DATA 192,16,208,247,96,32,220,36,32,253,206,32,209,225,173,168,2,141,255	<201>
530 DATA 35,162,0,160,36,138,133,253,169,18,133,254,169,253,133,185,76,216,255	<233>
540 DATA 32,220,36,32,253,206,32,209,225,169,1,133,185,169,0,32,213,255,173,255	<148>
550 DATA 35,141,168,2,96,72,138,72,152,72,173,29,145,16,37,45,30,145,170,41	<181>
560 DATA 2,240,32,44,17,145,32,52,247,32,225,255,208,18,32,82,253,32,240,38,32	<253>
570 DATA 249,253,32,24,229,32,220,36,108,2,192,76,86,255,76,222,254,169,187,141	<175>
580 DATA 24,3,169,38,141,25,3,96,120,32,240,38,173,0,3,141,58,39,173,1,3,141	<095>
590 DATA 59,39,169,45,141,0,3,169,39,141,1,3,32,220,36,169,39,133,44,141,130	<074>
600 DATA 2,169,64,133,43,141,129,2,32,68,198,88,76,116,196,142,57,39,32,220,36	<030>
610 DATA 174,57,39,108,58,39,28,58,196,0,0,0,0	<066>

Listing 1. DATA-Lader
zu »Hyper-Graphics«

Fortsetzung von Seite 60

gegen das »größte« nicht freie Zeichen ausgetauscht und dieses Zeichen wieder freigegeben, indem der Zähler für die freien Zeichen erhöht wird.

Zum Demo-Programm (Listing 4):
Es enthält folgende 4 Teile:

1. Eine 3D-Effekt-Grafik; die Formel hierzu ist dem oben erwähnten Data-Becker-Buch entnommen und wurde für »Hyper-Graphics« umgeschrieben.
2. Die für Grafikprogramme obligatorische Sinuskurve wird dargestellt.
3. Eine aus Rechtecken bestehende Zentralperspektive wird dargestellt.
4. Es wird mit hochauflösender Grafik in doppelthoher Schrift ein Text geschrieben und dann perspektivisch ein Oktaeder in Durchsichtszeichnung gezeichnet (dieser Programmteil ist Bestandteil eines Programmes, das auf einer Ausstellung unseres Chemie-Leistungskurses lief, und stieß im großen und ganzen auf positive Kritik).

Die einzelnen Programnteile werden durch Druck der angezeigten Zahlentaste gewählt, danach sind Rahmen-, Hintergrund- und Punktfarbe im bekannten Rahmen der VC-20-Möglichkeiten einzugeben, wobei auf korrekte Eingabe getestet wird. Da das Demo-Programm nur zum Ansehen dient, wurde, um die Abtipparbeit zu erleichtern, weitge-

hend auf REMs verzichtet, lediglich die Zeilen 1350 bis 1450 sind interessant, da hier die Routine zur Umwandlung von ASCII-Zeichen in Punkte zur gleichzeitigen Darstellung von Text und Grafik steht. Diese Routine ist aber leicht verständlich.

Um normal hohe Zeichen zu erhalten, streicht man einfach die Zeile 1450 und ersetzt in Zeile 1440 den Term $2 \cdot J + 16$ durch $J + 16$. Die 16 gibt die Höhe des Textes auf dem Bildschirm an, soll in Kleinschrift geschrieben werden, so muß zu AD 2048 addiert werden.

Sie sehen also, 53248 Bildpunkte sind nicht nur den »größeren« Computern vorbehalten.

Eingabehinweise:

Geben Sie zuerst das Programm »Hyper-Graphics« ein. Dazu steht Ihnen wahlweise Listing 1 oder Listing 2 zur Verfügung. Beachten Sie beim Basic-Lader (Listing 1) bitte die Eingabehinweise auf Seite 76. Nach dem Speichern starten Sie das »Hyper-Graphics« mit RUN. Für Interessierte haben wir in Listing 3 den (nicht einzugebenden) Quellcode abgedruckt. Das Demo-Programm aus Listing 4 laden und starten Sie bitte nur mit aktivierter Erweiterung.

