```
1010 x$=f$+str$(x)
1020 ifx=int(x)thenx$=x$+".00"
1030 ifmid$(x$,len(x$)-1,1)="."thenx$=x$+"0"
1040 x$=right$(x$,9)
1050 return
```

Zu Beginn des Programms wird ein Füllstring definiert. Zur Aufbereitung der Variablen X wird ins Unterprogramm verzweigt. Zeile 1000 rundet X auf die Anzahl der Nachkommastellen (hier 2). Zeile 1010 wandelt X in eine Stringvariable und stellt ihr den Füllstring voran. In den Zeilen 1020 und 1030 werden bei Bedarf die Nachkommastellen auf zwei aufgefüllt. Zeile 1040 schneidet den String der passenden Länge heraus, hier sechs Vor- und zwei Nachkommastellen plus Dezimalpunkt.

(H. G. Sander/tr)

GET-Befehl sinnvoll angewendet

Hier ist ein sehr praktisches Unterprogramm, wenn in einem Programm eine Taste an verschiedenen Stellen gedrückt werden soll (in einem Menü wählen — eine Frage mit 'J/N' beantworten — und so weiter).

Bevor man die Routine durch ein »GOSUB« aufruft, werden die zugelassenen Tasten in der Variable ZT$ definiert (zum Beispiel: ZT$="1234", oder ZT$="JN" oder ZT$=CHR$(...)).

In der Zeile 10000 wird zuerst der Zähler »OK« auf Null gestellt; in Zeile 10010 folgt der übliche »GET«-Befehl mit der Stringvariablen OK$.

In der Zeile 10020 wird überprüft, ob die gedrückte Taste einer der in ZT$ definierten entspricht, bis eventuell der ganze String überprüft worden ist (Zeile 10030). Ist nichts gefunden worden, wird auf einen neuen Tastendruck gewartet (Zeile 10040).

Ist der Test in Zeile 10020 positiv, wird ins Hauptprogramm zurückgesprungen — mit folgenden Informationen:
— OK$ enthält die gedrückte Taste
— OK enthält den Rang von OK$ im String ZT$ (für ein »ON OK GOTO/GOSUB«)
— AW enthält den ASCII-Wert von OK$ (kann natürlich weggelassen werden).

```
10000 OK=0
10010 GET OK$:IF OK$="" "THEN 10010
10020 OK=OK+1:IF MID$(ZT$,OK,1)=OK$ THEN AW=ASC(OK$):RETURN
10030 IF OK<LEN(ZT$) THEN 10020
10040 GOTO 10000
```

(G. Gartner/tr)

Tips & Tricks für Profis

Wie nützt man am besten den $C000-Bereich für Basic-Programme? Wir zeigen Ihnen eine wirklich geniale Lösung und viele weitere interessante und trickreiche C 64-Leckerbissen.

Übrigens: Vor kurzem wollten wir in der Redaktion einen »too many files«-Error hervorrufen. Wir verwendeten dazu folgenden Einzeiler:

```
10 FOR I=1 TO 20 : OPEN I,2 : NEXT
```

Völlig ungläubig starrten wir auf die nach dem »RUN« ausgegebene Fehlermeldung unseres C 64. Sie lautete nämlich... Halt! Probieren Sie es selbst einmal aus. Kennen Sie die Erklärung? Auflösung folgt in der nächsten Ausgabe.

Auffrisierter SYS-Befehl

Wer selbst Basic-Erweiterungen programmiert, wird bei zeitkritischen Befehlen (zum Beispiel einer Plot-Routine) schnell an die Grenzen des SYS-Befehls mit Parameterübergabe stoßen. Er ist aufgrund der Adreßumrechnung (Sprungadresse aus dem Basic-Text holen, umrechnen, auf Fehler testen und einen JMP ausführen) manchmal zu langsam. Eine besonders geniale Methode bietet sich als Alternative an: die USR-Funktion. Sie ist um einen wesentlichen Faktor schneller als der SYS-Befehl und eine Parameterübergabe ist ebenso möglich (USR(x),y,z). Ein weiterer Vorteil: Manche Compiler verweigern den SYS-Befehl mit nachfolgenden Parametern. Ein Compiler, der die USR-Funktion nicht verarbeitet, ist uns hingegen nicht bekannt. Bei Befehlserweiterungen mit mehreren Befehlen kann dem Maschinenprogramm über den Übergabeparameter x mitgeteilt werden, welcher Befehl gewünscht wird. Übrigens: Der USR-Vektor steht in 785/786 dezimal. Das Maschinenprogramm müßte dann so aussehen:

```
JSR $B7F7   USR-Argument nach $0014/$0015 holen...
LDA $14     und schon steht der Parameter x zur
            Weiterverarbeitung bereit!
```

Das High-Byte ($0015) der Übergabevariablen x wird bei dieser Methode natürlich nicht berücksichtigt, aber wer hat schon eine Befehlserweiterung mit mehr als 256 Befehlen...

