

Ein schneller »Drawline«-Algorithmus

Im folgenden wird eine Möglichkeit vorgestellt, schnell und einfach eine Strecke, die durch ihre beiden Endpunkte gegeben ist, zu plotten. Als Ausgabegerät können Bildschirm, Drucker und Plotter eingesetzt werden.

Der Algorithmus wird in einer Basic- (Listing 1) und einer Assembler-Version (Listing 2) für einen 6510-Mikroprozessor mit den Adressen für den Commodore 64 beschrieben. Die Programme können, da sie im Aufbau einfach sind und erklärt werden, ohne große Probleme für andere Systeme beziehungsweise andere Sprachen umgewandelt werden. Das Maschinenprogramm (Listing 3) geben Sie bitte mit dem MSE ein.

Ein Rasterbildschirm setzt sich aus einzelnen Punkten, die gesetzt oder auch gelöscht sein können, zusammen. Der Abstand der Punkte voneinander ist in der Richtung der Achsen immer gleich und jeweils eine Schrittweite groß (Bild 1).

Name	Adressen						Beschreibung
	2001	3032	8032	VC20	C 64	610/710	
CHKCOM	CE11	CDF8	BEF5	CEFD	AEFD	9730	Prüfe ob nächstes Zeichen im Basic-Text ein Komma ist, wenn nicht gebe »SYNTAX ERROR« aus
GETCOR	D6C4	D6C6	C921	D7EB	B7EB	B4E5	Holt die Koordinaten eines Punktes aus dem Basic-Text. Die Routine wertet auch Ausdrücke aus. Die X-Koordinate wird als 2-Byte-Wert in X0, die Y-Koordinate als Byte im X-Register übergeben.
X0	0066	0014	0014	0014	0014	0011	Hier wird die X-Koordinate von GETCOR abgelegt.
PLOT	—	—	—	—	—	—	Diese Routine ist keine Betriebssystemroutine. Sie übernimmt die Koordinaten von X0 und dem X-Register.

Tabelle 1. Benützte Subroutinen und Übergabe-speicherzellen

Name	Beschreibung
X0, Y0	Koordinaten des ersten Punktes
X1, Y1	Koordinaten des zweiten Punktes
CT	Schleifenzähler
IX	Inkrement oder Dekrement für X0 (-1,0,+1)
IY	Inkrement oder Dekrement für Y0 (-1,0,+1)
AX	wie IX für Steigungen > 1
AY	wie IY für Steigungen > 1
DX	Entfernung der Punkte entlang der X-Achse (= Anzahl der Punkte)
DY	Entfernung der Punkte entlang der Y-Achse
OF	Approximationsvariable zur Bestimmung ob Y0 gleichbleibt

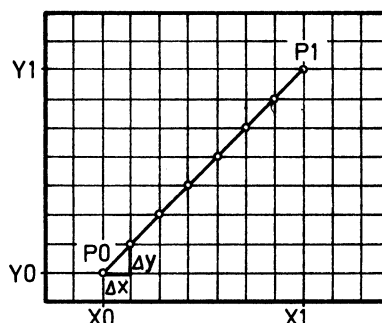
Tabelle 2. Die verwendeten Variablen

Um eine Gerade zwischen den Punkten P0 und P1 zu ziehen, muß daher schrittweise berechnet werden, welcher Punkt der Ideallinie (Bild 2) am nächsten ist und daher gesetzt werden muß.

Geschwindigkeitsvorteile durch einfache Berechnungen

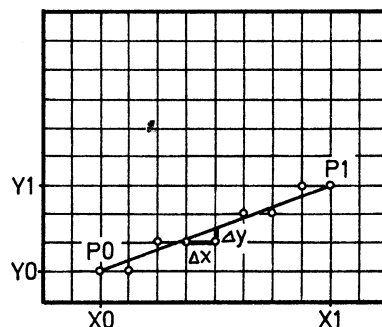
Für diese Berechnungen gibt es verschiedene Möglichkeiten, doch sind sie meistens mit Multiplikationen und Divisionen in der Approximationsschleife verbunden und daher weder schnell noch einfach zu programmieren. Der hier vorgestellte Algorithmus verwendet dagegen nur eine Division und in der Schleife nur mehr Addition, Subtraktion und eine Vergleichsoperation. Da die Schleifenoperationen außerdem nur mehr an Integerzahlen durchzuführen sind, ist er besonders schnell, und er läßt sich auch einfach programmieren. Zur Erklärung soll eine Gerade mit einer Steigung zwischen 0 und 1 (0 bis 45 Grad) betrachtet werden.

