

# Graf-quattro

von N.G. Barbieri

## Teil 3: Tricks über Tricks

Was ist eigentlich der Applesoft-Interpreter? Für den Applesoft-Anfänger ein unbekanntes Wesen im Apple, das für Dauerbelästigung mit „SYNTAX ERROR“ sorgt; für den Applesoft-Kenner ein persönlicher Gegner, den es unbedingt auszutricksen gilt; und für den Assemblerprogrammierer schlicht und einfach eine Ansammlung von Maschinenroutinen, die nach Möglichkeit durch eigene Routinen aufzurufen und zu erweitern sind. Dies setzt jedoch genaue Kenntnisse über Aufbau und Struktur des Interpreters voraus.



### BOX.COPY

Damit kommen wir zu der in diesem Beitrag abgedruckten Utility **BOX.COPY**. Wie bereits angekündigt, handelt es sich um das punktweise Kopieren des Inhalts eines in der HGR1-Seite definierten Rechtecks auf eine andere Stelle derselben

Seite oder wahlweise auf die HGR2-Seite. Bindet man diese Utility in die Routinen der vergangenen Beiträge ein, dann ist es von Applesoft aus sehr einfach: Cursoren einmal positionieren, dann ein zweites Mal, ein POKE 255,0 für Übertragung HGR1-HGR1 oder POKE 255,1 für Übertragung HGR1-HGR2, ein CALL 25297, und die Sache läuft schon. Will man nur diese Routine allein in eigene Programme einbauen, dann immer – in der Reihenfolge X1LOW, X1HIGH, Y1 und X2LOW, X2HIGH, Y2 – die Eckwerte der ursprünglichen Box in \$03C4 (964) bis \$03C9 (969) und die Eckwerte des Zielgebietes in \$03CA (970) bis \$03CF (975) poken, Seitenflag in \$00FF (255) nicht vergessen und nun **CALL 25297**.

Zu dem eigentlichen Assemblerprogramm, insbesondere für diejenigen, die in der Maschinensprache ihre ersten Schritte wagen, ist noch einiges zu sagen: Es handelt sich um eine reine „Bit-an/Bit-aus-Übertragung“ ohne Rücksicht auf das letzte Bit oder die Positionierung (d.h. Color-Bereich oder gerade und ungerade Hires-Spalte), so daß beim Übertragen von farbigen Ausschnitten komische Farbverschiebungen eintreten können. Demgegenüber klappt es bei schwarzweiß immer.

Das Programm besteht aus zwei Teilen: Der erste Teil untersucht die relative Position der zwei Rechtecke und entscheidet,

ob das Kopieren von links nach rechts und von oben nach unten oder umgekehrt verlaufen soll, da es für den Fall, daß sich Ursprungs- und Zielbox überschneiden, passieren könnte, daß man kopiert, was schon einmal kopiert worden ist. Das Problem ist ähnlich wie beim Moven von Speicherbereichen, nur daß hier die Überschneidung zweidimensional sein könnte. Der zweite Teil (ab LOOP) holt sich einfach einen Punkt von einer Stelle und bringt ihn an einer anderen Stelle unter, bis der ganze Inhalt der Box übertragen worden ist (siehe auch Kommentar).



### Drei Tricks für Anfänger

#### 16-Bit-Vergleich

Jeder kennt die CMP-Funktion (Compare) zum Verzweigen (Branch) je nach Carry- oder Zero-Flag, obwohl gelegentlich beim Bestimmen, was größer, gleich oder kleiner

### Neue Preise # 8

IBM-PC-KOMPAKT-PC-1000	55,00
Mainboard XT 16M + 256K RAM + 2M512 LK + 256K LK	42,00
Disk Controller (2 x 5.25" Disk)	28,00
Monochrome Graph. (8 par. Printer Leuchte (Hercules comp))	45,00
Color RGB + Video Graphics Adapter	52,00
Parallel Printer I (Epson)	75,00
320K RAM Leuchte	
Mult. IO Leuchte (Printer, Ser., Game, Uhr, optisch)	85,00
IBM-Mult. IO Leuchte	85,00
Mainboard XT 16M + 256K RAM + 2M512 LK + 256K LK	42,00
OK = Boot Disk	600,00
Mainboard wo jedoch 256K vollwertig	777,00
4MB-DISK Controller (2 x 5.25" Disk) (Epson comp)	85,00
Disk Controller (2 x 5.25" Disk) (Epson)	75,00
Monochrome Graph. (8 par. Printer Karte (Hercules comp))	510,00
Color RGB + Video Graphics Adapter	288,00
Parallel Printer-Karte (Epson) (Kabel (5m) + 75m)	135,00
32K-RAM-Karte (2M)	155,00
288K-Mult. IO Karte (10.25" disk) (Printer, Int. Game, optisch Port)	545,00
Real Time Clock (Epson) (OK + RAM-Disk/Speed-Über-Geh.)	785,00
288K-Mult. IO (2M) (2M) + 512K	495,00
Mult. IO Karte (Printer, optisch) (Epson)	495,00
Parallel Printer-Karte (Epson) (2M) (2M) (2M) (2M) (2M) (2M)	342,00
1024K-Mult. IO Karte (10.25" disk) (Printer, Int. Game, optisch Port)	495,00
256K-Diskdrive (5.25" Disk) (Epson)	495,00
XT 256K (io + Color RGB + Video-Graph-Karte)	2.232,00
XT 256K (io + Color RGB + Video-Graph-Karte)	2.545,00
XT 256K (io + Color RGB + Video-Graph-Karte + Parallel-Printer)	2.888,00
IBM-PC Systemeinheit + Laufwerk	2.078,00
IBM-deutsche CPU-Tastatur	498,00
IBM-Schreibgerät Color	1.058,00
PC 10-Cassette (2) (Tast.)	4475,00
GENE 16 C (256K RAM bis 64K) (2 x 256K-Disk)	4885,00
IBM-PC Systemeinheit + Laufwerk	2.078,00
IBM-deutsche CPU-Tastatur	498,00
IBM-Schreibgerät Color	1.058,00
PC 10-Cassette (2) (Tast.)	4475,00
GENE 16 C (256K RAM bis 64K) (2 x 256K-Disk)	4885,00
IBM-PC Systemeinheit + Laufwerk	2.078,00
IBM-deutsche CPU-Tastatur	498,00
IBM-Schreibgerät Color	1.058,00
PC 10-Cassette (2) (Tast.)	4475,00
GENE 16 C (256K RAM bis 64K) (2 x 256K-Disk)	4885,00



ner ist, Verwirrungen entstehen können. Belegen aber die zu vergleichenden Werte zwei Bytes (wie im Falle der X-Koordinaten), ist die einfachere Version erst ein Vergleich der High-Bytes und, wenn sie gleich sind, ein zweiter Vergleich der Low-Bytes, um die Sache endgültig klarzumachen. Der im Programm gezeigte Weg (Trick Nr. 1) ist kürzer und schneller. Erst ein Vergleich der zwei Low-Bytes, um das Carry-Flag entsprechend zu setzen, dann eine Subtraktion (mit Carry) der zwei High-Bytes, so daß am Ende die üblichen Branchen erfolgen können.

**16-Bit-Dekrement**  
 Was beim 16-Bit-Inkrement klappt, d.h. das High-Byte wird dann inkrementiert, wenn das Low-Byte den Wert \$00 einnimmt (ansonsten BEQ), ist beim Dekrementieren nicht gegeben, da der Wert \$00 des Low-Bytes auch mitgenommen werden muß. Also greift man normalerweise zur 16-Bit-Subtraktion, wobei vom High-Byte immer eine 0 mit Carry subtrahiert wird. Mit dem kleinen Kniff (siehe Trick Nr. 2) wird beim 16-Bit-Dekrement auch die 0 immer mitgenommen.