(Hartmut Kroos/tr)

LOAD-Schutz einmal anders

Kannten Sie den schon? Verwenden Sie zur Speicherung eines Programms auf Floppy folgenden Dateinamen:

```
SAVE CHR$(34),8
```

Im Directory erscheint dieses File dann mit dem Dateinamen "". Versuchen Sie nun einmal, dieses Programm wieder zu laden. Richtig! Dies geht nur mehr mit

```
LOAD CHR$(34),8
```

(Stefan Bültena/tr)

Hilfe beim Programme-Abtippen

Wenn man ein Programm abtippen möchte, sieht man sich öfters einer DATA-Wüste gegenüberstehen, beziehungsweise -sitzen. Das dabei am meisten vorkommende Zeichen ist zweifellos das Komma. Dieses liegt aber nun auf einer kleinen Taste auf der Tastatur, die, wenn man nicht dauernd auf die Tastatur schauen will, recht schwer zu erreichen ist. Die Folge ist: Man vertippt sich häufiger. Um das zu ändern, tue man folgendes:

1) ROM ins RAM kopieren

Zum Beispiel mit dem Einzeiler:

```
FOR I = 0 TO 8191 : POKE 40960 + I, PEEK (40960 + I) :
POKE 57344 + I,PEEK (57344 + I) : NEXT
```

Genialer geht es natürlich mit einem Maschinensprache-Monitor und einem Transfer-Befehl. Es müssen die Bereiche $A000 bis $BFFF nach $A000 und $E000 bis $FFFF nach $E000 verschoben werden.

2) Umschalten mit `POKE 1,53`

3) `POKE 60349,44`

Wenn man nun auf die SPACE-Taste drückt, entdeckt man, daß diese Kommanta produziert. Wer Lust hat, kann die Komma-Taste mit POKE 60336,32 mit der SPACE-Funktion belegen. Diese Umbelegung erleichtert das Abtippen um einiges.

(Peter Gorgs/tr)

Grafik-Erweiterung für den LoRes-Bildschirm

Manchmal ist es aus Platzgründen sinnvoll, für grafische Darstellungen nicht die HiRes-Grafik zu verwenden. Unsere kleine Basic-Erweiterung in Listing 1 erlaubt es, mit einem einfachen SYS-Aufruf im normalen Text-Modus einen Pixel (4x4 Punkte) zu setzen.

