Unterprogrammbibliothek Exsort — Sortieren mit Komfort

Exsort zeichnet sich zum einen durch die Sortiergeschwindigkeit aus. Zum anderen werden sowohl numerische als auch alphanumerische Felder auf- oder absteigend sortiert. Ein weiterer Clou: Ein zweites Feld kann abhängig vom ersten Feld mitsortiert werden.

Jeder Programmierer ärgert sich irgendwann einmal über das langsame Basic, das vor allem beim Suchen und Sortieren stört. Gute Sortierroutinen, in Assembler geschrieben, kann nicht jeder entwickeln. Viele Sort-Programme sind aber auch sehr einseitig: Entweder sortieren sie nur aufsteigend oder lediglich alphanumerische Felder. Exsort kann beides und noch mehr.

Vorteile:

- Zirka zehnmal so schnell wie die schnellste Basic-Version.
- Die Befehle können in jedem Basic-Programm angewendet werden.
- Unterprogramme in Basic, die oft nur ein bestimmtes Feld in einer Richtung sortieren können, entfallen.
- Die Erweiterung belegt keinen Basic-Speicher.
- Beim Sortieren von Strings kommt es nicht zu einem zeitraubenden Garbage-Collect, da die Descriptoren vertauscht werden.
- Ein zweites Feld, das Informationen über das erste Feld enthält, kann mitsortiert werden.
- Das zu sortierende Feld kann numerisch oder alphanumerisch sein.

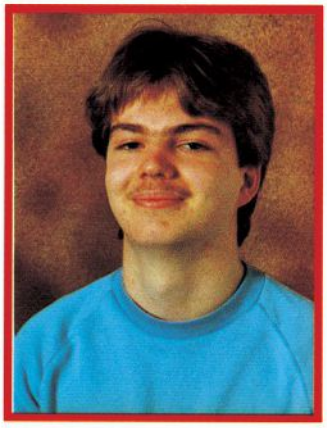
Nachteile:

— Es kann nicht mit Exbasic oder Simons Basic zusammen genutzt werden.
— Es kann nicht compiliert werden.

Weiter auf Seite 156.

```
0 REM ********************************
1 REM **MASCHINENCODE SORTIERROUTINE **
2 REM **                            **
3 REM **      VON MARCUS RICKERT    **
4 REM ********************************
100 DATA169,3,141,8,3,169,193,141,9,3,96
,160,1,177,122,1395
110 DATA201,83,240,3,76,228,167,200,177,
122,201,69,208,3,76,2054
120 DATA218,197,201,79,208,239,32,115,0,
32,115,0,32,115,0,1583
130 DATA32,253,174,32,236,196,32,36,195,
76,88,193,32,253,174,2002
140 DATA32,138,173,32,247,183,165,20,141
,0,192,165,21,141,1,1651
150 DATA192,32,253,174,32,138,173,32,247
,183,165,20,141,2,192,1976
160 DATA165,21,141,3,192,96,32,49,193,32
,253,174,32,158,183,1724
170 DATA142,4,192,224,2,144,7,162,120,16
0,197,76,250,196,32,1908
180 DATA199,195,142,6,192,140,7,192,141,
8,192,160,0,177,122,1873
190 DATA240,4,201,58,208,8,169,0,141,11,
192,76,162,193,32,1695
200 DATA253,174,32,236,196,32,36,195,32,
199,195,142,9,192,140,2063
210 DATA10,192,141,11,192,169,0,141,5,19
2,160,0,32,60,197,1502
220 DATA160,2,32,60,197,169,0,141,246,19
2,141,247,192,160,18,1957
230 DATA32,80,197,144,3,76,174,167,160,1
6,32,80,197,162,16,1536
240 DATA160,18,32,42,197,142,20,192,140,
21,192,162,20,160,246,1744
250 DATA32,154,196,240,219,48,217,173,16
,192,141,12,192,173,17,2022
260 DATA192,141,13,192,173,18,192,141,14
,192,173,19,192,141,15,1808
270 DATA192,32,239,195,160,6,174,8,192,2
32,138,162,12,32,1,1775
280 DATA197,32,47,196,240,23,72,173,4,19
2,208,6,104,16,14,1524
290 DATA76,27,194,104,48,8,162,12,32,227
,196,76,251,193,160,1766
300 DATA6,174,8,192,232,138,162,14,32,1,
197,32,47,196,240,1671
310 DATA23,72,173,4,192,208,6,104,48,14,
76,67,194,104,16,1301
320 DATA8,162,14,32,209,196,76,35,194,16
2,12,160,14,32,154,1460
330 DATA196,240,2,16,103,160,6,174,8,192
,232,138,162,12,32,1673
340 DATA1,197,134,251,132,252,160,6,174,
8,192,232,138,162,14,2053
350 DATA32,1,197,134,253,132,254,172,8,1
92,32,105,197,173,11,1893
360 DATA192,240,38,160,9,174,11,192,232,
138,162,12,32,1,197,1790
370 DATA134,251,132,252,160,9,174,11,192
,232,138,162,14,32,1,1894
380 DATA197,134,253,132,254,172,11,192,3
2,105,197,162,12,32,227,2112
390 DATA196,162,14,32,209,196,162,14,160
,12,32,154,196,48,3,1590
400 DATA76,251,193,162,14,160,16,32,42,1
97,142,20,192,140,21,1658
410 DATA192,162,18,160,12,32,42,197,142,
22,192,140,23,192,162,1688
420 DATA20,160,22,32,154,196,16,34,162,1
2,160,18,32,154,196,1368
430 DATA16,10,160,12,32,60,197,160,18,32
,60,197,173,14,192,1333
440 DATA141,18,192,173,15,192,141,19,192
,76,200,193,162,16,160,1890
450 DATA14,32,154,196,16,10,160,16,32,60
,197,160,14,32,60,1153
460 DATA197,173,12,192,141,16,192,173,13
,192,141,17,192,76,200,1927
470 DATA193,169,4,141,20,192,162,0,134,3
,134,4,32,115,0,1303
480 DATA240,63,201,58,240,59,201,36,240,
20,201,37,240,24,201,2061
490 DATA44,240,47,21,3,149,3,224,2,240,3
2,232,76,47,195,1555
500 DATA169,2,141,20,192,76,99,195,169,1
,141,20,192,165,3,1585
510 DATA9,128,133,3,165,4,9,128,133,4,76
,47,195,162,158,1354
520 DATA160,197,76,250,196,165,47,133,25
1,165,48,133,252,76,164,2313
530 DATA195,160,0,177,251,197,3,208,10,2
00,177,251,197,4,208,2238
540 DATA3,76,183,195,160,2,24,165,251,11
```