**HSCRN-Routine**  
 An sich hatte ich ursprünglich vor, diese Routine als separate Utility zur allgemeinen Benutzung zu gestalten, aber dann habe ich es mir anders überlegt. Trotz ausgedehntem Einsatz von Zeropage-Pointern ist das bitweise Übertragen auch so langsam genug! Überträgt man eine Box in der Größe der ganzen Grafikseite, d.h. 53760 Punkte, dauert dieses etwa 33 Sekunden. Eine gesonderte Routine hätte

für jeden einzelnen Punkt zusätzlich ein JSR und ein RTS gekostet, also ganze 12 Zyklen mehr, und das macht sich bemerkbar!

Ich möchte dem Leser aber eine separate HSCRN-Routine nicht vorenthalten. Die Routine **HSCRN** ist völlig relocativ, so daß man wie folgt vorgehen kann:

Ein BLOAD HSCRN, A... an jede beliebige Stelle. Danach die in Frage kommenden Koordinaten in der üblichen Reihenfolge (XLOW, XHIGH und Y) in \$0000, \$0001 und \$0002 poken, dann CALL... (Stelle von A = Anfangsadresse der Routine), danach ein N = PEEK (3). Ist N = 0, dann ist der angepeilte Punkt schwarz, ist N < > 0, entsprechend weiß. Einfach, nicht wahr?

Mit dieser Folge sind die Assembler-Routinen für das erste Modul der Serie Grafikquattro komplett. Jetzt heißt es zusammenstellen, also Diskette mit allen Assembler-Routinen in das Laufwerk legen und dann:

```

BLOAD XPLOD
BLOAD CURSOR 1
usw. bis
BLOAD BOX.COPY
Danach:
BSAVE GRAF.QUATTRO1, A$6000, L$466
    
```

Wer sich diese Mühe nicht machen will, kann auch die entsprechende Peeker-Diskette bestellen! Eigentlich könnte jetzt jeder Applesoft-Programmierer seinen eigenen Hires-Page-Editor selber schreiben. Nichtsdestoweniger wird mein eigener Editor demnächst im Peeker erscheinen.

## BOX.COPY

```

1          ORG $62D1      ;25297
2          *
3          * BOX.COPY
4          * -----
5          *
6          * von N.G. Barbieri/1985
7          *
8          * Aufrufen mit CALL 25297
9          *
10         * Kopiert den Inhalt einer Box
11         * bitweise auf eine andere
12         * Stelle derselben (HGR1) oder
13         * der anderen (HGR2) Grafik-
14         * seite.
15         *
16         * Applesoft-Routinen u. -Pointer
17         *
18         HPOSN EQU $F411
19         HPLOT EQU $F457
20         GBASL EQU $26
21         HMASK EQU $30
22         HCOLORZ EQU $E4
23         HPAG EQU $E6
24         *
25         * Cursorposition der
26         * ursprünglichen Box
27         *
28         CALXL EQU $3C4
29         CALXH EQU $3C5
30         CALY EQU $3C6
31         CA2XL EQU $3C7
32         CA2XH EQU $3C8
33         CA2Y EQU $3C9
34         *
35         * Cursorposition
36         * des neuen Platzes
37         *
38         CULXL EQU $3CA
39         CULXH EQU $3CB
40         CULY EQU $3CC
41         CU2XL EQU $3CD
42         CU2XH EQU $3CE
43         CU2Y EQU $3CF
44         *
45         * Zähler für X- u. Y-Richtungen
46         *
47         COUNXL EQU $1D
48         COUNXH EQU $1E
49         COUNY EQU $1F
50         *
51         * X,Y-Koordinaten des zu
52         * übertragenden Bits
53         *
54         XLOW EQU $00
55         XHIGH EQU $01
56         Y EQU $02
57         *
58         * Links/Rechts-Flag
59         *
60         XFLAG EQU $03
61         *
62         * X,Y-Zielkoordinaten
63         *
64         XLOW2 EQU $06
65         XHIGH2 EQU $07
66         Y2 EQU $08
67         *
68         * Oben/unten-Flag
69         *
70         YFLAG EQU $09
71         *
72         * Temporäre Ablage für beide
73         * X-Koordinaten u. X-Zähler
74         *
75         TEXL1 EQU $F9
76         TEXH1 EQU $FA
77         TEXL2 EQU $FB
78         TEXH2 EQU $FC
79         COXL EQU $FD
80         COXH EQU $FE
81         *
82         * Grafikseite-Übertragungsflag:
83         * wenn 0 auf HGR1, sonst auf HGR2
84         *
85         PFLAG EQU $FF
86         *
87         *

```

```

62D1: AD C6 03 88          LDA CALY
62D4: CD CC 03 89          CMP CULY
62D7: D0 06 90          BNE STEP1
91          *
92          * Wenn beide Y gleich, dann Flag
93          * auf 2!
94          *
62D9: A9 02 95          LDA #2
62DB: 85 09 96          STA YFLAG
62DD: D0 0C 97          BNE TESTX
98          *
99          * Feststellen, ob von oben nach
100         * unten oder umgekehrt und YFLAG
101         * entsprechend setzen.
102         *
62DF: B0 06 103         STEP1 BCS DOINC
62E1: A9 01 104         LDA #1
62E3: 85 09 105         STA YFLAG
62E5: D0 04 106         BNE TESTX
62E7: A9 00 107         DOINC LDA #0
62E9: 85 09 108         STA YFLAG
109         *
110         * Wenn Übertragung auf der an-
111         * deren Seite, dann egal
112         * welche Richtung!
113         *
62EB: A5 FF 114         TESTX LDA PFLAG
62ED: D0 27 115         BNE DOXINC
116         *
117         * Feststellen, ob von links nach
118         * rechts oder umgekehrt und XFLAG
119         * entsprechend setzen.
120         *
62EF: AD C5 03 121         LDA CALXH
62F2: CD CB 03 122         CMP CULXH
62F5: F0 04 123         BEQ TXLOW
62F7: B0 1D 124         BCS DOXINC
62F9: 90 15 125         BCC DOXDEC
62FB: AD C4 03 126         TXLOW LDA CALXL
62FE: CD CA 03 127         CMP CULXL
6301: D0 0B 128         BNE STEP2
129         *
130         * Wenn gleiche X- und Y-Werte u.
131         * gleiche Seite, warum übertragen?
132         * Jetzt wird klar, warum bei
133         * gleichen Y-Werten das YFLAG
134         * auf 2 steht!
135         *
6303: A5 09 136         LDA YFLAG
6305: C9 02 137         CMP #2
6307: F0 19 138         BEQ RET1
139         *
6309: 85 03 140         STA XFLAG
630B: 4C 1A 63 141         JMP STEP3
630E: B0 06 142         STEP2 BCS DOXINC
6310: A9 01 143         DOXDEC LDA #1
6312: 85 03 144         STA XFLAG
6314: D0 04 145         BNE STEP3
6316: A9 00 146         DOXINC LDA #0
6318: 85 03 147         STA XFLAG
148         *
631A: AD C6 03 149         STEP3 LDA CALY
631D: CD C9 03 150         CMP CA2Y
6320: D0 05 151         BNE DOIT
152         *
6322: A9 20 153         RET1 LDA #20
6324: 85 E6 154         STA HPAG
6326: 60 155         RTS
156         *
157         * Für den Y-Zähler testen, was
158         * wovon subtrahiert werden
159         * muß!
160         *
6327: 90 0C 161         DOIT BCC SUBY2
6329: 38 162         SEC
632A: ED C9 03 163         SBC CA2Y
632D: 85 1F 164         STA COUNY
632F: A5 09 165         LDA YFLAG
6331: F0 0F 166         BEQ MOD1
6333: D0 16 167         BNE MOD2
168         *
6335: 38 169         SUBY2 SEC
6336: AD C9 03 170         LDA CA2Y
6339: ED C6 03 171         SBC CALY
633C: 85 1F 172         STA COUNY
633E: A5 09 173         LDA YFLAG
6340: F0 09 174         BEQ MOD2
175         *
176         * Y-Startpunkte setzen.

