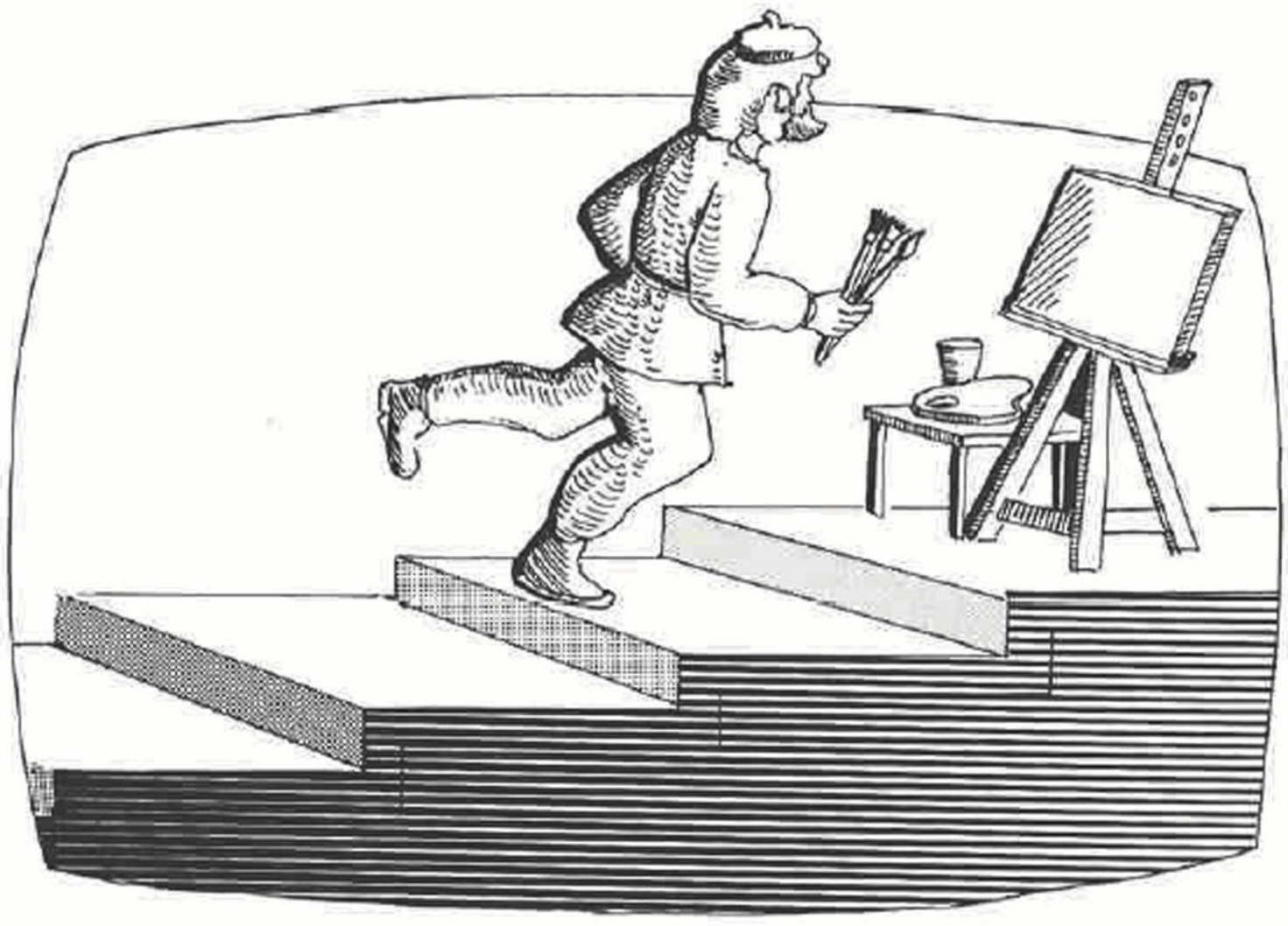
Double-Lores-Applesoft-Erweiterungen

von Karl-Walter Bott

Wenn Sie einen Apple IIe mit einer erweiterten 80-Zeichenkarte oder einen Apple IIc besitzen, dann können Sie mit der hier vorgestellten Software den Applesoft-Interpreter um sechs neue Grafik-Befehle erweitern.

Die erweitere 80-Zeichenkarte für den Apple IIe stellt zu den bereits vorhandenen 64K RAM auf dem Motherboard weitere 64K RAM zur Verfügung, die sich auf vielfältige Weise einsetzen lassen. Eine dieser Möglichkeiten besteht darin, die Auflösung der Lores-Grafik zu verdoppeln, so daß 80 mal 48 Kästchen darstellbar sind. Über die Organisation und die Programmierung der Double-Lores-Grafik wurde bereits im Peeker 1/84 ausführlich berichtet.
Die Programmierung von Double-Lores in Applesoft-BASIC erweist sich als sehr umständlich, da der Interpreter keine Befehle zur Erstellung von Double-Lores-

Grafiken enthält, wie man sie von der normalen niedrigauflösenden Grafik gewohnt ist.
Das hier vorgestellte Software-Paket implementiert die fehlenden Befehle, so daß Applesoft-Programmierer die Double-Lores-Grafik nutzen können, ohne detaillierte Kenntnisse über deren rechnerinterne Organisation zu besitzen.



Die neuen Befehle

&GR – Das &GR-Statement veranlaßt den Computer, auf doppelt-niedrigauflösende Grafik umzuschalten. Dabei wird der Bildschirm gelöscht und vier Textzeilen am unteren Bildschirmrand bereitgestellt. Die Darstellung auf dem Bildschirm erfolgt jetzt in einem 80-mal-40-Raster.
Will man die ganze Seite mit Grafik ausfüllen (80 mal 48 Punkte), so muß zunächst durch
POKE 49234,0
auf Full-Screen-Ausgabe umgeschaltet werden. Um die acht Grafikzeilen am unteren Bildschirmrand zu löschen, verwende man dann folgende Befehle:
POKE 49236,0: CALL 63538
POKE 49237,0: CALL 63538.
Hierdurch wird der ganze Bildschirm gelöscht und wie oben die Farbe schwarz eingestellt (COLOR = 0).

&TEXT – Dieser Befehl schaltet von Double-Lores-Grafik auf Textdarstellung um. Es wird der 40-Zeichen-Textmodus eingeschaltet und die erweiterte 80-Zeichenkarte desaktiviert.
Dieser Befehl sollte nur dann benutzt werden, wenn die 80-Zeichenkarte vor Aufruf von &GR nicht aktiv war, da ansonsten Softswitch-Probleme auftreten. Wurde die Double-Lores vom 80-Zeichenmodus aus aktiviert, genügt ein TEXT-Befehl zum Zurückschalten.

&PLOT – Der &PLOT-Befehl zeichnet ein Kästchen in der aktuellen Farbe auf dem Double-Lores-Bildschirm. Der erste Ausdruck, der dem Schlüsselwort „&PLOT" folgt, gibt die Spalte an, in der das Kästchen gezeichnet werden soll (zwischen 0...79, von links nach rechts); der zweite Ausdruck, durch ein Komma vom ersten getrennt, bestimmt die Zeile (0...47, von oben nach unten).

Beispiel: „&PLOT 10, 15" plottet einen Block in Spalte 10, Zeile 15.
Der normale PLOT-Befehl wurde dahingehend erweitert, daß jetzt – wie unter HGR – auch Linien zwischen zwei oder mehreren beliebigen Punkten gezeichnet werden können.
Beispiel: „&PLOT 0, 0 TO 79, 47" zeichnet eine diagonale Linie von links oben nach rechts unten in der aktuellen Farbe.
Liegen die angegebenen Koordinaten außerhalb des zulässigen Bereichs, wird das Programm unterbrochen und die Meldung „ILLEGAL QUANTITY ERROR" ausgegeben.

&HLIN – Das &HLIN-Statement zeichnet eine horizontale Linie in der aktuellen Farbe. Die beiden Ausdrücke, die dem Befehl „&HLIN" folgen, bestimmen die Spaltennummern, in denen der Anfang und das Ende der Linie liegen. Der Ausdruck, der dem Schlüsselwort „AT" folgt, bestimmt die Zeile, in welcher die horizontale Linie gezeichnet werden soll.
Beispiel: „&HLIN 20, 60 AT 10" zeichnet eine horizontale Linie in Zeile 10 von Spalte 20 bis Spalte 60.
Wird die zulässige Spalten- oder Zeilenzahl überschritten, wird eine Fehlermeldung ausgegeben und das Programm unterbrochen.

&VLIN – Das &VLIN-Statement zeichnet eine vertikale Linie. Die beiden folgenden Ausdrücke bestimmen Anfangs- und Endzeile der Linie, während der Ausdruck nach „AT" die Spalte bestimmt, in der die Linie gezeichnet wird.
Beispiel: „&VLIN 10, 30 AT 20" zeichnet eine vertikale Linie in Spalte 20 von Zeile 10 bis Zeile 30.
Bei Überschreiten der zulässigen Spalten- oder Zeilenzahl wird eine Fehlermeldung ausgegeben und der Programmablauf unterbrochen.

&SCRN – Der &SCRN-Befehl ermittelt die Farbe eines angegebenen Blocks auf dem Double-Lores-Bildschirm. Der Wert, den die Funktion zurückgibt, liegt zwischen 0 und 15, und wird der angegebenen Variablen zugewiesen, die wahlweise vom Typ Integer oder Real sein kann.
Beispiel: „&SCRN(A, 50, 10)" ermittelt die Farbe des Blocks in Spalte 50, Zeile 10. Die Variable A hat dann einen Wert zwischen 0 und 15.
Wenn die Variable nicht dem zulässigen Typ entspricht (z.B. A$) oder die zulässigen Grenzen für die Koordinatenangaben überschritten werden, erfolgt eine Fehlermeldung.

Der normale COLOR-Befehl kann weiterhin verwendet werden, um die gewünschte Farbe eines Blockes zu definieren, wobei der Ausdruck rechts vom Gleichheitszeichen zwischen 0 und 15 liegen muß.
Beispiel: „COLOR = 15" wählt die Farbe weiß aus.