Syntax: `SYS 49152,X-Wert,Y-Wert,Modus(,Farbe)`

Für X sind Werte von 0 bis 79, für Y von 0 bis 49 zugelassen. »Modus« gibt an, ob der Punkt gesetzt (Modus = 1), gelöscht (Modus = 0), oder getestet (Modus = 255) werden soll. Bei Modus gleich 255 steht das Ergebnis des Tests (1 = gesetzt, 0 = gelöscht) in Speicherzelle 2. Für den Parameter »Farbe« gelten die normalen Farb-POKEs (0 bis 15). Wenn auf »Punkt gesetzt/gelöscht« geprüft werden soll, muß »,Farbe« entfallen. (Georg Brandt/Andreas Wellie/ tr)

```
programm : lores              c000 c136
-----------------------------------

c000 : 20 fd ae 20 9e b7 86 f7  80
c008 : e0 50 90 03 4c 48 b2 20  a7
c010 : fd ae 20 9e b7 86 f8 e0  96
c018 : 32 b0 f1 20 fd ae 20 9e  36
c020 : b7 86 fd e8 f0 08 20 fd  82
c028 : ae 20 9e b7 86 fe a5 f8  6e
c030 : 4a 85 f8 90 03 a9 02 2c  6b
c038 : a9 00 a8 a5 f7 4a 85 f7  98
c040 : 98 90 02 09 01 85 02 a6  54
c048 : f8 bd cd c0 18 6d 88 02  bd
c050 : 85 fc bd f0 ec 85 fb a4  15
c058 : f7 b1 fb a6 fd f0 38 e8  16
c060 : f0 35 a2 0f dd e6 c0 f0  6f
c068 : 05 ca 10 f8 a2 00 a9 00  c6
c070 : a4 02 f0 07 18 69 10 88  50
c078 : 18 90 f7 85 02 8a 18 65  26
c080 : 02 aa bd f6 c0 a4 f7 91  5a
c088 : fb a5 fc 29 03 18 69 d8  02
c090 : 85 fc a5 fe 91 fb 60 48  e7
c098 : a9 00 a4 02 f0 07 18 69  25
c0a0 : 10 88 18 90 f7 aa 68 a0  c4
c0a8 : 00 dd f6 c0 f0 0b e8 c8  09
c0b0 : c0 10 d0 f5 a9 00 85 02  20
c0b8 : 60 24 fd f0 05 a9 01 18  9a
c0c0 : 90 f4 d9 e6 c0 f0 e7 b9  c4
c0c8 : e6 c0 4c 85 c0 00 00 00  de
c0d0 : 00 00 00 00 01 01 01 01  ef
c0d8 : 01 01 02 02 02 02 02 02  57
c0e0 : 02 03 03 03 03 03 20 6c  27
c0e8 : 7b 7c 7e 61 62 e2 e1 ec  0c
c0f0 : fc fe fb ff 7f a0 7e 7f  60
c0f8 : 61 e2 7e 61 fc e2 fb ec  47
c100 : fc a0 fb ec 7f a0 7c e1  9b
c108 : ff 7c e2 ec fe e2 e1 ec  04
c110 : a0 fe fb ff fb a0 7b 62  a6
c118 : 7b ff 61 61 62 ec fe ec  7b
c120 : fc fe a0 ff fc a0 6c 6c  23
c128 : 62 e1 7f fc 62 fb e1 a0  c9
c130 : fc fe fb fe 7f a0 a9 00  2e
```

Listing 1. Befehlserweiterung für eine 400-Punkte-Grafik im Textmodus

```
programm : g.c.-anzeige    cf00 cfc3
-----------------------------------

cf00 : 20 fd ae 20 9e b7 e0 28  4a
cf08 : b0 41 8e 9d cf 20 fd ae  03
cf10 : 20 9e b7 e0 19 b0 34 86  7e
cf18 : 28 a2 28 86 71 a9 00 85  dc
cf20 : 29 85 72 20 57 b3 8a 18  1a
cf28 : 6d 9d cf 90 01 c8 c8 c8  75
cf30 : c8 c8 8c 91 cf 8d 90 cf  fd
cf38 : 8c 94 cf 8d 93 cf 78 a2  93
cf40 : 4e a0 cf 8e 14 03 8c 15  5a
cf48 : 03 58 60 4c 48 b2 ba bc  97
cf50 : 05 01 bd 06 01 aa c9 b5  fe
cf58 : d0 05 98 c9 26 b0 0a 8a  2f
cf60 : c9 b6 d0 13 98 c9 06 b0  6c
cf68 : 0e ad 9e cf d0 14 ee 20  98
cf70 : d0 20 85 cf 4c 82 cf ad  1f
cf78 : 9e cf f0 06 20 85 cf ce  06
cf80 : 20 d0 4c 31 ea 49 ff 8d  56
cf88 : 9e cf a0 23 b9 9f cf be  f0
cf90 : 00 04 99 00 04 8a 99 9f  33
cf98 : cf 88 10 f0 60 00 00 ad  2f
cfa0 : a0 82 89 94 94 85 a0 97  9d
cfa8 : 81 92 94 85 8e ba a0 87  98
cfb0 : 81 92 82 81 87 85 a0 83  79
cfb8 : 8f 8c 8c 85 83 94 89 8f  83
cfc0 : 8e a0 ad 00 00 c2 ae c1  5e
```

Listing 2. Zeigt an, wann der C 64 eine Garbage-Collection durchführt

Will man die Rahmenfarbe jedoch belassen, so sind noch zwei POKEs einzugeben: POKE 53102,44: POKE 53119,44. Das Programm braucht nur einmal aufgerufen zu werden und bleibt die ganze Zeit durch den Gebrauch der Interrupttechnik einsatzbereit. Nach RUN/STOP-RESTORE und RESET muß die Routine wieder initialisiert werden. Listing 3 demonstriert die Anwendung des Programms.