```

```

177 *
6342: AE C9 03 178 MOD1 LDX CA2Y
6345: AC CF 03 179 LDY CU2Y
6348: 4C 51 63 180 JMP STEP4
181 *
634B: AE C6 03 182 MOD2 LDX CA1Y
634E: AC CC 03 183 LDY CU1Y
184 *
6351: 06 02 185 STEP4 STX Y
6353: 84 08 186 STY Y2
6355: E6 1F 187 INC COUNY
188 *
189 * Für den X-Zähler testen, was
190 * wovon subtrahiert werden
191 * muß!
192 *
193 * *** TRICK NR. 1 ***
194 *
6357: AD CA 03 195 LDA CU1XL
635A: CD CD 03 196 CMP CU2XL
635D: AD CB 03 197 LDA CU1XH
6360: ED CE 03 198 SBC CU2XH
6363: B0 17 199 BCS SUBX2
200 *
6365: 38 201 SEC
6366: AD CD 03 202 LDA CU2XL
6369: ED CA 03 203 SBC CU1XL
636C: 85 1D 204 STA COUNXL
636E: AD CE 03 205 LDA CU2XH
6371: ED CB 03 206 SBC CU1XH
6374: 85 1E 207 STA COUNXH
6376: A5 03 208 LDA XFLAG
6378: D0 17 209 BNE MOD3
637A: F0 28 210 BEQ MOD4
211 *
637C: 38 212 SUBX2 SEC
637D: AD CA 03 213 LDA CU1XL
6380: ED CD 03 214 SBC CU2XL
6383: 85 1D 215 STA COUNXL
6385: AD CB 03 216 LDA CU1XH
6388: ED CE 03 217 SBC CU2XH
638B: 85 1E 218 STA COUNXH
638D: A5 03 219 LDA XFLAG
638F: D0 13 220 BNE MOD4
221 *
222 * X-Startpunkte setzen!
223 *
6391: AE C7 03 224 MOD3 LDX CA2XL
6394: AC CB 03 225 LDY CA2XH
6397: 86 00 226 STX XLOW
6399: 84 01 227 STY XHIGH
639B: AE CD 03 228 LDX CU2XL
639E: AC CE 03 229 LDY CU2XH
63A1: 4C B4 63 230 JMP REST
231 *
63A4: AE C4 03 232 MOD4 LDX CA1XL
63A7: AC C5 03 233 LDY CA1XH
63AA: 86 00 234 STX XLOW
63AC: 84 01 235 STY XHIGH
63AE: AE CA 03 236 LDX CU1XL
63B1: AC CB 03 237 LDY CU1XH
238 *
63B4: 86 06 239 REST STX XLOW2
63B6: 84 07 240 STY XHIGH2
63BB: E6 1D 241 INC COUNXL
63BA: D0 02 242 BNE WEITER
63BC: E6 1E 243 INC COUNXH
244 *
245 * X-Koordinaten und X-Zähler
246 * in die Ablage bringen.
247 *
63BE: A6 00 248 WEITER LDX XLOW
63C0: A4 01 249 LDY XHIGH
63C2: 86 F9 250 STX TEXL1
63C4: 84 FA 251 STY TEXH1
63C6: A6 06 252 LDX XLOW2
63C8: A4 07 253 LDY XHIGH2
63CA: 86 FB 254 STX TEXL2
63CC: 84 FC 255 STY TEXH2
63CE: A6 1D 256 LDX COUNXL
63D0: A4 1E 257 LDY COUNXH
63D2: 86 FD 258 STX COXL
63D4: 84 FE 259 STY COXH
260 *
261 * Jetzt geht's los!
262 *
63D6: A9 20 263 LOOP LDA #20
63D8: 85 E6 264 STA HPAG
265 *

```

```

266 * HSCRN-Routine
267 * Ermitteln, ob ein angesprochener
268 * Punkt an oder aus ist ...
269 *
63DA: A6 00 270 LDX XLOW
63DC: A4 01 271 LDY XHIGH
63DE: A5 02 272 LDA Y
63E0: 20 11 F4 273 JSR HPOSN
63E3: B1 26 274 LDA (GBASL),Y
63E5: 25 30 275 AND HMASK
63E7: 29 7F 276 AND #57F
277 *
278 * ... und dann den entsprechenden
279 * schwarzen oder weißen Punkt auf
280 * die neue Stelle plotten.
281 *
63E9: F0 04 282 BEQ BLACK
63EB: A9 7F 283 LDA #57F ;weiß
63ED: D0 02 284 BNE POINT
63EF: A9 00 285 BLACK LDA #50 ;schwarz
63F1: 85 E4 286 POINT STA HCOLORZ
287 *
288 * Grafikseite-Flag (PFLAG) testen
289 * und entsprechende Vorkehrungen
290 * für die Übertragung treffen.
291 *
63F3: A5 FF 292 LDA PFLAG
63F5: F0 04 293 BEQ PAGE1
63F7: A9 40 294 LDA #540
63F9: 85 E6 295 STA HPAG
63FB: A6 06 296 PAGE1 LDX XLOW2
63FD: A4 07 297 LDY XHIGH2
63FF: A5 08 298 LDA Y2
6401: 20 57 F4 299 JSR HPLOTO
300 *
301 * X-Zähler bis auf 0 dekrementie-
302 * ren, dann nach DOY ...
303 *
6404: C6 1D 304 DEC COUNXL
6406: D0 06 305 BNE DOX
6408: A5 1E 306 LDA COUNXH
640A: F0 27 307 BEQ DOY
640C: C6 1E 308 DEC COUNXH
309 *
310 * Je nach XFLAG X-Koordinaten
311 * in- oder dekrementieren.
312 *
640E: A5 03 313 DOX LDA XFLAG
6410: D0 0E 314 BNE DECRX
6412: E6 00 315 INC XLOW
6414: D0 02 316 BNE NEXTX
6416: E6 01 317 INC XHIGH
6418: E6 06 318 NEXTX INC XLOW2
641A: D0 BA 319 BNE LOOP
641C: E6 07 320 INC XHIGH2
641E: D0 B6 321 BNE LOOP
322 *
323 * *** TRICK NR. 2 ***
324 *
6420: A5 00 325 DECRX LDA XLOW
6422: D0 02 326 BNE DOL1
6424: C6 01 327 DEC XHIGH
6426: C6 00 328 DOL1 DEC XLOW
6428: A5 06 329 LDA XLOW2
642A: D0 02 330 BNE DOL2
642C: C6 07 331 DEC XHIGH2
642E: C6 06 332 DOL2 DEC XLOW2
6430: 4C D6 63 333 JMP LOOP
334 *
335 * Y-Zähler dekrementieren.
336 * Ist er 0, dann erledigt,
337 * sonst ...
338 *
6433: C6 1F 339 DOY DEC COUNY
6435: F0 2A 340 BEQ RET2
341 *
342 * ... je nach YFLAG Y-Koordinaten
343 * in- oder dekrementieren und ...
344 *
6437: A5 09 345 LDA YFLAG
6439: F0 07 346 BEQ INY
643B: C6 02 347 DEC Y
643D: C6 08 348 DEC Y2
643F: 4C 46 64 349 JMP RSTX
6442: E6 02 350 INY INC Y
6444: E6 00 351 INC Y2
352 *
353 * ... die Start-X-Werte und den
354 * X-Zähler wieder für eine

```

```

355 * neue Runde herstellen.
356 *
6446: A6 F9 357 RSTX LDX TEXL1
6448: A4 FA 358 LDY TEXH1
644A: 86 00 359 STX XLOW
644C: 84 01 360 STY XHIGH
644E: A6 FB 361 LDX TEXL2
6450: A4 FC 362 LDY TEXH2
6452: 86 06 363 STX XLOW2
6454: 84 07 364 STY XHIGH2
6456: A6 FD 365 LDX COXL
6458: A4 FE 366 LDY COXH
645A: 86 1D 367 STX COUNXL
645C: 84 1E 368 STY COUNXH
645E: 4C D6 63 369 JMP LOOP
370 *
6461: A9 20 371 RET2 LDA #20
6463: 85 E6 372 STA HPAG
6465: 60 373 RTS