Technische Anmerkungen

Das Programm residiert ab Adresse $931A im Hauptspeicher und ist 742 Bytes

lang, kann aber durchaus auch mit einer anderen Startadresse assembliert werden. (Anm.: Dies ist insbesondere für ProDOS erforderlich. Unter DOS 3.3 liegt das Programm unterhalb von $9600, d.h. im Bereich $931A-$95FF unterhalb vom dritten DOS-Puffer bei Maxfiles 3. Man *könnte* es auch unterhalb von $9A00 legen und die DOS-Puffer entsprechend nach unten verschieben. Unter ProDOS *muß* es unterhalb von $9A00 liegen, d.h. im Bereich $971A-$99FF, weil es sonst durch einen Textfile-Zugriff zerstört würde. Zu diesem Zweck muß man über die GETBUFR-Routine 3 Pages anfordern und dann das Programm mit BRUN DOUBLE.LORES starten. Aus all diesen Gründen wird HIMEM durch DOUBLE.LORES nicht geändert. us)

Um die neuen Applesoft-Befehle in den Interpreter einzubinden, wurde der Ampersand-Vektor verwendet. Er bietet eine bequeme Möglichkeit, eigene Routinen ohne Geschwindigkeitsverlust an den Interpreter anzuhängen.

Jedesmal, wenn der Interpreter auf das &-Zeichen trifft, verzweigt er zur Adresse $03F5, in der normalerweise ein Sprung zu einem RTS-Befehl steht. An dieser Stelle wird nun ein Sprung zum eigenen Programm eingetragen.

Stößt der Interpreter auf das &-Zeichen, so verzweigt er zum Anwenderprogramm, das zunächst überprüft, welcher Token vorliegt, um zu dem entsprechenden Programmsegment zu springen. Nach Ausführung des jeweiligen Befehls wird der Textpointer bis zum nächsten Applesoft-Befehl vorgerückt und DOUBLE.LORES über einen RTS-Befehl verlassen.

Konflikte mit 40/80-Z/Z

Es empfiehlt sich, vor Aktivierung der Double-Lores die 80-Zeichenkarte mit „PRINT CHR$(21)" oder „ESC Ctrl-Q" abzustellen, um Speicherkonflikte zu vermeiden und das korrekte Arbeiten des &TEXT-Befehls zu gewährleisten.

Sollten Sie keine Double-Lores-Grafik auf dem Bildschirm sehen, kann das an Ihrer erweiterten 80-Zeichenkarte liegen. Entweder Sie besitzen eine Nachbau-80-Zeichenkarte, die nicht über die entsprechenden technischen Möglichkeiten verfügt, oder die Karte ist nicht korrekt installiert. Lesen Sie dazu das entsprechende Kapitel im Handbuch zur 80-Zeichenkarte.

Bei gemischter Darstellung (Text und Grafik) beachten Sie bitte folgendes:

DOUBLE.LORES

```
                        1           ORG  $931A
                        2  *
                        3  *    Double-Lores 1.Ø
                        4  *    ================
                        5  *
                        6  *    K.-W. Bott, 1985
                        7  *
                        8  *
                        9  *    Tokens
                        1Ø  *
                        11  TPLOT    EQU  $8D
                        12  TGR      EQU  $88
                        13  THLIN    EQU  $8E
                        14  TVLIN    EQU  $8F
                        15  TSCRN    EQU  $D7
                        16  TTEXT    EQU  $89
                        17  TO       EQU  $C1
                        18  AT       EQU  $C5
                        19  *
                        2Ø  *    Applesoft-Routinen
                        21  *
                        22  AMPER    EQU  $Ø3F5      ;Ampersand-Vektor
                        23  TABV     EQU  $FB5B      ;Cursor vertikal tabulieren
                        24  CLRTOP   EQU  $F836      ;Lores-Schirm löschen
                        25  PLOT     EQU  $F8ØØ      ;Block plotten
                        26  SCRN     EQU  $F871      ;Lores-Screen-Befehl
                        27  CHRGET   EQU  $ØØB1      ;Zeichen aus BASIC
                        28  GETBYT   EQU  $E6F8      ;Byte aus BASIC
                        29  SYNCHR   EQU  $DECØ      ;Zeichen im AREG
                        3Ø  *                        ;mit BASIC vergleichen
                        31  SNERR    EQU  $DEC9      ;"SYNTAX ERROR"
                        32  ILQUAN   EQU  $E199      ;"ILLEGAL QUANTITY"
                        33  PTRGET   EQU  $DFE3      ;Pointer auf Variable
                        34  *                        ;aus BASIC-Text
                        35  SNGFLT   EQU  $E3Ø1      ;YREG -> FAC
                        36  *
                        37  *    Konstanten
                        38  *
                        39  PAGE1    EQU  $C054      ;Hauptspeicher
                        4Ø  PAGE2    EQU  $C055      ;Zusatzspeicher
                        41  CHAR     EQU  $B8        ;aktuelles Zeichen im
                        42  *                        ;BASIC-Text
                        43  POINTER  EQU  $CE        ;Zeiger für Puffer
                        44  VARNAM   EQU  $81        ;letzte BASIC-Variable
                        45  SUBFLG   EQU  $14        ;Flag für Var.-behandlung
                        46  FAC      EQU  $9D        ;Float.-Akkumulator
                        47  WNDTOP   EQU  $22        ;obere Begrenzung des
                        48  *                        ;Scroll-Windows
                        49  *
                        5Ø  *    Speicherplatz für Koordinaten
                        51  *
                        52  X1       EQU  $Ø6
                        53  X2       EQU  $Ø7
                        54  Y1       EQU  $Ø9
                        55  Y2       EQU  $E3
                        56  XS       EQU  $D7        ;Schrittweite X
                        57  YS       EQU  $EB        ;Schrittweite Y
                        58  XD       EQU  $CF        ;Differenz X1 - X2
                        59  YD       EQU  $EF        ;Differenz Y1 - Y2
                        6Ø  *
                        61  *
                        62  * Ampersand-Vektor linken
                        63  *
931A: A9 4C             64  START    LDA  #$4C       ;JMP
931C: 8D F5 Ø3          65           STA  AMPER
931F: A9 2A             66           LDA  #<LINK
9321: 8D F6 Ø3          67           STA  AMPER+1
9324: A9 93             68           LDA  #>LINK
9326: 8D F7 Ø3          69           STA  AMPER+2
9329: 6Ø                7Ø           RTS
                        71  *
                        72  * Einsprung für jeden Ampersand-
                        73  * Befehl; Token im A-Register
                        74  *
932A: C9 88             75  LINK     CMP  #TGR
932C: FØ 2Ø             76           BEQ  GR
932E: C9 89             77           CMP  #TTEXT
933Ø: FØ 46             78           BEQ  TEXT
9332: C9 8D             79           CMP  #TPLOT
9334: FØ ØF             8Ø           BEQ  JPLOT
9336: C9 8E             81           CMP  #THLIN
9338: FØ 54             82           BEQ  HLIN
933A: C9 8F             83           CMP  #TVLIN
933C: FØ ØA             84           BEQ  JVLIN
933E: C9 D7             85           CMP  #TSCRN
934Ø: FØ Ø9             86           BEQ  JSCRN
```

1. Ein &PLOT-Befehl über die 39. Grafikzeile hinaus druckt ein Zeichen in die unteren vier Textzeilen.
2. Die erweiterte 80-Zeichenkarte sollte vor Aktivierung der Double-Lores-Grafik eingeschaltet sein und eingeschaltet bleiben.
3. War die 80-Zeichenkarte aktiviert, so ist der TEXT-Befehl zum Zurückschalten zu verwenden, ansonsten &TEXT.

Bei reiner Grafikdarstellung ist folgendes zu berücksichtigen:
1. Benutzen Sie die oben erwähnten POKE- und CALL-Befehle, um die unteren vier Zeilen für Grafik freizuhalten.
2. Jeder PRINT-Befehl druckt Zeichen in den Bildschirmspeicher, die als Farbblökke auftreten.
3. Selbst ein GET-Befehl erzeugt einen farbigen Block, da der Cursor sichtbar wird. Um dies zu vermeiden, kann die Eingabe eines Zeichens von der Tastatur z.B. wie folgt vorgenommen werden:

```
10 ON PEEK (49152) < 128 GOTO 10:A$
     = CHR$ (PEEK (49152) - 128):
     POKE 49168,0
```