Initialisierung mit: SYS 52992, Spalte, Zeile

(Joachim und Michael Kreutzer/tr)

```
60 PRINT"{CLR,2DOWN,3RIGHT}DIESER TEXT WIR
   D JETZT GLEICH WEGEN"                    <088>
70 PRINT"EINER GARBAGE COLLECTION UEBERSCH
   RIEBEN."                                 <237>
80 DIM A$(200,1):FOR I=1 TO 200:A$(I,1)=CH
   R$(I):NEXT                               <088>
120 PRINT"{3DOWN}<TASTE>":POKE 198,0:WAIT
    198,255                                 <064>
130 SYS 52992,2,3                           <047>
140 A=FRE(0):REM ERZWINGEN EINER GARBAGE C
    OLLECTION                               <194>
150 PRINT"{2DOWN}WIE SIE SEHEN BLIEB DER U
    RSPRUENGLICHE"                          <059>
160 PRINT"TEXT JEDOCH ERHALTEN."            <176>
```

Listing 3. Demo-Programm zur »Garbage-Collection-Anzeige«

Die verflixte Müllabfuhr

Bei einer Garbage Collection zeigt der C 64 keine Reaktion mehr. Man weiß dann nie so genau, ob der Computer abgestürzt oder am Rechnen ist. Besonders schlimm ist diese Tatsache bei der Programmentwicklung oder bei gekauften Basic-Programmen, die viele Daten zu verarbeiten haben (Adreßverwaltung, Vokabelprogramm, Haushaltsbuchhaltung..). Mit dem Programm »Garbage Anzeige« (Listing 2) wollen wir nun eine Möglichkeit angeben, diese Unsicherheit aus dem Weg zu räumen. Sobald der Computer dabei ist, den Stringmüll zu sortieren, zeigt das Programm automatisch einen Text. Ist die Garbage Collection beendet, sieht der Bildschirm genauso aus wie vorher, das heißt der überschriebene Text wird wieder in unveränderter Form auf den Bildschirm gebracht. Ferner kann man vorher genau bestimmen, wo der Text erscheinen soll (Spalte, Zeile). Zusätzlich wird bei jedem Aufruf die Rahmenfarbe um eins erhöht. Selbstverständlich wird diese bei Ende der Garbage Collection sofort wieder zurückgesetzt.