(Joachim Brocke/og)

```

..2400 BE BE BE A0 88      ..24D7 12 0F 03 0B 05      ..25AE 16 A0 24 20 1E      ..2685 FD CE 20 D1 E1
..2405 99 90 85 92 AD      ..24DC AD 05 90 C9 C0      ..25B3 CB 4C 62 C4 20      ..268A AD A8 02 8D FF
..240A 87 92 81 90 88      ..24E1 F0 2B A2 D0 BD      ..25B8 DC 24 A2 0E 4C      ..268F 23 A2 00 A0 24
..240F 89 83 93 A0 BC      ..24E6 FF 0F 9D FF 1F      ..25BD 37 C4 20 FD CE      ..2694 8A 85 FD A9 12
..2414 BC BC 3F 54 4F      ..24EB BD CF 10 9D CF      ..25C2 20 9E D7 E0 D0      ..2699 85 FE A9 FD 85
..2419 4F 20 4D 41 4E      ..24F0 20 BD FF 93 9D      ..25C7 B0 EE 8A 4A 4A      ..269E B9 4C D8 FF 20
..241E 59 20 43 48 41      ..24F5 9F 21 BD CF 94      ..25CC 4A 8D A9 02 8A      ..26A3 DC 24 20 FD CE
..2423 52 41 43 54 45      ..24FA 9D 6F 22 CA D0      ..25D1 29 07 8D AA 02      ..26A8 20 D1 E1 A9 01
..2426 52 53 00 1A 34      ..24FF E5 A2 70 BD 9F      ..25D6 20 FD CE 20 9E      ..26AD 85 B9 A9 00 20
..242D 4E 68 82 9C B6      ..2504 11 9D 3F 23 CA      ..25DB D7 8A 4A 4A 4A      ..26B2 D5 FF AD FF 23
..2432 D0 EA 04 1E 38      ..2509 D0 F7 20 5F E5      ..25E0 4A 8D AB 02 8A      ..26B7 8D A8 02 60 48
..2437 52 6C 86 10 10      ..250E A0 02 A2 16 BD      ..25E5 29 0F 8D AC 02      ..26BC 8A 48 98 48 AD
..243C 10 10 10 10 10      ..2513 FF 23 9D F9 11      ..25EA 60 AC AB 02 B9      ..26C1 1D 91 10 25 2D
..2441 10 10 10 11 11      ..2516 98 9D F9 95 CA      ..25EF 2A 24 85 FB B9      ..26C6 1E 91 AA 29 02
..2446 11 11 11 11 80      ..251D D0 F3 A0 06 B9      ..25F4 3A 24 85 FC AC      ..26CB F0 20 2C 11 91
..244B 40 20 10 08 04      ..2522 E3 ED 99 FF 8F      ..25F9 A9 02 B1 FB 60      ..26D0 20 34 F7 20 E1
..2450 02 01 09 13 1A      ..2527 88 D0 F7 A9 22      ..25FE A8 4A 4A 4A 4A      ..26D5 FF D0 12 20 52
..2455 21 FF CC 28 43      ..252C 8D 01 90 A9 30      ..2603 18 69 10 85 FE      ..26DA FD 20 F0 26 20
..245A 29 20 31 39 38      ..2531 8D 03 90 60 A9      ..2608 98 0A 0A 0A 0A      ..26DF F9 FD 20 18 E5
..245F 35 20 42 59 20      ..2536 1A A2 D0 9D FF      ..260D 18 85 FD AD AA      ..26E4 20 DC 24 6C 02
..2464 4A 4F 41 43 48      ..253B 0F 9D CF 10 9D      ..2612 02 AA BD 4A 24      ..26E9 C0 4C 56 FF 4C
..2469 49 4D 20 42 52      ..2540 FF 1F 9D CF 20      ..2617 AC AC 02 60 20      ..26EE DE FE A9 BB 8D
..246E 4F 43 4B 45 4C      ..2545 CA D0 F1 A2 60      ..261C BF 25 20 EB 25      ..26F3 18 03 A9 26 8D
..2473 87 24 4C 35 25      ..254A A9 00 9D 9F 11      ..2621 C9 1A F0 5E 8D      ..26F8 19 03 60 78 20
..2478 4C 70 25 4C 1B      ..254F 9D 9F 21 CA D0      ..2626 AE 02 20 FE 25      ..26FD F0 26 AD 00 03
..247D 26 4C 81 26 4C      ..2554 F7 A9 12 85 FC      ..262B 49 FF 31 FD 91      ..2702 8D 3A 27 AD 01
..2482 A2 26 4C DC 24      ..2559 A9 00 85 FB A2      ..2630 FD A0 00 B1 FD      ..2707 03 8D 3B 27 A9
..2487 AD 05 90 C9 CC      ..255E 0E A8 91 FB 88      ..2635 D0 49 C8 C0 10      ..270C 2D 8D 00 03 A9
..248C F0 33 A0 06 B9      ..2563 D0 FB E6 FC CA      ..263A D0 F7 AC A9 02      ..2711 27 8D 01 03 20
..2491 51 24 99 FF 8F      ..2568 D0 F6 A9 FF 8D      ..263F A9 1A 91 FB EE      ..2716 DC 24 A9 27 85
..2496 88 D0 F7 A2 D0      ..256D A8 02 60 20 BF      ..2644 A8 02 A2 D0 BD      ..271B 2C 8D 82 02 A9
..249B BD FF 1F 9D FF      ..2572 25 20 FD CE 20      ..2649 CF 10 CD A8 02      ..2720 40 85 2B 8D 81
..24A0 0F BD CF 20 9D      ..2577 9E D7 8E AD 02      ..264E F0 11 BD FF 0F      ..2725 02 20 44 C6 58
..24A5 CF 10 BD 9F 21      ..257C 20 EB 25 C9 1A      ..2653 F0 03 CA D0 F0      ..272A 0C 74 C4 8E 39
..24AA 9D FF 93 BD 6F      ..2581 D0 0C AD A8 02      ..2658 AD AE 02 9D FF      ..272F 27 20 DC 24 AE
..24AF 22 9D CF 94 CA      ..2586 C9 1B 90 20 CE      ..265D 0F 4C 67 26 AD      ..2734 39 27 6C 3A 27
..24B4 D0 E5 A2 70 BD      ..258B A8 02 91 FB 20      ..2662 AE 02 9D CF 10
..24B9 3F 23 9D 9F 11      ..2590 FE 25 11 FD 91      ..2667 A5 FD 85 FB A5
..24BE CA D0 F7 60 28      ..2595 FD A5 FC 18 69      ..266C FE 85 FC AD A8
..24C3 03 29 20 31 39      ..259A 84 85 FC AC A9      ..2671 02 20 FE 25 A0
..24C6 38 35 20 02 19      ..259F 02 AD AD 02 91      ..2676 00 B1 FD 91 FB
..24C9 20 0A 0F 01 03      ..25A4 FB 60 A9 00 85      ..267B C8 C0 10 D0 F7
..24D2 08 09 0D 20 02      ..25A9 3E 20 DC 24 A9      ..2680 60 20 DC 24 20

```

Listing 2. Hexdump
zu »Hyper-Graphics«

SPRUNGTABELLE:

.2472 JMP \$2487	ZU 'GRAPHIK EINSCHALTEN'
.2475 JMP \$2535	ZU 'GRAPHIKSPEICHER LOESCHEN'
.2478 JMP \$2570	ZU 'PUNKT SETZEN'
.247B JMP \$261B	ZU 'PUNKT LOESCHEN'
.247E JMP \$2681	ZU 'GRAPHIK ABSPEICHERN'
.2481 JMP \$26A3	ZU 'GRAPHIK LADEN'
.2484 JMP \$24DC	ZU 'GRAPHIK AUSSCHALTEN'

GRAPHIK EINSCHALTEN:

.2487 LDA \$9005	HOLT ZEIGER AUF CHARACTERGENERATOR
.248A CMP #\$CC	PRUEFT AUF 'GRAPHIK ON'-MODUS
.248C BEQ \$24C1	SCHON IM GRAPHIKMODUS? DANN ZURUECK INS BASIC
.248E LDY #\$06	ZAEHLER FUER ANZAHL DER VIDEOCONTROLLERWERTE
.2490 LDA \$2451,Y	HOLT VIDEOCONTROLLERWERTE AUS TABELLE
.2493 STA \$8FFF,Y	LAEDT VIDEOCONTROLLER
.2496 DEY	ZAEHLER ERNIEDRIGEN
.2497 BNE \$2490	UNGLEICH NULL? DANN VORGANG WIEDERHOLEN
.2499 LDX #\$D0	ZAEHLER FUER GRAPHIK-KOPIE LADEN
.249B LDA \$1FFF,X	BYTE AUS VIDEORAMKOPIE, 1.TEIL, HOLEN
.249E STA \$0FFF,X	...UND INS VIDEORAM BRINGEN
.24A1 LDA \$20CF,X	BYTE AUS VIDEORAMKOPIE, 2.TEIL, HOLEN
.24A4 STA \$10CF,X	...UND INS VIDEORAM BRINGEN
.24A7 LDA \$219F,X	BYTE AUS FARBRAMKOPIE, 1.TEIL, HOLEN
.24AA STA \$93FF,X	...UND INS FARBRAM BRINGEN
.24AD LDA \$226F,X	BYTE AUS FARBRAMKOPIE, 2.TEIL, HOLEN
.24B0 STA \$94CF,X	...UND INS FARBRAM BRINGEN
.24B3 DEX	ZAEHLER ERNIEDRIGEN
.24B4 BNE \$249B	UNGLEICH NULL? DANN NAECHSTES ZEICHEN KOPIEREN
.24B6 LDX #\$70	ZAEHLER FUER KOPIE DER ZEICHEN 26-32 DES CHARACTERGENERATORS
.24B8 LDA \$233F,X	BYTE AUS KOPIE DIESES CHARACTERGENERATORTEILS HOLEN
.24BB STA \$119F,X	...UND IN CHARACTERGENERATOR EINSETZEN
.24BE DEX	ZAEHLER ERNIEDRIGEN
.24BF BNE \$24B8	UNGLEICH NULL? DANN NAECHSTES ZEICHEN KOPIEREN
.24C1 RTS	ZURUECK INS BASIC

GRAPHIK LOESCHEN:

.2535 LDA #\$1A	AKKU MIT CODE FUER GRAPHIKLEERZEICHEN LADEN
.2537 LDX #\$D0	ZAEHLER FUER VIDEORAM LADEN
.2539 STA \$0FFF,X	1.TEIL DES VIDEORAMS LOESCHEN
.253C STA \$10CF,X	2.TEIL DES VIDEORAMS LOESCHEN
.253F STA \$1FFF,X	1.TEIL DER VIDEORAMKOPIE LOESCHEN
.2542 STA \$20CF,X	2.TEIL DER VIDEORAMKOPIE LOESCHEN
.2545 DEX	ZAEHLER ERNIEDRIGEN
.2546 BNE \$2539	UNGLEICH NULL? DANN VORGANG WIEDERHOLEN
.2548 LDX #\$60	ZAEHLER FUER ZEICHEN 26-32 DES CHARACTERGENERATORS
.254A LDA #\$00	AKKU MIT PUNKTFREIEM BYTE LADEN
.254C STA \$119F,X	BYTE DER ZEICHEN 26-32 DES CHARACTERGENERATORS LOESCHEN
.254F STA \$219F,X	BYTE IN DER KOPIE DIESES TEILS LOESCHEN
.2552 DEX	ZAEHLER ERNIEDRIGEN
.2553 BNE \$254C	UNGLEICH NULL? DANN VORGANG WIEDERHOLEN
.2555 LDA #\$12	AKKU MIT HIGH-BYTE DES CHARACTERGENERATORSTARTS LADEN
.2557 STA \$FC	AKKU ALS VEKTOR-HIGHBYTE IN ZEROPAGE BRINGEN
.2559 LDA #\$00	AKKU MIT LOW-BYTE DES CHARACTERGENERATORS LADEN
.255B STA \$FB	...UND ALS VEKTOR-LOWBYTE IN ZEROPAGE BRINGEN
.255D LDX #\$0E	X-REG. MIT ANZAHL DER PAGES DES CHARACTERGENERATORS LADEN
.255F TAY	Y-REG. ALS ZAEHLER DER BYTES PRO PAGE
.2560 STA (\$FB),Y	CHARACTERGENERATORBYTE LOESCHEN
.2562 DEY	ZAEHLER FUER BYTES INNERHALB EINER PAGE ERNIEDRIGEN
.2563 BNE \$2560	UNGLEICH NULL? DANN NAECHSTES ZEICHEN LOESCHEN

Listing 3. Quell-Code-Listing zu »Hyper-Graphics«

```
.2565 INC $FC      VEKTOR AUF NAECHSTE MEMORYPAGE DES CHARACTERGENERATOR SETZEN
.2567 DEX          ZAEHLER FUER ANZAHL DER ZU BEARBEITENDEN PAGES ERNIEDRIGEN
.2568 BNE $2560    UNGLEICH NULL? DANN NAECHSTE PAGE LOESCHEN
.256A LDA #$FF     CODE FUER OBERSTES FREIES ELEMENT DES CHARACTERGENERATORS
.256C STA $02A8    ...AN FREIEN PLATZ IN PAGE 2 BRINGEN
.256F RTS          ZURUECK INS BASIC
```

PUNKT SETZEN:

```
.2570 JSR $25BF    HOLT X- UND Y-KOORDINATE AUS BASICTEXT UND WANDELT UM
.2573 JSR $CEFD    TESTET AUF KOMMA IM BASICTEXT
.2576 JSR $D79E    HOLT BYTEWERT (HIER FARBCODE) AUS BASICTEXT INS X-REGISTER
.2579 STX $02AD    FARBCODE ZWISCHENSPEICHERN
.257C JSR $25EB    HOLT AKTUELL ZU BEARBEITENDES ZEICHEN AUS VIDEORAM
.257F CMP #$1A     PRUEFT AUF LEERZEICHEN
.2581 BNE $258F    KEIN LEERZEICHEN? DANN KEIN NEUES ZEICHEN SETZEN
.2583 LDA $02A8    CODE FUER OBERSTES FREIES ZEICHEN HOLEN
.2586 CMP #$1B     VERGLEICHT MIT CODE FUER LEERZEICHEN +1
.2588 BCC $25AA    KLEINER? DANN FEHLERMELDUNG 'TOO MANY CHARACTERS' AUSGEBEN
.258A DEC $02A8    ZEIGER AUF OBERSTES FREIES ELEMENT DES CHARACTERGEN. ERNIEDR.
.258D STA ($FB),Y  NEUES ZEICHEN INS VIDEORAM BRINGEN
.258F JSR $25FE    BERECHNET VEKTOR AUF CHARACTERGENERATOR & HOLT AKTUELLES BYTE
.2592 ORA ($FD),Y  SETZT PUNKT IM AKTUELLEN BYTE
.2594 STA ($FD),Y  SCHREIBT BEARBEITETES BYTE IN CHARACTERGENERATOR
.2596 LDA $FC      HOLT HIGH-BYTE DES VIDEORAMVEKTORS
.2598 CLC          ADDITION VORBEREITEN
.2599 ADC #$84     WANDELT VEKTOR AUF VIDEORAM IN VEKTOR AUF FARBRAM UM
.259B STA $FC      SETZT HIGH-BYTE DES VEKTORS AUF DAS FARBRAM
.259D LDY $02A9    HOLT BILDSCHIRMSPALTE
.25A0 LDA $02AD    HOLT FARBCODE
.25A3 STA ($FB),Y  BRINGT FARBCODE INS FARBRAM
.25A5 RTS          ZURUECK INS BASIC
```