4. Für das Zurückschalten gilt das gleiche wie oben.

Das Programm **DOUBLE.LORES.DEMO** zeigt eine Anwendung der neuen Befehle und die Umschaltung auf reine Grafik-Ausgabe.

```
9342: 4C C9 DE  87              JMP  SNERR
                88  *
9345: 4C 1B 94  89  JPLOT       JMP  SPLOT
9348: 4C C0 93  90  JVLIN       JMP  VLIN
934B: 4C F1 93  91  JSCRN       JMP  SCREEN
                92  *
                93  * &GR
                94  * ----
                95  *
                96  * Double-Lores initialisieren
                97  * und Bildschirm löschen
                98  *
934E: AD 50 C0  99  GR          LDA  $C050      ;GRAFIK
9351: AD 56 C0  100             LDA  $C056      ;LORES
9354: AD 53 C0  101             LDA  $C053      ;MIXSET
9357: 8D 01 C0  102             STA  $C001      ;80STORE
935A: 8D 0D C0  103             STA  $C00D      ;80COL
935D: AD 5E C0  104             LDA  $C05E      ;AN3
9360: A9 14     105             LDA  #20
9362: 85 22     106             STA  WNDTOP     ;Scroll-Window begrenzen
9364: A9 17     107             LDA  #23        ;Cursor in Zeile 23
9366: 20 5B FB  108             JSR  TABV       ;setzen
9369: AD 55 C0  109             LDA  PAGE2
936C: 20 36 F8  110             JSR  CLRTOP     ;Zusatzspeicher löschen
936F: AD 54 C0  111             LDA  PAGE1
9372: 20 36 F8  112             JSR  CLRTOP     ;Hauptspeicher löschen
9375: 4C B1 00  113             JMP  CHRGET     ;Textptr. vorrücken
                114 *                           ;und Return nach BASIC
                115 *
                116 * &TEXT
                117 * ----
                118 *
                119 * Zurück in den Textmodus
                120 *
9378: 8D 00 C0  121 TEXT        STA  $C000      ;40STORE
937B: 8D 0C C0  122             STA  $C00C      ;40COL
937E: AD 5F C0  123             LDA  $C05F      ;AN3 OFF
9381: AD 51 C0  124             LDA  $C051      ;TEXT
9384: AD 54 C0  125             LDA  PAGE1
9387: A9 00     126             LDA  #0
9389: 85 22     127             STA  WNDTOP     ;volles Scroll-Window
938B: 4C B1 00  128             JMP  CHRGET     ;Txtptr. ink. -> BASIC
                129 *
                130 * &HLIN X1, X2 AT Y
                131 * ----------
                132 *
938E: 20 B1 00  133 HLIN        JSR  CHRGET     ;TXTPTR vorrücken
9391: 20 F8 E6  134             JSR  GETBYT     ;Param. X1 aus BASIC-Text
9394: E0 50     135             CPX  #80        ;X1 > 79?
9396: B0 25     136             BCS  ERROR
```

```
9398: 86 Ø6     137          STX  X1
939A: A9 2C     138          LDA  #','
939C: 2Ø C0 DE  139          JSR  SYNCHR     ;Komma?
939F: 2Ø F8 E6  14Ø          JSR  GETBYT     ;Param. X2 aus BASIC-Text
93A2: EØ 5Ø     141          CPX  #8Ø        ;X2 > 79?
93A4: 86 Ø7     142          STX  X2
93A6: A9 C5     143          LDA  #AT
93A8: 2Ø C0 DE  144          JSR  SYNCHR     ;'AT'?
93AB: 2Ø F8 E6  145          JSR  GETBYT     ;Param. Y aus BASIC-Text
93AE: EØ 3Ø     146          CPX  #48        ;Y > 47?
93BØ: BØ ØB     147          BCS  ERROR
93B2: 86 Ø9     148          STX  Y1
93B4: 86 E3     149          STX  Y2
93B6: 2Ø 8A 94  15Ø          JSR  LINPLOT    ;Linie plotten
93B9: AD 54 C0  151          LDA  PAGE1      ;sicher ist sicher ...
93BC: 6Ø        152          RTS             ;zurück zu BASIC
93BD: 4C 99 E1  153 ERROR    JMP  ILQUAN
                154 *
                155 * &VLIN Y1, Y2 AT X
                156 * ═══════════════
                157 *
93C0: 2Ø B1 ØØ  158 VLIN     JSR  CHRGET     ;Txtptr. vorrücken
93C3: 2Ø F8 E6  159          JSR  GETBYT     ;Param. Y1 aus BASIC
93C6: EØ 3Ø     16Ø          CPX  #48        ;Y1 > 47?
93C8: BØ F3     161          BCS  ERROR
93CA: 86 Ø9     162          STX  Y1
93CC: A9 2C     163          LDA  #','       ;Komma?
93CE: 2Ø C0 DE  164          JSR  SYNCHR
93D1: 2Ø F8 E6  165          JSR  GETBYT     ;Param. Y2 aus BASIC
93D4: EØ 3Ø     166          CPX  #48        ;Y2 > 47?
93D6: BØ E5     167          BCS  ERROR
93D8: 86 E3     168          STX  Y2
93DA: A9 C5     169          LDA  #AT        ;'AT'?
93DC: 2Ø C0 DE  17Ø          JSR  SYNCHR
93DF: 2Ø F8 E6  171          JSR  GETBYT     ;Param. X aus BASIC
93E2: EØ 5Ø     172          CPX  #8Ø        ;X > 79
93E4: BØ D7     173          BCS  ERROR
93E6: 86 Ø6     174          STX  X1
93E8: 86 Ø7     175          STX  X2
93EA: 2Ø 8A 94  176          JSR  LINPLOT    ;Linie zeichnen
93ED: AD 54 C0  177          LDA  PAGE1
93FØ: 6Ø        178          RTS             ;zurück zu BASIC
                179 *
                18Ø * &SCRN (VAR, X, Y)
                181 * ═══════════════
                182 *
                183 * Farbwert des Blockes X, Y bestimmen
                184 * und in BASIC-Variable übertragen
                185 *
93F1: 2Ø B1 ØØ  186 SCREEN   JSR  CHRGET     ;Txtptr. vorrücken
93F4: 2Ø AD 95  187          JSR  GETADR     ;Adr. und Typ der
                188 *                        ;Variablen feststellen
93F7: A9 2C     189          LDA  #','       ;Komma?
93F9: 2Ø C0 DE  19Ø          JSR  SYNCHR
93FC: 2Ø 58 94  191          JSR  GETVAL     ;Werte X, Y aus BASIC
93FF: A5 Ø6     192          LDA  X1
94Ø1: 4A        193          LSR             ;A = A / 2
94Ø2: 9Ø Ø5     194          BCC  S2         ;gerader Wert -> PG2
94Ø4: AE 54 C0  195          LDX  PAGE1
94Ø7: BØ Ø3     196          BCS  OKAY       ;ungerade -> PG1
94Ø9: AE 55 C0  197 S2       LDX  PAGE2
94ØC: A8        198 OKAY     TAY             ;X -> YREG
94ØD: A5 Ø9     199          LDA  Y1         ;Y -> AREG
94ØF: 2Ø 71 F8  2ØØ          JSR  SCRN       ;Farbe im AREG
9412: 2Ø D2 95  2Ø1          JSR  GIVBACK    ;Farbcode der Variablen
```

```
                2Ø2 *                        ;zuordnen
9415: AD 54 C0  2Ø3          LDA  PAGE1
9418: 4C B1 ØØ  2Ø4          JMP  CHRGET     ;Txtptr. vorrücken
                2Ø5 *                        ;Return nach BASIC
                2Ø6 *
                2Ø7 *
                2Ø8 * &PLOT X1, Y1 TO X2, Y2 ....
                2Ø9 * ═══════════════════
                21Ø *
941B: 2Ø B1 ØØ  211 SPLOT    JSR  CHRGET     ;Txtptr. vorrücken
941E: A2 ØØ     212          LDX  #Ø
942Ø: A1 B8     213          LDA  (CHAR,X)
9422: C9 C1     214          CMP  #TO
9424: FØ ØE     215          BEQ  CTO        ;keine Param. holen
9426: 2Ø 58 94  216          JSR  GETVAL
9429: 2Ø 73 94  217          JSR  EXPLOT     ;Punkt plotten
                218 *
942C: A2 ØØ     219 PLOTTO   LDX  #Ø
942E: A1 B8     22Ø          LDA  (CHAR,X)
943Ø: C9 C1     221          CMP  #TO
9432: DØ 2Ø     222          BNE  EXIT       ;kein 'TO' -> Ende
                223 *
9434: 2Ø B1 ØØ  224 CTO      JSR  CHRGET     ;Txtptr. vorrücken
                225 *
                226 * Neue Param. holen und Xn, Yn <--> Xn+1, Yn+1
                227 *
9437: A5 Ø6     228          LDA  X1
9439: 48        229          PHA
943A: A5 Ø9     23Ø          LDA  Y1
943C: 48        231          PHA
943D: 2Ø 58 94  232          JSR  GETVAL
944Ø: A5 Ø6     233          LDA  X1
9442: 85 Ø7     234          STA  X2
9444: A5 Ø9     235          LDA  Y1
9446: 85 E3     236          STA  Y2
9448: 68        237          PLA
9449: 85 Ø9     238          STA  Y1
944B: 68        239          PLA
944C: 85 Ø6     24Ø          STA  X1
                241 *
944E: 2Ø 8A 94  242          JSR  LINPLOT    ;Linie plotten
9451: 4C 2C 94  243          JMP  PLOTTO     ;... zum Anfang
                244 *
9454: AD 54 C0  245 EXIT     LDA  PAGE1
9457: 6Ø        246          RTS             ;-> BASIC
                247 *
                248 * GETVAL holt die X- und Y-Parameter
                249 * aus dem BASIC-Text
                25Ø *
9458: 2Ø F8 E6  251 GETVAL   JSR  GETBYT     ;Byte holen
945B: EØ 5Ø     252          CPX  #8Ø        ;X > 79?
945D: BØ 11     253          BCS  FEHLER
945F: 86 Ø6     254          STX  X1
9461: A9 2C     255          LDA  #','
9463: 2Ø C0 DE  256          JSR  SYNCHR     ;Syntax ok,?
9466: 2Ø F8 E6  257          JSR  GETBYT     ;Byte holen
9469: EØ 3Ø     258          CPX  #48        ;Y > 47?
946B: BØ Ø3     259          BCS  FEHLER
946D: 86 Ø9     26Ø          STX  Y1
946F: 6Ø        261          RTS
                262 *
947Ø: 4C 99 E1  263 FEHLER   JMP  ILQUAN
                264 *
                265 * EXPLOT plottet einen Punkt mit
                266 * den Koordinaten X1, Y1 auf dem
```

```
                    267  * Double-Lores-Schirm
                    268  *
9473: A5 Ø6         269  EXPLOT   LDA  X1          ;X -> AREG
9475: 4A            27Ø          LSR             ;X = X / 2
9476: 9Ø Ø5         271          BCC  PAG2        ;gerade -> Zusatssp.
9478: AC 54 CØ      272          LDY  PAGE1
947B: BØ Ø3         273          BCS  CONT
947D: AC 55 CØ      274  PAG2     LDY  PAGE2
9480: A8            275  CONT     TAY             ;X -> YREG
9481: A5 Ø9         276          LDA  Y1          ;Y -> AREG
9483: 2Ø ØØ F8      277          JSR  PLOT        ;Block plotten
9486: AD 54 CØ      278          LDA  PAGE1
9489: 6Ø            279          RTS
                    280  *
                    281  * LINPLOT plottet eine Linie
                    282  * zwischen zwei Punkten
                    283  *
948A: A9 Ø1         284  LINPLOT  LDA  #1
948C: 85 D7         285          STA  XS          ;STEP X = 1
948E: 85 EB         286          STA  YS          ;STEP Y = 1
                    287  *
9490: 38            288          SEC
