

Tips & Tricks für Einsteiger

Unter dieser neuen Überschrift wollen wir den »Newcomern« unter den Computer-Fans helfen, durch kleine Tricks und Kniffe zu echten Profis zu werden.

Bevor Sie sich jedoch in unsere kleinen Tips und Listings stürzen, eine Bitte an Sie: Bestimmt haben auch Sie noch Schwierigkeiten mit dem C 64, die Sie aus Mangel an Erfahrung selber nicht beseitigen können. Schreiben Sie uns! Aber auch, wenn Sie die Probleme bereits gelöst haben und vielleicht jetzt darüber lachen. Schicken Sie die Lösung. Denken Sie daran, daß es anderen ebenso gehen könnte.

Simons Basic-Modul abschalten

Wer häufig, aber eben nicht immer mit Simons Basic programmiert, kann das Modul mit »SYS 64760« abschalten. Allerdings stehen dann statt 38911 nur 30719 Byte für Basic zur Verfügung. Man erspart sich aber das dauernde Rausziehen und Reinsticken des Moduls. Eingeschaltet wird Simons Basic wieder mit »SYS 64738«, durch einen Reset-Taster oder durch Aus- und Einschalten des C 64. (Jürgen Ehnes/tr)

Der bunte Cursor

Durch »POKE 53265,80« wird der Cursor rot, die Schrift bleibt allerdings hellblau. Farbige und reverse PRINT-Anweisungen werden zerstört. Läßt sich gut vor INPUT einsetzen. »Ge-SHIFT-ete« Zeichen erscheinen hellblau mit weißem Hintergrund. (Volker Hilt/tr)

Berechnung von PI

Im Mathematikunterricht habe ich die näherungsweise Berechnung der Kreiszahl PI kennengelernt. Das Näherungsverfahren beruht auf der Umfangsberechnung regulärer Tangenten- und Sehnenvielecke eines Kreises. Dieses Verfahren, das ich in einem Einzeller realisiert habe, entwickelte der englische Mathematiker Gregory im 17. Jahrhundert. Das Programm (Listing 1) berechnet eine Intervallschachtelung.

```

10 FOR I=0 TO 12
20 W=- (U=0)
30 O=2*SQR(3)*W+O
40 U=3*W+U
50 O=2*O*U/(O+U)
60 U=SQR(O*U)
70 PRINT 6*2^I,U;O
80 NEXT

```

Listing 1. Dieses Programm berechnet näherungsweise die Kreiszahl PI

Programmerklärung: Die FOR-NEXT-Schleife bricht die Berechnungen nach dem 13. Durchgang (FOR I=0 TO 12) ab, da PI vom Basic-Interpreter nicht mehr genauer berechnet werden kann. An der Variablen W erkennt der C 64, ob U=0 ist. Im ersten Durchlauf ist U=0, da nach einem »RUN« automatisch alle Variablen gelöscht werden. Das bedeutet, daß der Term

»U=0« wahr ist (-1). In den folgenden Durchläufen ist U ≠ 0 (ungleich Null). Das bedeutet, daß der Term »U ≠ 0« nicht wahr, beziehungsweise falsch ist (siehe Bild 1 und 2).

Indem ich vor den Term »U=0« ein Minuszeichen setze, erreiche ich folgende Belegung der Variablen W:

W=1: 1. Durchlauf (U=0)
W=0: x-ter Durchlauf (U ≠ 0)

Dadurch wird bewirkt, daß die Variablen nur einmal initialisiert werden. Zu dem Produkt aus W und der Initialisierungszahl addiere ich die Variable selbst, damit die Variablenwerte in den nachfolgenden Durchläufen nicht verlorengehen (siehe Bild 3). Dann folgt die eigentliche Berechnung:

$$O = \frac{2 \cdot O \cdot U}{O + U}; U = \sqrt{U \cdot O}$$

Bevor ein neuer Programmdurchlauf gestartet wird, werden N, U und O auf dem Bildschirm ausgegeben. Das n-Eck (Viel-eck) hat zuerst sechs Ecken, dann 12, 24, 48 und so weiter, wird also an einem Kreis angenähert. Das bedeutet, daß N nach jeder Berechnung mit 2 multipliziert wird (n := 2n). Durch die Formel $6 \cdot 2^{I-1}$ berechne ich N für den I+1-ten Durchgang (siehe Bild 4). (Gerhard Muth/tr)

```

READY.
? 1=5
0

READY.
? 7=7
-1

READY.
Q = (1=5) : ? Q
0

READY.
Q = (7=7) : ? Q
-1

READY.

```

```

U = W * 3 + U = U;
0, da W=0

O = W * 2 * SQR(3) + O = O;
0, da W=0

```

Bild 3. So werden die Variablen bei jedem Schleifendurchlauf neu definiert

| I | 6 * 2 ^{I-1} | N |
|-----|----------------------|------|
| 0 | 6 * 1 | = 6 |
| 1 | 6 * 2 | = 12 |
| 2 | 6 * 4 | = 24 |
| 3 | 6 * 8 | = 48 |
| ... | ... | ... |

Bild 1. Beispiel für »wahre« und »falsche« Aussagen. Das »?« ist die Abkürzung für »PRINT«

```

0 : falsch
-1 : wahr

```

Bild 4. Die Wertetabelle unseres kleinen Programms

I : Schleifenzähler
W : Kennvariable
O : obere Grenze
U : untere Grenze
N : Anzahl der Ecken des Vielecks