Wie man in Bild 2 unschwer erkennen kann, ist der Abstand der Punkte P0 und P1 entlang der X-Achse gleich der Anzahl der zu setzenden Punkte. $DX = X1 - X0$, das heißt, es sind DX-Approximationen durchzuführen, um die Gerade zu zeichnen. Für jeden folgenden Punkt ist als X0 und IX (= 1) zu erhöhen, während Y0 gleichbleibt (IY = 0) oder ebenfalls um 1 (IY = 1) erhöht wird. Der Abstand der beiden Punkte entlang der Y-Achse ist demnach: $DY = Y1 - Y0$. Es bleibt also nur mehr festzustellen, wann IY = 0, beziehungsweise IY = 1 zu sein hat. Dazu wird vor Beginn der Schleife ein Approxima-



$$\begin{aligned} DX &= X1 - X0 \\ DY &= Y1 - Y0 \\ DX &= DY \\ \Delta x &= 1 \\ IX &= \Delta x \\ \Delta y &= 1 \\ IY &= \Delta y \end{aligned}$$

Bild 1. Eine Strecke mit der Steigung 1 dargestellt auf dem Bildschirm



$$\begin{aligned} DX &= X1 - X0 \\ DY &= Y1 - Y0 \\ DX &= DY \\ \Delta x &= 1 \\ IX &= \Delta x \\ 0 \leq \Delta y \leq 1 \\ IY &= 0 \text{ oder } 1 \end{aligned}$$

Bild 2. Eine Strecke mit der Steigung 1 und den optimierten Rasterpunkten

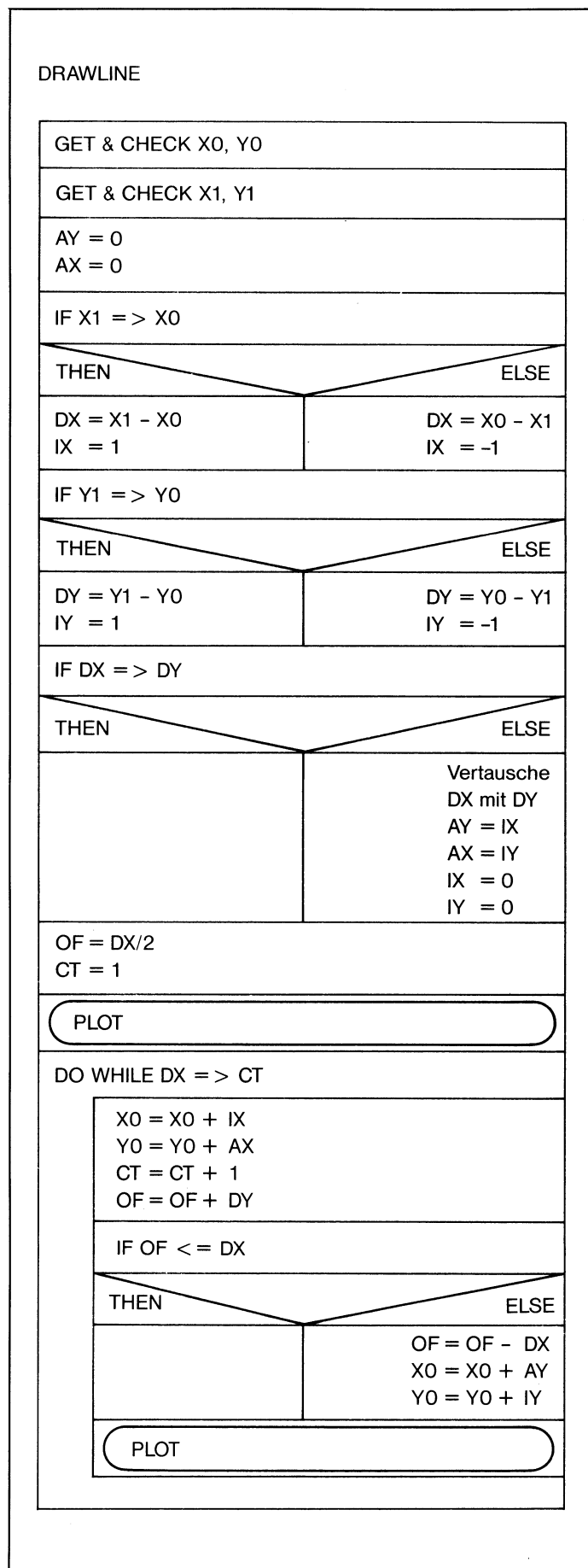


Bild 3. Nassi-Shneiderman-Diagramm des Programms »Draw«. Durch dieses Programm kann sehr schnell eine Strecke, die durch ihre beiden Endpunkte gegeben ist, berechnet werden.

tionswert OF berechnet. Da beim Idealfall für eine Steigung von 1 (Bild 1) $IY = IX$, das heißt, immer 1 ist, und $DX = DY$ ist, wird $OF = DX/2$.