43007 Bytes free!

»BASIC $B000« (Listing 4) verschiebt den Basic-Interpreter von $A000-$BFFF nach $B000-$CFFF und ändert alle benötigten Parameter beziehungsweise Werte und Adressen. Somit meldet sich anschließend eine Basic-Version mit 43007 statt 38911 Bytes free. Der Vorteil gegenüber anderen Basic-RAM-Erweiterungen liegt darin, daß die 43007 Bytes an einem Stück liegen, und somit in einer beliebigen Aufteilung für Variablen oder Programm zur Verfügung stehen. Die zusätzlichen 4 KByte RAM sind für Adventures oder Datenverwaltungsprogramme äußerst nützlich. Als Beispielprogramm diene das Adventure »Buch der Weisheit« aus dem Sonderheft 3/85, Spiele. Dieses Adventure ist normalerweise nur in einer compilierten Fassung lauffähig, sonst meldet es sich mit einem »OUT OF MEMORY ERROR«. Mit Basic $B000 ist auch die Basic-Version lauffähig. Es muß allerdings darauf hingewiesen werden, daß nur reine Basic-Programme ohne Maschinen-Routinen mit Basic $B000 verwendet werden dürfen!

```
programm : basic $b000      0801 0a8d
-------------------------------------

0801 : 1a 08 c2 07 9e 32 35 36  6e
0809 : 30 3a 12 20 42 41 53 49  ed
0811 : 43 20 24 42 30 30 30 20  3b
0819 : 00 00 00 a9 a0 85 44 a9  e9
0821 : b0 85 46 a9 e0 85 48 a0  f7
0829 : 00 84 43 84 45 84 47 b1  c6
0831 : 43 91 45 b1 47 91 47 c8  74
0839 : d0 f5 e6 44 e6 46 e6 48  13
0841 : d0 ed a9 35 85 01 a9 b0  81
0849 : 8d 01 fd 8d 8b fd 8d 71  4a
0851 : fe a0 0d a2 81 20 c6 08  7b
0859 : a9 b3 a0 29 a2 65 20 c6  8c
0861 : 08 a9 e4 a0 48 a2 54 20  b6
0869 : c6 08 a9 c8 8d d7 08 a9  c2
0871 : b0 a0 82 a2 a0 8e 8a fd  0b
0879 : 20 c6 08 a9 b3 a0 8a a2  e3
0881 : d0 20 f9 08 a9 e0 a0 00  05
0889 : a2 e5 8c 89 fd 8e d6 fd  1e
0891 : 20 f9 08 a9 a2 8d 88 fd  9a
0899 : a9 b3 a2 d0 a0 00 20 36  d5
08a1 : 09 a0 8a 20 3d 09 a9 e0  25
08a9 : a2 e3 a0 3d 20 36 09 a0  26
08b1 : 00 20 3d 09 a9 e3 a2 e4  40
08b9 : a0 5e 20 36 09 a0 7b 20  1b
08c1 : 3d 09 6c fc ff 85 44 8e  98
08c9 : d9 08 b1 43 c9 e0 b0 04  e9
08d1 : 69 10 91 43 c8 c8 ea c0  0f
08d9 : aa d0 ef 60 c8 d0 02 e6  dc
08e1 : 44 a5 46 c5 44 d0 08 c4  b7
08e9 : 45 d0 04 ba e8 e8 9a 60  f0
08f1 : c0 00 d0 02 c6 44 88 60  97
08f9 : 85 44 86 46 b1 43 c9 a9  ba
0901 : d0 2d 20 dd 08 20 dd 08  35
0909 : b1 43 c9 a0 d0 1b 20 dd  04
0911 : 08 b1 43 29 f0 c9 a0 f0  aa
0919 : 04 c9 b0 d0 09 b1 43 18  a3
0921 : 69 10 91 43 d0 09 20 f1  19
0929 : 08 20 f1 08 20 f1 08 20  b1
0931 : dd 08 4c fd 08 85 44 86  b0
0939 : 46 84 45 60 a2 4a b1 43  e8
0941 : dd 7d 09 f0 0f ca d0 f8  ba
0949 : a2 30 dd c7 09 f0 0e ca  59
0951 : d0 f8 f0 03 20 dd 08 20  8b
0959 : dd 08 4c 3d 09 20 dd 08  0e
0961 : 20 dd 08 b1 43 29 f0 c9  7d
0969 : 90 f0 08 c9 a0 f0 04 c9  e2
0971 : b0 d0 e4 b1 43 18 69 10  b3
0979 : 91 43 4c 58 09 69 29 c9  de
0981 : e0 c0 49 a9 a2 a0 09 e9  70
0989 : 65 25 06 24 c5 e4 c4 c6  ab
0991 : 45 e6 a5 a6 a4 46 05 26  64
0999 : 66 e5 85 86 84 75 35 16  19
09a1 : d5 d6 55 f6 b5 b4 56 15  9a
09a9 : 36 76 f5 95 94 b6 96 90  c5
09b1 : b0 f0 30 d0 10 50 70 71  28
09b9 : 31 d1 51 b1 11 f1 91 61  07
09c1 : 21 c1 41 a1 01 e1 81 6d  47
09c9 : 2d 0e 2c cd ec cc ce 4d  cd
09d1 : ee f9 20 ad ae ac 4e 0d  1d
09d9 : 2e 6e ed 8d 8e 8c 7d 3d  29
09e1 : 1e dd de 5d fe bd bc 5e  df
09e9 : 1d 3e 7e fd 9d 79 39 d9  c3
09f1 : 59 b9 be 19 4c 99 6c 00  3d
09f9 : 00 00 00 00 00 00 00 a9  4d
0a01 : 37 85 01 a2 00 bd 48 0a  b3
0a09 : 20 d2 ff e8 e0 29 d0 f5  36
0a11 : 20 9f ff a5 cb c9 27 f0  3f
0a19 : 19 c9 22 d0 f3 a9 b1 8d  28
0a21 : 36 08 a9 b0 8d 49 f0 8d  de
0a29 : 1e f4 a9 90 8d 4e f0 8d  68
0a31 : 23 f4 a2 00 bd 78 0a 20  7f
0a39 : d2 ff e8 e0 12 d0 f5 4c  79
0a41 : 1c 08 2a 2a 2a 2a 2a 93  f5
0a49 : 11 11 11 11 48 41 42 45  6b
0a51 : 4e 20 53 49 45 20 48 59  d7
0a59 : 50 52 41 2d 4c 4f 41 44  95
0a61 : 20 47 45 4c 41 44 45 4e  e8
0a69 : 20 3f 20 28 4a 2f 4e 29  e0
0a71 : 00 00 00 2a 2a 2a 2a 93  7a
0a79 : 11 11 11 11 20 42 49 54  5b
0a81 : 54 45 20 57 41 52 54 45  ed
0a89 : 4e 00 00 00 cc 00 02 c8  3e
```

Listing 4. Durch einen genialen Trick stehen Ihnen »echte« 43007 Bytes für Basic zur Verfügung.