FEHLERMELDUNG 'TOO MANY CHARACTERS' AUSGEBEN:

```
.25A6 LDA #$00     FLAG FUER 'CONT GESPERRT'
.25A8 STA $3E      CONT SPERREN
.25AA JSR $24DC    GRAPHIK AUSSCHALTEN
.25AD LDA #$16     ZEIGER AUF ASCII-TEXT (LOW-BYTE) DES FEHLERTEXTES
.25AF LDY #$24     ZEIGER AUF ASCII-TEXT (HIGH-BYTE) DES FEHLERTEXTES
.25B1 JSR $CB1E    ASCII-TEXT AUSGEBEN
.25B4 JMP $C462    'ERROR' AUSGEBEN + READY-MELDUNG
```

FEHLERMELDUNG 'ILLEGAL QUANTITY ERROR' AUSGEBEN:

```
.25B7 JSR $24DC    GRAPHIK AUSSCHALTEN
.25BA LDX #$0E     CODENUMMER FUER 'ILLEGAL QUANTITY' HOLEN
.25BC JMP $C437    FEHLERMELDUNG AUSGEBEN
```

X- & Y-KOORDINATEN AUS BASICTEXT HOLEN:

```
.25BF JSR $CEFD    TESTET AUF KOMMA IM BASICTEXT
.25C2 JSR $D79E    HOLT BYTE-WERT (HIER X-KOORDINATE) AUS BASICTEXT
.25C5 CPX #$D0     VERGLEICHT MIT MAXIMALWERT +1
.25C7 BCS $25B7    GROESSER/GLEICH? DANN ZU 'ILLEGAL QUANTITY ERROR' AUSGEBEN
.25C9 TXA
.25CA LSR          7
.25CB LSR          1- X-KOORDINATE/8 BERECHNEN
.25CC LSR          1
.25CD STA $02A9    ...UND ZWISCHENSPEICHERN
.25D0 TXA
.25D1 AND #$07     REST VON (X-KOORDINATE/8) BERECHNEN
.25D3 STA $02AA    ...UND ZWISCHENSPEICHERN
.25D6 JSR $CEFD    TESTET AUF KOMMA IM BASICTEXT
.25D9 JSR $D79E    HOLT BYTE-WERT (HIER Y-KOORDINATE) AUS BASICTEXT INS X-REG.
```

Listing 3. Quell-Code-Listing zu »Hyper-Graphics« (Fortsetzung)

```
.25DC TXA
.25DD LSR      7
.25DE LSR      7- Y-KOORDINATE/16 BERECHNEN
.25DF LSR      1
.25E0 LSR      7
.25E1 STA $02AB ... UND ZWISCHENSPEICHERN
.25E4 TXA
.25E5 AND #$0F  REST VON Y-KOORDINATE/16 BERECHNEN
.25E7 STA $02AC ...UND ZWISCHENSPEICHERN
.25EA RTS      ZURUECK ZUR PUNKTSETZ- BZW. -LOESCHROUTINE
```

AKTUELL ZU BEARBEITENDES ZEICHEN AUS VIDEORAM HOLEN:

```
.25EB LDY $02AB BILDSCHIRMZEILE HOLEN
.25EE LDA $242A,Y VIDEORAMVEKTOR (LOW-BYTE) AUS TABELLE HOLEN
.25F1 STA $FB    ...UND IN ZEROPAGE BRINGEN
.25F3 LDA $243A,Y VIDEORAMVEKTOR (HIGH-BYTE) AUS TABELLE HOLEN
.25F6 STA $FC    ...UND IN ZEROPAGE BRINGEN
.25F8 LDY $02A9  HOLT BILDSCHIRMSPALTE
.25FB LDA ($FB),Y HOLT AKTUELLES ZEICHEN AUS VIDEORAM
.25FD RTS      ZURUECK ZUR HAUPTROUTINE
```

VEKTOR AUF CHARACTERGENERATOR UND PUNKT BERECHNEN:

```
.25FE TAY      AKTUELLES ZEICHEN ZWISCHENSPEICHERN
.25FF LSR      7
.2600 LSR      7-HIGH-BYTE DES VEKTORS AUF DEN CHARACTERGENERATOR BERECHNEN
.2601 LSR      1
.2602 LSR      7
.2603 CLC      ADDITION VORBEREITEN
.2604 ADC #$10  OFFSET FUER ANFANG DES CHARACTERGENERATORS ADDIEREN
.2606 STA $FE   HIGH-BYTE DES VEKTORS AUF DEN CHARACTERGENERATORS SETZEN
.2608 TYA      AKTUELLES ZEICHEN HOLEN
.2609 ASL      7
.260A ASL      7-LOW-BYTE DES VEKTORS AUF DEN CHARACTERGENERATOR BERECHNEN
.260B ASL      1
.260C ASL      7
.260D CLC
.260E STA $FB   LOW-BYTE DES VEKTORS AUF DEN CHARACTERGENERATOR SETZEN
.2610 LDA $02AA HOLT PUNKTNUMMER
.2613 TAX      IN INDEX WANDELN
.2614 LDA $244A,X HOLT BIT=PUNKT AUS TABELLE
.2617 LDY $02AC ZEILE DES AKTUELLEN ZEICHENS HOLEN
.261A RTS      ZURUECK ZUR HAUPTROUTINE
```