9491: A5 Ø7         289          LDA  X2
9493: E5 Ø6         29Ø          SBC  X1
9495: 85 CF         291          STA  XD          ;XD = X2 - X1
                    292  *
9497: 38            293          SEC
9498: A5 E3         294          LDA  Y2
949A: E5 Ø9         295          SBC  Y1
949C: 85 EF         296          STA  YD          ;YD = Y2 - Y1
                    297  *
                    298  * Setzen der Schrittweite für Liniengenerator
                    299  *
949E: 24 CF         3ØØ          BIT  XD
94AØ: 1Ø Ø6         3Ø1          BPL  LAB1
94A2: 24 EF         3Ø2          BIT  YD
94A4: 3Ø Ø8         3Ø3          BMI  CASE1       ;DX < Ø & DY < Ø
94A6: 1Ø Ø9         3Ø4          BPL  CASE2       ;DX < Ø & DY >= Ø
94A8: 24 EF         3Ø5  LAB1     BIT  YD
94AA: 3Ø ØB         3Ø6          BMI  CASE3       ;DX >= Ø & DY < Ø
94AC: 1Ø ØC         3Ø7          BPL  FERTIG      ;DX >= Ø & DY >= Ø
94AE: 2Ø 8D 95      3Ø8  CASE1    JSR  MIYS        ;YS = -1
94B1: 2Ø 9B 95      3Ø9  CASE2    JSR  MIXS        ;XS = -1
94B4: 4C BA 94      31Ø          JMP  FERTIG
94B7: 2Ø 8D 95      311  CASE3    JSR  MIYS        ;YS = -1
                    312  *
94BA: A5 CF         313  FERTIG   LDA  XD
94BC: C5 EF         314          CMP  YD
94BE: 1Ø 47         315          BPL  DDA         ;XD >= YD
                    316  *
                    317  * BRENSENHAM-Algorithmus für YD > XD
                    318  * (Modifizierter Digital-Differential-Analyzer DDA)
                    319  *
94CØ: A5 CF         32Ø          LDA  XD
94C2: ØA            321          ASL
94C3: 38            322          SEC
94C4: E5 EF         323          SBC  YD
94C6: 8D A9 95      324          STA  ETERM       ;ETERM = 2 * YD - XD
94C9: A6 EF         325          LDX  YD
94CB: E8            326          INX              ;Schleifenzähler initialisieren
94CC: CA            327  LOOP2    DEX
94CD: FØ 35         328          BEQ  READY2      ;Linie fertig?
94CF: 2Ø 73 94      329          JSR  EXPLOT
94D2: AD A9 95      33Ø          LDA  ETERM
94D5: 3Ø 19         331          BMI  NOCOR2      ;ETERM >= Ø
```

```
94D7: FØ 17         332          BEQ  NOCOR2      ;ETERM = Ø
94D9: 18            333          CLC
94DA: A5 Ø6         334          LDA  X1
94DC: 65 D7         335          ADC  XS          ;X1 = X1 + XS
94DE: 85 Ø6         336          STA  X1
94EØ: 38            337          SEC
94E1: A5 CF         338          LDA  XD
94E3: E5 EF         339          SBC  YD
94E5: ØA            34Ø          ASL
94E6: 18            341          CLC
94E7: 6D A9 95      342          ADC  ETERM
94EA: 8D A9 95      343          STA  ETERM       ;ETERM = ETERM + 2 * (XD - YD)
94ED: 4C FA 94      344          JMP  LAB3
94FØ: A5 CF         345  NOCOR2   LDA  XD
94F2: ØA            346          ASL
94F3: 18            347          CLC
94F4: 6D A9 95      348          ADC  ETERM
94F7: 8D A9 95      349          STA  ETERM       ;ETERM = ETERM + 2 * XD
94FA: 18            35Ø  LAB3     CLC
94FB: A5 EB         351          LDA  YS
94FD: 65 Ø9         352          ADC  Y1
94FF: 85 Ø9         353          STA  Y1          ;Y1 = Y1 + YS
9501: 4C CC 94      354          JMP  LOOP2
9504: 4C 73 94      355  READY2   JMP  EXPLOT
                    356  *
                    357  * BRENSENHAM-Algorithmus XD >= YD
                    358  * hier ETERM eine 16-Bit-Signed-Number
                    359  *
9507: A5 EF         36Ø  DDA      LDA  YD
9509: 8D AB 95      361          STA  TMP
950C: A9 ØØ         362          LDA  #Ø
950E: 8D AC 95      363          STA  TMP+1
9511: ØE AB 95      364          ASL  TMP
9514: 2E AC 95      365          ROL  TMP+1
9517: 38            366          SEC
9518: AD AB 95      367          LDA  TMP
951B: E5 CF         368          SBC  XD
951D: 8D A9 95      369          STA  ETERM
952Ø: A9 ØØ         37Ø          LDA  #Ø
9522: E9 ØØ         371          SBC  #Ø
9524: 8D AA 95      372          STA  ETERM+1     ;ETERM = 2 * YD - YX
9527: A6 CF         373          LDX  XD          ;XREG = Zähler
9529: E8            374          INX
952A: CA            375  LOOP     DEX
952B: FØ 5D         376          BEQ  READY1      ;Linie fertig
952D: 2Ø 73 94      377          JSR  EXPLOT      ;PLOT (X, Y)
9530: AD AA 95      378          LDA  ETERM+1
9533: 3Ø 39         379          BMI  NOCOR
9535: DØ Ø5         38Ø          BNE  MARK
9537: AD A9 95      381          LDA  ETERM
953A: FØ 32         382          BEQ  NOCOR
953C: 18            383  MARK     CLC
953D: A5 Ø9         384          LDA  Y1
953F: 65 EB         385          ADC  YS
9541: 85 Ø9         386          STA  Y1          ;Y1 = Y1 + 1
9543: 38            387          SEC
9544: A5 EF         388          LDA  YD
9546: E5 CF         389          SBC  XD
9548: 8D AB 95      39Ø          STA  TMP
954B: A9 ØØ         391          LDA  #Ø
954D: E9 ØØ         392          SBC  #Ø
954F: 8D AC 95      393          STA  TMP+1
9552: ØE AB 95      394          ASL  TMP
9555: 2E AC 95      395          ROL  TMP+1
9558: 18            396          CLC
```

```
9559: AD AB 95  397          LDA  TMP
955C: 6D A9 95  398          ADC  ETERM
955F: 8D A9 95  399          STA  ETERM
9562: AD AC 95  400          LDA  TMP+1
9565: 6D AA 95  401          ADC  ETERM+1
9568: 8D AA 95  402          STA  ETERM+1     ;ETERM = ETERM + 2 * (YD - XD)
956B: 4C 80 95  403          JMP  LAB2
956E: A5 EF     404  NOCOR   LDA  YD
9570: 0A        405          ASL
9571: 18        406          CLC
9572: 6D A9 95  407          ADC  ETERM
9575: 8D A9 95  408          STA  ETERM
9578: AD AA 95  409          LDA  ETERM+1
957B: 69 00     410          ADC  #0
957D: 8D AA 95  411          STA  ETERM+1     ;ETERM = ETERM + 2 * DY
9580: 18        412  LAB2    CLC
9581: A5 D7     413          LDA  XS
9583: 65 06     414          ADC  X1
9585: 85 06     415          STA  X1          ;X1 = X1 + XS
9587: 4C 2A 95  416          JMP  LOOP
958A: 4C 73 94  417  READY1  JMP  EXPLOT
                418  *
958D: A9 FF     419  MIYS    LDA  #$FF
958F: 85 EB     420          STA  YS          ;YS = -1
9591: A5 EF     421          LDA  YD
9593: 49 FF     422          EOR  #%11111111
9595: 18        423          CLC
9596: 69 01     424          ADC  #1
9598: 85 EF     425          STA  YD          ;YD = ABS (YD)
959A: 60        426          RTS
                427  *
959B: A9 FF     428  MIXS    LDA  #$FF
959D: 85 D7     429          STA  XS          ;XS = -1
959F: A5 CF     430          LDA  XD
95A1: 49 FF     431          EOR  #%11111111
95A3: 18        432          CLC
95A4: 69 01     433          ADC  #1
95A6: 85 CF     434          STA  XD          ;XD = ABS (XD)
95A8: 60        435          RTS
95A9: 00        436  ETERM   HEX  00          ;Error-Term für DDA
95AA: 00        437          HEX  00
95AB: 00        438  TMP     HEX  00          ;Hilfsspeicher für 16-Bit-Mult.
95AC: 00        439          HEX  00
                440  *
                441  * Adresse für Variable, die den Farbcode
                442  * erhalten wird, feststellen und
                443  * Variablentyp (Real, Integer) bestimmen
                444  *
95AD: A9 00     445  GETADR  LDA  #0
95AF: 85 14     446          STA  SUBFLG      ;Subscript auf 0
95B1: 20 E3 DF  447          JSR  PTRGET      ;Adr. der Variablen
                448  *                       ;feststellen
95B4: 85 CE     449          STA  POINTER
95B6: 84 CF     450          STY  POINTER+1
                451  *
95B8: A5 81     452          LDA  VARNAM      ;ergibt Variablentyp
95BA: 30 0A     453          BMI  INTA1
95BC: A5 82     454          LDA  VARNAM+1
95BE: 30 0C     455          BMI  TYPERR      ;es war ein String!
95C0: A9 01     456          LDA  #1
95C2: 8D FF 95  457          STA  TYPFLG      ;Real-Variable
95C5: 60        458          RTS
                459  *
95C6: A9 02     460  INTA1   LDA  #2          ;Integer-Variable
95C8: 8D FF 95  461          STA  TYPFLG
```

```
95CB: 60        462          RTS
                463  *
95CC: AD 54 C0  464  TYPERR  LDA  PAGE1
95CF: 4C C9 DE  465          JMP  SNERR
                466  *
                467  * Wertzuweisung der BASIC-Variable mit Farbcode
                468  * im A-Register
                469  *
95D2: 48        470  GIVBACK PHA              ;Farbwert retten
95D3: AD FF 95  471          LDA  TYPFLG
95D6: C9 01     472          CMP  #1
95D8: F0 0B     473          BEQ  REAL1       ;Real-Var.
                474  *
95DA: 68        475          PLA
95DB: A0 01     476          LDY  #1
95DD: 91 CE     477          STA  (POINTER),Y ;High-Byte
95DF: A9 00     478          LDA  #0
95E1: 88        479          DEY
95E2: 91 CE     480          STA  (POINTER),Y ;Low-Byte
95E4: 60        481          RTS
                482  *
95E5: 68        483  REAL1   PLA
95E6: A8        484          TAY
95E7: 20 01 E3  485          JSR  SNGFLT      ;YREG -> Float -> FAC
95EA: A0 00     486          LDY  #0
95EC: B9 9D 00  487  REAL2   LDA  FAC,Y       ;FAC -> Var.
95EF: 91 CE     488          STA  (POINTER),Y
95F1: C8        489          INY
95F2: C0 05     490          CPY  #5
95F4: D0 F6     491          BNE  REAL2
95F6: A5 9E     492          LDA  FAC+1
95F8: A0 01     493          LDY  #1
95FA: 29 7F     494          AND  #%01111111 ;SIGN auf +
95FC: 91 CE     495          STA  (POINTER),Y
95FE: 60        496          RTS
                497  *
95FF: 00        498  TYPFLG  HEX  00

742 Bytes
```

