

Synthetische Steuerzeichen

Mit den »synthetischen Steuerzeichen« stehen Ihnen Möglichkeiten offen, die Sie bisher nicht für möglich gehalten haben: beispielsweise ein neuartiger Listschutz, Grafik und Breitschrift-Schmalschrift-Wechsel im Listing.

Nach dem Einschalten des VC20 oder des C64 und der »READY«-Meldung des Computers kann man sich nach Herzenslust auf dem Bildschirm austoben – zunächst ohne Programm. Sie können vorab Texte gestalten, Grafiken entwerfen, dabei die Farben wechseln, hier Zeichen ergänzen und da Stellen abändern. Komfortabel wird dieses Austesten der Möglichkeiten auf der »elektronischen Tafel« jedoch erst durch die Cursor-Steuertasten CLR/HOME, INST/DEL, RVS ON/OFF sowie durch die Farbwahltasten. Man erreicht damit ein hohes Maß an Flexibilität im Ansteuern jeder beliebigen Bildschirmstelle.

Schreibt man nun unversehens ein Anführungszeichen (engl.: quote) auf den Schirm, so verhält sich der Computer plötzlich anders: die Cursor-Tasten wirken nicht mehr – man kommt mit ihnen nicht mehr aus der Zeile heraus. Auch die Farbumschaltung mißlingt, das Bildschirmlöschen mit CLR/HOME versagt, lediglich DEL funktioniert noch. Und dies alles in voller Absicht! Warum? Statt der Ausführung der Steuerbefehle zieht es der Computer vor, sich die Anweisungen in Form reverser Steuerzeichen zu »merken«. Wozu? Nun, um Steuerbefehle dieser Art programmierbar zu machen. (Man stelle sich nur einmal vor, der Cursor sei nicht per Software steuerbar.) Nach dem ersten Anführungszeichen befindet sich der Computer offenbar in einem anderen Verarbeitungsmodus, genannt Quote-Modus, in dem er (mit Ausnahme von DEL) alle Steuerbefehle als Reverszeichen »speichert« und damit zur späteren Ausführung bereitstellt. Der Quote-Modus wird verlassen, sobald das zweite Anführungszeichen eingetippt oder die RETURN-Taste betätigt wird. Da die Steuerzeichen – eingeschachtelt in Gänsefüßchen – vom Computer wie Texte behandelt werden, können sie in Programmen auch als solche verarbeitet werden. Dazu stehen alle Stringoperationen zur Verfügung, die auch bei »normalen« Texten verwendet werden. Wirkung zeigen Steuerzeichen jedoch erst dann, wenn man sie mittels PRINT aktiviert. Fazit: Ein Basic-Listing kann normalerweise keine reversen Zeichen enthalten – es sei denn, es handelt sich um Steuerzeichen in Strings.

Steuerbefehle ohne »Gänsefüßchen«

Wir haben uns als Commodore-Anwender sicherlich schon längst an das reverse Q für CURSOR UP oder an das Herzchen für CLR/HOME gewöhnt. Doch es geht auch anders. Das muß es auch, wenn der Drucker zum Beispiel keine Steuerzeichen ausgeben kann. In diesem Fall bedient man sich der CHR\$(X)-Funktion, die durch die Befehlsfolge

```
PRINT CHR$(X)
```

aktiviert wird. Mit X = 147 wird beispielsweise ebenfalls der Bildschirm gelöscht. Wie kommt das? Tippen Sie bitte ein:

```
PRINT ASC("}{CLR/HOME}")
```

Bitte tippen Sie die Ausdrücke, die in den geschweiften Klammern

stehen, nicht als Buchstabenfolge ein, sondern zum Beispiel für PRINT ASC("}{CLR/HOME}") nach dem ersten Gänsefüßchen die Control-Taste und dann gleichzeitig die CLR-/HOME-Taste. Verfahren Sie bei den folgenden Beispielen dementsprechend.

Nach erfolgtem RETURN lesen Sie: 147. Das reverse Herzchen wird vom Computer also als Zeichen interpretiert, das den ASCII- beziehungsweise CHR\$-Code 147 trägt. Ein weiteres Beispiel: Die Zeilen

```
PRINT " { CTRL-RVS ON} TEST"
```

und

```
PRINT CHR$(18) "TEST"
```

bewirken dasselbe, weil

```
PRINT ASC("}{CTRL-RVS ON}")
```

zeigt, daß der CHR\$-Code des reversen R eben 18 ist.