```

405 Bytes

### HSCRN

```

1 *
2 * HSCRN
3 *
4 *
5 * von N.G. Barbieri/1985
6 *
7 XLOW EQU $00
8 XHIGH EQU $01
9 Y EQU $02
10 FLAG EQU $03
11 GBASL EQU $26
12 HMASK EQU $30
13 HPOSN EQU $F411
14 *
15 * Ermittelt, ob ein Hires-Punkt
16 * weiß oder schwarz ist.
17 * Resultat in FLAG: 0 = schwarz,
18 * nicht 0 = weiß.
19 *
8000: A6 00 20 LDX XLOW
8002: A4 01 21 LDY XHIGH
8004: A5 02 22 LDA Y
8006: 20 11 F4 23 JSR HPOSN
8009: B1 26 24 LDA (GBASL), Y
800B: 25 30 25 AND HMASK
800D: 29 7F 26 AND #7F
800F: 85 03 27 STA FLAG
8011: 60 28 RTS

```

18 Bytes



**Der nächste Peeker  
Heft 9/1985  
erscheint am  
26. 8. 1985**

## Peeker

Einzelbezug DM 28,-  
Fortsetzungsbezug DM 20,-  
(Jederzeit kündbar, jedoch mindestens  
6 Disketten)

(\* = nur auf Diskette, nicht im Peeker  
gelistet! Seitenangaben beziehen sich  
auf Beginn des Listings)

Hüthig Software Service  
Postfach 10 28 69 · 6900 Heidelberg 1

PRIM.TEST \*  
PRIM.TOOLKIT.SOURCE \*

**Disk #2**  
(Heft 1-2, 1985, DOS-Format)

T.RAMDISKLC (1-2/85, S. 14)  
RAMDISKLC

T.IBS.RAMDISKDRIVER (1-2/85, S. 20)  
IBS.RAMDISKDRIVER

T.AP20.RAMDISKTEST  
AP20.RAMDISKTEST

T.QUICKCOPY (1-2/85, S. 26)  
QUICKCOPY

QUICKCOPY.PUFFER  
PRODOS.COPYA  
T.PRODOS.COPYOBJ \*  
PRODOS.COPYOBJ

PRODOS.PATCH (1-2/85, S. 31)

T.APPLESOFT.FRE (1-2/85, S. 36)

T.LC.FRE  
LC.FRE  
FRE.TEST  
T.RAM.FRE \*  
RAM.FRE

T.SCHIRMDISK (1-2/85, S. 44)  
SCHIRMDISK.LISA.SOURCE  
SCHIRMDISK

T.VIDEXT  
VIDEXT.LISA.SOURCE  
VIDEXT

GETPAS (1-2/85, S. 70)  
T.GETPAS.ASS \*

GETPAS.ASS  
GETDOS.PASCAL.SOURCE  
COPYDUPDIR.PASCAL.SOURCE

PRODOS.EDITOR.MACROS  
(1-2/85, S. 86)

**Disk #3**  
(Heft 1-2, 1985, CP/M-Format)

STEUER.84 (1-2/85, S. 47)  
PASS.BAS  
MENUE.BAS  
HELP.BAS \*

A.BAS  
B.BAS  
C.BAS  
D.BAS  
E.BAS  
F.BAS  
G.BAS  
H.BAS  
I.BAS

**Disk #1**  
(Heft 1+2, 1984)

T.DISASSEMBLER.65C02 (1/84, S. 15)  
DISASSEMBLER.65C02

T.ACCEL.WAIT (1/84, S. 22)

ACCEL.WAIT  
T.ACCEL.BOOT  
ACCEL.BOOT  
ACCEL.LC.KOPIERER  
T.ACCEL.LC.KOPIE  
ACCEL.LC.KOPIE  
T.ACCEL.ROM.KOPIE1  
ACCEL.ROM.KOPIE1  
T.ACCEL.ROM.KOPIE2  
ACCEL.ROM.KOPIE2

TURTLE.GRAFIK.MIT.REMS (1/84, S.29)  
TURTLE.GRAFIK.OHNE.REMS \*

DOUBLE.LORES.SOFTSWITCH.DEMO  
(1/84, S. 37)

DOUBLE.LORES.APPLESOFT.DEMO  
AMPER.DOUBLE.LORES.DEMO  
T.AMPER.DOUBLE.LORES  
AMPER.DOUBLE.LORES  
T.DOUBLE.LORES  
DOUBLE.LORES

HIRES (1/84, S. 41)

T.PRINTHIRES  
PRINTHIRES

DHGR.APISOFT.DEMO (2/84, S. 30)  
AMPER.DOUBLE.HIRES.BAS  
AMPER.DOUBLE.HIRES  
T.AMPER.DOUBLE.HIRES  
DHGR.LINEPLOTTER

INSTRING.TEST (2/84, S. 43)

INSTRING.OBJ  
T.INSTRING.OBJ  
INSTRING.LISA.SOURCE

LOESCHEN.EINES.ARRAYS  
(2/84, S. 52)

ULTRATERM.ENGLISCH \* (2/84, S. 60)  
ULTRATERM.DEUTSCH \*

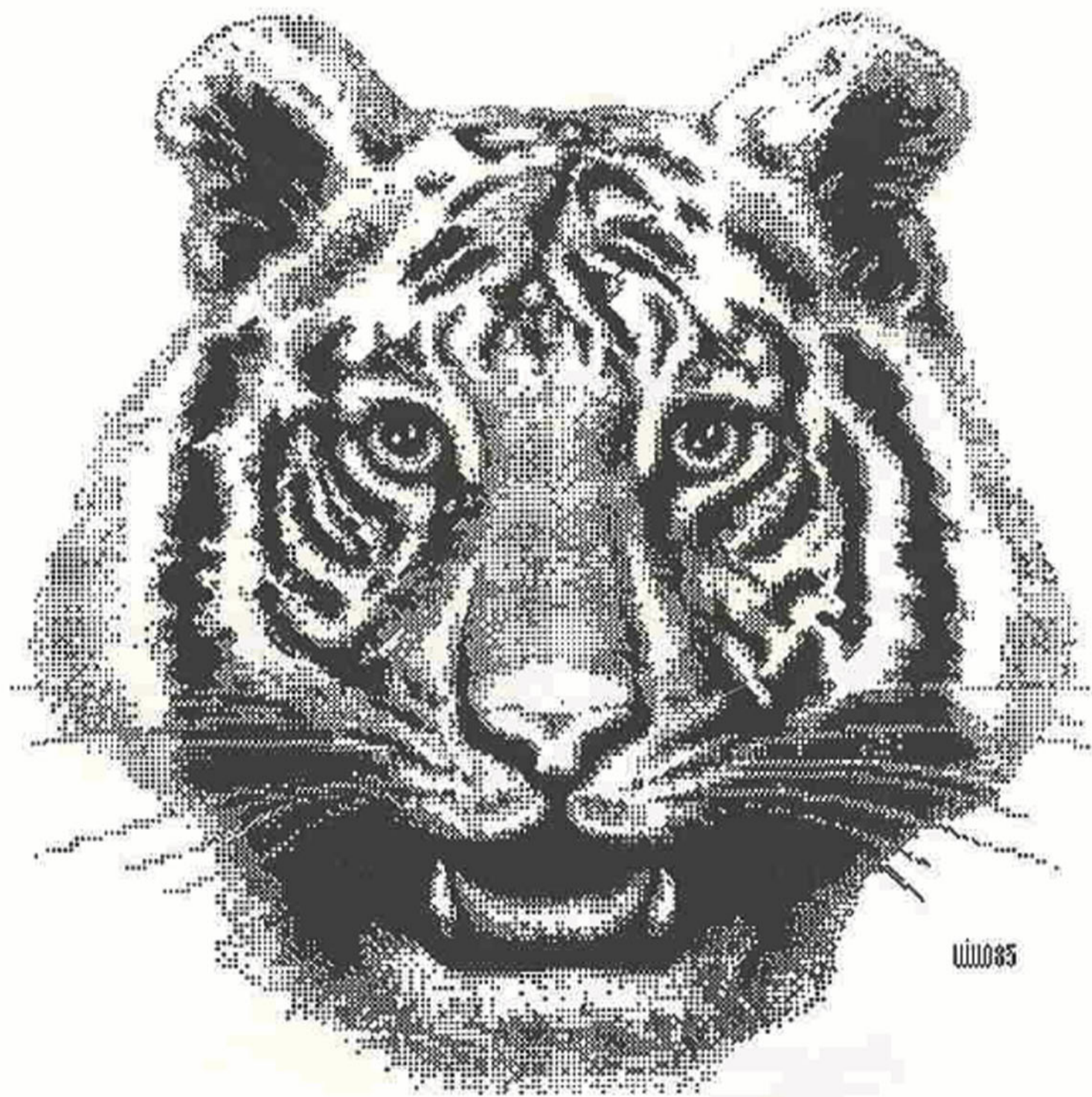
PRIMZAHLEN.OVERMEYER \*  
(2/84, S. 70)  
PRIM.OBJ0 \*  
PRIM.OBJ1 \*