Bild 5. Die verwendeten Variablen und ihre Bedeutung

Fehlersuche für Einsteiger

Vor allem Anfänger haben Schwierigkeiten, die manchmal nur schwer verständlichen englischen Fehlermeldungen des C 64 richtig zu deuten. Aber gerade während der ersten Schritte in Basic ist es wichtig zu wissen, was man denn eigentlich falsch gemacht hat. Dieses Programm (Listing 2) übersetzt für Sie die Meldungen des Basic-Interpreters und bringt die fehlerhafte Programmzeile gleich auf den Bildschirm. Nach der Eingabe mit dem MSE speichern Sie das Listing erst einmal auf Diskette oder Kassette. Laden Sie es dann absolut, also mit LOAD "FEHLER",8,1 für Diskette beziehungsweise mit LOAD "FEHLER",1,1 für Datasette. Danach

| programm : fehler | c000 c000 c2d8 | c0f8 : 4c 45 4e 41 4d 45 46 45 45 | c1f0 : 54 52 49 4e 47 20 3c 3d 6a |
|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| c000 : a9 16 8d 00 03 a9 c0 Bd b3 | c100 : 48 4c 54 00 55 4e 45 52 05 | c1f8 : 3e 20 5a 41 48 4c 56 45 d0 | |
| c008 : 01 03 60 a9 8b 8d 00 03 03 | c108 : 4c 41 55 42 54 45 20 47 11 | c200 : 52 57 45 43 48 53 45 4c 84 | |
| c010 : a9 e3 8d 01 03 60 8a 10 ac | c110 : 45 52 41 45 54 4e 55 4d 1f | c208 : 54 00 53 54 52 49 4e 47 f3 | |
| c018 : 03 4c 74 a4 0a aa bd 99 13 | c118 : 4d 45 52 00 4e 45 58 54 b6 | c210 : 20 5a 55 20 4c 41 4e 47 4d | |
| c020 : c2 85 fb bd 9a c2 85 fb e9 | c120 : 20 4f 48 4e 45 20 46 4f d1 | c218 : 00 46 41 4c 53 43 48 45 10 | |
| c028 : 20 cc ff a9 00 85 13 20 9c | c128 : 52 00 54 45 58 54 46 45 04 | c220 : 20 44 41 54 45 4e 20 45 0f | |
| c030 : d7 aa a9 62 a0 c0 20 1e e0 | c130 : 48 4c 45 52 00 52 45 54 Ba | c228 : 49 4e 45 53 20 46 49 4c 46 | |
| c038 : ab a5 fa a4 fb 20 1e ab 99 | c138 : 55 52 4e 20 4f 48 4e 45 49 | c230 : 45 53 00 5a 55 20 4b 4f 8c | |
| c040 : a4 3a c8 f0 03 20 c2 bd 09 | c140 : 20 47 4f 53 55 42 00 46 36 | c238 : 4d 50 4c 49 5a 49 55 52 93 | |
| c048 : a9 0d 20 16 e7 ea ea a5 0f | c148 : 45 48 4c 45 4e 44 45 20 c9 | c240 : 54 45 52 20 41 55 53 44 64 | |
| c050 : 39 85 14 a5 3a 85 15 20 6a | c150 : 44 41 54 41 00 55 4e 45 e1 | c248 : 52 55 43 4b 00 4b 45 49 81 | |
| c058 : a7 a6 a9 80 20 90 ff 4c ec | c158 : 52 4c 41 55 42 54 45 52 4c | c250 : 4e 20 43 4f 4e 54 20 4d 0c | |
| c060 : 80 a4 11 12 12 12 2a 2a 68 | c160 : 20 42 45 52 45 49 43 48 79 | c258 : 4f 45 47 4c 49 43 48 00 75 | |
| c068 : 2a 20 20 46 45 48 4c 45 c6 | c168 : 00 45 52 47 45 42 4e 49 bb | c260 : 46 45 48 4c 45 4e 44 45 47 | |
| c070 : 52 20 3a 20 20 2a 2a 2a b5 | c170 : 53 20 5a 55 20 47 52 4f 39 | c268 : 20 44 45 46 20 41 4e 57 b8 | |
| c078 : 92 0d 9a 00 00 00 5a 55 4c | c178 : 53 53 00 53 50 45 49 43 ba | c270 : 45 49 53 55 4e 47 00 56 a5 | |
| c080 : 20 56 49 45 4c 45 20 46 c2 | c180 : 48 45 52 20 56 4f 4c 4c ad | c278 : 45 52 49 46 59 20 50 52 7e | |
| c088 : 49 4c 45 53 00 46 49 4c a3 | c188 : 00 45 58 49 53 54 49 45 f2 | c280 : 47 2e 00 4c 4f 41 44 20 b8 | |
| c090 : 45 20 4f 46 46 45 4e 00 4a | c190 : 52 54 20 4e 49 43 48 54 57 | c288 : 50 52 2e 00 21 55 4e 54 2c | |
| c098 : 46 49 4c 45 20 4e 49 43 5f | c198 : 00 4e 49 43 48 54 44 49 45 | c290 : 45 52 42 52 45 43 48 55 13 | |
| c0a0 : 48 54 20 4f 46 46 45 4e 4d | c1a0 : 4d 45 4e 53 49 4f 4e 49 69 | c298 : 4e 47 00 7e c0 98 06 | |
| c0a8 : 00 46 49 4c 45 20 4e 49 c8 | c1a8 : 45 52 54 00 5a 57 45 49 33 | c2a0 : c0 a9 c0 bd c0 d4 c0 e5 9e | |
| c0b0 : 43 48 54 20 47 45 46 55 93 | c1b0 : 4d 41 4c 20 44 49 4d 45 03 | c2a8 : c0 f6 c0 04 c1 c1 2a ec | |
| c0b8 : 4e 44 45 4e 00 47 45 52 37 | c1b8 : 4e 53 49 4f 4e 49 45 52 d5 | c2b0 : c1 35 c1 47 c1 55 c1 69 06 | |
| c0c0 : 45 41 54 20 4e 49 43 48 8c | c1c0 : 54 00 44 55 52 43 48 20 71 | c2b8 : c1 7b c1 89 c1 99 c1 4c 61 | |
| c0c8 : 54 20 56 4f 52 48 41 4e b5 | c1c8 : 4e 55 4c 4c 20 47 45 54 57 | c2c0 : c1 c2 c1 d5 c1 ef c1 0a c4 | |
| c0d0 : 44 45 4e 00 4b 45 49 4e eb | c1d0 : 45 49 4c 54 00 44 49 52 43 | c2c8 : c2 19 c2 33 c2 4d c2 60 90 | |
| c0d8 : 20 45 49 4e 47 41 42 45 c9 | c1d8 : 45 4b 54 4d 4f 44 55 53 95 | c2d0 : c2 77 c2 83 c2 8c 2c 00 | |
| c0e0 : 46 49 4c 45 00 4b 45 49 89 | c1e0 : 20 4e 49 43 48 54 20 45 14 | da | |
| c0e8 : 4e 20 41 55 53 47 41 42 3a | c1e8 : 52 4c 41 55 42 54 00 53 c9 | | |
| c0f0 : 45 46 49 4c 45 00 46 49 34 | | | |