Zu OF wird für jeden neuen Punkt DY addiert. Solange OF kleiner als DX bleibt, bleibt $IY = 0$, wird OF gleich oder größer, so wird $IY = 1$, das heißt, Y0 um 1 erhöht. Damit diese Abfrage auch für die folgenden Punkte möglich ist, muß OF um DX vermindert werden.

Rasterpunkte optimieren

Wenn DX-Punkte gesetzt worden sind, sind alle Punkte der Geraden berechnet und die Schleife kann verlassen werden.

Soweit also für eine Gerade mit positiver Steigung zwischen 0 und 45 Grad. Wie sieht es aber aus, wenn zum Beispiel X1 kleiner X0 ist? Nun, dann wird eben $IX = -1$, X0 also für jeden Approximationsschritt um 1 erniedrigt. Und wie sieht es für eine Steigung größer 1 aus? Auch dieses Problem läßt sich einfach lösen. Es werden für die Rechnung einfach die beiden Achsen vertauscht.

Der Ablauf beider Programme ist im Nassi-Shneiderman-Diagramm (Bild 3) dargestellt. Zum leichteren Vergleich sind alle Variablenbezeichnungen in den Programmen identisch.

Die Basic-Version gibt nur die Koordinaten der berechneten Punkte aus, da hierfür keine »Setze-Punkt«-Routine geschrieben wurde. Man kann aber die Wirkungsweise des Algorithmus schön verfolgen.

Das Assemblerprogramm ist, wie schon oben gesagt, für den C 64 geschrieben mit einer möglichen Auflösung von 320 x 200 Punkten. Dadurch können die Werte für Y in einem Byte untergebracht werden, während für X zwei Byte benötigt werden. OF, DX und der Schleifenzähler CT benötigen deshalb ebenfalls 2 Byte. Die Länge der Werte muß beim Umstricken für ein anderes System berücksichtigt werden, da alle durchzuführenden Operationen dementsprechend 1 oder 2 Byte lang sind.

In dieser Assemblerversion werden nur 2 Subroutinen aus dem Betriebssystem verwendet. Sie sind in Tabelle 1 beschrieben. Zusätzlich sind die Adressen für die anderen Commodore Computer angegeben. Die Subroutine »PLOT« muß, wenn sie sich im Betriebssystem wie bei den Commodoresystemen nicht findet, extra geschrieben werden. Für den C 64 findet man im 64'er geeignete Programme. Die Routine CHKCOM überprüft, ob das nächste Zeichen des Basic-Textes (oder Eingabepuffers) ein Komma ist. Ist dies nicht der Fall, kehrt das Unterprogramm nicht zurück, sondern gibt »syntax error« aus und bricht das Programm ab. Beim Vorfinden eines Kommas wird das dem Komma folgende Zeichen in den Akku geholt. Ist dieses Zeichen eine Ziffer, dann wird übrigens das Carryflag auf »1« gesetzt. GETCOR holt zuerst eine Integerzahl (Übergabe in \$14 und \$15), prüft anschließend auf Komma, und holt dann das folgende Byte aus dem Text. Die Übergabe erfolgt im X-Register. Diese Routine ist wie geschaffen für die Eingabe von Koordinaten, da auch Formeln und Variablenwerte übernommen werden.

Kein Problem: Variablen

In Tabelle 2 sind alle verwendeten Variablen aufgelistet und beschrieben. Aufgerufen wird diese Version mit SYS aaaa,X0,Y0,X1,Y1, wobei aaaa die Startadresse der Routine, X0/Y0 und X1/Y1 die Koordinaten der beiden Punkte sind. Für die Koordinaten können auch Ausdrücke verwendet werden, da die Betriebssystemroutine »GETCOR« auch Ausdrücke auswertet.