# Graf-quattro

von Nino G. Barbieri

## Teil 4: Der Grafik-Editor

Zeige mir Dein BASIC, und ich werde Dir sagen, wer Du bist. Es ist immer interessant, ein BASIC-Programm vom stilistischen Standpunkt aus zu betrachten und Rückschlüsse auf den jeweiligen Autor zu ziehen. Es gibt den Superpingeligen und den totalen Chaoten, den überschwenglichen REM-Benutzer und den „Je-weniger-desto-besser“-Typ. Die stilistischen Eigenarten dieses hier abgedruckten BASIC-Programms sind aber mehr auf die vorhandenen Gegebenheiten als auf den Charakter des Autors zurückzuführen. Gewiß, BASIC ist langsam. Compilieren ist wegen Platzmangels oft nicht möglich. Der beste Weg ist also, alles was zeitlich möglich ist, in Assembler zu schreiben und von BASIC aus mit PEEK, POKE und CALL zu steuern. Das macht fast immer aus einer Schnecke einen Flitzer. In den vorangegangenen Beiträgen ist eine Ansammlung von Maschinensprache-Routinen abgedruckt worden; wir wollen sie jetzt aus einem Applesoft-Programm heraus steuern.



### 1. Das Programm

Zwischen Anfang des Applesoft-Programms (2048) und Beginn der ersten Grafikseite (8192) sind genau 6144 Bytes verfügbar. Aus diesem Grund mußte das Programm **GRAFIK.EDITOR** auf REMs verzichten und so gehalten werden, daß ein Maximum an Effizienz mit einem Minimum an Platzbedarf zu erreichen war. Wenn das Programm im Speicher ist und läuft, bleiben nur noch ca. 100 Bytes übrig. Dieses bedeutet, daß nicht sehr viel Platz

für Einfügungen oder Verbesserungen des Lesers bleibt, es sei denn, die verhältnismäßig aufwendigen Dialoge der LOAD- und SAVE-Routinen (Zeile 426-488) werden verkürzt oder herausgenommen. Schließlich kann man einen Grafikseite-SAVE und -LOAD auch „zu Fuß“ machen. Die Programmbeschreibung wird jetzt im Telegrammstil folgen, sonst müßte ich den ganzen Peeker in Anspruch nehmen (ich höre auch so unseren Chefredakteur stöhnen: „Zu lang, viel zu lang!“). Ich werde mich daher auf das Erläutern der einzelnen

Befehle beschränken, mit knappen Hinweisen auf das Programmgeschehen. Alle Befehle erfolgen per Knopfdruck, mit Ausnahme derjenigen, die die Beantwortung einer Frage verlangen.

Nach Laden des Assembler-Teils (GRAF-QUATTRO.1, als Zusammenstellung aller bisheriger Routinen auf der Sammeldisk #8 enthalten und hier nochmals als Hex-Dump gelistet), Initialisierung der notwendigen Variablen und Einschalten der Grafik (Zeile 100-110) springt das Programm nach Zeile 294, wo die Hauptsteuerschlei-

fe (bis 332) steht. Achtung: Der Grafikspeicher wird aus gutem Grund nicht gelöscht.

## 2. Allgemeine Befehle

**I, J, K, M** – Richtungssteuerung des Cursor 1, für die rechte Hand, in den jeweiligen logischen Richtungen (Zeile 298-304).

**W, A, S, Y** – Richtungssteuerung des Cursor 2, für die linke Hand (Zeile 306-312). Achtung, diejenigen, die keine deutsche Tastatur angeschlossen haben, müssen in Zeile 310 den Wert 89 (ASCII Y) mit dem Wert 90 (ASCII Z) austauschen!

Die Cursoren und evtl. andere Gebilde werden von den GOSUBs zwischen 164 und 224 gesteuert, die kontinuierlich von der Hauptschleife angesprochen werden.

**Ctrl-R = Repetieren** – Diese Taste (Zeile 314) bewirkt einen Sprung zu den Zeilen 366-382. Dieser Bereich sorgt für das Kopieren oder Übertragen von einem viereckigen Ausschnitt der Grafik auf die sichtbare Grafikseite 1 oder auf die unsichtbare Grafikseite 2. Beim Drücken der Taste Ctrl-R erscheint, wenn nicht schon vorhanden (s.a. Befehl V), ein blinkendes Viereck, begrenzt durch die Stellung der Cursoren. Dabei wird die Frage gestellt, ob die Übertragung auf Seite 1 oder 2 erfolgen soll. Nach Beantwortung der Frage das blinkende Viereck mit Hilfe der rechten oder linken Handsteuerung auf die gewünschte Übertragungsstelle schieben und erneut Ctrl-R drücken. Während der Positionierungszeit sorgt die Flagge U dafür, daß die Hauptschleife bis Zeile 316 begrenzt ist, so daß alle übrigen Befehle inaktiv werden.

**D = Doppelte Cursor-Steuerung** – Beim Drücken der Taste D (Zeile 318) wird die Flagge DO ein- und ausgeschaltet. Ist die Flagge auf 1, bewegen sich beide Cursoren synchron, gleichviel ob man mit der linken oder rechten Hand steuert. Die Routinen in den Zeilen 184-194 und in Zeile 218 sorgen dafür, daß beim Erreichen der Grafikränder keine weitere Verschiebung der Cursoren möglich ist.

**B = Box** – Diese Taste bewirkt eine in den vordefinierten Farben gefüllte Box (Zeile 320). Die Begrenzung der Box wird von der diagonal entgegengesetzten Stellung der Cursoren definiert.

**9 und 0 = Ellipse und gefüllte Ellipse** – (Zeile 322, Routinen in den Zeilen 142-158). Taste 9 bewirkt das Zeichnen einer Ellipse, wobei die X- und Y-Achse von der

jeweiligen Stellung der Cursoren definiert werden. Die Ellipse wird in der vordefinierten Farbe gezeichnet. Taste 0 bildet eine gefüllte Ellipse in den vordefinierten Boxfarben. Am einfachsten ist es, Viereck vorzuschalten (s. Befehl V), um Stellung und Dimensionen der Ellipse zu bestimmen. Ist das Viereck ein Quadrat, dann bekommen wir keine Ellipse mehr, sondern einen Kreis.

**P = Plotten** – Beim Drücken der Taste P wird in der angegebenen Farbe das sichtbare, blinkende Gebilde auf dem Bildschirm gezeichnet.