Listing 2. »Deutsche Fehler«

müssen Sie NEW eingeben (RETURN-Taste nicht vergessen), damit Sie später keinen »OUT OF MEMORY ERROR« (auf Deutsch: Speicher voll) erhalten. Starten Sie dann das Programm mit SYS 49152. Wenn Sie die Erweiterung nicht mehr brauchen, geben Sie einfach SYS 49163 ein.

(Florian Gudermann/tr)

Geniale Bildschirm-Hardcopy

Diese extrem kurze Programmroutine (Listing 3), die sich natürlich auch in eigene Programme einbauen lässt, bringt den Bildschirminhalt zu Papier. Angenommen, Sie möchten einen kleinen Text, oder einen Brief ausdrucken, so gibt es zwei Möglichkeiten:

1. Sie laden ein Textverarbeitungsprogramm (falls Sie eines besitzen) und quälen sich durch zig Menüs, bis Sie auf dem Drucker endlich das gewünschte Resultat haben, oder
2. Sie tippen dieses kurze Programm ein, schreiben Ihren Text einfach auf den Bildschirm und starten das Programm mit »RUN«.

Das Prinzip besteht darin, den Bildschirm als Eingabegerät zu definieren, und dann die einzelnen Zeichen direkt an den Drucker zu senden. Dabei gehen Sie so vor:

— Löschen Sie den Bildschirm mit der Taste SHIFT-CLR und schreiben Sie Ihren Text (maximal 23 Zeilen). Beachten Sie, daß Sie niemals »RETURN« drücken, da sonst der C 64 Ihre Eingaben als Befehle interpretieren würde, was eine Fehlermeldung zur Folge hätte.

— Wenn Sie fertig sind, schreiben Sie in die vorletzte Zeile ein »RUN« und drücken die »RETURN«-Taste (hoffentlich haben Sie vorher Ihren Drucker eingeschaltet). Das »RUN« wird jetzt automatisch gelöscht und der Text zu Papier gebracht.

Erklärung des Programms:

| Zeile | Bedeutung |
|-------|--|
| 10 | definiert den Bildschirm (Sekundäradresse 3) als Eingabegerät mit der Kanalnummer 1 |
| 20 | definiert den Drucker (Sekundäradresse 4) als Ausgabegerät mit der Kanalnummer 4 |
| 30 | das störende »RUN« auf dem Bildschirm wird gelöscht, damit es nicht mit aufs Papier kommt, und der Cursor wird in die obere linke Bildschirmposition gesetzt (ab hier werden die Zeichen eingelesen) |
| 40 | erste Schleife; wird 25mal durchlaufen (der Bildschirm hat 25 Zeilen) |
| 50 | zweite Schleife; wird 40mal durchlaufen (40 Zeichen pro Zeile) |
| 60 | holt ein Zeichen aus Kanal 1, der in der Zeile 10 definiert wurde, und legt dieses in der Variablen A\$ ab |
| 70 | für jede Bildschirmzeile werden alle Zeichen erst in der Variablen B\$ gesammelt |
| 80 | zweite Schleife weiterführen |
| 90 | jetzt wurde eine ganze Bildschirmzeile eingelesen, der Text in dieser Zeile kann also ausgedruckt werden (Kanal 4 »zeigt ja wegen Zeile 20 zum Drucker) |
| 100 | die Sammelvariable wird wieder gelöscht, |
| 110 | die nächste Bildschirmzeile eingelesen |
| 120 | und |
| 130 | die Kanäle aus den Zeilen 10 und 20 werden wieder geschlossen. |

Das war's auch schon!

(Peter Tiesat/tr)

»RUN« mit zwei Tastendrücken

Preisfrage: Wie erreiche ich einen Programmstart auf dem C 64 ohne RUN (Return) eintippen zu müssen?

Antwort: Bitte drücken Sie

A und dann SHIFT zusammen mit RUN/STOP

Auf dem Bildschirm erscheint:

ALOAD

?SYNTAX ERROR

READY.

RUN

(Ein eventuell vorhandenes Basic-Programm wird gestartet.)