Ausgehend von der Tatsache, daß jedem Steuerzeichen ein bestimmter Code in der CHR\$-Liste (siehe Handbuch) zugeordnet ist, wurde die Idee geboren, daß diese Zuordnung – um es mathematisch auszudrücken – umkehrbar eindeutig sein müsse. Das heißt, zu jedem Steuerbefehl müßte auch ein entsprechendes reverses Steuerzeichen gehören, das denselben Zweck erfüllt.

Nun gibt es zum Beispiel die Möglichkeit, über

```
PRINT CHR$(14)
```

auf Kleinbuchstaben umzuschalten. Gibt es dafür auch ein Steuerzeichen? Auf konventionellem Weg hieße die Antwort klipp und klar: nein. Aber ...

Zu Beginn war alles nur Spielerei, bis sich die erstaunlichen Möglichkeiten und Anwendungen häuften. Deshalb: spielen Sie nun bitte mit bei den folgenden Tricks.

Steuerzeichen auf illegalem Weg

Bevor wir das große Geheimnis lüften, wollen wir erst einmal an ganz und gar legalen Steuerzeichen üben, wie man sie illegal eingibt. Halten Sie sich bitte zunächst streng an das angegebene Rezept, auch wenn es noch andere Eingabeformen gibt. Das gilt auch für die späteren Beispiele. Unsere erste Übung soll darin bestehen, den Bildschirm ohne CHR\$(147) und ohne die Taste CLR/HOME zu löschen – im Direktmodus, versteht sich. Dazu geben Sie bitte hintereinander in einer Zeile ein:

1. Schritt:

```
PRINT
```

2. Schritt:

Erstes Anführungszeichen setzen. Jetzt befinden wir uns im Quote-Modus, den wir sofort wieder verlassen wollen.

Daher:

3. Schritt:

Zweites Anführungszeichen setzen.

4. Schritt:

Mit DEL eine Stelle zurückgehen. Dabei wird das zweite Anführungszeichen gelöscht. Da wir uns nicht mehr im Quote-Modus befinden, reagiert der Computer auf Steuerbefehle.

5. Schritt:

Mit CTRL-RVS ON auf Revers-Modus schalten. Das ist notwendig, da wir ja ein reverses Steuerzeichen quasi künstlich erzeugen wollen!

6. Schritt:

SHIFT S eingeben. Beim geSHIFTeten S bekommen wir eines der S-Taste zugeordneten Grafikzeichen, in diesem Fall das gewünschte Herzchen. Nun verlassen wir den Revers-Modus:

7. Schritt:

CTRL-RVS OFF. Wenn Sie wollen, können Sie noch ein Anführungszeichen zur optischen Abrundung anhängen. Was jetzt auf dem Bildschirm steht, sieht so aus, als hätten

wir nie den legalen Pfad verlassen, als hätten wir immer mit der CLR/HOME-Taste gearbeitet. Dieses Steuerzeichen ist jedoch künstlich entstanden – wird es auch wirken wie ein »echtes«? Das Drücken der RETURN-Taste überzeugt uns schnell davon, daß der Computer uns diese umständliche Manipulation nicht übelgenommen hat: es funktioniert tatsächlich!

Bedenken Sie bei der Eingabe, daß nur zwei Anführungszeichen auf dem Schirm erscheinen, obwohl insgesamt drei eingegeben wurden. Der Computer befindet sich also nach dem dritten Gänsefüßchen wieder im Quote-Modus.

An der Stelle, an der soeben mit SHIFT S das Steuerzeichen für CLR/HOME generiert wurde, kann nun jedes beliebige reverse Zeichen erzeugt werden. Diejenigen davon, die auch eine Steuerfunktion ausüben, aber nicht auf normalem Wege über direkte Tastendrücke erzeugt werden können, wollen wir »synthetische Steuerzeichen« nennen. Und nun folgen nützliche Beispiele mit diesen Steuerzeichen.

Die Synthetischen kommen

Wenn es – und das war die Frage, die alles ins Rollen brachte – anstelle der Umschaltung auf Kleinbuchstaben mit CHR\$(14) ein äquivalentes Steuerzeichen gibt, dann muß es zwischen dem reversen E für WHT (Code 5) und dem reversen Q für CURSOR DOWN (Code 17) zu suchen sein. Nummeriert man das Alphabet entsprechend durch, so findet man den Buchstaben N unter der Codezahl 14. Das reverse N müßte demnach das Steuerzeichen für die Umschaltung auf Kleinbuchstaben sein. Probieren wir es aus:

```
? ""{DEL RVS ON} N "" {RVS OFF} → RETURN
```

Es klappt! Der Computer schluckt diesen ungewöhnlichen Befehl und führt ihn aus.