(Michael Bauer/og)

```

100 REM          DRAWLINE          <006>
110 REM DIESE ROUTINE BERECHNET DIE KOORDI
    NATEN          <151>
120 REM DER PUNKTE AUF EINER STRECKE, DIE
    DURCH          <124>
130 REM DIE ENDPUNKTE GEGEBEN IST. MIT EIN
    ER GE-         <028>
140 REM EIGNETEN 'SETPOINT'-ROUTINE KANN D
    ER             <098>
150 REM BILDSCHIRM ODER EIN PLOTTER ANGEST
    EUERT          <184>
160 REM WERDEN. DIE GRENZEN DER WERTE FUER
    X UND          <093>
170 REM Y ENTSPRECHEN DEN WERTEN FUER DEN
    HI-RES        <165>
180 REM BILDSCHIRM DES COMMODORE 64.      <165>
190 REM          <252>
200 REM DAS ORIGINAL DIESES PROGRAMMES STA
    MMT VON        <131>
210 REM MIKE HIGGINS ERSCHIENEN IN BYTE 19
    81 IM          <129>
220 REM HEFT 8 AUF DEN SEITEN 414 - 416   <084>
230 REM          <036>
240 REM          <046>
250 INPUT"KOORDINATEN 1. PUNKT";X0,Y0    <024>
260 IF X0>319 OR Y0>199 THEN 250         <158>
270 IF X0<0 OR Y0<0 THEN 250             <010>
280 INPUT"KOORDINATEN 2. PUNKT";X1,Y1    <203>
290 IF X1>319 OR Y1>199 THEN 280         <224>
300 IF X1<0 OR Y1<0 THEN 280             <251>
310 REM SETPAR          <244>
320 S1=0:S2=1:A1=1:A2=0                  <070>
330 IF X1=>X0 THEN D1=X1-X0:GOTO 360      <158>
340 A1=-1                                <208>
350 D1=X0-X1                             <203>

360 IF Y1=>Y0 THEN D2=Y1-Y0:GOTO 390      <206>
370 D2=Y0-Y1                             <008>
380 S2=-1                                 <072>
390 IF D1=>D2 THEN 470                    <220>
400 N1=D1:REM VERTAUSCHE D1 UND D2        <225>
410 D1=D2                                  <117>
420 D2=N1                                  <136>
430 S1=A1                                  <037>
440 A1=0                                    <121>
450 A2=S2                                  <123>
460 S2=0                                    <221>
470 D3=D1/2                                <137>
480 N1=1:REM SCHLEIFENZAehler            <014>
490 GOTO 580:REM PLOTTE URSPRUNGSPUNKT    <057>
500 X0=X0+A1                                <018>
510 Y0=Y0+A2                                <066>
520 D3=D3+D2                                <050>
530 N1=N1+1                                <158>
540 IF D3<=D1 THEN 580                    <255>
550 D3=D3-D1                                <206>
560 X0=X0+S1                                <098>
570 Y0=Y0+S2                                <146>
580 PRINT X0,Y0                            <102>
590 IF D1=>N1 THEN 500                    <118>
600 END                                    <094>