Erklärung: Das Betriebssystem hält eine Routine bereit, die die Zeichenfolge

LOAD(cr) RUN (cr)

in den Tastaturpuffer einliest ((cr) bedeutet »carriage return« und heißt »Return-Taste ausführen«).

```

10 OPEN 1,3
20 OPEN 4,4
30 PRINT"UP,3SPACE,HOME";
40 FOR I=1 TO 25
50 FOR J=1 TO 40
60 GET#1,A$
70 B$=B$+A$
80 NEXT J
90 PRINT#4,B$
100 B$=""
110 NEXT I
120 CLOSE 4
130 CLOSE 1
@ 64'er
    
```

Diese Routine wird durch Drücken von SHIFT und RUN/STOP aktiviert. (Sie ist besonders für Datasettenbenutzer als »Autostart« interessant.) Um nur das »RUN(cr)« zu aktivieren, muß man das »LOAD(cr)« gewissermaßen außer Betrieb setzen. Dieses geschieht dadurch, daß man durch die vorherige Eingabe eines Buchstabens (zum Beispiel »A«) das Wort »ALOAD« erzeugt. Da das (cr) danach aus dem Tastaturpuffer geholt wird, versucht der Basic-Interpreter diesen »ALOAD«-Befehl auszuführen. Dieser ist aber nicht im Befehlssatz vorhanden — ein »Syntax Error« wird ausgegeben. Danach geht das Betriebssystem wieder dazu über, den Tastaturpuffer abzufragen. Dort steht immer noch »RUN(cr)«. Dieses wird nun in den Bildschirmspeicher geschrieben und ausgeführt — ein Basic-Programm wird gestartet.

(Bernd Kleine-Voßbeck/tr)

Der Piepser

Dieses Maschinenprogramm für den C 64 (Listing 4) bewirkt, daß immer dann, wenn eine Taste gedrückt wird, ein Ton erklingt (außer bei der »RESTORE«-Taste). Nützlich zum Beispiel bei schnellen Tipparbeiten auf ausgeleierter Tastatur als Bestätigung des »korrekten« Tastendrucks. Das Programm kopiert den Basic-Interpreter und das Kernel ins darunterliegende RAM und schaltet auf RAM um (auch mit POKE 1,53 möglich). Dann wird in die Tastaturabfrage des Betriebssystems ein JMP-Befehl zu der Routine ab \$C000 (49152) eingefügt. Diese läßt einen Ton erklingen (sie wird nur bei gedrückter Taste angesprungen) und springt wieder in die Betriebssystem-Routine zurück. Das Programm steht ab \$C000=49152, wird aber mit SYS 49184 gestartet. Ist das Programm einmal aktiviert, so kann der Bereich ab 49184 genutzt werden. Experimente mit Frequenz und ADSR-Hüllkurve sind natürlich möglich.

Start: SYS 49194

Nach Drücken von RUN/STOP-RESTORE: POKE 1,53

Abschalten: POKE 1,55

(Cadrio/tr)

Zahlen rechtsbündig

Wer hat sich noch nicht geärgert, daß die Zahlenausgabe im normalen Commodore-Basic immer linksbündig erfolgt? (Zum besseren Verständnis gebe man einmal »PRINT 3:PRINT 12« ein!) Wer keine komplizierte Print-Using-Routine benutzen will, kann sich in vielen Fällen mit dem hier vorgestellten Trick behelfen:

Die Kennziffer des dekadischen Logarithmus gibt die Anzahl der Vorkommastellen -1 an. In Verbindung mit der »PRINT TAB«-Funktion bietet sich die Möglichkeit, Zahlen formatiert auszugeben:

PRINT TAB (A-(INT(LOG(X)/LOG(10)))); INT (X*10^{1B})/10^{1B}
wobei

A die Anzahl der Vorkommastellen,
B die Anzahl der Nachkommastellen,
X die auszugebende Zahl bedeutet.

Natürlich ist zu beachten, daß für X-Werte kleiner gleich Null keine reellen Lösungen existieren. Auch bei Zahleneingaben die kleiner als 0,001 sind, wird die vorgestellte Print-Using-Hilfe nicht funktionieren.

(Heinrich Stender/tr)

Listschatz geknackt

Der Listschatz aus der 64'er, Ausgabe 11/85, läßt sich knacken, indem man nach dem Laden einen »POKE 2067,73< eingibt.

(Sascha Lengen/tr)

| | | | |
|------|-----|-----------|-----------------------------|
| C000 | LDA | #\$10 | ; Wellenform, Gate-Bit off |
| | STA | \$D404 | |
| | LDA | #\$80 | ; Attack, Decay |
| | STA | \$D405 | |
| | LDA | #\$00 | ; Sustain, Release |
| | STA | \$D406 | |
| | LDA | #\$67 | ; Frequenz Low-Byte |
| | STA | \$D400 | |
| | LDA | #\$11 | ; Frequenz High-Byte |
| | STA | \$D401 | |
| | LDA | #\$11 | ; Gate-Bit on |
| | STA | \$D404 | |
| | LDA | #\$FF | ; Lautstärke maximal |
| | STA | \$D418 | |
| | LDA | #\$81 | ; 2 Befehle aus Tasten- |
| | STA | \$F5 | abfrage |
| | JMP | \$EA9F | ; zurück ins Betriebssystem |
| C02A | LDY | #\$00 | ; Kopieren des Basic-ROM |
| | LDA | #\$20 | |
| | STA | \$03E8 | |
| | LDA | #\$00 | |
| | STA | \$BB | |
| | LDA | #\$AO | |
| | STA | \$BC | |
| C039 | LDA | (\$BB), Y | |
| | STA | (\$BB), Y | |
| | INC | \$BB | |
| | BEQ | M01 | |
| | JMP | \$C039 | |
| M01 | INC | \$BC | |
| | DEC | \$03E8 | |
| | BEQ | M02 | |
| | JMP | \$C039 | |
| M02 | LDA | #\$20 | ; Kopieren des KERNEL |
| | STA | \$03E8 | |
| | LDA | #\$00 | |
| | STA | \$BB | |
| | LDA | #\$EO | |
| | STA | \$BC | |
| C05B | LDA | (\$BB), Y | |
| | STA | (\$BB), Y | |
| | INC | \$BB | |
| | BEQ | M03 | |
| | JMP | C05B | |
| M03 | INC | \$BC | |
| | DEC | \$03E8 | |
| | BEQ | M04 | |
| | JMP | C05B | |
| M04 | LDA | #\$35 | ; Umschalten auf |
| | STA | \$01 | ; RAM |
| | LDA | #\$4C | ; Code für JMP |
| | STA | \$EA9B | |
| | LDA | #\$00 | ; Low-Byte von C000 |
| | STA | \$EA9C | |
| | LDA | #\$C0 | ; High-Byte von C000 |
| | STA | \$EA9D | |
| | RTS | | |