Dieser Erfolg motivierte eine Suche nach allen verfügbaren synthetischen Steuerzeichen; die Ausbeute fiel jedoch zunächst recht mager aus. Die nachfolgende Tabelle 1 zeigt Ihnen die synthetischen Zeichen und ihre Wirkungen bei Bildschirmausgaben (beim Drucker sieht's etwas anders aus – dazu aber später mehr).

Steuerzeichen	CHR\$-Code	Wirkung
H	8	Die manuelle Umschaltung zwischen Groß- und Kleinbuchstaben mit Hilfe der COMMODORE- und SHIFT-Tasten wird gesperrt.
I	9	Die obige Sperre wird wieder gelöst.
M	13	RETURN. Alles, was nach dem reversen M in der Zeile steht, wird nicht mehr gedruckt. Leider ist eine echte Simulation der RETURN-Taste zum Ändern von Basic-Zeilen unter Programmkontrolle nicht möglich.
N	14	Umschalten auf Kleinbuchstaben.
T	20	DEL. Hiermit können Sie Teile oder komplette Programmzeilen

Die Zeichen dieser Tabelle müssen als reverse Zeichen wie oben erzeugt werden (Ausnahmen: T für DEL und SHIFT T für INS).

Auch wenn die Eingabe der synthetischen Steuerzeichen umständlich erscheint, man wird sich schnell daran gewöhnt

haben und auf ihre Vorteile nicht mehr verzichten wollen – nicht nur, weil sie gegenüber der CHR\$-Funktion Speicherplatz sparen.

So unscheinbar und unwichtig das SHIFT RETURN ist, so gewaltig sind die Möglichkeiten, die uns das zugehörige synthetische Steuerzeichen eröffnet. Es handelt sich um das reverse, geschiftete M (Code 141).

Das Super-Steuerzeichen

Gibt man »? " "{DEL}{ TEST {CTRL-RVS ON} {SHIFT M} {CTRL-RVS OFF} LAUF "« ein, so zeigt sich nach dem Drücken der RETURN-Taste, daß der Computer nach dem Teilstring »TEST« einen Wagenrücklauf (carriage return) mit Zeilenvorschub (line feed) durchführt und »LAUF« direkt darunter ausgibt. Genau dieses Verhalten erwartet man von SHIFT RETURN. {SHIFT M} kann jedoch mehr – viel mehr! Das nächste Beispiel soll es beweisen:

```
10 ? ""{DEL} TEST {CTRL-RVS ON} {SHIFT M} R {CTRL-RVS OFF} LAUF "
```

Geben Sie bitte dieses einzeilige Programm ein und starten Sie es. Wie erwartet erscheint »TEST« und darunter revers »LAUF«. Nun listen Sie bitte dieses Programm. Erstaunt? Das ist tatsächlich neu! Das Steuerzeichen für die Umschaltung in den Revers-Modus erscheint nicht im Listing, sondern wird entgegen aller bisherigen Kenntnisse ausgeführt. »LAUF« wird im Listing revers ausgegeben. Durch das SHIFT-RETURN-Zeichen wird der Basic-Interpreter offenbar veranlaßt, nachgestellte Steuerzeichen auch im Listing wirksam werden zu lassen. Es zeigt sich, daß diese Schlüsselfunktion alle Steuerbefehle, seien es Farbumschaltungen oder Cursor-Bewegungen, aktiviert. Ungeahnte Möglichkeiten eröffnen sich nun zur optischen Aufbereitung, das heißt Strukturierung und Gestaltung von Listings auf dem Bildschirm.