```

Listing 1. Das Basic-Programm »Drawline«

PROGRAMM : DRAW C000 C34B

```

C000 : 4C 2D C0 4C 44 C0 4C 5B CF
C008 : C0 4C 78 C0 4C 92 C0 4C 19
C010 : 95 C0 4C 80 C1 4C 83 C1 38
C018 : 4C A8 C1 4C A8 C1 4C CE 4A
C020 : C1 4C D1 C1 4C 60 C1 4C 1B
C028 : 48 C1 4C 70 C1 AD 11 D0 E1
C030 : 8D 5B C3 AD 18 D0 8D 5C 08
C038 : C3 A9 3B 8D 11 D0 A9 18 BF
C040 : 8D 18 D0 60 A0 00 A9 20 0A
C048 : 84 FD 85 FE 98 91 FD C8 AC
C050 : D0 FB E6 FE A5 FE C9 40 B1
C058 : D0 F2 60 20 FD AE 20 9E D0
C060 : B7 A0 00 A9 04 84 FD 85 04
C068 : FE 8A 91 FD C8 D0 FB E6 A0
C070 : FE A5 FE C9 08 D0 F2 60 CD
C078 : A0 00 A9 20 84 FD 85 FE D3
C080 : B1 FD 49 FF 91 FD C8 D0 50
C088 : F7 E6 FE A5 FE C9 40 D0 47
C090 : EF 60 A9 00 2C A9 80 85 37
C098 : 97 20 FC C0 80 F3 8A 4A 00
C0A0 : 4A 4A A8 B9 2F C1 8D 57 56
C0A8 : C3 B9 16 C1 8D 5B C3 8A C5
C0B0 : 29 07 18 6D 57 C3 8D 57 89
C0B8 : C3 A5 14 29 F8 8D 59 C3 61
C0C0 : 18 A9 00 6D 57 C3 85 FD 00
C0C8 : A9 20 6D 58 C3 85 FE 18 7C
C0D0 : A5 FD 6D 59 C3 85 FD A5 A6
C0D8 : FE 65 15 85 FE A5 14 29 3F
C0E0 : 07 49 07 AA A9 01 CA 30 D1
C0E8 : 03 0A D0 FA A0 00 24 97 4D
C0F0 : 10 05 49 FF 31 FD 2C 11 AB
C0F8 : FD 91 FD 60 20 FD AE 20 36
C100 : EB 87 E0 C8 80 0C A5 15 44
C108 : C9 01 90 07 D0 04 A4 14 3F
C110 : C0 40 60 A4 14 60 00 01 E3
C118 : 02 03 05 06 07 08 0A 0B 0D
C120 : 0C 0D 0F 10 11 12 14 15 95
C128 : 16 17 19 1A 1B 1C 1E 00 5F
C130 : 40 80 C0 00 40 80 C0 00 EC
C138 : 40 80 C0 00 40 80 C0 00 F4
C140 : 40 80 C0 00 40 80 C0 00 FC

C148 : 20 FD AE 20 D4 E1 A2 00 FD
C150 : A0 40 A9 00 85 FD A9 20 AA
C158 : 85 FE A9 FD 20 D8 FF 60 10
C160 : 20 FD AE 20 D4 E1 A9 61 F4
C168 : 85 B9 A9 00 20 D5 FF 60 A6
C170 : AD 5B C3 8D 11 D0 AD 5C 74
C178 : C3 8D 18 D0 20 44 E5 60 9E
C180 : A9 00 2C A9 80 85 97 20 3C
C188 : FD AE 20 9E 87 E0 C8 80 BF
C190 : EE 8E 4B C3 86 FD A0 3F 6A
C198 : 8C 4C C3 A0 01 8C 4D C3 80