programm : tastenton c000 c084

```
c000 : a9 10 8d 04 d4 a9 70 8d 0d
c008 : 05 d4 a9 00 8d 06 d4 a9 91
c010 : cf 8d 00 d4 a9 22 8d 01 24
c018 : d4 a9 11 8d 04 d4 a9 ff 44
c020 : 8d 18 d4 a9 81 85 f5 4c d8
c028 : 9f ea a0 00 a9 20 8d e8 08
c030 : 03 a9 00 85 bb a9 a0 85 4f
c038 : bc b1 bb 91 bb e6 bb f0 b2
c040 : 03 4c 39 c0 e6 bc ce e8 31
c048 : 03 f0 03 4c 39 c0 a9 20 8e
c050 : 8d e8 03 a9 00 85 bb a9 b6
c058 : e0 85 bc b1 bb 91 bb e6 65
c060 : bb f0 03 4c 5b c0 e6 bc af
c068 : ce e8 03 f0 03 4c 5b c0 0b
c070 : a9 35 85 01 a9 4c 8d 9b a0
c078 : ea a9 00 8d 9c ea a9 c0 32
c080 : 8d 9d ea 60 4c 45 20 46 9e
```

Listing 4:
»Der Piepser«:
akustische Rück-
meldung eines
Tastendrucks.
Oben der
Source-Code,
darunter das
MSE-Listing.

Sauberer Schriftbild auf dem MPS 801

Ein weit verbreiteter und derzeit sehr preisgünstiger Matrixdrucker für den C 64 beziehungsweise VC 20 ist der MPS 801 beziehungsweise GP 100 VC, dessen Matrix mit der des älteren VC 1515 für den VC 20 identisch ist. Diese Drucker haben ein sehr gut lesbares Schriftbild bei ausschließlicher Verwendung von Großbuchstaben (bei Listings meist ausreichend). Bei Verwendung von Groß- und Kleinschreibung fällt aber sehr störend das Fehlen von Unterlängen auf, wodurch die Zeichen p,q,j und so weiter hochgerutscht erscheinen. Und beim Schreiben deutscher Texte ist zwar durch eigene Zeichendefinition ein deutscher Zeichensatz mit Umlauten möglich und in vielen Textsystemen auch realisiert. Dabei fehlen dann jedoch die Oberlängen bei den großen Umlauten Ä, Ö, Ü. Bedingt ist dies durch die Zeichenmatrix, die zwar in der Breite von 6 Punkten sehr gut lesbar ist, in der Höhe von 7 Punkten aber für Unter- oder Oberlängen (Ä, Ö, Ü) nicht ausreicht. Die folgende Maschinensprache-Routine (Listing 5) versetzt die genannten Drucker dennoch in die Lage, mit Hilfe der Grafikcodes echte Ober- und Unterlängen zu drucken, außerdem auch die deutschen Sonderzeichen. Des Weiteren werden statt inverser Zeichen unterstrichene gedruckt, wodurch in Basic-Listing (CMD-Modus) die Steuerzeichen besser lesbar werden. Ein besonderer »Leckerbissen« ist das Anführungszeichen, das abwechselnd unten und oben erscheint. Die Druckroutine funktioniert bei Normalschrift (CHR\$(14)) und Breitschrift (CHR\$(15)), wobei auch das Mischen von Schmal- und Breitschrift innerhalb einer Zeile unverändert möglich ist.

Das Programm nimmt Rücksicht auf den geringen Zeichenspuffer des Matrix-Druckers und ist so ausgelegt, daß der Drucker möglichst schnell arbeitet. Daher wird kein völlig eigener Zeichensatz verwendet, sondern nur der vorhandene entsprechend modifiziert.

Das Programm ist für den C 64 konzipiert und kann im Bereich \$D000, also über dem I/O-Port arbeiten, so daß es weder Basic-Speicherplatz belegt noch den gleichzeitigen Betrieb von eigenen Betriebssystem-Erweiterungen, Toolkits oder 80-Zeichen-Karte beeinträchtigt! In diesem Falle steht jedoch der Kassettenpuffer für die Datasette nicht mehr zu Verfügung, da dort ein Ein- und Aussprungprogramm nach \$D000 steht. Aufgrund des universellen Basic-Laders ist das Programm allerdings auch in anderen Speicherbereichen zu betreiben und somit auch im VC 20 ohne Änderung (mindestens 3 KByte-RAM-Erweiterung erforderlich). Bei allen Speicherbereichen außer \$D000 kann auch weiterhin ein Kassettenrecorder betrieben werden.