DOUBLE.LORES.DEMO

```
10 PRINT CHR$ (21): PRINT CHR$ (4)"BRUN DOUBLE.LORES"
15 REM DOUBLE.LORES.DEMO zeichnet eine Kugel
20 TEXT : GOSUB 85
25 XE = 79:YE = 47:A = XE / 2:B = YE / 2:F1 = 7:F2 = 23
30 FOR C = 1 TO 15: COLOR= C:R = C + C
35 FOR X = A - R TO A + R: IF X < 0 OR X > XE GOTO 60
40 H = R * R - (X - A) * (X - A): IF H < = 0 GOTO 60
45 Y1 = B + SQR (H) - F1:Y2 = B - SQR (H) + F1
50 IF Y1 > 0 AND Y1 < = YE THEN & PLOT X,Y1
55 IF Y2 > 0 AND Y2 < = YE THEN & PLOT X,Y2
60 NEXT X
65 NEXT C
70 GET X$
75 & TEXT : HOME : END
80 REM Umschaltung auf Full-Screen
85 & GR : POKE 49234,0
86 POKE 49236,0: CALL 63538
87 POKE 49237,0: CALL 63538
90 RETURN
```

Applesoft-Interpreter-Erweiterungen für Double-Hires

von Dr. Wolfgang Braun

Die normalen Applesoft-Hires-Routinen für hochauflösende Grafik sind bekanntlich nicht in der Lage, die Fähigkeit der Apple IIe und IIc zur Darstellung doppelt-hochauflösender Grafik mit 560 * 192 Bildpunkten auszunutzen. Die normalen Hires-Befehle können nur Grafiken mit 280 * 192 Bildpunkten erzeugen.

Warum sollte es aber nicht möglich sein, die normalen Hires-Routinen so zu modifizieren, daß sie auch im doppelt-hochauflösenden Modus funktionieren? Da dem Apple die Fähigkeiten zum Zeichnen von Punkten, Geraden und Linienfolgen (Shapes) in farbiger Darstellung von Hause aus bereits mitgegeben sind, müßte man nur das Vorhandene nutzen und ergänzen und dadurch den erforderlichen Programmieraufwand und den Speicherbedarf gering halten.

Die Aufgabe kann allerdings erst dann als befriedigend gelöst gelten, wenn folgende Bedingungen erfüllt sind:

1. Es müssen sämtliche Applesoft-Befehle, die sich auf die normale Hires-Grafik beziehen, auch für die Double-Hires-Grafik verfügbar sein.
2. Die Befehle müssen in ihrer ursprünglichen Form, d.h. ohne Ampersand-Vektor (&), in Applesoft-Programmen angewendet werden können.

Meine Bemühungen resultierten in einem Assembler-Programm von 700 Bytes Länge, wobei davon nur 480 Bytes für die Erweiterungen der normalen Grafikbefehle im Speicher verfügbar sein müssen. Die restlichen 220 Bytes werden nur zur Initialisierung der Erweiterungen benötigt und danach wieder für Applesoft freigegeben. Das im folgenden abgedruckte Assembler-Programm **AS.DOUBLE.HIRES** leistet das Gewünschte, wenn man es mit „BRUN AS.DOUBLE.HIRES“ aufruft, bevor ein Applesoft-Programm eingegeben oder gestartet wird.

Danach meldet sich Applesoft zurück, und für den Benutzer hat sich nichts verändert, außer daß sich in den Hires-Befehlen der zulässige Bereich für die horizontale X-Koordinate von 0-279 auf 0-559 verdoppelt hat. Jedes Programm, das die für die normale Hires-Grafik ausgelegten Befehle benutzt, wird auch jetzt noch laufen, allerdings werden die Bilder auf die halbe Breite zusammengeschrumpft sein. Um die Erweiterungen voll zu nutzen, sollten die Werte der X-Koordinaten verdoppelt werden.

Es müssen jedoch folgende Einschränkungen gemacht werden:
Die bisherige HGR-Seite 2 ist im Double-Hires-Modus prinzipiell nicht mehr nutzbar, weil die Software-Schalter $C054 und $C055 zum Umschalten zwischen den HGR-Seiten 1 (Speicherbereich $2000-$3FFF) und 2 (Speicherbereich $4000-$5FFF) ihre Funktionen geändert haben. Daher ist der Befehl HGR2 nicht mehr verfügbar.
Unter ProDOS sind die Erweiterungen nicht lauffähig, weil sie die Language-Card benutzen und daher mit ProDOS kollidieren würden. Alle Grafik-Programme, die auf die LC zugreifen (z.B. mit in die LC geschobenem DOS) laufen ebenfalls nicht.

1. Die Grundlagen der Erweiterungen

Um das gesteckte Ziel zu erreichen, war es erforderlich, die normalen Hires-Routinen, die sich im Speicherbereich $F3D8-$F772 des Interpreters befinden, zu modifizieren. Dies geht natürlich nicht im ROM, sondern nur im RAM. Daher wird der gesamte Speicherbereich von $D000-$FFFF, der den Interpreter einschließlich der Hires-Routinen und auch das Monitorprogramm enthält, in die LC kopiert und diese danach aktiviert. Jetzt können an den Hires-Routinen Änderungen vorgenommen werden, ohne daß sich, von der Grafikfähigkeit abgesehen, für den Benutzer etwa ändert.
Um die Hires-Routinen zu manipulieren, mußten zunächst einmal diejenigen Stellen ausfindig gemacht werden, an denen der normale Programmablauf unterbrochen und „Umleitungen“ zu Hilfsroutinen im Arbeitsspeicher angebracht werden konnten. Eine sehr gute Darstellung der Funktionsweise der Applesoft-Hires-Routinen (1) erleichterte mir diese Arbeit. Es ergaben sich insgesamt zehn Ansatzpunkte, an denen drei oder mehr Bytes durch je einen JSR- bzw. JMP-Befehl und wenn nötig noch durch andere Befehle ersetzt wurden. Ein JSR-Befehl leitet den Programmablauf aus der normalen Hires-Routine heraus zu einer Hilfsroutine, nach deren Ausführung das Programm wieder in die Hires-Routine zurückkehrt.

Von zentraler Bedeutung für die Realisierung der doppelt-hochauflösenden Grafik ist die Kenntnis der Software-Schalter, mit deren Hilfe der Double-Hires-Modus eingeschaltet werden kann (2, 3). Im normalen Hires-Modus ist der Bildschirm in 40 Spalten unterteilt, und in jeder Spalte können mit Hilfe eines einzelnen Bytes sieben Bits (Bildpunkte) gesetzt werden. Damit ergeben sich 7 * 40 = 280 Bildpunkte in der Horizontalen.

Im Double-Hires-Modus gehören zu jeder der 40 Spalten zwei adreßgleiche Speicherseiten: eine im Hauptspeicher (MAINRAM) und die andere im Hilfsspeicher der 80-Zeichenkarte (Auxiliary-RAM oder AUXRAM). Damit können in jeder der 40 Spalten 14 Bildpunkte gesetzt werden, links die sieben im AUXRAM, rechts daneben die sieben im MAINRAM. Das ergibt 14 * 40 = 560 Bildpunkte in der Horizontalen. Die vertikale Punktanzahl von 192 bleibt unverändert.

Ob ein Punkt in die linke oder in die rechte Hälfte einer der 40 Spalten gesetzt wird, läßt sich mit den bereits erwähnten Soft-

ware-Schaltern bestimmen. Mit z.B. „LDA $C055" wird die AUXRAM-Speicherseite aktiviert und zugleich die MAINRAM-Speicherseite desaktiviert. Wird danach ein Bit im Bildschirmspeicher gesetzt, so erscheint der Punkt in der linken Hälfte der entsprechenden Spalte. Entsprechendes gilt mit „LDA $C054" für die rechte Hälfte der Spalte.

Den normalen Hires-Routinen fehlt nun die Fähigkeit, diese Software-Schalter zu betätigen, weil es letztere zum Zeitpunkt des Erscheinens von Applesoft noch nicht gab. Man kann die Hires-Routinen jedoch dadurch überlisten, daß man sie vor der Übergabe der Koordinaten eines im Double-Hires-Modus zu zeichnenden Punktes unterbricht, die X-Koordinate in den normalen Hires-Modus umrechnet, die richtige Speicherseite selektiert und erst dann die nunmehr „normalen" Koordinaten an die Hires-Routinen zur weiteren Verarbeitung übergibt. Der zu zeichnende Punkt wird dann in der richtigen Spaltenhälfte und an der richtigen Position erscheinen. Die normalen Hires-Routinen „merken" also gar nicht, daß sie im Double-Hires-Modus arbeiten.
Die Transformation der X-Koordinate eines Punktes vom doppelten zum einfachen Hires-Modus kann durch folgenden einfachen Ausdruck, in dem XDIV7 der ganzzahlige Anteil von X, dividiert durch 7, und REST der Rest bei dieser Division (X MOD 7) bedeutet, geschehen:

normale X-Koordinate = 7 * INT (XDIV7 / 2) + REST

Ist XDIV7 geradzahlig, so muß AUXRAM, ist XDIV7 ungeradzahlig, so muß MAINRAM aktiviert werden. Der Wert von REST, eine Zahl zwischen 0 und 6, bestimmt die Position des gesetzten Bits im Speicher-Byte. Im Programm wird allerdings ein etwas kürzerer Algorithmus verwendet.