C1A0 : 88 84 15 84 14 4C FA C1 53
C1A8 : A9 00 2C A9 80 85 97 20 64
C1B0 : FD AE 20 8A AD 20 F7 87 89
C1B8 : 20 08 C1 80 C2 80 4D C3 88
C1C0 : 8C 4C C3 A2 C7 86 FD A0 A1
C1C8 : 00 8C 4B C3 F0 D7 A9 00 CE
C1D0 : 2C A9 80 85 97 20 FC C0 92
C1D8 : 80 E1 8E 4B C3 8D 52 C3 FF
C1E0 : 8C 51 C3 20 FC C0 80 D3 4A
C1E8 : 86 FD 8C 4C C3 AC 51 C3 88
C1F0 : 84 14 8D 4D C3 AD 52 C3 06
C1F8 : 85 15 A0 01 8C 56 C3 8C F4
C200 : 53 C3 8C 51 C3 88 8C 52 D9
C208 : C3 8C 55 C3 8C 54 C3 88 68
C210 : AD 4D C3 C5 15 90 09 D0 A9
C218 : 1E AD 4C C3 C5 14 80 17 86
C220 : 38 A5 14 ED 4C C3 8D 4F A5
C228 : C3 A5 15 ED 4D C3 8D 50 88
C230 : C3 8C 56 C3 4C 48 C2 38 CA
C238 : AD 4C C3 E5 14 8D 4F C3 2B
C240 : AD 4D C3 E5 15 8D 50 C3 C8
C248 : A5 FD CD 4B C3 80 0F 38 37
C250 : AD 4B C3 E5 FD 8D 4E C3 5D
C258 : 8C 53 C3 4C 64 C2 ED 4B B3
C260 : C3 8D 4E C3 AD 50 C3 D0 04
C268 : 24 AD 4F C3 CD 4E C3 80 6F
C270 : 1C AE 4E C3 8D 4E C3 8E 67
C278 : 4F C3 AD 56 C3 8D 54 C3 60
C280 : AD 53 C3 8D 55 C3 C8 8C 29
C288 : 56 C3 8C 53 C3 AD 50 C3 C0

C290 : 4A 8D 4D C3 AD 4F C3 6A A6
C298 : 8D 4C C3 4C 31 C3 AD 56 5A
C2A0 : C3 30 0E 18 65 14 85 14 37
C2A8 : A5 15 69 00 85 15 4C BE E2
C2B0 : C2 38 A5 14 E9 01 85 14 5F
C2B8 : A5 15 E9 00 85 15 18 AD 1F
C2C0 : 4B C3 6D 55 C3 8D 4B C3 50
C2C8 : 18 AD 4C C3 6D 4E C3 8D 86
C2D0 : 4C C3 AD 4D C3 69 00 8D 85
C2D8 : 4D C3 EE 51 C3 D0 03 EE 99
C2E0 : 52 C3 AD 4D C3 CD 50 C3 9C
C2E8 : 90 47 D0 08 AD 4F C3 CD 51
C2F0 : 4C C3 80 3D 38 AD 4C C3 9B
C2F8 : ED 4F C3 8D 4C C3 AD 4D 64
C300 : C3 ED 50 C3 8D 4D C3 AD F4
C308 : 54 C3 30 0E 18 65 14 85 14
C310 : 14 A5 15 69 00 85 15 4C 82
C318 : 27 C3 38 A5 14 E9 01 85 83
C320 : 14 A5 15 E9 00 85 15 18 3A
C328 : AD 4B C3 6D 53 C3 8D 4B 39
C330 : C3 AE 4B C3 20 9E C0 AD EB
C338 : 52 C3 CD 50 C3 90 09 AD 29
C340 : 4F C3 CD 51 C3 80 01 60 95
C348 : 4C 9E C2 FF 00 FF 00 FF 94