Sortierter Lebenslauf:



Ich wurde am 26.10.1967 in Köln geboren und wohne zur Zeit in Bergisch Gladbach. Ich besuche die elfte Klasse des Johann-Gottfried-Herder-Gymnasiums in Köln-Buchheim. Zum Computern kam ich durch meine Liebe zur Mathematik. Es hat mit kleinen Taschenrechnern angefangen, ging über die ersten programmierbaren Rechner (32,128,512 Schritte) zum ersten Heimcomputer (ZX81) und schließlich zum C 64, den ich im Oktober letzten Jahres als Geburtstagsgeschenk erhielt. Später folgten Floppy und Drucker 1526. Auf dem ZX81 machte ich Bekanntschaft mit Assembler (Z-80) und programmierte kleine Programme, die aber oft nur 100—300 Bytes lang waren. Erst auf dem C 64 begann ich längere Maschinenprogramme zu schreiben, da ich mit meinem Monitor arbeiten konnte.

Zur Entstehungsgeschichte des Programms:

Ich brauchte für ein Datenverwaltungsprogramm eine Sortierroutine. Ich kannte zu diesem Zeitpunkt nur das BUBBLE-Sort-Verfahren. Auf die Basic-Version programmierte ich eine in Assembler, die zwar bei Dateien unter 50 Einträgen annehmbare Zeiten lieferte, aber bei größeren Datenmengen zeitlich versagte. Dann bekam ich von einem Freund eine Kopie des QUICKSORT-Algorithmus, der schon in Basic sehr schnell ist. Nachdem ich das Sortiersystem begriffen hatte, konnte ich es in Maschinensprache umschreiben und es mit einigen Extras versehen. Das Ergebnis ist mein Programm »Exsort«, das ich inzwischen in mein Datenverwaltungsprogramm integriert habe.