PUNKT LOESCHEN:

```
.261B JSR $25BF HOLT X- UND Y-KOORDINATE AUS BASICTEXT
.261E JSR $25EB HOLT AKTUELLES ZEICHEN AUS VIDEORAM
.2621 CMP #$1A  VERGLEICHT MIT LEERZEICHEN
.2623 BEQ $2680 GLEICH? DANN FERTIG!
.2625 STA $02AE AKTUELLES ZEICHEN RETTEN
.2628 JSR $25FE BERECHNET VEKTOR AUF CHARACTERGENERATOR UND HOLT PUNKT
.262B EOR #$FF  BYTE INVERTIEREN
.262D AND ($FD),Y BIT=PUNKT LOESCHEN
.262F STA ($FD),Y BYTE IN CHARACTERGENERATOR EINSETZEN
.2631 LDY #$00  INDEX AUF ANFANG DES AKTUELLEN CHARACTERS
.2633 LDA ($FD),Y EIN BYTE DES AKTUELLEN ZEICHENS HOLEN
.2635 BNE $2680 UNGLEICH NULL (NICHT ALLE PUNKTE GELOESCHT)? DANN FERTIG
.2637 INY      INDEX ERHOEHEN
.2638 CPY #$10  ZEICHEN GANZ DURCHSUCHT?
.263A BNE $2633 NEIN, DANN NAECHSTES BYTE TESTEN
.263C LDY $02A9 BILDSCHIRMSPALTE HOLEN
.263F LDA #$1A  CODE FUER LEERZEICHEN
```

Listing 3. Quell-Code-Listing zu »Hyper-Graphics« (Fortsetzung)

```
.2641 STA ($FB),Y    LEERZEICHEN IN VIDEORAM BRINGEN
.2643 INC $02A8      EIN ZEICHEN FREIGEBEN
.2646 LDX #$D0       ZAEHLER FUER BYTES PRO VIDEORAMHAELFTEN INITIALISIEREN
.2648 LDA $02AE       ZEICHEN AUS ZWISCHENSPEICHER HOLEN
.264B CMP $10CF,X    GLEICH ZEICHEN IN VIDEORAM (2.TEIL) ?
.264E BEQ $2661      JA,DANN ZU 'ZEICHEN VERTAUSCHEN' FUER 2.VIDEORAMTEIL
.2650 CMP $0FFF,X    GLEICH ZEICHEN IN VIDEORAM (1.TEIL) ?
.2653 BEQ $2658      GLEICH, DANN ZU 'ZEICHEN VERTAUSCHEN' FUER 1.VIDEORAMTEIL
.2655 DEX            ZAEHLER ERNIEDRIGEN
.2656 BNE $2648      UNGLEICH NULL? DANN VORGANG WIEDERHOLEN
.2658 LDA $02AE       ZEICHEN HOLEN
.265B STA $0FFF,X    ...UND INS VIDEORAM BRINGEN
.265E JMP $2667      ZU 'ZEICHEN KOPIEREN'
.2661 LDA $02AE       ZEICHEN HOLEN
.2664 STA $10CF,X    ...UND INS VIDEORAM BRINGEN
.2667 LDA $FD        7
.2669 STA $FB        7- VEKTOR AUF CHARACTERGENERATOR KOPIEREN
.266B LDA $FE        1
.266D STA $FC        1
.266F LDA $02A8      FREIGEgebenES ZEICHEN HOLEN
.2672 JSR $25FE      VEKTOR AUF CHARACTERGENERATOR ZU DIESEM ZEICHEN BERECHNEN
.2675 LDY #0         INDEX AUF ANFANG DER ZEICHEN
.2677 LDA ($FD),Y    BYTE DES ALTEN CHARACTERS HOLEN
.2679 STA ($FB),Y    ...UND IN FREIGEWORDENES ZEICHEN EINSETZEN
.267B INY            ZAEHLER ERHOEHEN
.267C CMP #$10       MIT 16 VERGLEICHEN
.267E BNE $2677      UNGLEICH? DANN NAECHSTES BYTE KOPIEREN
.2680 RTS            ZURUECK INS BASIC
```

GRAPHIK ABSPEICHERN:

```
.2681 JSR $24DC      GRAPHIK AUSSCHALTEN
.2684 JSR $CEFD      PRUEFT AUF KOMMA IM BASICTEXT
.2687 JSR $E1D1      HOLT FILENAME UND DEVICENUMBER
.268A LDA $02A8      ZEIGER AUF OBERSTES FREIES ZEICHEN HOLEN
.268D STA $23FF      ...UND IN ABZUSPEICHERNDEN RAM-BEREICH BRINGEN
.2690 LDX #$00       LOW-BYTE DER ENDADRESSE DES GRAPHIKSPEICHERS
.2692 LDY #$24       HIGH-BYTE DER ENDADRESSE DES GRAPHIKSPEICHERS
.2694 TXA            LOW-BYTE DER ANFANGSADRESSE DES GRAPHIKSPEICHERS
.2695 STA $FD        ALS VEKTOR-LOW-BYTE IN ZEROPAGE BRINGEN
.2697 LDA #$12       HIGH-BYTE DER STARTADRESSE DES GRAPHIKSPEICHERS HOLEN
.2699 STA $FE        ...UND ALS VEKTOR-HIGH-BYTE IN ZEROPAGE BRINGEN
.269B LDA #$FD       ADRESSE DES VEKTORS AUF START DES GRAPHIKSPEICHERS
.269D STA $B9        ...FUER SAVE-ROUTINE ZUGAENGLICH MACHEN
.269F JMP $FFD8      ZUR 'SAVE'-ROUTINE DES BETRIEBSSYSTEMS
```

GRAPHIK EINLADEN:

```
.26A2 JSR $24DC      GRAPHIK AUSSCHALTEN
.26A5 JSR $CEFD      PRUEFT AUF KOMMA IM BASICTEXT
.26A8 JSR $E1D1      HOLT FILENAME UND DEVICENUMBER
.26AB LDA #$01       KONSTANTE FUER 'ABSOLUT LADEN'
.26AD STA $B9        ... FUER LOAD-ROUTINE SETZEN
.26AF LDA #$00       FLAG FUER LOAD
.26B1 JSR $FFD5      ZUR LOAD-ROUTINE DES BETRIEBSSYSTEMS
.26B4 LDA $23FF      ZEIGER AUF OBERSTES FREIES ZEICHEN HOLEN
.26B7 STA $02A8      ... UND AN DIE VORGESEHENE STELLE BRINGEN
.26BA RTS            ZURUECK INS BASIC
```

NMI-ROUTINE:

```
.26BB PHA            AKKU RETTEN
.26BC TXA            X-REGISTER IN AKKU UEBERTRAGEN
.26BD PHA            X-REGISTER RETTEN
```