```

Listing 3. Das Maschinen-Programm »Draw« geben Sie bitte mit dem MSE ein.

212 zeilen 37 symbole

```

0000      1 ;algorithmus basiert auf einen artikel von mike
0000      2 ;higgins erschienen in der byte august 1981 s.414-416
0000      3 ;
0000      4 ;das programm fuer den cbm 64 wurde von
0000      5 ;michael bauer aindorferstr. 86 8000 muenchen 21
0000      6 ;geschrieben.
0000      7 ;
0014      8 x0 = $14      ;x-koordinate
00fd      9 y1 = $fd      ;y-koordinate
aefd     10 chkcom = $aefd ;prueft ob ein komma folgt,
0000     11 ;            wenn ja, holt das naechste
0000     12 ;            zeichen, wenn nein -> syntax
0000     13 ;            error
b7eb     14 getcor = $b7eb ;holt eine adresse (= 2 byte)
0000     15 ;            und ein byte. die adresse wird
0000     16 ;            in x0 und x0+1, das byte im
0000     17 ;            x-register uebergeben.
0000     18 plot = $0000    ;die adresse dieser routine
0000     19 ;            muss hier eingesetzt werden.
0000     20 ;            plot muss in der lage sein
0000     21 ;            einen punkt, dessen x und y
0000     22 ;            koordinaten uebergeben werden,
0000     23 ;            zu setzen.
0000     24 ;
c178     25      * = $c178
c178     26 ;
c178 60   27 ill rts
c179     28 ;***** einsprung: aufruf sys49529,x0,y0,x1,y1
c179 20efc2 29 drawlin jsr gcoord ;hole koordinaten 1. punkt
c17c b0fa   30 bcs ill      ;ignoriere wert wenn ausserhalb
c17e 8e09c3 31 stx y0
c181 8d10c3 32 sta ct+1
c184 8c0fc3 33 sty ct
c187 20efc2 34 jsr gcoord ;hole koordinaten 2. punkt
c18a b0ec   35 bcs ill      ;ignoriere wert wenn ausserhalb
c18c 86fd   36 stx y1
c18e 8c0ac3 37 sty x1
c191 ac0fc3 38 ldy ct
c194 8414   39 sty x0
c196 8d0bc3 40 sta x1+1
c199 ad10c3 41 lda ct+1
c19c 8515   42 sta x0+1
c19e a001   43 setpar ldy $001 ;(yr) = 1
c1a0 8c14c3 44 sty ix
c1a3 8c11c3 45 sty iy
c1a6 8c0fc3 46 sty ct
c1a9 88     47 dey      ;(yr) = 0
c1aa 8c10c3 48 sty ct+1
c1ad 8c13c3 49 sty ax
c1b0 8c12c3 50 sty ay
c1b3 88     51 dey      ;(yr) = - 1
c1b4     52 ;          berechne dx
c1b4 ad0bc3 53 begin lda x1+1 ;x1 => x0 ?
c1b7 c515   54 cmp x0+1
c1b9 9009   55 bcc draw01 ;nein ->
c1bb d01e   56 bne draw02 ;ja ->
c1bd ad0ac3 57 lda x1
c1c0 c514   58 cmp x0
c1c2 b017   59 bcs draw02 ;ja ->
c1c4 38     60 draw01 sec ;dx = x0 - x1
c1c5 a514   61 lda x0
c1c7 ed0ac3 62 sbc x1
c1ca 8d0dc3 63 sta dx
c1cd a515   64 lda x0+1
c1cf ed0bc3 65 sbc x1+1
c1d2 8d0ec3 66 sta dx+1
c1d5 8c14c3 67 sty ix      ;ix = -1
c1d8 4cecc1 68 jmp draw03
c1db 38     69 draw02 sec ;dx = x1 - x0
c1dc ad0ac3 70 lda x1
c1df e514   71 sbc x0
c1e1 8d0dc3 72 sta dx
c1e4 ad0bc3 73 lda x1+1
c1e7 e515   74 sbc x0+1
c1e9 8d0ec3 75 sta dx+1
c1ec     76 ;          berechne dy
c1ec a5fd   77 draw03 lda y1 ;y1 => y0 ?
c1ee cd09c3 78 cmp y0
c1f1 b00f   79 bcs draw04 ;ja ->
c1f3 38     80 sec      ;dy = y0 - y1
c1f4 ad09c3 81 lda y0
c1f7 e5fd   82 sbc y1
c1f9 8d0cc3 83 sta dy
c1fc 8c11c3 84 sty iy      ;iy = -1
c1ff 4c08c2 85 jmp draw05
c202 ed09c3 86 draw04 sbc y0 ;dy = y1 - y0
c205 8d0cc3 87 sta dy
c208 ad0ec3 88 draw05 lda dx+1 ;dx < dy ?
c20b d024   89 bne draw07 ;nein ->
c20d ad0dc3 90 lda dx
c210 cd0cc3 91 cmp dy
c213 b01c   92 bcs draw07 ;nein ->
c215 ae0cc3 93 ldx dy      ;vertausche die achsen
c218 8d0cc3 94 sta dy
c21b 8e0dc3 95 stx dx
c21e ad14c3 96 lda ix      ;ay = ix
c221 8d12c3 97 sta ay
c224 ad11c3 98 lda iy      ;ax = iy
c227 8d13c3 99 sta ax
c22a c8     100 iny      ;(yr) = 0
c22b 8c14c3 101 sty ix      ;ix = 0
c22e 8c11c3 102 sty iy      ;iy = 0
c231 ad0ec3 103 draw07 lda dx+1 ;of = dx / 2
c234 4a     104 lsr a
c235 8d0bc3 105 sta of+1 ;= 0
c238 ad0dc3 106 lda dx
c23b 6a     107 ror a
c23c 8d0ac3 108 sta of
c23f 4cd5c2 109 jmp plotit ;plotte 1. punkt
c242     110 ;***** approximationsschleife