Beispiel 1:

Reverse REM-Zeile ohne sichtbares REM-Statement und ohne Zeilennummer

```
10 REM ""{DEL} {CTRL-RVS ON} {SHIFT M} {SHIFT Q} R
{CTRL-RVS OFF} TESTLAUF
20 FOR I=1 TO 10
30 PRINT "TESTLAUF":NEXT
```

Steuerzeichen	CHR\$-Code	Wirkung
		optisch verschwinden lassen. Dieses Steuerzeichen wirkt auch beim Listen eines Programms auf dem Schirm. Anwendung: partieller Listschutz.
SHIFT M	141	SHIFT RETURN. Das Super-Zeichen! Mehr dazu im Text.
SHIFT N	142	Umschalten auf Großbuchstaben.
SHIFT T	148	INS. Kann zum Ändern von Text an bereits gedruckten Text benutzt werden. Beispiel: ? "{SHIFT T}X" liefert ein X auf dem Schirm. Bei mehrfacher Abarbeitung wird stets ein weiteres X an die bereits vorhandenen angehängt.

Tabelle 1. Steuerzeichen und ihre Wirkung

Die Zeilen 20 und 30 sind Beiwerk, damit die REM-Zeile nicht so allein dasteht. Listen Sie das Programm bitte. Sie sehen die reverse Überschrift »TESTLAUF« ohne REM und ohne Zeilennummer und darunter wie gewohnt die Programmzeilen 20 und 30. Wie funktioniert dieser Trick? Das reverse

{SHIFT M} bewirkt einen Sprung zum Anfang der nächsten Zeile und gibt die nachfolgenden Steuerzeichen zur Ausführung frei. Mit dem reversen {SHIFT Q} geht der Cursor eine Zeile nach oben, also auf »1« der Zeilennummer 10. Reverses R schaltet auf reverse Zeichen um, in denen dann der REM-Text »TESTLAUF« ausgegeben wird. Dabei überschreibt der Computer die Zeilennummer sowie das REM-Statement. Das alles vollzieht sich so schnell, daß es sich der Beobachtung entzieht. Der Anwender sieht nur noch die reverse Überschrift und kann demzufolge an dieser Zeile auch nichts mehr ändern.

Sollten Sie die Original-REM-Zeile noch auf dem Bildschirm haben, so hängen Sie an das »F« von »TESTLAUF« noch das Steuerzeichen für CURSOR DOWN an. Sie werden sehen, daß beim Listen sogar eine Leerzeile zwischen Überschrift und Programm entsteht. Selbstverständlich können derartige REM-Zeilen beliebig im Programm verstreut sein. Wirkung: Im Programmlauf werden unter der reversen Überschrift »Quadratzahlen« die Ergebnisse ausgegeben, während beim Listen des Programms der PRINT-Befehl verschwiegen wird und man eine REM-Zeile wie in Beispiel 1 vermutet. Damit trägt eine PRINT-Zeile zur optischen Strukturierung eines Listings bei – ein Effekt, der bislang in dieser Form unmöglich schien. Nun einige Erläuterungen zu den Beispielsprogrammen 2 bis 4:

Beispiel 2. Farbige REM-Zeile (ändern Sie nur Zeile 10).

Beispiel 3. Wirkung: Grüne Überschrift ohne Zeilennummer und ohne REM, gelbes Listing.

Beispiel 4. Wirkung: Die bekannten Herzchen sorgen nach dem Listen jeder Zeile für ein Löschen des Bildschirms, so daß die beiden Programmzeilen weder einzeln noch insgesamt sichtbar bleiben und die Codezahlen A und B dem Anwender vorenthalten werden.

Beispiel 2:

Farbige REM-Zeile (ändern Sie nur Zeile 10)

```
10 REM ""{DEL} {CTRL-RVS ON} {SHIFT M} {SHIFT Q} {↑}
{CTRL-RVS OFF} TESTLAUF {CTRL-RVS ON} {SHIFT π}
{CTRL-RVS OFF}
```

Beispiel 3:

Listschutz

```
10 A=4711 : REM ""{DEL} {CTRL-RVS ON} {SHIFT M}
{SHIFT S}
20 B=0815 : REM ""{DEL} {CTRL-RVS ON} {SHIFT M}
{SHIFT S}
```

reverses SHIFT M für SHIFT RETURN

reverses SHIFT S für CLR/HOME

Beispiel 4:

REM-Ersatz

```
10 PRINT ""{DEL} {CTRL-RVS ON} {SHIFT M} {SHIFT S} R
{CTRL-RVS OFF} QUADRATZAHLEN
```

reverses SHIFT M für SHIFT RETURN

reverses SHIFT S für CLR/HOME

reverses R für RVS ON

```
20 FOR I=1 TO 10
```

```
30 PRINT I,I*I : NEXT
```

Wirkung: Im Programmlauf werden unter der reversen Überschrift »Quadratzahlen« die Ergebnisse ausgegeben, während beim Listen des Programms der PRINT-Befehl verschwiegen wird und man eine REM-Zeile wie in Beispiel 1 vermutet. Damit trägt eine PRINT-Zeile zur optischen Strukturierung eines Listings bei – ein Effekt, der bislang in dieser Form unmöglich schien.