Listing 3. Quell-Code-Listing zu »Hyper-Graphics«

```

.26BE TYA      Y-REGISTER IN AKKU UEBERTRAGEN
.26BF FHA      Y-REGISTER RETTEN
.26C0 LDA $911D AKKU MIT IFR DES VIA1 LADEN
.26C3 BPL $26EA IRQ-BIT GESETZT? DANN ZURUECK
.26C5 AND $911E NICHT FREIGELEGENE INTERRUPTS AUSSIEBEN
.26C8 TAX      ERGEBNIS RETTEN
.26C9 AND #$02 SCHIEBEREGISTERFLAG ISOLIEREN
.26CB BEQ $26ED ZUR NMI-ROUTINE FUER SERIELLE SCHNITTSTELLE
.26CD BIT $9111
.26D0 JSR $F734 ERHOEHT UHRZEIT UND FRAGT STOP-TASTE AB
.26D3 JSR $FFE1 FRAGT AUCH DIE STOP-TASTE AB (IM BETRIEBSSYSTEM SO VORGESEHEN)
.26D6 BNE $26EA NICHT GDRUECKT? DANN ZURUECK
.26D8 JSR $FD52 BETRIEBSSYSTEMVEKTOREN NEU SETZEN
.26DB JSR $26F0 NMI-VEKTOR NEU SETZEN
.26DE JSR $FD9F INITIALISIERT I/O-REGISTER
.26E1 JSR $E518 INITIALISIERT VIDEOCONTROLLER UND LOESCHT BILDSCHIRM
.26E4 JSR $24DC SCHALTET AUF HYPERGRAPHIC-TEXTMODUS (= GRAPHIK AUSSCHALTEN)
.26E7 JMP ($C002) ZUM BASIC-KALTSTART
.26EA JMP $FF56 RUECKKEHR VOM INTERRUPT
.26ED JMP $FEDE ZUR NMI-ROUTINE FUER SERIELLE SCHNITTSTELLE
.26F0 LDA #$BB LOW-BYTE DES HYPERGRAPHIC-NMI-VEKTORS HOLEN
.26F2 STA $0318 ...UND EINSETZEN
.26F5 LDA #$26 HIGH-BYTE DES HYPERGRAPHIC-NMI-VEKTORS HOLEN
.26F7 STA $0319 ...UND SETZEN
.26FA RTS      RUECKSPRUNG

```

HYPERGRAPHICS START:

```

.26FB SEI      INTERRUPT SPERREN
.26FC JSR $26F0 HYPERGRAPHIC-NMI-VEKTOR SETZEN
.26FF LDA $0300 LOWBYTE DES VEKTORS AUF AUSGABE EINER FEHLERMELDUNG HOLEN
.2702 STA $273A ... UND RETTEN
.2705 LDA $0301 DAZUGEHOEERIGES HIGH-BYTE HOLEN
.2708 STA $273B ... UND ENTSPRECHEND SETZEN
.270B LDA #$2D LOW-BYTE DES NEUEN FEHLERMELDUNGS-AUSGABEVEKTORS HOLEN
.270D STA $0300 ... UND SETZEN
.2710 LDA #$27 ENTSPRECHENDES HIGH-BYTE HOLEN
.2712 STA $0301 ...UND ENTSPRECHEND SETZEN
.2715 JSR $24DC HYPERGRAPHIC-MODUS AUF TEXT STELLEN (= GRAPHIK AUSSCHALTEN)
.2718 LDA #$27 HIGH-BYTE DES HYPERGRAPHICS-BASIC-STARTS HOLEN
.271A STA $2C ... UND IN DIE DAFUER VORGESEHENEN SPEICHERSTELLEN BRINGEN
.271C STA $0282
.271F LDA #$40 ENTSPRECHEND MIT DEM LOW-BYTE VERFAHREN
.2721 STA $FB "
.2723 STA $0281 "
.2726 JSR $C644 BASIC-BEFEHL NEW AUSFUEHREN
.2729 CLI      IRQ WIEDER FREIGEBEN
.272A JMP $C474 ZUM READY-EINSPRUNG

```

AUSGABE EINER FEHLERMELDUNG:

```

.272D STX $2739 FEHLERNUMMER RETTEN
.2730 JSR $24DC GRAPHIK AUSSCHALTEN
.2733 LDX $2739 FEHLERNUMMER HOLEN
.2736 JMP ($273A) ZU 'FEHLERMELDUNG AUSGEBEN'

```

GRAPHIK AUSSCHALTEN:

```

.24DC LDA $9005 HOLT ZEIGER AUF CHARACTERGENERATOR & VIDEORAM
.24DF CMP #$C0 GRAPHIK AUS?
.24E1 BEQ $2510 JA, DANN FERTIG
.24E3 LDX #$D0 7
.24E5 LDA $0FFF,X 1
.24E8 STA $1FFF,X 1

```

Listing 3. Quell-Code-Listing zu »Hyper-Graphics« (Fortsetzung)

```
.24EB LDA $10CF,X |
.24EE STA $20CF,X | SIEHE $2499 - $24B4
.24F1 LDA $93FF,X | (HIER UMGEKEHRTE FUNKTION)
.24F4 STA $219F,X |
.24F7 LDA $94CF,X |
.24FA STA $226F,X |
.24FD DEX |
.24FE BNE $24E5 |
.2500 LDA #$70 |
.2502 LDA $119F,X |
.2505 STA $233F,X | SIEHE $24B6 - $24BF
.2508 DEX | (HIER UMGEKEHRTE FUNKTION)
.2509 BNE $2502 |
.250B JSR $E55F BILDSCHIRM LOESCHEN
.250E LDY #$02 CODE FUER FARBE ROT IM FARBRAM
.2510 LDX #$16 ANZAHL DER ZEICHEN PRO ZEILE
.2512 LDA $23FF,X HOLT BILDSCHIRMCODE AUS TABELLE
.2515 STA $11F9,X BRING ZEICHEN AUF BILDSCHIRM
.2518 TYA HOLT FARBCODE
.2519 STA $95F9,X SETZT FARBCODE
.251C DEX ZAEHLER ERNIEDRIGEN
.251D BNE $2512 UNGLEICH NULL? DANN VORGANG WIEDERHOLEN
.251F LDY #$06 ZAEHLER FUER ANZAHL DER VIDEOCONTROLLER-WERTE
.2521 LDA $EDE3,Y HOLT VIDEOCONTROLLER-WERTE AUS DEM ROM
.2524 STA $8FFF,Y SETZT VIDEOCONTROLLER-WERTE
.2527 DEX ZAEHLER ERNIEDRIGEN
.2528 BNE $2521 UNGLEICH NULL? DANN VORGANG WIEDERHOLEN
.252A LDA #$22 BILDSCHIRM VERTIKAL ZENTRIEREN
.252C STA $9001 "
.252F LDA #$30 KONSTANTE FUER 24-ZEILEN-BILDSCHIRM HOLEN
.2531 STA $9003 ... UND IN VIDEOCONTROLLER SETZEN
.2534 RTS ZURUECK INS BASIC
```