**2 = Cursor-Schrittweite** – Bei Betätigung erfolgt die Frage nach der gewünschten Cursor-Schrittweite (Zeile 324, die Befragung in Zeile 292).

**Ctrl-D = Disk** – Das gleichzeitige Drücken der Tasten Ctrl und D (Zeile 326) bewirkt einen Sprung zu den Zeilen 426-488, wo die LOAD- und SAVE-Routinen für die Grafikseiten ihren Platz finden. Bei Platzbedarf können die Zeilen 326 und 426-488 entfernt werden. Danach muß aber das LOAD und SAVE der Grafik manuell erfolgen.

## 3. Zirkel-Befehle

**Ctrl-K = Zirkel** – (Zeile 328) ruft den Übergang auf die sekundäre Steuerschleife hervor (Zeile 334-354), die einen Zirkel simuliert.

Zwischen Zeile 258 und 286 ist eine dritte Steuerschleife, die gemeinsam von der Haupt- und Zirkelschleife benutzt wird. Die folgenden Befehle werden nach Betätigung von Ctrl-K gültig.

**Ctrl-Z = Zurück zur Hauptschleife** – Diese Taste (Zeile 356) ist zu betätigen, wenn man mit der Kreisroutine fertig ist. Bei der Kreisroutine bildet einer der Cursoren immer das Zentrum des Kreises, der andere Cursor rotiert in dem gegebenen Abstand bei Betätigen der nachfolgenden Tasten.

**J und K = Bogen um Mittelpunkt-Cursor** – Das Zeichnen eines Bogens, eines kompletten Kreises oder eines regelmäßigen Polygons ist, je nach Betätigung anderer untenstehender Steuertasten, gegeben. Dabei bewegt die Taste J den Cursor im entgegengesetzten Uhrzeigersinn, die Taste K im Uhrzeigersinn.

**P = Plotten** – Die Taste P schaltet das Plotten der Kreissegmente in der vorgegebenen Farbe und definiertem Winkelgradschritt ein und aus. Einmal drücken, und der rotierende Cursor hinterläßt eine Linie,

nochmals drücken, und der Cursor bewegt sich, ohne zu zeichnen (Zeile 346).

**W = Wechseln** – Beim Drücken der Taste W wird der jeweilige rotierende Cursor zum Mittelpunkt-Cursor und der ehemalige Drehpunkt-Cursor zum rotierenden Cursor. In dieser Art und Weise kann man sehr einfach Kreisornamente zeichnen (Zeile 348).

**2 = Cursor-Schrittweite** – Diese Taste (Zeile 350) hat dieselbe Funktion wie in der Hauptsteuerschleife, nur, daß die Bewegung des jeweiligen rotierenden Cursors in Winkelgraden angegeben wird. Eine Bewegung von jeweils 10 Grad reicht aus, um einen einigermaßen sauberen Kreis zu ziehen. Werden größere Grad-schritte angegeben, zeichnet der Zirkel ein Gebilde, welches immer mehr zu einem Polygon neigt. Gibt man z.B. eine Schrittweite von 120 Grad ein, zeichnet unser Zirkel ein gleichschenkliges Dreieck. 90 Grad Schrittweite bildet ein Quadrat, 72 Grad ein Pentagon usw. Bei der Gradeingabe sind Werte mit Dezimalstellen erlaubt.

## 4. Gemeinsame Steuerbefehle

Sowohl die Hauptschleife als auch die Zirkelschleife benutzen die Utilities dieser gemeinsamen Schleife (Zeile 258-286). Folgende Befehle sind vorgesehen:

**X = Kontinuierliches Zeichnen** – Die Taste X (Zeile 260) bewirkt das Ein- und Ausschalten des kontinuierlichen Zeichnens. Gezeichnet wird in der vorgegebenen Farbe (mit Ausnahme der Farbe 8 = XOR).

**L = Linie** – Mit dieser Taste (Zeilen 262 und 264) wird eine blinkende Linie zwischen den zwei Cursoren ein- und ausgeschaltet. Diese Linie verhält sich wie ein Gummiband, d.h. sie verbindet immer die zwei Cursoren, gleichgültig, wie sie bewegt werden. Mit Hilfe der beiden Funktionen P und X kann man einige hübsche Dinge realisieren. Will man z.B. eine dicke Linie zeichnen, bringt man die Cursoren in den gewünschten Abstand und schaltet die Funktion D (Doppel-Cursor-Steuerung) und X ein. Bei Schrittweite 1 wird mit Hilfe der Richtungstasten (Tasten gedrückt lassen) eine Linie in der gewünschten Stärke gezeichnet. Ist die Cursor-Schrittweite eine andere als 1, wird eine Reihe von Linien in regelmäßigen Abständen geplottet. Will man die blinkende Linie nur an bestimmten Stellen zeichnen, ist die Funktion P anstatt X zu benutzen. Ist die Zirkelroutine eingeschaltet, ist das Ergebnis dasselbe, nur daß anstatt einer



gradlinigen Bewegung eben die Radien des jeweiligen Kreises oder Bogens gezeichnet werden.

**V = Viereck** – (Zeile 266 und 268). Diese Taste schaltet ein blinkendes Viereck zwischen den zwei diagonal entgegengesetzten Cursors ein und aus. Das Viereck verhält sich in derselben Art und Weise wie die Linie, d.h. in Verbindung mit den Funktionen D, X und P. Es lohnt sich, mit verschiedenen gleichzeitig eingeschalteten Funktionen ein bißchen zu experimentieren. Probieren Sie z.B. einen größeren Kreis mit eingeschalteter Viereckfunktion (V- und X-Taste) zu zeichnen. Viereck ermöglicht besseres Dimensionieren und Positionieren bei der Benutzung der Funktionen: Ausschnitt übertragen, Box, Ellipse und gefüllte Ellipse.

**Leertaste = Textzeilen** – Die Leertaste (Zeile 270) schaltet die vier Textzeilen unter der Grafik ein und aus. Sind die Textzeilen eingeblendet, werden automatisch Informationen sichtbar, und das Programm wird langsamer, da bei jedem Tastendruck die neue Situation angezeigt wird (Zeile 384-424). Die Fußnoten sind in drei Spalten aufgeteilt und zeigen folgendes an (immer von oben nach unten):

*Linke Spalte:*

- XCOL. 1 = die Farbe, die benutzt wird beim Einschalten der Funktion X.
- PCOL. 2 = die Farbe der Funktion P.
- BCOL. 1 und 2 = die Farben, mit denen Box, Ellipse und Hintergrund gefüllt werden.

*Mittlere Spalte:*

Der Reihe nach: Funktionen L, V, X und D an oder aus. Diese Angaben haben reinen Erinnerungswert.

*Rechte Spalte bei der Hauptschleife:*

- X-,Y-Position des Cursors 1;
- X-,Y-Position des Cursors 2;
- horizontaler und vertikaler Abstand der zwei Cursors;
- aktuelle Cursor-Schrittweite.

*Rechte Spalte bei Zirkelschleife:*

- X-,Y-Position des stehenden Cursors (Kreismitte);
- absolute Winkelangabe zwischen stehendem und rotierendem Cursor (sogar nach DIN, d.h. 0 Grad rechts, 90 Grad oben, 180 Grad links und selbstverständlich 270 Grad unten);
- die jeweilige rotierende Cursor-Bewegung in Grad;
- Angabe, ob die Kreiszeichnerfunktion an oder aus ist.

**1 = Cursor-Schrittweite 1/10** – Die Taste 1 (Zeile 272) bewirkt eine Umschaltung der Cursor-Schrittweite von 1 auf 10 und umgekehrt. Bei der Kreisroutine natürlich nicht in Punkten, sondern in Grad.