Sie werden sicherlich Spaß an der Vielseitigkeit dieser neuen Methode gewinnen und eine Vielzahl von Anwendungen austüfeln.

Bisher wurde die Erzeugung synthetischer Steuerzeichen und ihre Wirkung bei Bildschirmbetrieb dargestellt. Jetzt soll von neuen Möglichkeiten für Druckeranwender die Rede

sein. Dabei wird auf den Drucker VC 1515 Bezug genommen, mit dem die Steuerzeichen ausgetestet wurden. Es ist nicht auszuschließen, daß der eine oder andere Druckertyp unterschiedlich reagieren wird. Die wichtigsten Steuerfunktionen dürften jedoch auf allen Geräten die gleichen Reaktionen hervorrufen.

Benutzen Sie die synthetischen Steuerzeichen, um Ihre Druckerlistings übersichtlicher zu machen.

Bevor es nun zur Sache geht, ein kurzer Abriß des vorangegangenen Abschnitts:

Zur Cursor-Steuerung für RVS ON/OFF oder zum Beispiel für Farbumschaltungen stehen dem VC 20- beziehungsweise C 64-Anwender zwei Möglichkeiten zur Verfügung. Erstens kann die CHR\$()-Funktion genutzt werden (Beispiel ?CHR\$(19) bewirkt HOME) und zweitens erlauben reverse Steuerzeichen eine direkte und kurze Eingabe der Steuerbefehle (Beispiel: das reverse S veranlaßt ebenfalls HOME). Nun existieren jedoch einige CHR\$()-Codes, für die äquivalente Steuerzeichen nicht über zugehörige Tasten abgerufen werden können. So schaltet CHR\$(14) beispielsweise auf Kleinbuchstaben um. Mit Hilfe der ASC-Funktion zeigt sich, daß das reverse N den CHR\$()-Code 14 trägt. Damit kann dieses Zeichen ebenfalls zur Steuerung herangezogen werden. Da außer den konventionellen Steuerzeichen keinerlei reverse Symbole in Strings vorkommen können, muß das reverse N quasi künstlich erzeugt werden (daher der Name »Synthetische Steuerzeichen«). Aber das ist kein Problem! Wir haben bereits ein Eingabeverfahren kennengelernt, das ich aber jetzt nicht wiederholen möchte. Statt dessen möchte ich Ihnen ein anderes Verfahren zeigen, das vielleicht ein wenig übersichtlicher ist als das bereits beschriebene.

Geben Sie die Programmzeile, die ein im String stehendes Steuerzeichen erhalten soll, wie gewohnt ein. Reservieren Sie dabei jedoch mittels Space die Stelle im Textstring, wo später das synthetische Steuerzeichen stehen wird. Schließen Sie die Eingabe der Zeile mit Betätigung der Return-Taste ab. Nun können sie ohne Schwierigkeiten den Cursor auf die bewußte Stelle bewegen, durch gleichzeitiges Drücken von CTRL und RVS ON in den Revers-Mode schalten (es erscheint kein reverses R!) und schließlich den freigehaltenen Platz mit dem entsprechenden Reversezeichen belegen (in unserem Beispiel also mit dem reversen N). Die so ergänzte Programmzeile wird jetzt durch erneutes Drücken der Return-Taste verlassen. Fertig. Auf diese Weise lassen sich sowohl die altbekannten als auch die neuen Steuerzeichen leicht erzeugen und entsprechend ins Programm einfügen. In den weiteren Ausführungen werde ich die synthetischen Steuerzeichen vereinfachend in geschweiften Klammern darstellen (zum Beispiel reverses N = {N}).