Ich hoffe, daß es vielen Lesern des 64'er-Magazins eine komfortable Hilfe ist. M. Rickert

```
3,251,170,8,200,40,1841
550 DATA165,252,113,251,133,252,138,133,
251,165,251,197,49,208,212,2770
560 DATA165,252,197,50,208,206,162,143,1
60,197,76,250,196,160,4,2426
570 DATA177,251,201,1,208,1,96,162,185,1
60,197,76,250,196,24,2185
580 DATA160,6,177,251,237,2,192,8,136,17
7,251,40,237,3,192,2069
590 DATA176,7,162,0,169,18,76,59,164,24,
165,251,105,7,170,1553
600 DATA165,252,105,0,168,173,20,192,96,
24,173,16,192,109,18,1703
610 DATA192,170,173,17,192,109,19,192,74
,141,21,192,138,106,141,1877
620 DATA20,192,174,8,192,232,138,160,6,1
62,20,32,1,197,134,1668
630 DATA40,132,41,173,8,192,201,4,208,4,
138,76,162,187,174,1740
640 DATA8,192,138,168,177,40,149,97,202,
136,16,248,96,134,40,1841
650 DATA132,41,173,8,192,201,2,240,9,201
,1,240,65,138,32,1675
660 DATA91,188,96,160,1,177,40,133,3,200
,177,40,133,4,160,1603
670 DATA0,76,98,196,177,3,209,98,240,8,1
76,3,169,1,96,1550
680 DATA169,255,96,200,152,160,0,209,40,
240,8,197,97,240,4,2067
690 DATA168,76,83,196,177,40,197,97,240,
5,176,229,76,91,196,2047
700 DATA169,0,96,165,97,141,21,192,165,9
8,141,20,192,160,0,1657
710 DATA177,40,141,23,192,200,177,40,141
,22,192,162,20,160,22,1709
720 DATA189,0,192,217,0,192,208,11,189,1
,192,217,1,192,208,2009
730 DATA3,76,124,196,189,1,192,48,23,185
,1,192,48,164,56,1498
740 DATA189,0,192,249,0,192,189,1,192,24
9,1,192,144,152,76,2018
750 DATA91,196,185,1,192,16,144,76,183,1
96,56,189,0,192,233,1950
760 DATA1,157,0,192,189,1,192,233,0,157,
1,192,96,254,0,1665
770 DATA192,208,3,254,1,192,96,165,122,5
6,233,1,133,122,165,1943
780 DATA123,233,0,133,123,96,134,34,132,
35,76,71,164,133,40,1527
790 DATA189,0,192,133,113,189,1,192,133,
114,169,0,133,41,152,1751
800 DATA72,32,87,179,134,40,132,41,104,1
68,24,165,40,121,0,1339
810 DATA192,170,165,41,121,1,192,168,96,
56,185,0,192,253,0,1832
820 DATA192,72,185,1,192,253,1,192,168,1
04,170,96,174,5,192,1997
830 DATA185,0,192,157,25,192,185,1,192,1
57,136,192,232,142,5,1993
840 DATA192,96,174,5,192,202,16,2,56,96,
189,25,192,153,0,1590
850 DATA192,189,136,192,153,1,192,142,5,
192,24,96,177,251,170,2112
860 DATA177,253,145,251,138,145,253,136,
16,243,96,255,87,82,79,2356
870 DATA78,71,32,83,79,82,84,73,78,71,32
,68,73,82,69,1055
880 DATA67,84,73,79,206,65,82,82,65,89,3
2,78,79,84,32,1197
890 DATA70,79,85,78,196,87,82,79,78,71,3
2,65,82,82,65,1231
900 DATA89,32,78,65,77,197,87,82,79,78,7
1,32,73,78,68,1186
910 DATA69,216,79,78,76,89,32,79,78,69,3
2,68,73,77,69,1184
920 DATA78,83,73,79,78,32,65,82,82,65,21
7,0,255,0,255,1444
930 DATA0,255,32,255,0,32,115,0,32,115,0
,32,115,0,32,1015
940 DATA253,174,32,236,196,32,36,195,32,
49,193,32,199,195,142,1996
950 DATA6,192,140,7,192,141,8,192,32,253
,174,32,158,173,162,1862
960 DATA2,160,0,32,154,196,16,7,162,174,
160,197,76,250,196,1782
970 DATA173,8,192,201,2,240,20,32,141,17
3,173,8,192,201,4,1760
980 DATA240,23,32,170,177,133,97,132,98,
76,57,198,32,163,182,1810
990 DATA133,97,165,34,133,98,165,35,133,
99,174,8,192,232,138,1836
1000 DATA160,6,162,0,32,1,197,142,9,192,
140,10,192,174,9,1426
```