Bedienungsanleitung:

Wenn Sie mit Hypra-Load arbeiten wollen, so laden Sie dies zuerst und starten es. Ansonsten oder anschließend laden Sie »BASIC $B000« und starten es mit »RUN«. Danach wird »Bitte warten« ausgegeben, und nach zirka sieben Sekunden meldet sich »BASIC $B000« mit »43007 Bytes free«. Mit »SYS 64738«, also einem Software-Reset, wird nicht auf das normale Basic umgestellt, sondern »BASIC $B000« neu initialisiert. Nur mit einem Reset-Taster oder »POKE 1,55« kommt man wieder in den Normalmodus. Danach läßt sich das »BASIC $B000« nur mit einem erneuten Laden und Starten wiederbeleben. Da Turbo-Tape-Versionen einen Teil des Bereichs von $C000 bis $CFFF belegen, kann man diese leider nicht mit »BASIC $B000« verwenden, da es selbst diesen Bereich benötigt. Somit ist eine Benutzung von »BASIC $B000« nur mit einer Diskettenstation zu empfehlen. Bei einer Benutzung von Hypra-Load schalten Sie vorher bitte den Drucker aus und laden keine Programme über 169 Blocks beziehungsweise 43007 Bytes Länge.

(Robert Bartz/tr)

Hi-Eddi und Panasonic-Drucker

Mit dieser geänderten Druckroutine läßt sich der KX-P1090 genauso wie ein Epson-Drucker ansprechen. Die Einstellung der DIP-Schalter ist bis auf Schalter 1 identisch mit der Originaleinstellung des Druckers.

1. Änderungsmöglichkeit:

Im Basic-Lader aus der 64'er, Ausgabe 1/85, Seite 66, müssen in Zeile 580 zwei DATA-Werte geändert werden. Die neue Zeile heißt:

```
580 DATA 27,75,255,255,255
```

2. Änderungsmöglichkeit:

Bei dem Maschinenprogramm müssen die Adressen $0E75 und $0E76 überschrieben werden. Dies geschieht am besten mit dem SMON. SMON laden und starten. Die ursprüngliche HI-PRINT-Version laden.

Nun eingeben: `M0E70 0E77`

Jetzt werden in der angezeigten Zeile das 6. und 7. Byte geändert in 4B und FF. Nachdem <Return> gedrückt wurde, kann die geänderte Version auf Diskette gespeichert werden.

(Werner Tüllmann/tr)

Hypra-Platos und FX 80

Für den FX 80 habe ich das »2. PRG« etwas verlängert, um die Zeichengenauigkeit in X-Richtung zu verbessern. Die Zeilenbreite von im Mittel genau 1/10 Zoll erreiche ich dadurch, daß der Vorschub innerhalb von fünf Zeilen dreimal den Wert 22/216 Zoll (1., 3. und 4. Zeile) und zweimal den Wert 21/216 Zoll annimmt. Zusätzlich wird am Ende jeder Zeile LF ausgegeben. Dadurch kann beim Görlitz-Interface die Sekundäradresse 4 (Linearkanal) benutzt werden. (Falls das nicht erwünscht ist, überschreiben Sie $9CED bis $9CF1 mit EA (NOP).) Die dafür notwendigen Änderungen können mit dem MSE eingegeben werden.

Geändert werden muß die Zeile:

```
9A74: 4C C2 9C EA EA 20 33 f3 0A
```

und angefügt werden die Zeilen:

```
9CC0: 00 16 EE C0 9C AD C0 9C 12
9CC8: C9 02 F0 09 C9 05 D0 08 08
9CD0: A9 00 8D C0 0C CE C1 9C 75
9CD8: A9 1B 20 D2 FF A9 33 20 CC
9CE0: D2 FF AD C1 9C 20 D2 FF 6B
9CE8: A9 0D 20 D2 FF A9 0A 20 30
9CF0: D2 FF A9 16 8D C1 9C 4C E1
9CF8: 79 9A AD 66 95 C9 52 D0 89
9D00: 03 4C 81 99 4C E4 9B 20 57
```

(Prof. Erich Pröve/tr)

»DEF FN« sinnvoll eingesetzt

Eine der nützlichsten Funktionen des mageren C 64-Basic führt bei den meisten C 64-Fans zweifellos ein Schattendasein: die DEF FN-Funktion. Das kleine Demo-Programm in Listing 5 zeigt, wie sich fast ausschließlich mit dieser Funktion die Echtzeituhr der CIAs auslesen läßt.