**Ctrl-S = Seiten austauschen (swappen)**

– Bei Betätigung der Taste Ctrl-S (Zeile 274) werden die Grafikseite 1 und 2 miteinander vertauscht. Das ist bequem, wenn man mit zwei Grafikseiten gleichzeitig arbeiten will oder wenn ein Teil der Grafikseite 2 (Hilfsseite) auf die Grafikseite 1 (Arbeitsseite) zu übertragen ist. Also zuerst swappen, dann übertragen und nochmals swappen.

**Ctrl-T = Test 1** – Die Taste Ctrl-T (Zeile 276) bewirkt ein schnelles Hin- und Herschalten der beiden Grafikseiten, so daß sie quasi gleichzeitig zu sehen sind. Diese Funktion ist nützlich, wenn man den Unterschied zwischen der ersten und zwei-

ten Grafik genau betrachten will, z.B. einen Bewegungsschritt bei Animation. Abstellen mit irgendeiner anderen Taste.

**T = Test 2** – Diese Funktion (Zeile 278) macht nur die unsichtbare Seite zur Kontrolle sichtbar; irgendeine andere Taste führt zum ursprünglichen Zustand zurück.

**Ctrl-H = Hintergrund** – Die Taste Ctrl-H (Zeile 280) füllt die Grafikseite mit den angegebenen Farben. Sind die angegebenen Farben beide mit 8 definiert worden, wird die Grafikseite negatiert. Vorsicht: Mit Ausnahme der Negativierungsfunktion zerstört der Befehl Ctrl-H den Inhalt der Grafikseite.

**Taste F = Farbe 1** – Beim Drücken der Taste F (Zeile 282 und GOSUB nach 230 und 232) wird die Frage nach der gewünschten Linienfarbe gestellt. Die Farben werden, wie beim Apple üblich, zwischen 0 und 7 definiert. Dieser Farbbereich gilt gleichzeitig für die Funktion X und P. Neu in diesem Grafiksystem ist die Farbe 8 = XOR, d.h. immer negativ zum Hintergrund. Die Farbe 8 gilt nur für die Funktion P, da sie für die Funktion X zu unerwünschten Effekten führen könnte.

**Ctrl-F = Farbe 2** – Drückt man Ctrl-F (Zeile 284, GOSUB nach 234), wird zweimal nach einer Farbdefinition gefragt. Hier kann man entweder zweimal dieselbe oder auch verschiedene Farben angeben, einschließlich der Farbe 8. Diese Farbdefinition gilt für Box, gefüllte Ellipse und Hintergrund. Gefüllt wird alternativ in den zwei angegebenen Farben. Auch hier lohnt es sich zu probieren, z.B. alternativ mit Farbe 8 und einer anderen, um durchscheinende Effekte zu erreichen.

Ich habe versucht, die Tastenfunktionen nach einer gewissen Mnemonik zu verteilen (z.B. F für Farbe, L für Linie usw.), so daß sie schnell zu erlernen und einfach zu behalten sind. Wer nicht zufrieden ist, kann selbstverständlich seine eigenen Funktionstasten programmieren: Einfach in der angegebenen Zeile den gewünschten ASCII-Wert einsetzen.

Eine letzte Anmerkung: Zeile 252 muß ohne Leertasten abgetippt werden, sonst paßt sie nicht in den Eingabepuffer.

## 5. Wie geht's weiter?

Da sich die Veröffentlichung der kompletten Serie noch über viele Monate hinziehen würde, wird sie mit diesem Grafik-Editor zunächst abgeschlossen. Statt dessen soll „Graf-quattro“ entweder als Pecker-Sonderheft oder in Buchform erscheinen. Wir werden Sie rechtzeitig informieren.





```

208 IF X < 0 THEN X = 0
210 IF X > 279 THEN X = 279
212 IF Y < 0 THEN Y = 0
214 IF Y > 191 THEN Y = 191
216 RETURN
218 IF DO THEN XU = X1 - X2:YU = Y1 - Y2: RETURN
220 X1 = PEEK (970) + PEEK (971) * 256:Y1 = PEEK (972):X2
= PEEK (973) + PEEK (974) * 256:Y2 = PEEK (975):
RETURN
222 IF PEEK (769) < > 8 THEN POKE 768,1: GOSUB 164: POKE
768,0
224 GOTO 178
226 F = PEEK (T) - 176: IF F < 0 OR F > 8 THEN 226
228 POKE S,0: RETURN
230 HOME : POKE - 16301,0: PRINT " Linie-Farbe (1/8) ? ":
GOSUB 226: HOME : POKE - 16302,0: POKE 769,F: IF F = 8
THEN POKE 768,0:F = 0: RETURN
232 C = F: HCOLOR=C:F = 0: RETURN
234 HOME : POKE - 16301,0: PRINT "1. Box-Farbe (1-8) ? ":
GOSUB 226: POKE 770,F: PRINT : PRINT "2. Box-Farbe
(1-8) ? ": GOSUB 226: POKE 771,F:F = 0: HOME : POKE -
16302,0: RETURN
236 POKE - 16302,0:S1 = 1
238 POKE - 16299,0: FOR I = 1 TO 50: NEXT : POKE -
16300,0: FOR I = 1 TO 50: NEXT : IF PEEK (T) < 128
THEN 238
240 POKE S,0: RETURN
242 POKE - 16299,0: IF PEEK (T) < 128 THEN 242
244 POKE - 16300,0: GOTO 240
246 GOSUB 178: CALL 25177: HCOLOR=C: POKE 768,0: GOSUB
178: RETURN
248 POKE - 16301,0: HOME : PRINT " Übertragen auf Seite
(1-2) ? ": GOSUB 226:UE = F - 1: IF UE < 0 OR UE > 1
THEN 248
250 POKE - 16302,0: RETURN
252 CALL 24576: HPLLOT PEEK (964) + PEEK (965) * 256, PEEK
(966) TO PEEK (964) + PEEK (965) * 256, PEEK (969) TO
PEEK (967) + PEEK (968) * 256, PEEK (969) TO PEEK
(967) + PEEK (968) * 256, PEEK (966) TO PEEK (964) +
PEEK (965) * 256, PEEK (966): RETURN
254 X1 = PEEK (964) + PEEK (965) * 256:Y1 = PEEK (966):X2
= PEEK (967) + PEEK (968) * 256:Y2 = PEEK (969):
RETURN
256 IF S1 THEN RETURN
258 IF F = 80 THEN GOSUB 222
260 IF F = 88 THEN POKE 768,1 * ( PEEK (768) = 0): IF NOT
PEEK (768) AND NOT KR THEN F = 80: GOTO 258
262 IF F = 76 THEN CALL LI:L = 1 * (L = 0)
264 IF F = 76 AND KR THEN CALL LI
266 IF F = 86 THEN CALL VI:V = 1 * (V = 0)
268 IF F = 86 AND KR THEN CALL VI
270 IF F = 32 THEN S1 = 1 * (S1 = 0): POKE - 16301 - S1,0
272 IF F = 49 THEN FA = 1 + 9 * (FA = 1)
274 IF F = 19 THEN GOSUB 160: CALL 25258: GOSUB 160
276 IF F = 20 THEN GOSUB 236:F = 0
278 IF F = 84 THEN GOSUB 242:F = 0
280 IF F = 8 THEN GOSUB 246: IF KR THEN GOSUB 178
282 IF F = 70 THEN GOSUB 230
284 IF F = 6 THEN GOSUB 234
286 RETURN
288 HOME : POKE - 16301,0: PRINT "Winkel-Schritt (Max.
120) ": INPUT FA: IF FA < 0 OR FA > 120 THEN 288
290 RETURN
292 HOME : POKE - 16301,0: INPUT "Cursor-Schritt (1-191) ?
":FA: POKE - 16302,0: IF FA < 1 OR FA > 191 THEN 292
294 IF NOT S1 THEN GOSUB 164: GOSUB 384
296 GOSUB 164:F = PEEK (T) - 128: IF F < 0 THEN 296
298 POKE S,0: IF F = 75 THEN Z = 1:K1 = Z
300 IF F = 74 THEN Z = 2:K1 = Z
302 IF F = 77 THEN Z = 3:K1 = Z
304 IF F = 73 THEN Z = 4:K1 = Z
306 IF F = 83 THEN Z = 1:K2 = Z
308 IF F = 65 THEN Z = 2:K2 = Z
310 IF F = 89 THEN Z = 3:K2 = Z
312 IF F = 87 THEN Z = 4:K2 = Z
314 IF F = 18 THEN 366
316 IF U THEN GOSUB 256: GOTO 294
318 IF F = 68 THEN DO = 1 * (DO = 0): GOSUB 218
320 IF F = 66 THEN GOSUB 178: CALL 25007: GOSUB 178
322 IF F = 48 OR F = 57 THEN GOSUB 178: GOSUB 142: GOSUB
178
324 IF F = 50 THEN 292
326 IF F = 4 THEN 426
328 IF F = 11 THEN KR = 1: GOTO 334
330 GOSUB 258
332 GOTO 294
334 GOSUB 178:P = 0:K3 = 0: GOSUB 360: GOTO 352
336 GOSUB 164:F = PEEK (T) - 128: IF F < 0 THEN 354
338 POKE S,0: GOSUB 260
340 IF F = 26 THEN 356
342 IF F = 75 THEN GOSUB 112
344 IF F = 74 THEN GOSUB 116
346 IF F = 80 THEN P = 1 * (P = 0)