Etwas komplizierter liegen die Dinge beim Zeichnen von Geraden mit HPLOT TO oder von Shapes mit DRAW AT, weil im ersten Fall nur die Koordinaten der Start- und Zielpunkte und im zweiten Fall sogar nur die Koordinaten des Startpunktes vorgegeben sind und die Koordinaten der dazwischenliegenden bzw. nachfolgenden Punkte nicht.
In beiden Fällen wird zunächst der vorgegebene Startpunkt geplottet. Danach berechnet eine normale HGR-Routine die Spaltennummer S des nächsten zu zeichnenden Punktes. Das Ergebnis ist nur von der Steigung der Geraden abhängig. Hat sich S im Vergleich mit dem zuletzt gezeichneten Punkt um 1 erhöht bzw. erniedrigt, so wird im normalen Hires-Modus der nächste Punkt in die danebenliegende nächste Spalte gezeichnet.
Im Double-Hires-Modus würde dieser Fall jedoch dazu führen, daß der zuletzt gezeichnete Punkt und der als nächstes zu zeichnende Punkt um 7 Bildpunkte auseinanderliegen, denn die eine Hälfte der Spalte wurde wegen der fehlenden Speicherseitenumschaltung übersprungen.
Es muß also jedesmal, bevor der nächste Punkt gezeichnet werden kann, überprüft werden, ob sich dessen Spaltennummer im Vergleich zu der des Vorgängers geändert hat. Für die Differenz DS der Spaltennummern benachbarter Punkte gibt es 5 mögliche Fälle (erste Spalte = $00, letzte Spalte = $27):

DS = $00 – Beide Punkte liegen in derselben Spaltenhälfte.
DS = $01 – Der nächste Punkt liegt in der benachbarten Spaltenhälfte rechts.
DS = $FF – Der nächste Punkt liegt in der benachbarten Spaltenhälfte links.
DS = $D9 – Der nächste Punkt liegt auf dem linken Rand (Wrap-around, Spalte $00, AUXRAM).
DS = $27 – Der nächste Punkt liegt auf dem rechten Rand (Wrap-around, Spalte $27, MAINRAM).

Nur im ersten Fall ist keine Seitenumschaltung erforderlich; in allen anderen Fällen muß vor dem Zeichnen des nächsten Punktes von einer Seite auf die andere umgeschaltet werden. Zusätzlich muß in den Fällen DS = $01 und DS = $FF die durch die Hires-Routinen bewirkte Erhöhung der Spaltennummer genau dann unterdrückt werden, wenn der Übergang von der einen Spaltenhälfte zur anderen innerhalb derselben Spalte stattfindet.

2. Erläuterungen zum Programm

Der Initialisierungsteil reicht von Zeile 48 bis 188. Die eigentliche Erweiterung für die Applesoft-Hires-Routinen beginnt in der Zeile 208.
In den Zeilen 48 bis 66 wird der ROM-Speicherbereich $D000-$FFFF in die LC kopiert und diese aktiviert. In den Zeilen 71 bis 133 werden die JSR-Befehle, die in den Zeilen 163 bis 188 zu finden sind, an den unmittelbar ablesbaren Stellen in die Hires-Routinen eingetragen. Mit den Zeilen 138 bis 141 wird der zulässige Wertebereich der X-Koordinaten auf 0-559 erweitert, und schließlich wird in den Zeilen 146 bis 154 zum Schutz der Erweiterungsroutinen vor Überschreiben durch Stringvariablen HIMEM auf $9421 gesetzt und die 80-Zeichenkarte aktiviert. Damit ist der Speicherbereich des Initialisierungsteils wieder für Applesoft-Programme zugänglich, und die Hilfsroutinen für die Hires-Befehle mit einigen Variablen stehen ab $9421 bereit.

3. Hinweise zu den einzelnen Befehlen

3.1. HGR

Sobald der Interpreter in einem Applesoft-Programm den Befehl HGR erkennt, springt er zu der Adresse $F3E2. Die ersten 4 Bytes dort wurden ersetzt durch die beiden Befehle

```
JSR  PHGR
     RTS.
```

Demnach springt das Programm aus dem Interpreter heraus nach PHGR (Zeile 395 des Assembler-Programms). Dort werden die Software-Schalter für den Double-Hires-Modus bedient und dann zum Löschen des Bildschirmes nacheinander MAINRAM und AUXRAM aktiviert und jeweils mit Hilfe der Bildschirmlöschroutine HCLR ($F3F2) gelöscht. Danach kehrt das Programm in den Interpreter zurück. In der Zeile 395 kann man „LDA MIX" ($C053) durch „LDA $C052" ersetzen, um mit HGR den Bildschirm auf Grafik ohne Text umzuschalten.

Die Anweisungen HCOLOR und TEXT behalten ihre ursprüngliche Bedeutung bei.

3.2. HPLOT X, Y

Nach einem HPLOT-Befehl springt der Interpreter zur Adresse $F6FE. In der Folge würde dann bei Adresse $F705 die normale Routine HPLOT ($F457) zum Zeichnen eines Punktes aufgerufen. Der Befehl „JSR $F457" ist jetzt jedoch ersetzt durch „JSR PPLOT". Daher verzweigt das Programm jetzt nach Zeile 208 mit den Koordinaten des zu zeichnenden Punktes in den Registern X, Y und A. Die nächste Routine XTRANSF transformiert die X-Koordinate von doppelter auf einfache Hires-Grafik, und die nachfolgende Routine PAGER aktiviert dann die richtige Speicherseite. Danach werden die Register – jetzt mit den normalen X-Koordinaten und der unveränderten Y-Koordinate – geladen, und die normale Plot-Routine HPLOT zum Zeichnen eines Punktes wird aufgerufen.

3.3. HPLOT X1, Y1 TO X2, Y2

Die Einsprungadresse ist dieselbe wie oben. Die Routine zum Zeichnen einer Geraden zwischen zwei Punkten, HLINE ($F53A), würde bei $F71B aufgerufen. Hier verzweigt das Programm statt dessen zur Hilfsroutine PHLINE in der Zeile 252, die sich selbst erklärt. Die Routine zur

Berechnung der Spaltennummer des nächsten zu zeichnenden Punktes ist INTX ($F465), die von HLINE in $F57D aufgerufen würde. Dort wird statt dessen zur Hilfsroutine PINTX (Zeile 244) verzweigt, wo die Spaltennummer des zuletzt gezeichneten Punktes abgespeichert und die normale Routine INTX aufgerufen wird. Danach steht die neue Spaltennummer im Y-Register. Die Routine SWITCH vergleicht dann die neue und die alte Spaltennummer, aktiviert die entsprechende Speicherseite und kehrt nach HLINE zurück. Daraufhin wird der nächste Punkt gezeichnet wird.
Die Hilfsroutinen PDECRX (Zeile 223) und PINCRX (Zeile 231) verhindern, daß die letzten 7 Bildpunkte vor dem rechten und linken Rand des Bildschirmes nicht geplottet werden.
Auch der Befehl „HPLOT TO X, Y" arbeitet im Double-Hires-Modus in der gleichen Weise, vorausgesetzt daß vorher ein Startpunkt gezeichnet worden ist.

3.4. DRAW/XDRAW N AT X, Y

Die Einsprungstellen für diese beiden Befehle sind $F769 bzw. $F76F. Die Abläufe sind bei beiden Befehlen bis auf die eigentlichen Zeichenroutinen identisch.
Ab Speicherstelle $F763 steht der Befehl „JSR HPOSN" ($F411). HPOSN setzt an Hand der Koordinaten X, Y die internen Cursor-Daten (Adresse des Bildschirm-Bytes, Spaltennummer sowie die Bit-Position) des Startpunktes. Vor dem Aufruf von HPOSN muß daher die X-Koordinate transformiert und die richtige Speicherseite selektiert werden. Dies übernimmt die Routine PDRAW1 (Zeile 314), die jetzt HPOSN ersetzt. Danach kehrt das Programm wieder nach $F766 in die Hires-Routine zurück.
Bis dahin ist auf dem Bildschirm allerdings noch nichts geschehen. Bei den Befehlen DRAW bzw. XDRAW sind LRUD1/LRUD2 ($F4B3/$F4B4) bzw. LRUDX1/LRUDX2 ($F49C/$F49D) die Routinen, die die Punkte auf dem Bildschirm setzen. Vor dem Setzen des Bildschirm-Bytes ist (wie bei HPLOT TO) die richtige Speicherseite zu selektieren. Dazu werden an den Stellen $F49D bzw. $F4B4 Sprünge zu der Hilfsroutine PDRAW (Zeile 331) eingetragen, wo dann das Erforderliche durch SWITCH veranlaßt wird.

Die Befehle ROT und SCALE bleiben in ihren Funktionen unverändert.

4. Schlußbemerkung

Die erforderliche Hardware ist ein Apple IIe mit erweiterter 80-Zeichenkarte, auf der die Drahtbrücke installiert sein muß, oder ein IIc.
Zur Demonstration der Funktionsfähigkeit der beschriebenen Erweiterungen ist ein Applesoft-Programm **AS.DOUBLE.HIRES.DEMO** abgedruckt, in dem die wesentlichen Hires-Befehle im Double-Hires-Modus aufgerufen werden.

Literatur:

(1) C. K. Mesztenyi, All About Applesoft Nr.1, 1981, Seite 92.
(2) K.-W. Bott, Peeker, Heft 2/84, Seite 24.
(3) U. Stiehl, Apple Assembler, Hüthig-Verlag, 1984.

Kurzhinweise

1. Zweck:
Erweiterung der Applesoft-HGR-Befehle für doppelt-hochauflösende Grafik.
2. Konfiguration:
Apple IIe mit 80-Zeichenkarte oder IIc;
DOS 3.3 (48K, da LC benutzt wird).
3. Test:
RUN AS.DOUBLE.HIRES.DEMO
4. Sammeldisk:
AS.DOUBLE.HIRES.DEMO
(Applesoft-Demoprogramm)
AS.DOUBLE.HIRES
(Maschinenprogramm)
T.AS.DOUBLE.HIRES
(Big-Mac-Quelltext)

```
AS.DOUBLE.HIRES setzt einen Apple IIc oder
Apple IIe mit 64K-Karte und DOS 3.3 voraus.
Kein ProDOS, kein gemovtes DOS 3.3 verwenden!
```