Nebenbei bemerkt: Es gibt unglaublich viele Anwendungsbeispiele für die FN-Anweisung, und sei es nur die LoHi-Zerlegung einer Zahl. Die Joysticks lassen sich damit auslesen, Sprites steuern, Grafikpunkte setzen, und die Register des SID kann man durch gezielte Umrechnungen wesentlich komfortabler programmieren...

Listing 5 ist durch die REM-Zeilen ausreichend dokumentiert, so daß hier auf weitere Erklärungen verzichtet werden kann.

(Wolf Dieter Busch/tr)

```
10 TD=56328:
      REM ECHTZEITUHR SEC/10-REGISTER      <012>
20 POKE TD+6,PEEK(TD+6)AND 127:
      REM 50 HZ EINSTELLEN                 <054>
30 DEF FN U4(X)=(X AND 15):
      REM UNTERE 4 BITS VON X              <186>
40 DEF FN O4(X)=(X AND 240)/16:
      REM OBERE 4 BITS                     <223>
50 DEF FN DC(X)=FN U4(X)+FN O4(X)*10:
      REM WERT BEI BCD-CODIERUNG           <001>
60 DEF FN DI(X)=FN DC(PEEK(X)AND 127):
      REM BCD-INHALT VON X OHNE BIT 7      <074>
70 DEF FN H(X)=FN DI(TD+3):
      REM STUNDEN                          <245>
80 DEF FN M(X)=FN DI(TD+2):
      REM MINUTEN                          <093>
90 DEF FN S(X)=FN DI(TD+1):
      REM SEKUNDEN                         <044>
100 DEF FN DH(X)=X-INT(X/10)*10+INT(X/10)*
    16: REM DEZIMAL NACH BCD               <174>
110 INPUT"UHRZEIT HH,MM,SS";HH,MM,SS       <154>
120 IF HH>12 THEN HH=HH-12                 <053>
130 POKE TD+3,FN DH(HH):REM UHR STELLEN    <177>
140 POKE TD+2,FN DH(MM)                    <029>
150 POKE TD+1,FN DH(SS)                    <009>
160 POKE TD+0,0                            <139>
170 PRINT"{CLR}"                           <158>
180 PRINT"{HOME}"FN H(X)"{LEFT}"FN M(X)"{L
    EFT}"FN S(X)"{LEFT}"PEEK(TD)"{DEL}";   <244>
190 GOTO 180                               <246>
```

Listing 5. Eine sinnvolle Anwendung der FN-Anweisung

Der Super-Autostart

Darauf haben Sie schon lange gewartet: Einen Autostart-Generator, der viele sinnvolle Eigenschaften aufweist. Dazu gehören: Kurzes Listing (sowohl des Generator-Programms als auch des Autostarts selber), einfach in der Anwendung, RUN/STOP-RESTORE- und Reset-Schutz für das fertige Programm und eine eingebaute Codier- und Decodier-Funktion.

Der Autostart-Generator in Listing 6 hat alle genannten Funktionen. Die Anwendung ist äußerst einfach: Abtippen, speichern, absolut laden, »NEW« eintippen. Dann das zu bearbeitende (Basic-)Programm laden und den Autostart mit folgender Zeile aktivieren:

```
SYS 49152,Code,"Haupt-Name","Lader-Name"
```