Listing 3. Quell-Code-Listing zu »Hyper-Graphics« (Schluß)

```
10 REM***** <064>
20 REM* HYPER-GRAPHIK DEMO * <194>
30 REM* (C) 1985 BY JOACHIM BROCKE * <026>
40 REM* DIETRICH-BONHOEFFERSTR.-5 * <013>
50 REM* 4790 PADERBORN * <104>
60 REM***** <114>
70 REM FESTLEGUNG DER STARTADRESSEN DER EINZELNEN ROUTINEN <254>
80 IN=9330:REM'GRAPHIC ON' <024>
90 CL=IN+3:REM'GRAPHIC CLEAR' <102>
100 SE=IN+6:REM'POINT SET' <194>
110 RS=IN+9:REM'RESET POINT' <166>
120 GS=IN+12:REM'GSAVE' <174>
130 GL=IN+15:REM'GLOAD' <181>
140 OF=IN+18:REM'GRAPHIC OFF' <201>
150 REM MENUE <082>
160 PRINT"(CLR)HYPER-GRAPHICS DEMO " <082>
170 PRINT"(DOWN)BITTE WAEHLEN: " <151>
180 PRINT"(DOWN)3D-EFFEKT-BILD (-1-) " <051>
190 PRINT"(DOWN)SINUSKURVE (5SPACE) (-2-) " <255>
200 PRINT"(DOWN)RECHTECKE (6SPACE) (-3-) " <233>
210 PRINT"(DOWN)TEXT + GRAPHIK (-4-) " <253>
220 GET X$ <102>
230 IF VAL(X$)<1 OR VAL(X$)>4 THEN 220 <012>
```

Listing 4. Demo-Programm für »Hyper-Graphics«

240 REM WAHL DER FARBEN	<043>
250 PRINT "{DOWN}NACH DEM PLOTTEN{6SPACE}TASTE DRUECKEN"	<162>
260 INPUT "RAHMENFARBE";RF	<253>
270 IF RF<0 OR RF>7 THEN PRINT "FEHLER!":GOTO 260	<039>
280 INPUT "HINTERGRUNDFARBE";HF	<107>
290 IF HF<0 OR HF>15 THEN PRINT "FEHLER!":GOTO 280	<214>
300 INPUT "PUNKTFARBE";PF	<128>
310 IF PF<0 OR PF>7 OR PF=HF THEN PRINT "FEHLER!":GOTO 300	<057>
320 POKE 646,PF:REM ZEICHENFARBE IM TEXT-MODUS FESTLEGEN	<158>
330 POKE 36879,8+RF+16*HF:REM RAHMEN- UND HINTERGRUNDFARBE SETZEN	<037>
340 REM VERZWEIGENINGEWAEHLTESUNTERPROGRAMM:	<090>
350 ON VAL(X\$)GOSUB 1200,400,470,1310:	<218>
360 POKE 198,0:WAIT 198,1:REM AUF TASTE WARTEN	<120>
370 SYS 0F:REM GRAPHIK AUSSCHALTEN	<209>
380 RUN:REM NEUSTART	<012>
390 REM*** DIE UNTERPROGRAMME ***	<233>
400 REM SINUSKURVE	<009>
410 SYS IN:SYS CL	<031>
420 X=-1:FOR I=-1TO1STEP 2*1/208	<237>
430 X=X+1:Y=SIN(I)	<159>
440 SYS SE,X,Y*127+127,PF	<009>
450 NEXT I	<024>
460 RETURN	<008>
470 REM RECHTECKE	<194>
480 SYS IN:SYS CL	<101>
490 FOR I=16 TO 208-16	<020>
500 SYS SE,I,16,PF	<126>
510 SYS SE,I,255-16,PF	<116>
520 NEXT	<022>
530 FOR I=31 TO 208-31	<189>
540 SYS SE,I,31,PF	<160>
550 SYS SE,I,255-31,PF	<030>
560 NEXT	<062>
570 FOR I=16 TO 255-16	<100>
580 SYS SE,16,I,PF	<240>
590 SYS SE,208-16,I,PF	<075>
600 NEXT	<102>
610 FOR I=31 TO 255-31	<011>
620 SYS SE,31,I,PF	<021>
630 SYS SE,208-31,I,PF	<083>
640 NEXT	<142>
650 FOR I=64 TO 208-64	<230>
660 SYS SE,I,64,PF	<033>
670 SYS SE,I,255-64,PF	<038>
680 NEXT	<182>
690 FOR I=71 TO 208-71	<029>
700 SYS SE,I,71,PF	<068>
710 SYS SE,I,255-71,PF	<254>
720 NEXT	<222>
730 FOR I=86 TO 255-86	<085>
740 SYS SE,86,I,PF	<082>
750 SYS SE,208-86,I,PF	<007>
760 NEXT	<006>
770 FOR I=90 TO 255-90	<110>
780 SYS SE,90,I,PF	<185>
790 SYS SE,208-90,I,PF	<005>
800 NEXT	<008>
810 FOR I=86 TO 208-86	<169>
820 SYS SE,I,86,PF	<201>
830 SYS SE,I,255-86,PF	<040>
840 NEXT	<088>
850 FOR I=90 TO 208-90	<192>
860 SYS SE,I,90,PF	<230>