Listing 1. »Exsort data«

Unterprogrammbibliothek:

```
1010 DATA192,172,10,192,32,47,196,240,42
,162,0,160,2,32,154,1633
1020 DATA196,240,26,162,0,32,227,196,56,
173,9,192,109,8,192,1818
1030 DATA141,9,192,173,10,192,105,0,141,
10,192,76,75,198,169,1683
1040 DATA255,160,255,76,134,198,173,1,19
2,172,0,192,32,145,179,2164
1050 DATA169,0,133,13,133,14,169,73,133,
69,169,78,133,70,32,1388
1060 DATA231,176,166,71,164,72,32,212,18
7,76,174,167,255,0,255,2238
1100 ZN=100:Z=49400:RESTORE:REM ** EINLE
SEN DER WERTE **
1110 PR=0
1120 FORS=0TO14:READX:PR=PR+X:POKEZ+S,X:
NEXTS
1130 READX:IFX=PRTHEN1150
1140 PRINTCHR$(14)"■PRUEFSUMMEN-FEHLER I
N ♦EILE";ZN:END
1150 ZN=ZN+10:Z=Z+15
1160 IFZ<>50855THEN1110
1170 PRINT"FERTIG"
1180 INPUT"■CASSETTE(1) ODER DISKETTE(8)
";PN
1190 POKE251,PN:REM ** ABSPEICHERN ALS A
BSOLUTPROGRAMM **
1200 POKE252,PEEK(45):POKE253,PEEK(46)
1210 POKE43,248:POKE44,192
1220 POKE45,172:POKE46,198
1230 IFPEEK(251)=1THENSAVE"EXSORT",1,2
1240 IFPEEK(251)<>1THENSAVE"@:EXSORT",PE
EK(251)
1250 POKE43,1:POKE44,8:POKE45,PEEK(252):
POKE46,PEEK(253)
READY.
```

Listing 1. »Exsort data« (Schluß)

1. Befehl »so«

Syntax: so, (feldname), (anfangsindex), (endindex), (sortierungsrichtung)

Dieser Befehl sortiert ein beliebiges eindimensionales Feld innerhalb von zwei Grenzen mit einer vom Benutzer gewählten Sortierungsrichtung.

Beispiel 1:

Das Feld heißt ax$, alphanumerisch aufsteigend sortieren (von Index 100 bis Index 5000).

Befehl: so,ax$,100,5000,1 (1= aufsteigend)

Beispiel 2:

Das Feld heißt qe%, numerisch absteigend sortieren (von Index 0 bis zu dem Index, der in der Variable »en« enthalten ist).

Befehl: so,qe%,0,en,0 (0=absteigend)

Option: Manchmal ist es notwendig, daß Daten, die in einem zweiten Feld vorhanden sind, entsprechend dem ersten Feld sortiert werden.

Syntax: so,(feldname 1), (anfangsindex), (endindex), (sortierungsrichtung), (feldname2)

Beispiel: Das Feld fe$ soll alphanumerisch aufsteigend von Index 0 bis Index 10 sortiert werden. Die Daten in dem Realfeld »nr« sollen entsprechend dem ersten Feld sortiert werden.

Befehl: so,fe$,0,10,1,nr

2. Befehl: »se«

Syntax: se,(feldname),(anfangsindex),(endindex),(element)

Dieser Befehl durchsucht ein beliebiges eindimensionales Feld innerhalb von zwei Grenzen nach einem Element.

Beispiel: Es soll die Zahl - 12 in dem Feld rt% von Index 0 bis Index 100 gesucht werden.