»Code« ist eine beliebige Zahl zwischen 0 und 255. »Haupt-Name« und »Lader-Name« sind die zukünftigen Namen des

```
programm : autostart        c000 c131
-----------------------------------

c000 : 20 fd ae 20 9e b7 8e ac  0a
c008 : c0 20 fd ae 20 9e ad 20  1b
c010 : a3 b6 8d 6f c0 c9 0d 90  0f
c018 : 03 4c 71 a5 20 bd ff 20  83
c020 : 11 c1 a2 08 86 ba 20 35  e5
c028 : c0 a9 2b a6 2d a4 2e 20  4d
c030 : d8 ff 4c bf c0 a5 2b 85  04
c038 : fb a5 2c 85 fc a0 00 b1  fa
c040 : fb 4d ac c0 91 fb c8 d0  e3
c048 : f6 a5 fc e6 fc c5 2e d0  85
c050 : ee 60 a2 ea 8e 28 03 bd  26
c058 : 77 02 4d 00 03 9d 80 7f  42
c060 : ca 30 f4 a2 04 bd 10 fd  3e
c068 : 9d 04 80 ca 10 f7 a9 0c  00
c070 : a2 0d a0 80 20 bd ff a9  14
c078 : 00 85 9d 20 d5 ff 86 2d  78
c080 : 98 a6 2b 86 fb a4 2c 84  a6
c088 : fc 20 57 a6 a8 b1 fb 4d  e2
c090 : 00 03 91 fb c8 d0 f6 a5  30
c098 : 2e e7 fc 10 f0 20 53 e4  22
c0a0 : 4c ae a7 e2 fc 5e fe 43  cf
c0a8 : 48 44 38 36 00 45 a6 02  b0
c0b0 : 28 43 29 38 36 20 42 59  eb
c0b8 : 20 43 48 44 4c 79 00 20  e5
c0c0 : 35 c0 20 79 00 c9 2c d0  2d
c0c8 : f3 20 73 00 20 9e ad 20  96
c0d0 : a3 b6 c9 00 d0 05 a2 08  11
c0d8 : 4c 37 a4 20 bd ff a2 52  f8
c0e0 : a0 c0 86 ac 84 ad a2 bc  d1
c0e8 : a0 c0 86 ae 84 af a9 61  8f
c0f0 : 85 b9 20 d5 f3 20 8f f6  81
c0f8 : a9 08 20 b1 ff a9 61 20  f7
c100 : 93 ff a9 a6 20 dd ed a9  ce
c108 : 02 20 dd ed a0 00 4c 24  d3
c110 : f6 86 fb 84 fc a8 88 b1  73
c118 : fb 4d ac c0 99 b0 c0 88  30
c120 : 10 f5 a0 03 b9 a3 c0 4d  0a
c128 : ac c0 99 a3 c0 88 10 f4  8a
c130 : 60 ff ff ff ff ff ff ff  90
```

Listing 6. Der Super-Autostart, den Sie schon immer suchten

codierten Hauptteils beziehungsweise des Lade-Programms auf der Diskette. Der Lader ist später mit »,8,1« in den C 64 zu lesen.

Läßt man »,"Lader-Name"« weg, so wird nur der codierte Hauptteil neu gespeichert (wenn man Änderungen am Hauptteil vorgenommen hat). Wichtig ist dann nur, daß die Code-Zahl des Hauptprogramms mit der des Laders übereinstimmt. Im Zweifelsfall sollte man lieber den alten Lader löschen und beide Teile neu generieren.

(Christoph Dautzenberg/tr)

Tips & Tricks zum C 128

Weiter geht's mit den Tips und Tricks. Diesmal dabei: Variablendump, ein FIND-Befehl, C 64-Modus im 2-MHz-Takt, ein Trick zur 1571 und vieles mehr.

Tips und Tricks nutzen jedem Anwender. Wie könnte man seine Maschine sonst ausnutzen, wenn man nicht die kleinen Kniffe wüßte, die dem Computer die Feinheiten entlocken. Lesen Sie hier, wie Sie Ihren C 128 bändigen.

Cursor auch bei GET

Mit dieser kleinen Zeile, die als Erste in einem Programm stehen muß, können Sie beim C 128 auf dem 80-Zeichen-Bildschirm den Cursor bei GET blinken lassen:

```
0 PRINT CHR$(27)+"U"
```

(Thomas Tschink/dm)

Komfortable Joystick-Abfrage

Dieses Programm ermöglicht eine komfortable Joystickabfrage. Statt umständlichen IF..THEN-Anweisungen werden einfach indizierte Variablen benutzt.

Zum Programm

In den Zeilen 40 und 50 werden die Variablen A(X) und B(X) definiert. Der Index X entspricht den Richtungen 1 bis 8 des Joysticks.

In Zeile 60 wird die Variable J mit dem Zustand des Joysticks in Port 1 belegt. Ist der Feuerknopf gedrückt (Richtung + 128), springt das Programm wieder zu Zeile 60 (es würde sonst zu einer Fehlermeldung kommen).

In Zeile 70 werden jetzt die Werte der Variablen A(J) und B(J) zu der X- und Y-Position dazuaddiert. Dabei kommt uns ein Gesetz der Mathematik zugute: Addiert man zu einer positiven eine negative Zahl, kommt dies einer Subtraktion gleich (zum Beispiel: 100 + (-20) = 80).

Mit X und Y können jetzt Sprites, Shapes oder ähnliches bewegt werden (hier ist es ein Sprite).

Ein Beispiel für die Arbeitsweise

Wird der Joystick nach Rechts-Oben bewegt, ist J = 2; also ist A(J) = 1 und B(J) = -1. Angenommen, X = 100 und Y = 90, so ist X + A(J) = 101 ((100 + 1 = 101)) und Y + B(J) = 89 ((90 + (-1) = 89)).