```

```

c242 ad14c3 111 drawlop lda ix ;ix = -1 ?
c245 300e   112 bmi draw08 ;ja -> weiter
c247 18     113 clc      ;x0 = x0 + ix
c248 6514   114 adc x0
c24a 8514   115 sta x0
c24c a515   116 lda x0+1
c24e 6900   117 adc $000
c250 8515   118 sta x0+1
c252 4c62c2 119 jmp draw11
c255 38     120 draw08 sec ;x0 = x0 - ix
c256 a514   121 lda x0
c258 e901   122 sbc $001
c25a 8514   123 sta x0
c25c a515   124 lda x0+1
c25e e900   125 sbc $000
c260 8515   126 sta x0+1
c262 18     127 draw11 clc ;y0 = y0 + ax
c263 ad09c3 128 lda y0
c266 ad13c3 129 adc ax
c269 8d09c3 130 sta y0
c26c 18     131 clc      ;of = of + dy
c26d ad0ac3 132 lda of
c270 6d0cc3 133 adc dy
c273 8d0ac3 134 sta of
c276 ad0bc3 135 lda of+1
c279 6900   136 adc $000
c27b 8d0bc3 137 sta of+1
c27e ee0fc3 138 inc ct      ;ct = ct + 1
c281 d003   139 bne draw06
c283 ee10c3 140 inc ct+1
c286 ad0bc3 141 draw06 lda of+1 ;of <= dx ?
c289 cd0ec3 142 cmp dx+1
c28c 9047   143 bcc plotit ;ja -> zeichne punkt
c28e d008   144 bne draw09
c290 ad0dc3 145 lda dx
c293 cd0ac3 146 cmp of
c296 b03d   147 bcs plotit ;ja -> zeichne punkt
c298 38     148 draw09 sec ;of = of - dx
c299 ad0ac3 149 lda of
c29c ed0dc3 150 sbc dx
c29f 8d0ac3 151 sta of
c2a2 ad0bc3 152 lda of+1
c2a5 ed0ec3 153 sbc dx+1
c2a8 8d0bc3 154 sta of+1
c2ab ad12c3 155 lda ay      ;ay = -1 ?
c2ae 300e   156 bmi draw10 ;ja -> weiter
c2b0 18     157 clc      ;x0 = x0 + ay
c2b1 6514   158 adc x0
c2b3 8514   159 sta x0
c2b5 a515   160 lda x0+1
c2b7 6900   161 adc $000
c2b9 8515   162 sta x0+1
c2bb 4ccbc2 163 jmp draw12
c2be 38     164 draw10 sec ;x0 = x0 - ay
c2bf a514   165 lda x0
c2c1 e901   166 sbc $001
c2c3 8514   167 sta x0
c2c5 a515   168 lda x0+1
c2c7 e900   169 sbc $000
c2c9 8515   170 sta x0+1
c2cb 18     171 draw12 clc ;y0 = y0 + iy
c2cc ad09c3 172 lda y0
c2cf 6d11c3 173 adc iy
c2d2 8d09c3 174 sta y0
c2d5 ae09c3 175 plotit ldx y0 ;uebergebe y0 im (x:r) an plot
c2d8 200000 176 jsr plot ;zeichne punkt
c2db ad10c3 177 lda ct+1 ;ct <= dx ?
c2de cd0ec3 178 cmp dx+1
c2e1 9009   179 bcc nexptnt ;nein -> naechster punkt
c2e3 ad0dc3 180 lda dx
c2e6 cd0fc3 181 cmp ct
c2e9 b001   182 bcs nexptnt
c2eb 60     183 rts      ;linie ist fertig gezeichnet
c2ec 4c42c2 184 nexptnt jmp drawlop
c2ef     185 ;
c2ef 20fdae 186 gcoord jsr chkcom ;hole und pruefe parameter
c2f2 20ebb7 187 jsr getcor
c2f5 e0c8   188 cpx $200 ;y > 200 ?
c2f7 b00c   189 bcs finish
c2f9 a515   190 lda x0+1 ;x > 320 ?
c2fb c901   191 cmp $001
c2fd 9007   192 bcc finis ;< 1
c2ff d004   193 bne finis ;> 1
c301 a414   194 ldy x0 ;= 1
c303 c040   195 cpy $40
c305 60     196 finish rts
c306 a414   197 finis ldy x0
c308 60     198 rts
c309     199 ;
c309     200 ;arbeitsvariable
c309     201 ;
c309     202 y0 = *      ;y-koordinate
c30a     203 x1 = y0+1 ;x-koordinate
c30a     204 of = x1    ;approximationswert
c30c     205 dy = of+2 ;abstand y1 zu y0
c30d     206 dx = dy+1 ;anzahl der punkte
c30f     207 ct = dx+2  ;schleifenzaehler
c311     208 iy = ct+2 ;inkrement
c312     209 ay = iy+1 ;wie iy
c313     210 ax = ay+1 ;wie ix
c314     211 ix = ax+1 ;inkrement
c309     212 .end
x0 = 0014 y1 = 00fd chkcom = aefd getcor = b7eb
plot = 0000 ill = c178 drawlin = c179 setpar = c19e
begin = c1b4 draw01 = c1c4 draw02 = c1db draw03 = c1ec
draw04 = c202 draw05 = c208 draw06 = c231 draw07 = c238
draw08 = c255 draw11 = c262 draw12 = c2c2b plotit = c2d5
gcoord = c2ef finish = c305 finis = c306 dy = c30c y0 = c309
x1 = c30a of = c30a dx = c30d
ct = c30f iy = c311 ay = c312 ax = c313
ix = c314

```

Listing 2. Quellcode zu »Draw«.