AS.DOUBLE.HIRES.DEMO

```
100 REM AS.DOUBLE.HIRES.DEMO
110 PRINT CHR$ (4);"BRUN AS.DOUBLE.HIRES"
120 HGR : HCOLOR= 3
130 POKE - 16302,0: REM NOMIX
140 REM Shape-Tabelle ab $300 = 768
150 FOR I = 1 TO 57: READ Z: POKE 767 + I,Z: NEXT
160 REM Anfangsadresse der Tabelle in $00E8/$E9
170 POKE 232,0: POKE 233,03
180 SCALE= 1: ROT= 1
190 REM Das Wort 'Text' schreiben
200 FOR I = 2 TO 5
210 DRAW I AT 50 + I * 7,86: NEXT
220 REM Rahmen zeichnen
230 HPLOT 0,0 TO 559,0 TO 559,191
240 HPLOT TO 0,191 TO 0,0
250 REM Bewegung
260 SCALE= 85
270 FOR K = 1 TO 64: ROT= K
280 XDRAW 1 AT 279,96: NEXT
290 REM Mit beliebiger Taste abbrechen
300 IF PEEK (49152) < 127 THEN GOTO 270
310 TEXT : POKE 49168,0
320 DATA 5,0,14,0,16,0,23,0,33,0,48,0,57,0,4,0
330 DATA 45,45,159,54,54,38,0
340 DATA 146,146,32,100,173,62,183,45,4,0
350 DATA 146,146,13,24,14,88,40,40,248,27,21,85,170,4,0
360 DATA 137,54,54,46,12,24,248,4,0
```