```

```

348 IF F = 87 THEN GOSUB 360
350 IF F = 50 THEN GOSUB 288
352 IF NOT S1 THEN GOSUB 384
354 GOTO 336
356 GOSUB 178: IF DO THEN GOSUB 218
358 KR = 0:FA = 10: GOTO 330
360 IF K3 THEN 364
362 XM = X1:YM = Y1:X = X2:Y = Y2: GOSUB 130:K3 = 1:
RETURN
364 XM = X2:YM = Y2:X = X1:Y = Y1: GOSUB 130:K3 = 0:
RETURN
366 IF U THEN 372
368 DX = DO:DO = 1: GOSUB 218: POKE 255,1: GOSUB 240: IF
NOT UE THEN 376
370 GOSUB 382: CALL 25258: GOSUB 160: GOTO 294
372 IF NOT UE THEN 376
374 U = 0: GOSUB 160: CALL 25258: POKE 230,32: POKE -
16299,0: CALL 25297: GOSUB 160: GOSUB 254: POKE -
16300,0:DO = DX: GOTO 294
376 IF U THEN 380
378 DX = DO:DO = 1: GOSUB 218: POKE 255,0: GOSUB 382:
GOSUB 252: GOSUB 160: GOTO 294
380 U = 0: GOSUB 160: GOSUB 252: CALL 25297: GOSUB 160:
GOSUB 254:DO = DX: GOTO 294
382 U = 1: FOR I = 964 TO 969: POKE I, PEEK (I + 6): NEXT
: GOSUB 160:V = 1: RETURN
384 HOME : PRINT "XCOL, 1=":C: PRINT "PCOL, 2=": PEEK
(769): PRINT "BCOL, 1=": PEEK (770): PRINT "BCOL, 2=":
PEEK (771):
386 VTAB 21: HTAB 12: PRINT "Lin, ": IF L THEN PRINT
"an": GOTO 390
388 PRINT "aus"
390 VTAB 22: HTAB 12: PRINT "Vier, ": IF V THEN PRINT
"an": GOTO 394
392 PRINT "aus"
394 VTAB 23: HTAB 12: PRINT "Plot ": IF PEEK (768) THEN
PRINT "an": GOTO 398
396 PRINT "aus"
398 VTAB 24: HTAB 12: PRINT "2XCu, ": IF DO THEN PRINT
"an": GOTO 402
400 PRINT "aus":
402 IF KR THEN 414
404 VTAB 21: HTAB 22: PRINT "Cur. 1: X=":X1:" Y="Y1
406 VTAB 22: HTAB 22: PRINT "Cur. 2: X=":X2:" Y="Y2
408 VTAB 23: HTAB 22: PRINT "HLIN.=": ABS (X1 - X2):"
VLIN.=": ABS (Y1 - Y2):
410 VTAB 24: HTAB 22: PRINT "Cur.-Step =" :FA:
412 RETURN
414 VTAB 21: HTAB 22: PRINT "Mitte : X=":XM:" Y="YM:
416 VTAB 22: HTAB 22: PRINT "Winkel : ": INT (360 - W):
418 VTAB 23: HTAB 22: PRINT "Step : ":FA:
420 VTAB 24: HTAB 22: PRINT "Kr,-Plot : ": IF P THEN
PRINT "an": GOTO 424
422 PRINT "aus":
424 RETURN
426 F = FRE (0): GOSUB 160
428 HOME : POKE - 16301,0: PRINT "S = SAVE": PRINT "L =
LOAD": PRINT "Z = Zurück":
430 F = PEEK (T): IF F < 128 THEN 430
432 POKE S,0
434 IF F < > 218 THEN 438
436 POKE - 16302,0: GOTO 106
438 ONERR GOTO 106
440 IF F < > 204 THEN 462
442 HOME : PRINT "Laden v. Disk (Name der Grafik) ?"
444 PRINT : INPUT NS
446 HOME : PRINT "Laden : ":NS: PRINT : PRINT "auf Seite
(1-2) ":
448 F = PEEK (T) - 176: IF F < 1 OR F > 2 THEN 448
450 HOME : PRINT "Lade ";NS;" auf Seite ";F
452 PRINT : PRINT "OK (J/N) ? ":SE = F * 2000
454 F = PEEK (T): IF F < 128 THEN 454
456 POKE S,0: IF F = 206 THEN 436
458 IF F < > 202 THEN 454
460 PRINT CHR$ (13) + CHR$ (4):"BLOAD" + NS + ",AS" + STR$
(SE): GOTO 436
462 IF F < > 211 THEN 436
464 HOME : PRINT "Speichern (Name der Grafik) ?"
466 PRINT : INPUT NS
468 HOME : PRINT "Speichern : ":NS: PRINT : PRINT "von
Seite (1-2) ":
470 F = PEEK (T) - 176: IF F < 1 OR F > 2 THEN 470
472 SE = F * 2000: IF F = 2 THEN CALL 25258
474 HOME : PRINT "Speichere ";NS;" von Seite ";F
476 PRINT : PRINT "OK (J/N) ? ":
478 F = PEEK (T): IF F < 128 THEN 478
480 POKE S,0: IF F = 206 THEN 436
482 IF F < > 202 THEN 478
484 PRINT CHR$ (13) + CHR$ (4):"BSAVE" + NS +
",AS2000,LS1FF8"
486 IF SE = 4000 THEN CALL 25258
488 GOTO 436

```



## Schwierigkeiten mit den Graf-quattro-Cursoren.

Es haben sich bei der Redaktion Leser gemeldet, die Schwierigkeiten mit den Graf-quattro-Cursoren hatten. Obwohl die Routinen getreulich abgetippt wurden, erschien nach den vorgeschriebenen CALLs eine Ansammlung von „wild gewordenen Linien“ und nicht die versprochenen +- und x-Cursoren. Hierzu gibt es eine einfache Erklärung, die ich in meinem Beitrag zu erwähnen vergessen habe (mea culpa): Die Cursoren sind Shapes, und Shapes brauchen zwei Parameter, nämlich ROT und SCALE.

Also, bevor Sie die Cursoren in Betrieb setzen, den Befehl SCALE = 1 nicht vergessen. Dieser Befehl setzt nur den Zero-Page-Pointer \$E7 (231) auf den Wert 1, so daß auch ein POKE 231,1 zu demselben Ergebnis führt. Im Teil 4 auf S. 6 in diesem Heft ist dieser Patch nicht erforderlich.

Ich hoffe, hiermit dem Leser manchen Frust zu ersparen.

*N. G. Barbieri*