Befehl: se,rt%,0,100, - 12

Wenn das Element gefunden wird, enthält die Variable »in« den jeweiligen Index. Wird das Element nicht gefunden, so enthält »in« den Wert —1.

```
0 IFK=0THENK=1:LOAD"EXSORT?",8,1:REM LAD
EN VON EXSORT
1 SYS49400:REM STARTEN VON EXSORT
100 REM ************************
110 REM ***   EXSORT DEMO   ***
120 REM ************************
130 REM
140 REM ******************
150 REM * 1.BEFEHL: "SO" *
160 REM ******************
165 PRINT"■ERSTER BEFEHL: 'SO'"
170 INPUT"■ZAHL DER ZU SORTIERENDEN ELEM
ENTE";A
175 PRINT"■ANFUELLEN DES FELDES 'ZA' MIT
 ZUFALLS-    ZAHLEN"
180 DIMZA(A)
190 REM *** DAS FELD ZA WIRD MIT ZUFALLS
ZAHLEN BELEGT ***
200 FORS=1TOA
210 :ZA(S)=RND(1)*10000-5000
220 NEXTS
230 IT=TI:REM ZEIT SPEICHERN
240 PRINT"■SORTIERBEGINN"
250 REM
260 REM *** AUFRUF DES BEFEHLS "SO" ***
270 SO,ZA,1,A,1
280 REM   SO   =   BEFEHL
290 REM   ZA   =   FELDNAME
300 REM   1    =   ANFANGSINDEX
310 REM   A    =   ENDINDEX
320 REM   1    =   SORTIERUNGSRICHTUNG(A
UFSTEIGEND)
330 REM
340 IT=TI-IT
350 PRINT"■SORTIERENDE"
355 FORS=1TO1000:NEXTS
360 REM *** AUSGABE DER SORTIERTEN ELEME
NTE ***
370 FORS=1TOA
380 :PRINTS,TAB(6)ZA(S)
390 NEXTS
400 PRINT"■♦EIT:"IT/60"SEC"
410 PRINT"■BITTE TASTE DRUECKEN"
420 GETT$:IFT$=""THEN420
430 REM
440 REM ******************
450 REM * 2.BEFEHL: "SE" *
460 REM ******************
470 REM
480 CLR:DIMFE$(10000)
485 PRINT"■ZWEITER BEFEHL 'SE'"
490 REM ** IN 50 BELIEBIGE ELEMENTE  **
500 REM ** DES FELDES FE$ WIRD DAS   **
510 REM ** WORT "HALLO" GESCHRIEBEN  **
520 REM
525 PRINT"■IN 50 BELIEBIGE ELEMENTE VON
FE$ WIRD   'HALLO' GESCHRIEBEN"
530 FORS=1TO50
540 :FE$(RND(1)*10000)="HALLO"
550 NEXTS
560 PRINT"■IN FOLGENDEM ELEMENTEN STEHT
'HALLO':"
570 REM
580 REM ** AUSDRUCKEN JEDES INDEXES **
590 REM ** IN DEM "HALLO" STEHT     **
600 REM
610 IN=-1:IT=TI
```

Listing 2. »Exsort demo«

Exsort — Sortieren mit Komfort

```
620 REM ** AUFRUF DES BEFEHLS "SE" **
630 SE,FE$,IN+1,10000,"HALLO"
640 REM  SE       =   BEFEHL
650 REM  FE$      =   FELDNAME
660 REM  IN+1     =   ANFANGSINDEX
670 REM  10000    =   ENDINDEX
680 REM  "HALLO"  =   ELEMENT
690 REM ** BEI RUECKKEHR AUS "SE"  **
700 REM ** ENTHAELT "IN" DEN INDEX **
710 REM ** ODER (WENN DAS ELEMENT  **
720 REM ** NICHT GEFUNDEN WURDE)   **
730 REM ** DEN WERT -1             **
740 IFIN=-1ORIN=10000THEN760
750 PRINTIN,:GOTO630
760 PRINT:PRINT"■ZEIT:"(TI-IT)/60"SEC"
770 PRINT"■BITTE TASTE DRUECKEN"
780 GETT$:IFT$=""THEN780
790 REM
800 REM ****************************
810 REM * 1.BEFEHL 'SO' MIT OPTION *
820 REM ****************************
830 REM
835 PRINT"■ERSTER BEFEHL MIT OPTION"
840 DATANULL,ZWEI,VIER,SECHS,ACHT,ZEHN,E
INS,DREI,FUENF,SIEBEN,NEUN
850 DATA0,2,4,6,8,10,1,3,5,7,9
860 CLR:DIMNR(10),NR$(10)
870 REM ** EINLESEN IN FELD NR$ **
880 FORS=0TO10
890 :READX$:NR$(S)=X$
900 NEXTS
910 REM ** EINLESEN IN FELD NR  **
920 FORS=0TO10
930 :READX:NR(S)=X
940 NEXTS
950 REM ** AUSGABE FELD VOR SORTIERUNG *
*
960 PRINT"■INDEX NR$ VORHER NR * NR$ NAC
HHER NR"
970 FORS=0TO10
980 :PRINTS;TAB(6)NR$(S)TAB(16)NR(S)
990 NEXTS
1000 REM ** AUFRUF DES BEFEHL "SO" MIT O
PTION **
1010 SO,NR$,0,10,0,NR
1020 REM SO    =   BEFEHL
1030 REM NR$   =   FELDNAME 1
1040 REM 0     =   ANFANGSINDEX
1050 REM 10    =   ENDINDEX
1060 REM 0     =   SORTIERUNGSRICHUTNG(AB
STEIGEND)
1070 REM NR    =   FELDNAME 2
1080 REM
1090 PRINT"■SORTIEREN VON NR$ ABSTEIGEND
"
1095 PRINT"■NR WIRD ENTSPRECHEND MITSORT
IERT"
1097 PRINT"■BITTE TASTE DRUECKEN"
1098 GETT$:IFT$=""THEN1098
1100 REM ** AUSGABE FELD NACH SORTIERUNG
 **
1110 PRINT"■■■■";
1120 FORS=0TO10
1130 :PRINTTAB(22)NR$(S)TAB(33)NR(S)
1140 NEXTS
1150 PRINT"■■■■■■"
READY.
```