Selbstverständlich kann die Drucker-Routine auch in Textverarbeitungsprogramme eingebaut werden, wozu es allerdings Maschinensprache-Kenntnisse bedarf.

Das Programm ist in Basic-DATAs geschrieben, um es universell und kompatibel zu halten. So kann jeder, der es einmal abgetippt hat, es an VC 20- und C 64-Besitzer ohne Anpassungsprobleme weitergeben.

Zur Bedienung des Programms: Nach dem Start mit »RUN« bietet es fünf Möglichkeiten für den Speicherbereich oder den verwendeten Computer an. Je nach gewähltem Bereich setzt es aufgrund der DATA-Zeilen die Maschinensprache an die richtige Stelle. Die DATA-Prüfsummen werden vor Ansprung der Maschinensprache geprüft. Nach einer Wartezeit von 30 Sekunden wird dann noch angeboten, die Lage der deutschen Umlaute auf der Tastatur zu ändern, im Original ist ä=CHR\$(60), Ä=CHR\$(62), ö=CHR\$(91), Ö=CHR\$(93), ü=CHR\$(64), Ü=CHR\$(42), ß=CHR\$(92). Um dies zu ändern, müssen neue ASCII-Codes eingegeben werden. Achtung: Es ist nicht zulässig, daß für verschiedene Umlaute der gleiche Code eingegeben wird (zum Beispiel alles Null). Die

Routine hängt sich sonst auf. Werden keine deutschen Sonderzeichen gewünscht, so können für diese die Codes über 200 eingegeben werden, die in der Praxis kaum vorkommen. Wenn das Programm durchgelaufen ist, werden POKEs angeboten, mit denen die Drucker-Routine ein- und ausgeschaltet werden kann. Achtung: Beide POKEs müssen jeweils wie angezeigt in einer Befehlszeile mit Doppelpunkt getrennt eingegeben werden, keinesfalls einzeln! Nach Ausschalten muß mit »PRINT #4,CHR\$(15);« (oder CHR\$(14)) der Drucker zunächst wieder in den Normalzustand gebracht werden (steht sonst auf CHR\$(8) = Grafik). Das Basic-Programm endet, indem es den Befehl »NEW« und die Abfrage »PRINT FRE(0)« vorschlägt, um zu dokumentieren, wieviel freier Speicherraum noch verfügbar ist. Auch nach »NEW« kann mit den genannten POKEs die Unterroutine beliebig an- und ausgeschaltet werden. Achtung: RUN/STOP-RESTORE schaltet die Druckeroutine ab.

Noch zwei Hinweise: Wird mit »OPEN4,4« ein Druckerfile erstmals eröffnet, so empfiehlt es sich, mit »PRINT #4« erst einmal eine Leerzeile zu drucken. Beim CMD-Modus entfällt dies. Am Ende eines Druckvorgangs ist die letzte Zeile stets ohne Unterlängen gedruckt, da der Computer noch auf die Oberlängen der nächsten Zeile wartet. Kommt keine nächste Zeile mehr, so sollte mit »PRINT #4« eine Leerzeile gedruckt werden. Nach Beendigung eines »CMD« muß daher manchmal nicht nur ein- sondern zweimal »PRINT #4« eingegeben werden. (Barnim König/tr)

```

100 PRINT" {DOWN}DRUCKROUTINE GP100 VC20/C
  BM64." <226>
101 PRINT" {DOWN}BARNIM KOENIG 8/1985" <228>
102 PRINT" {DOWN}BEREICHSWAHL STARTADRESSE
  ." <178>
103 PRINT" {DOWN}1: CBM 64 $D000 (OHNE CASS
  ETTE)" <122>
104 PRINT" 2: CBM 64 $C000 (MIT CASSETTE)." <235>
105 PRINT" {DOWN}3: VC 20 3K, WENN VORHANDE
  N ($0400)" <225>
106 PRINT" 4: VC 20 ROM-BEREICH $A000" <068>
107 PRINT" {DOWN}5: 20 ODER 64 AM BASIC-END
  E.{DOWN}" <235>
108 INPUT SA:IF SA<1 OR SA>5 THEN 102 <176>
109 PRINT" {DOWN}BITTE 30 SEC. WARTEN." <128>
110 :
111 POKE 828,0:IF SA=1 THEN POKE 828,1:POK
  E 829,PEEK(55):POKE 830,PEEK(56):GOTO
  113 <026>
112 IF SA<5 THEN 115 <073>
113 SA=PEEK(56)-7:IF PEEK(55)>0 THEN SA=SA
  -1 <190>
114 POKE 56,SA:POKE 55,0:CLR:SA=PEEK(56):G
  OTO 118 <216>
115 IF SA=2 THEN SA=192 <005>
116 IF SA=3 THEN SA=4 <133>
117 IF SA=4 THEN SA=160 <024>
118 SR=SA:IF PEEK(828)THEN SR=208 <107>
119 SD=SA*256:P=0 <205>
120 READ X:P=P+X:IF X=-1 THEN 127 <094>
121 IF X<300 THEN 126 <176>
122 IF X<600 THEN X=X-492+SR:GOTO 126 <232>
123 IF PEEK(828)=0 THEN X=PEEK(X):GOTO 126 <222>
124 IF X=806 THEN X=89 <058>
125 IF X=807 THEN X=3 <000>
126 POKE SD,X:SD=SD+1:GOTO 120 <231>
127 IF P<>187301 THEN PRINT" {DOWN}DATA-FE
  HLER BLOCK 1 {DOWN}":END <129>
128 SD=SA*256+1024:FOR X=SD TO SD+426:POKE
  X,0:NEXT X <054>
129 SD=SA*256+1535:P=0 <056>
130 READ X:P=P+X:IF X=-1 THEN 133 <076>
131 POKE SD,X:SD=SD+1:GOTO 130 <240>
132 IF P<>26405 THEN PRINT" {DOWN}DATA-FEH
  LER BLOCK 2 {DOWN}":END <121>
133 :
134 INPUT" {DOWN}ZEICHEN AENDERN";IN$:IF L
  EFT$(IN$,1)<>"J"THEN 144 <232>
135 PRINT CHR$(14)" {DOWN}ASCII-CODES FUER