AS.DOUBLE.HIRES

```
BSAVE AS.DOUBLE.HIRES, A$9344, L$02BC
```

```
1   ********************************
2   *                              *
3   *       AS.DOUBLE.HIRES        *
4   *                              *
5   * Modifikation der Applesoft-  *
6   * Hires-Routinen für doppelt-  *
7   * hochauflösende Grafik mit    *
8   *   560 * 192 Bildpunkten      *
9   *                              *
10  * von Dr.W. Braun, Juni 1985   *
11  *                              *
12  ********************************
13
14            ORG  $9344
15
16  HIMEM     EQU  $73
17  DX        EQU  $D0
18  QDRNT     EQU  $D3
19  E         EQU  $D4
20  X0        EQU  $E0
21  HNDX      EQU  $E5
22  HPAG      EQU  $E6
23
24  * Software-Schalter
25
26  STORE80   EQU  $C001
27  COL80     EQU  $C00D
28  STAT      EQU  $C01C
29  GRAFIK    EQU  $C050
30  MIX       EQU  $C053
31  MAINRAM   EQU  $C054
32  AUXRAM    EQU  $C055
33  HIRES     EQU  $C057
34  AN3       EQU  $C05E
35
36  * Applesoft-HGR-Routinen
37
38  INTX      EQU  $F465
39  HPLOT     EQU  $F457
40  HCLR      EQU  $F3F2
```

```
                        41  HLINENEU EQU  $F55A
                        42  HPOSN    EQU  $F411
                        43
                        44  * Move-Routinen
                        45
                        46  * ROM $D000-$FFFF in LC moven *
                        47
9344: 8D 89 C0          48           STA  $C089
9347: 8D 89 C0          49           STA  $C089
934A: A9 00             50           LDA  #$00
934C: 85 E0             51           STA  X0
934E: A9 D0             52           LDA  #$D0
9350: 85 E1             53           STA  X0+1
9352: A0 00             54           LDY  #$00
9354: B1 E0             55  NEXT     LDA  (X0),Y
9356: 91 E0             56           STA  (X0),Y
9358: C8                57           INY
9359: D0 F9             58           BNE  NEXT
935B: E6 E1             59           INC  X0+1
935D: A5 E1             60           LDA  X0+1
935F: D0 F3             61           BNE  NEXT
                        62
                        63  * Read/Write LC Bank 1
                        64
9361: 8D 8B C0          65           STA  $C08B
9364: 8D 8B C0          66           STA  $C08B
                        67
                        68  * Hilfsroutinen in die Apple-
                        69  * soft-HGR-Routinen eintragen
                        70
9367: A0 04             71           LDY  #$04       ;PDECRX
9369: B9 F8 93          72  LOOP1    LDA  P1,Y
936C: 99 71 F4          73           STA  $F471,Y
936F: 88                74           DEY
9370: 10 F7             75           BPL  LOOP1
                        76
9372: A0 02             77           LDY  #$02       ;PINCRX
9374: B9 FD 93          78  LOOP2    LDA  P2,Y
9377: 99 93 F4          79           STA  $F493,Y
937A: 88                80           DEY
937B: 10 F7             81           BPL  LOOP2
                        82
937D: A0 02             83           LDY  #$02       ;PINT
937F: B9 00 94          84  LOOP3    LDA  P3,Y
9382: 99 7D F5          85           STA  $F57D,Y
9385: 88                86           DEY
9386: 10 F7             87           BPL  LOOP3
                        88
9388: A0 02             89           LDY  #$02       ;PPLOT
938A: B9 03 94          90  LOOP4    LDA  P4,Y
938D: 99 05 F7          91           STA  $F705,Y
9390: 88                92           DEY
9391: 10 F7             93           BPL  LOOP4
                        94
9393: A0 09             95           LDY  #$09       ;PHLINE
9395: B9 06 94          96  LOOP5    LDA  P5,Y
9398: 99 15 F7          97           STA  $F715,Y
939B: 88                98           DEY
939C: 10 F7             99           BPL  LOOP5
                       100
939E: A9 0C            101           LDA  #$0C       ;RTS
93A0: 8D 0E F7         102           STA  $F70E
                       103
93A3: A0 03            104           LDY  #$03       ;PHGR
93A5: B9 10 94         105  LOOP6    LDA  P6,Y
93A8: 99 E2 F3         106           STA  $F3E2,Y
93AB: 88               107           DEY
93AC: 10 F7            108           BPL  LOOP6
                       109
93AE: A0 02            110           LDY  #$02       ;PDRAW1
93B0: B9 14 94         111  LOOP7    LDA  P7,Y
93B3: 99 63 F7         112           STA  $F763,Y
93B6: 88               113           DEY
93B7: 10 F7            114           BPL  LOOP7
                       115
93B9: A0 03            116           LDY  #$03       ;PDRAW
93BB: B9 17 94         117  LOOP8    LDA  P8,Y
93BE: 99 9D F4         118           STA  $F49D,Y
93C1: 99 B4 F4         119           STA  $F4B4,Y
93C4: 88               120           DEY
93C5: 10 F4            121           BPL  LOOP8
                       122
93C7: A0 02            123           LDY  #$02       ;AUSDRAW
93C9: B9 1B 94         124  LOOP9    LDA  P9,Y
93CC: 99 5A F6         125           STA  $F65A,Y
93CF: 88               126           DEY
93D0: 10 F7            127           BPL  LOOP9
                       128
93D2: A0 02            129           LDY  #$02       ;AUSXDRAW
93D4: B9 1E 94         130  LOOP10   LDA  P10,Y
93D7: 99 B6 F6         131           STA  $F6B6,Y
93DA: 88               132           DEY
```

```
93DB: 10 F7            133           BPL  LOOP10
                       134
                       135  * X-Bereich von (0..279)
                       136  * auf (0..559) erweitern
                       137
93DD: A9 02            138           LDA  #>560
93DF: 8D C4 F6         139           STA  $F6C4
93E2: A9 30            140           LDA  #<560
93E4: 8D CA F6         141           STA  $F6CA
                       142
                       143  * HIMEM zum Schutz der Hilfs-
                       144  * routinen neu setzen
                       145
93E7: A9 21            146           LDA  #<XSTARTL
93E9: 85 73            147           STA  HIMEM
93EB: A9 94            148           LDA  #>XSTARTL
93ED: 85 74            149           STA  HIMEM+1
                       150
                       151  * 80-Zeichenkarte über Outport aktivieren
                       152
93EF: A9 03            153           LDA  #03
93F1: 20 95 FE         154           JSR  $FE95
                       155
                       156  * LC-Schreibschutz
                       157
93F4: 8D 88 C0         158           STA  $C088
93F7: 60               159           RTS
                       160
                       161  * Hilfsroutinen
                       162
93F8: 20 4F 94         163  P1       JSR  PDECRX
93FB: EA               164           NOP
93FC: EA               165           NOP
                       166
93FD: 20 5C 94         167  P2       JSR  PINCRX
                       168
9400: 20 6C 94         169  P3       JSR  PINTX
                       170
9403: 20 2D 94         171  P4       JSR  PPLOT
                       172
9406: 20 76 94         173  P5       JSR  PHLINE
9409: 4C 08 F7         174           JMP  $F708
940C: 8D 54 C0         175           STA  MAINRAM
940F: 60               176           RTS
                       177
9410: 20 72 95         178  P6       JSR  PHGR
9413: 60               179           RTS
                       180
9414: 20 E4 94         181  P7       JSR  PDRAW1
                       182
9417: 20 0B 95         183  P8       JSR  PDRAW
941A: EA               184           NOP
                       185
941B: 4C F0 95         186  P9       JMP  AUSDRAW
                       187
941E: 4C F7 95         188  P10      JMP  AUSXDRAW
                       189
                       190  * Hierher springen die Apple-
                       191  * soft-HGR-Routinen
                       192
9421: 00               193  XSTARTL  HEX  00
9422: 00               194  XSTARTH  HEX  00
9423: 00               195  XZIELL   HEX  00
9424: 00               196  XZIELH   HEX  00
9425: 00               197  XDIV7    HEX  00
9426: 00               198  REST     HEX  00
9427: 00               199  SEITE    HEX  00
9428: 00               200  XL       HEX  00
9429: 00               201  XH       HEX  00
942A: 00               202  YSTART   HEX  00
942B: 00               203  YZIEL    HEX  00
942C: 00               204  YOLD     HEX  00
                       205
                       206  * Plotten eines einzelnen Punktes
                       207
942D: 8E 21 94         208  PPLOT    STX  XSTARTL
9430: 8C 22 94         209           STY  XSTARTH
9433: 8D 2A 94         210           STA  YSTART
9436: 8E 28 94         211  PLOTS    STX  XL
9439: 8C 29 94         212           STY  XH
943C: 20 15 95         213           JSR  XTRANSF
943F: 20 98 95         214           JSR  PAGER
9442: AE 28 94         215           LDX  XL
9445: AC 29 94         216           LDY  XH
9448: AD 2A 94         217           LDA  YSTART
944B: 20 57 F4         218           JSR  HPLOT
944E: 60               219           RTS
                       220
                       221  * Randspalte plotten
                       222
944F: 88               223  PDECRX   DEY
9450: 10 09            224           BPL  LBL14
```

```
9452: A0 FF     225           LDY  #$FF
9454: AD 1C C0  226           LDA  STAT
9457: 10 02     227           BPL  LBL14
9459: A0 27     228           LDY  #$27
945B: 60        229 LBL14     RTS
                230
945C: C8        231 PINCRX    INY
945D: C0 28     232           CPY  #$28
945F: D0 0A     233           BNE  LBL12
9461: 48        234           PHA
9462: AD 1C C0  235           LDA  STAT
9465: 10 03     236           BPL  LBL7
9467: A0 28     237           LDY  #$28
9469: 18        238           CLC
946A: 68        239 LBL7      PLA
946B: 60        240 LBL12     RTS
                241
                242 * Speicherseite schalten
                243
946C: 8C 2C 94  244 PINTX     STY  YOLD
946F: 20 65 F4  245           JSR  INTX
9472: 20 A6 95  246           JSR  SWITCH
9475: 60        247           RTS
                248
                249 * Differenz der X-Koordinaten von
                250 * Start- und Zielpunkt berechnen
                251
9476: 8E 23 94  252 PHLINE    STX  XZIELL
9479: 8C 24 94  253           STY  XZIELH
947C: 8D 2B 94  254           STA  YZIEL
947F: 8E 28 94  255           STX  XL
9482: 8C 29 94  256           STY  XH
9485: 38        257           SEC
9486: AD 23 94  258           LDA  XZIELL
9489: ED 21 94  259           SBC  XSTARTL
948C: 48        260           PHA
948D: AD 24 94  261           LDA  XZIELH
9490: ED 22 94  262           SBC  XSTARTH
9493: 85 D3     263           STA  QDRNT
9495: B0 0A     264           BCS  LBL20
9497: 68        265           PLA
9498: 49 FF     266           EOR  #$FF
949A: 69 01     267           ADC  #$01
949C: 48        268           PHA
949D: A9 00     269           LDA  #$00
949F: E5 D3     270           SBC  QDRNT
94A1: 85 D1     271 LBL20     STA  DX+1
94A3: 85 D5     272           STA  E+1
94A5: 68        273           PLA
94A6: 85 D0     274           STA  DX
94A8: 85 D4     275           STA  E
                276
                277 * Startpunkt zeichnen
                278
94AA: AE 21 94  279           LDX  XSTARTL
94AD: AC 22 94  280           LDY  XSTARTH
94B0: AD 2A 94  281           LDA  YSTART
94B3: 20 36 94  282           JSR  PLOTS
                283
                284 * X-Koordinaten des Ziel-
                285 * punktes transformieren
                286
94B6: AD 23 94  287           LDA  XZIELL
94B9: AE 24 94  288           LDX  XZIELH
94BC: 8D 28 94  289           STA  XL
94BF: 8E 29 94  290           STX  XH
94C2: 20 15 95  291           JSR  XTRANSF
                292
                293 * Zielpunkt -> Startpunkt
                294 * für nächstes TO
                295
94C5: AD 23 94  296           LDA  XZIELL
94C8: AE 24 94  297           LDX  XZIELH
94CB: AC 2B 94  298           LDY  YZIEL
94CE: 8C 2A 94  299           STY  YSTART
94D1: 8D 21 94  300           STA  XSTARTL
94D4: 8E 22 94  301           STX  XSTARTH
                302
                303 * Register laden
                304 * und Linie zeichnen
                305
94D7: AD 28 94  306           LDA  XL
94DA: AE 29 94  307           LDX  XH
94DD: AC 2B 94  308           LDY  YZIEL
94E0: 20 5A F5  309           JSR  HLINENEU
94E3: 60        310           RTS
                311
                312 * Interne Cursordaten für DRAW setzen
                313
94E4: 8E 21 94  314 PDRAW1    STX  XSTARTL
94E7: 8C 22 94  315           STY  XSTARTH
94EA: 8E 28 94  316           STX  XL
```

```
94ED: 8C 29 94  317           STY  XH
94F0: 8D 2A 94  318           STA  YSTART
94F3: 20 15 95  319           JSR  XTRANSF
94F6: 20 98 95  320           JSR  PAGER
94F9: AE 28 94  321           LDX  XL
94FC: AC 29 94  322           LDY  XH
94FF: AD 2A 94  323           LDA  YSTART
9502: 20 11 F4  324           JSR  HPOSN
9505: A5 E5     325           LDA  HNDX
9507: 8D 2C 94  326           STA  YOLD
950A: 60        327           RTS
                328
                329 * Speicherumschaltung bei DRAW, XDRAW
                330
950B: 08        331 PDRAW     PHP
950C: 20 A6 95  332           JSR  SWITCH
950F: 28        333           PLP
9510: A5 D1     334           LDA  DX+1
9512: 29 04     335           AND  #$04
9514: 60        336           RTS
                337
                338 * X-Koordinate auf einfache
                339 * HIRES-Grafik transformieren
                340 *
                341 * Nach XTRANSF befindet sich
                342 * die transformierte X-Koor-
                343 * dinate in XL/XH
                344
9515: A0 00     345 XTRANSF   LDY  #$00
9517: 8C 27 94  346           STY  SEITE
951A: 38        347           SEC
951B: AD 28 94  348 LOOP30    LDA  XL
951E: E9 07     349           SBC  #$07
9520: 8D 28 94  350           STA  XL
9523: AD 29 94  351           LDA  XH
9526: E9 00     352           SBC  #$00
9528: 8D 29 94  353           STA  XH
952B: C8        354           INY
952C: F0 03     355           BEQ  LBL6
952E: B0 EB     356           BCS  LOOP30
9530: 88        357           DEY
9531: 8C 25 94  358 LBL6      STY  XDIV7
9534: 18        359           CLC
9535: AD 28 94  360           LDA  XL
9538: 69 07     361           ADC  #$07
953A: 8D 26 94  362           STA  REST
                363
                364 * XL = (XKOOR1 + REST) * 0,5
                365 * oder
                366 * XL = (XKOOR1 + REST - 6) * 0,5
                367
953D: AD 25 94  368           LDA  XDIV7
9540: 4A        369           LSR
9541: 2E 27 94  370           ROL  SEITE
9544: AD 26 94  371           LDA  REST
9547: 18        372           CLC
9548: 6D 21 94  373           ADC  XSTARTL
954B: 8D 28 94  374           STA  XL
954E: A9 00     375           LDA  #$00
9550: 6D 22 94  376           ADC  XSTARTH
9553: 8D 29 94  377           STA  XH
9556: AC 27 94  378           LDY  SEITE
9559: F0 0E     379           BEQ  NULL
955B: 38        380           SEC
955C: AD 28 94  381           LDA  XL
955F: E9 06     382           SBC  #$06
9561: 8D 28 94  383           STA  XL
9564: AD 29 94  384           LDA  XH
9567: E9 00     385           SBC  #$00
9569: 18        386 NULL      CLC
956A: 4A        387           LSR
956B: 8D 29 94  388           STA  XH
956E: 6E 28 94  389           ROR  XL
9571: 60        390           RTS
                391
                392 * Simulation des HGR-Befehles für
                393 * doppelt-hochauflösende Grafik
                394
9572: AD 53 C0  395 PHGR      LDA  MIX
9575: AD 57 C0  396           LDA  HIRES
9578: AD 50 C0  397           LDA  GRAFIK
957B: 8D 01 C0  398           STA  STORE80
957E: 8D 0D C0  399           STA  COL80
9581: AD 5E C0  400           LDA  AN3
9584: A9 20     401           LDA  #$20
9586: 85 E6     402           STA  HPAG
9588: 8D 54 C0  403           STA  MAINRAM
958B: 20 F2 F3  404           JSR  HCLR
958E: 8D 55 C0  405           STA  AUXRAM
9591: 20 F2 F3  406           JSR  HCLR
9594: 8D 54 C0  407           STA  MAINRAM
9597: 60        408           RTS
```

```
                    409
                    410  * Speicherumschaltung für das
                    411  * Plotten eines Punktes
                    412
9598: AD 27 94      413  PAGER    LDA  SEITE
959B: F0 05         414           BEQ  LBL1
959D: 8D 54 C0      415           STA  MAINRAM
95A0: D0 03         416           BNE  LBL2
95A2: 8D 55 C0      417  LBL1     STA  AUXRAM
95A5: 60            418  LBL2     RTS
                    419
                    420  * Speicherumschaltung für das
                    421  * Zeichnen von Geraden mit
                    422  * HPLOT, DRAW oder XDRAW
                    423
95A6: 98            424  SWITCH   TYA
95A7: 38            425           SEC
95A8: ED 2C 94      426           SBC  YOLD
95AB: C9 00         427           CMP  #$00
95AD: F0 40         428           BEQ  LBL3
95AF: C9 01         429           CMP  #$01
95B1: D0 14         430           BNE  LBL4
95B3: AD 1C C0      431           LDA  STAT       ;DY = 1
95B6: 10 09         432           BPL  LBL5
95B8: AC 2C 94      433           LDY  YOLD       ;AUXRAM on
95BB: 8C 54 C0      434           STY  MAINRAM
95BE: 4C EC 95      435           JMP  LBL10
95C1: 8D 55 C0      436  LBL5     STA  AUXRAM
95C4: 4C EC 95      437           JMP  LBL10
```

```
95C7: C9 FF         438  LBL4     CMP  #$FF
95C9: D0 14         439           BNE  LBL8
95CB: AD 1C C0      440           LDA  STAT       ;DY = -1
95CE: 30 09         441           BMI  LBL9
95D0: AC 2C 94      442           LDY  YOLD
95D3: 8D 55 C0      443           STA  AUXRAM
95D6: 4C EC 95      444           JMP  LBL10
95D9: 8D 54 C0      445  LBL9     STA  MAINRAM
95DC: 4C EC 95      446           JMP  LBL10
95DF: C9 D9         447  LBL8     CMP  #$D9
95E1: D0 06         448           BNE  LBL11
95E3: 8D 55 C0      449           STA  AUXRAM     ;Spalte $27->$0
95E6: 4C EC 95      450           JMP  LBL10
95E9: 8D 54 C0      451  LBL11    STA  MAINRAM    ;Spalte $0->$27
95EC: 8C 2C 94      452  LBL10    STY  YOLD
95EF: 60            453  LBL3     RTS
                    454
                    455  * Rücksprung in Interpreter
                    456
95F0: F0 03         457  AUSDRAW  BEQ  LB1
95F2: 4C 26 F6      458           JMP  $F626
95F5: F0 05         459  LB1      BEQ  LB2
                    460
95F7: F0 03         461  AUSXDRAW BEQ  LB2
95F9: 4C 82 F6      462           JMP  $F682
95FC: 8D 54 C0      463  LB2      STA  MAINRAM
95FF: 60            464           RTS

700 Bytes
```