Listing 2. »Exsort demo« (Schluß)

Fehlermeldungen:

■ type mismatch:
Sie versuchten, einen String in einem numerischen Feld zu suchen (oder umgekehrt).

■ wrong index:
Beim Suchen war der Anfangsindex größer als der Endindex.

■ bad subscript:
Index außerhalb des zulässigen Bereiches.

■ only one dimension array:
Sie können nur eindimensionale Felder durchsuchen oder sortieren.

■ array not found:
Das Feld war nicht durch einen DIM-Befehl dimensioniert worden.

■ wrong array name:
Geben Sie bitte nur die ersten beiden Buchstaben des Feldnamen ein (plus % oder $ wenn nötig). Es wird dann sicher funktionieren.

■ wrong sorting direction error:
Sie haben einen anderen Wert als 0 oder 1 als Sortierungsrichtung angegeben.

Zu den Programmen:

Listing 1

Das Programm »Exsort data« erstellt das Maschinenprogramm aus DATA-Zeilen und speichert es als »Exsort«-Absolutprogramm auf Diskette oder Kassette. Sie können es dann jederzeit durch LOAD »Exsort«, 8,1 absolut laden. Dabei geht ein Basic-Programm nicht verloren (siehe auch Listing 2, Demo-Programm).

Listing 2

Das Programm »Exsort demo« lädt das Absolutprogramm »Exsort« nach und startet es. Danach folgt eine Demonstration der beiden Befehle.

Um »Exsort« zu laden, muß Zeile 0 des Basic-Programms lauten:

0 if k=0 then k=1 : load"exsort",8,1 (für Diskette)

0 if k=0 then k=1 : load"exsort",1,1 (für Kassette)

In Zeile 1 muß stehen:

1 sys 49400

Da die Erweiterung nur einmal geladen und gestartet werden muß, kann sie bei späteren Starts des Programms übersprungen werden. (Marcus Rickert/gk)

Fortsetzung von Seite 31.

Wie super ist die Supergraphik?

gramme angehängt werden; so ist zum Beispiel ein komfortables Nachladen der Spritedaten möglich. RENUM ist ein Zeilennumerierungs-Befehl; er gleicht auch alle Sprungadressen hinter GOTO, GOSUB an. Ihre Funktionstasten können Sie mit KEY belegen. Sie sind relativ sinnvoll vorbelegt; so ist mit F5/F7 eine Umschaltung zwischen Textseite und Grafikseite möglich. DTASET ist ein gezieltes RESTORE; da hier auch arithmetische Ausdrücke vorkommen dürfen, ist eine Angleichung beim RENUM-Befehl nicht implementiert. Neben der Joystickabfrage ist auch eine Paddleabfrage über PADDLE realisiert worden. Als letztes noch der Befehl POS=; mit ihm kann im Textmodus der Cursor einfach auf beliebige Spalten und Zeilen gesetzt werden.

Ein anderes Detail, das mir allerdings sehr gut gefallen hat: Bei etwaigen Fehlermeldungen wird von der Grafikseite wieder automatisch auf die Textseite umgeschaltet. Das hält Sie beim Austesten eines Programms immer auf dem Laufenden.

Supergraphik 64 ist ein Programm von Data Becker für 99 Mark, das einerseits durch seine Befehlsvielfalt, andererseits durch einige kleine Details beeindruckt. Einige Mängel schwächen den positiven Eindruck allerdings ein wenig ab. Man sollte aber auf jeden Fall beim Kauf einer Grafikerweiterung Supergraphik 64 in Erwägung ziehen, insbesondere, wenn man es nicht so sehr auf komfortables Sprite-Handling, sondern eher auf ein gutes Werkzeug für hochauflösende Grafiken abgesehen hat.

(Boris Schneider/aa)