```

Listing 5. Sauberer Schriftbild auf dem MPS 801

```

SONDERZEICHEN EINGEBEN.":SD=SA*256      <188>
136 INPUT"AE";A:POKE SD+1655,A          <150>
137 INPUT"BE";A:POKE SD+447,A:POKE SD+761,
    A:POKE SD+1616,A                   <247>
138 INPUT"DE";A:POKE SD+1656,A          <122>
139 INPUT"OE";A:POKE SD+443,A:POKE SD+767,
    A                                     <134>
140 INPUT"UE";A:POKE SD+1657,A          <223>
141 INPUT"YE";A:POKE SD+449,A:POKE SD+773,
    A                                     <225>
142 INPUT"SS";A:POKE SD+1539,A:POKE SD+165
    2,A                                 <020>
143 :                                    <119>
144 IF PEEK(B28) THEN 149               <117>
145 PRINT"(2DOWN)AN: POKE806, 112 :POKE807
    ,":SA+3                            <186>
146 PRINT"(DOWN)AUS:POKE806,":PEEK(806);":
    POKE807,":PEEK(807);"."
147 POKE 806,112:POKE 807,SA+3:GOTO 158 <205>
148 :                                    <124>
149 SD=B32:P=0                         <064>
150 READ X:P=P+X:IF X=-1 THEN 152       <096>
151 POKE SD,X:SD=SD+1:GOTO 150         <012>
152 IF P<>10030 THEN PRINT"(2DOWN)DATA-FEH
    LER BLOCK 3(2DOWN)":END           <057>
153 SYS 887:POKE 55,PEEK(829):POKE 56,PEEK
    (830):CLR                          <217>
154 PRINT"(2DOWN)AN : POKE806, 64(2SPACE):
    POKE807, 3"                        <149>
155 PRINT"(DOWN)AUS: POKE806,":PEEK(806);"
    :POKE807,":PEEK(807)              <151>
156 POKE 861,PEEK(806):POKE 862,PEEK(807) <076>
157 POKE 806,64:POKE 807,3             <255>
158 PRINT"(2DOWN)DRUCKROUTINE EINGESCHALTE
    T."
159 PRINT"(2DOWN)NEW":PRINT"(2DOWN)?FRE(0)
    &UP)":END                           <151>
160 :                                    <136>
161 DATA 142,161,496,140,162,496,141,160,4
    96,201,13,208,10,72,169,0,141,166 <169>
162 DATA 496,104,76,210,492,201,10,240,242
    ,72,173,165,496,208,36,104,201,14 <196>
163 DATA 208,5,141,164,496,24,96,201,15,24
    0,247,201,18,208,5,141,166,496,24 <203>
164 DATA 96,201,146,208,10,72,169,0,141,16
    6,496,104,24,96,104,24,201,32,144 <045>
165 DATA 9,24,201,160,176,32,201,128,144,2
    8,72,173,165,496,208,3,104,24,96,169 <070>
166 DATA 18,141,166,496,104,9,64,32,139,49
    2,169,0,141,166,496,76,199,492,201 <056>
167 DATA 34,208,23,173,165,496,240,10,169,
    0,141,165,496,169,34,76,139,492,169 <154>
168 DATA 162,234,234,234,141,165,496,72,17
    3,164,496,201,14,208,17,72,173,163 <080>
169 DATA 496,41,1,240,5,169,32,32,165,492,
    104,32,165,492,104,24,174,163,496 <144>
170 DATA 224,80,144,12,72,32,210,492,104,2
    34,172,162,496,174,163,496,157,0 <255>
171 DATA 496,173,166,496,240,3,157,80,496,
    238,163,496,24,173,160,496,174,161 <073>
172 DATA 496,172,162,496,96,169,0,141,167,
    496,141,168,496,141,170,496,169,0 <065>
173 DATA 162,13,157,175,496,202,208,250,17
    4,167,496,189,0,497,201,14,240,81 <081>
174 DATA 162,9,221,254,497,240,6,202,208,2
    48,76,21,493,138,10,10,10,141,8,493 <069>
175 DATA 162,6,189,72,498,157,176,496,202,
    208,247,169,1,141,168,496,174,167 <115>
176 DATA 496,189,1,497,162,9,221,254,497,2
    40,6,202,208,248,76,118,493,138,10 <043>
177 DATA 10,10,141,50,493,162,6,189,24,498
    ,157,182,496,202,208,247,169,1,141 <232>
178 DATA 168,496,76,118,493,189,1,497,162,
    9,221,254,497,240,6,202,208,248,76 <043>
179 DATA 118,493,138,10,10,10,141,92,493,1
    62,6,189,72,498,72,138,10,170,104 <151>
180 DATA 157,175,496,157,176,496,72,138,74
    ,170,104,202,208,234,169,1,141,168 <210>
181 DATA 496,174,167,496,189,80,497,240,18
    ,162,6,169,2,29,176,496,157,176,496 <233>
182 DATA 202,208,245,169,1,141,168,496,174
    ,167,496,189,81,497,240,18,162,6,169 <221>
183 DATA 2,29,182,496,157,182,496,202,208,
    245,169,1,141,168,496,174,167,496 <010>
184 DATA 189,0,496,201,14,240,113,32,186,4
    93,76,197,493,201,93,208,2,169,62 <226>