Listing 4. Demo-Programm für »Hyper-Graphics« (Fortsetzung)



```

870 SYS SE,I,255-90,PF <160>
880 NEXT <128>
890 FOR I=64 TO 255-64 <214>
900 SYS SE,64,I,PF <114>
910 SYS SE,208-64,I,PF <145>
920 NEXT <168>
930 FOR I=71 TO 255-71 <013>
940 SYS SE,71,I,PF <088>
950 SYS SE,208-71,I,PF <165>
960 NEXT <208>
970 FOR I=100 TO 208-100 <034>
980 SYS SE,I,100,PF <174>
990 SYS SE,I,255-100,PF <029>
1000 NEXT <248>
1010 FOR I=102 TO 208-102 <076>
1020 SYS SE,I,102,PF <222>
1030 SYS SE,I,255-102,PF <199>
1040 NEXT <034>
1050 FOR I=100 TO 255-100 <112>
1060 SYS SE,100,I,PF <088>
1070 SYS SE,208-100,I,PF <103>
1080 NEXT <074>
1090 FOR I=102 TO 255-102 <154>
1100 SYS SE,102,I,PF <130>
1110 SYS SE,208-102,I,PF <175>
1120 NEXT <114>
1130 X=-1:FOR I=1 TO 102 <210>
1140 Y=I <010>
1150 SYS SE,I,Y,PF <255>
1160 SYS SE,208-I,Y,PF <135>
1170 SYS SE,I,255-Y,PF <013>
1180 SYS SE,208-I,255-Y,PF <089>
1190 NEXT I:RETURN <118>
1200 REM 3-EFFEKT-BILD <212>
1210 SYS IN:SYS CL <069>
1220 DEF FN A(Z)=90*EXP(-Z*Z/1500) <110>
1230 G=100:K=50:FOR X=-103 TO 0:L=-50:H=5*INT(SQR(10816-X*X)/5) <176>
1240 FOR Y=H TO -H STEP-5:Z=25+FN A(SQR(X*X+Y*Y))-.6*Y <060>
1250 X1=G+X:Y1=256-Z-K:X2=G-X <081>
1260 IF Z>L THEN L=Z:SYS SE,X1+3,Y1,6:SYS SE,X2+3,Y1,6 <046>
1270 NEXT Y,X <159>
1280 RETURN <068>
1290 SYS OF:POKE 36879,27:PRINT" {BLUE} <199>
1300 RETURN <088>
1310 REM TEXT+GRAPHIK <002>
1320 INPUT"(CLR)ASCII-TEXT(3SPACE)OKTAEDER(10LEFT)";TE$ <045>
1330 IF LEN(TE$)>16 OR TE$=""THEN 1320 <200>
1340 SYS CL:SYS IN <126>
1350 REM ASCII-TEXT (A-Z UND SPACE BIS ?)IN PUNKTE WANDELN <207>
1360 FOR I=1 TO LEN(TE$) <157>
1370 A=ASC(MID$(TE$,I,1)):IF A>63 THEN A=A-64 <067>
1380 X=I*8:REM STARTKOORDINATE DES TEXTES (X-RICHTUNG) <006>
1390 AD=32768+A*8:REM ADRESSE DES ZEICHENS IM CHARACTERGENERATOR <033>
1400 FOR J=0 TO 7 <207>
1410 A=PEEK(AD+J):REM EINES DER 8 BYTES DES ZEICHENS HOLEN <150>
1420 FOR K=0 TO 7 <235>
1430 P=A AND 2^(7-K):REM EINES DER 8 BITS PRO ZEILE DES ZEICHENS ISOLIER
EN <248>
1440 IF P THEN SYS SE,X+K,2*J+16,PF:REM BIT GESETZT?DANN PUNKT SETZEN <204>
1450 IF P THEN SYS SE,X+K,2*J+17,PF:REM BIT GESETZT?DANN PUNKT SETZEN <218>
1460 NEXT K,J,I <220>
1470 SYS IN:GOSUB 1490:REM OKTAEDER ZEICHNEN <075>
1480 RETURN <012>

```

Listing 4. Demo-Programm für »Hyper-Graphics« (Fortsetzung)

```

1490 REM OKTAEDER                                <219>
1500 FOR X=66 TO 133                              <225>
1510 SYS SE,X+30,127,PF                          <150>
1520 SYS SE,X+30,128,PF                          <193>
1530 SYS SE,(X AND 252)+68,127-18,PF             <030>
1540 NEXT X                                       <220>
1550 Y=127:FOR X=67 TO 101                       <159>
1560 SYS SE,(X+30)AND 254,Y,PF                   <123>
1570 SYS SE,X+97,Y,PF                            <004>
1580 SYS SE,X+96,Y,PF                            <012>
1590 Y=Y-.5                                       <107>
1600 NEXT X                                       <024>
1610 REM SYSSE,118,118,1                         <148>
1620 Y=128:FOR X=66 TO 118                       <157>
1630 SYS SE,X+30,Y,PF                            <044>
1640 SYS SE,X+30,Y+1,PF                          <034>
1650 SYS SE,X+30,256-Y,PF                        <062>
1660 SYS SE,X+30,255-Y,PF                        <040>
1670 Y=Y+1.5                                     <014>
1680 NEXT X                                       <104>
1690 Y=110:FOR X=102 TO 118                      <049>
1700 SYS SE,X+30,Y,PF                            <114>
1710 Y=Y+6                                       <043>
1720 NEXT X                                       <144>
1730 Y=207:FOR X=118 TO 133                     <211>
1740 SYS SE,X+30,Y-1,PF                          <150>
1750 SYS SE,X+30,Y-4,PF                          <000>
1760 SYS SE,X+30,Y-2,PF                          <202>
1770 SYS SE,X+30,Y-3,PF                          <244>
1780 SYS SE,X+30,Y-5,PF                          <062>
1790 Y=Y-5                                       <091>
1800 NEXT X                                       <226>
1810 Y=206:FOR X=118 TO 169                     <244>
1820 SYS SE,X+30,Y,PF                            <236>
1830 SYS SE,X+30,Y-1,PF                          <242>
1840 Y=Y-1.9                                     <222>
1850 NEXT X                                       <020>
1860 Y=110:FOR X=102 TO 118                     <221>
1870 SYS SE,X+30,Y,PF                            <030>
1880 Y=Y-3.8                                     <133>
1890 NEXT X                                       <060>
1900 Y=49:FOR X=148 TO 164                     <000>
1910 SYS SE,X,Y,PF                               <193>
1920 SYS SE,X,Y+1,PF                             <071>
1930 SYS SE,X,Y+2,PF                             <085>
1940 SYS SE,X,Y+3,PF                             <099>
1950 SYS SE,X,Y+4,PF                             <113>
1960 Y=Y+4.8                                     <246>
1970 NEXT X                                       <140>
1980 Y=49:FOR X=148 TO 200                     <137>
1990 SYS SE,X,Y,PF                               <017>
2000 Y=Y+1.15                                    <192>
2010 NEXT X                                       <180>
2020 RETURN                                       <044>

```

© 64'er

Listing 4. Demo-Programm für »Hyper-Graphics« (Schluß)