```

```

185 DATA 201,42,240,250,96,162,3,221,79,49
    8,240,6,202,208,248,76,239,493,138 <045>
186 DATA 10,10,10,24,105,80,141,223,493,16
    2,6,189,88,498,29,176,496,157,176 <161>
187 DATA 496,202,208,244,169,1,141,168,496
    ,174,167,496,189,1,496,32,186,493 <200>
188 DATA 162,3,221,79,498,240,6,202,208,24
    8,76,103,494,138,10,10,24,105,80 <203>
189 DATA 141,18,494,162,6,189,104,498,29,1
    82,496,157,182,496,202,208,244,169 <154>
190 DATA 1,141,168,496,76,103,494,189,1,49
    6,32,186,493,162,3,221,79,498,240 <023>
191 DATA 6,202,208,248,76,103,494,138,10,1
    0,10,24,105,80,141,69,494,162,6,189 <006>
192 DATA 104,498,72,138,10,170,104,72,29,1
    75,496,157,175,496,104,29,176,496 <164>
193 DATA 157,176,496,72,138,74,170,104,202
    ,208,226,169,1,141,168,496,173,168 <039>
194 DATA 496,208,27,238,170,496,238,167,49
    6,238,167,496,173,167,496,201,80,176 <251>
195 DATA 3,76,221,492,169,8,32,806,807,76,
    183,494,173,170,496,240,15,169,14 <081>
196 DATA 32,806,807,169,32,32,806,807,206,
    170,496,208,248,169,8,32,806,807,162 <038>
197 DATA 1,189,176,496,9,128,32,806,807,23
    2,224,13,208,243,162,0,142,168,496 <118>
198 DATA 76,111,494,169,0,141,167,496,169,
    13,32,806,807,173,169,496,32,806,807 <213>
199 DATA 169,17,32,806,807,174,167,496,189
    ,0,496,201,14,240,2,169,15,205,169 <044>
200 DATA 496,240,6,141,169,496,32,806,807,
    201,14,208,4,238,167,496,232,189,0 <232>
201 DATA 496,162,14,221,111,498,240,74,202
    ,208,248,201,62,208,2,169,193,201 <005>
202 DATA 93,208,2,169,207,201,42,208,2,169
    ,213,32,806,807,238,167,496,24,173 <108>
203 DATA 167,496,205,163,496,144,179,169,8
    ,32,806,807,169,0,141,163,496,141 <082>
204 DATA 167,496,162,160,189,255,495,157,2
    55,496,169,0,157,255,495,202,208,242 <136>
205 DATA 24,169,13,32,806,807,76,200,492,1
    38,10,10,10,24,105,120,141,83,495 <029>
206 DATA 169,8,32,806,807,162,1,138,72,189
    ,232,498,32,806,807,174,169,496,224 <231>
207 DATA 14,208,3,32,806,807,104,170,232,2
    24,8,208,231,173,169,496,76,10,495 <254>
208 DATA 0,72,165,154,201,4,240,4,104,76,8
    06,807,104,76,0,492,-1 <197>
209 :                                    <185>
210 DATA 71,74,80,81,92,44,59,162,89,0,128
    ,132,132,132,131,128,0,0,128,130,132 <131>
211 DATA 131,128,128,0,0,135,128,128,128,1
    28,128,0,0,128,128,128,128,128,128 <173>
212 DATA 0,0,135,128,128,128,128,128,128,0,0,1
    28,130,129,128,128,128,0,0,128,130 <134>
213 DATA 129,128,128,128,0,0,128,128,131,1
    28,131,128,0,0,128,132,131,128 <019>
214 DATA 128,0,62,39,34,0,0,0,0,0,128,16
    0,128,160,128,128,0,0,128,128,192 <227>
215 DATA 160,128,128,0,0,128,224,128,224,1
    28,128,0,71,74,80,81,92,44,59,60,91 <229>
216 DATA 64,39,162,34,89,0,0,0,184,196,196
    ,168,252,128,0,0,128,128,253,128 <233>
217 DATA 128,0,0,252,168,196,196,184,128,0
    ,0,184,196,196,168,252,128,0,0,254 <111>
218 DATA 201,201,182,128,128,0,0,128,128,1
    92,128,128,128,0,0,128,128,208,128 <032>
219 DATA 128,128,0,0,160,213,212,189,192,1
    28,0,0,184,197,196,197,184,128,0,0 <163>
220 DATA 189,192,192,189,192,128,0,0,128,1
    29,128,128,128,128,0,0,128,128,192 <141>
221 DATA 128,192,128,0,0,128,129,128,129,1
    28,128,0,0,188,192,160,252,128,128 <253>
222 DATA 0,66,65,82,78,73,77,32,32,75,79,6
    9,78,73,71,56,53,-1 <174>
223 :                                    <199>
224 DATA 72,165,154,201,4,240,8,104,108,93
    ,3,234,234,234,234,104,32,98,3,32 <221>
225 DATA 0,208,76,111,3,32,111,3,32,202,24
    1,76,98,3,72,165,1,141,113,3,41,252 <109>
226 DATA 120,133,1,104,96,72,169,55,133,1,
    104,88,96,32,98,3,165,56,234,141,132 <162>
227 DATA 3,162,0,189,0,128,157,0,208,232,2
    08,247,173,135,3,201,214,240,9,238,132 <240>
228 DATA 3,238,135,3,76,130,3,76,111,3,-1 <164>

```

© 64'er

Listing 5. (Schluß)