Die Drucker-»Synthies«

Nun zum eigentlichen Thema: Man könnte fast wetten, daß einige Druckerbesitzer unter unseren Lesern inzwischen beim Experimentieren mit synthetischen Steuerzeichen nicht schlecht gestaunt haben. Denn auch hier eröffnen sich neue Möglichkeiten, an die bislang nicht zu denken war. So sind zwar keine spektakulären Effekte in laufenden Programmen zu erzielen, dafür jedoch hat man erstmals die Möglichkeit, Druckerlistings zu manipulieren. Zuvor möchte ich Ihnen jedoch eine Liste der Steuerzeichen (Tabelle 2) geben. Es ist zu beachten, daß die mit *) gekennzeichneten Steuerzeichen nicht zu den synthetischen zählen, da sich diese direkt über Tasten eingeben lassen. Sie wurden nur der Vollständigkeit halber mit in die Tabelle aufgenommen.

Steuerzeichen	CHR\$-Code	Wirkung
{H}	8	Umschaltung auf Grafik-Modus
{J}	10	Zeilenvorschub
{M}	13	RETURN. Nachfolgende Zeichen werden nicht mehr gedruckt.
{N}	14	Umschaltung auf Breitschrift
{O}	15	Umschaltung auf normale Schriftbreite
{P}	16	Festlegung der Druckstartposition
*){Q}	17	Umschaltung auf KLeinbuchstaben
*){R}	18	RVS ON
{Z}	26	Zeichenwiederholung (Grafik)
{}	27	Punktadresse für Druckstart
{SHIFT M}	141	SHIFT RETURN. Nachfolgende Zeichen werden in einer neuen Zeile gedruckt.
*){SHIFT Q}	145	Umschaltung auf Großbuchstaben
*){SHIFT R}	146	RVS OFF

Tabelle 2. Liste der Steuerzeichen

Nun zu den Erläuterungen derjenigen Zeichen, die sinnvolle Anwendungen erlauben.

Grafik im Listing

Das reverse H gestattet die Umschaltung des Druckers in den Grafik-Modus. Alle nachfolgenden Zeichen im String werden – sofern ihr jeweiliger CHR\$-Code größer als 127 ist – als Grafikinformatoren interpretiert. In diesem Modus bleibt der Drucker so lange, bis er mit {N} oder {O} auf Schriftbetrieb zurückgeschaltet wird. Betrachten Sie bitte Beispiel 1.

Zwischen »TEST« und »LAUF« befindet sich in Bild 2 ein selbstdefiniertes Grafikzeichen, das sowohl im Programm- als auch im Listing ausgegeben wird. Auch wenn Sie es zunächst nicht glauben – Sie sehen ein Original-Listing und nicht etwa gePRINTete Textzeilen. Ich will Ihnen den Trick verraten. In einer Vergrößerung sieht das Grafiksymbolsymbol aus wie im Bild 3 gezeigt.

Die Umrechnung des dualen Bitmusters in Dezimalzahlen liefert diejenigen CHR\$-Codes, durch die das Zeichen spaltenweise beim Drucken entsteht. Man muß also nur wiederum synthetische Zeichen finden, die die gesuchten CHR\$-Codes tragen. (Tabelle 3).

"TEST {H}	{SHIFT H}	{SHIFT MINUS}	{SHIFT T}	{SHIFT T}	{SHIFT T}
8	136	156	148	148	148
{SHIFT T}	{SHIFT T}	{SHIFT MINUS}	{SHIFT H}	{SHIFT *}	{SHIFT *}
148	148	156	136	128	128
{O}LAUF "					
15					

Tabelle 3. Aus dem Beispiel in Bild 3 ergeben sich diese Charakterstring-Codes.

Mit dieser Folge von reversen Zeichen erzielen Sie also den Ausdruck, der im Bild 2 abgebildet ist.

Folgt dem {H} keine Grafikinformatoren, das heißt ist der CHR\$-Code kleiner als 128, so kommt es zum Abbruch des Listings. Diese Tatsache kann man sich bewußt zunutze machen, wenn man ohne größeren Programmieraufwand einen Drucker-Listenschutz in sein Programm einbauen will. Der Bildschirmbetrieb leidet darunter nicht, da {H} dort nur eine Verriegelung der Umschaltung zwischen Groß- und Kleinschrift bewirkt.

Zeilenvorschub

Die Beispiele in Bild 4 und 5 zeigen, daß {J} einen Zeilenvorschub mit Rückwagenlauf verursacht. In Bild 4 befindet sich

das reverse J zwischen »TEST« und »LAUF«, bei Bild 5 am Schluß des Strings. Da man das zweite Anführungszeichen eines Strings weglassen kann, sofern keine weiteren Basic-Befehle mehr in dieser Zeile folgen sollen, ergibt sich so die Möglichkeit, echte Leerzeilen mit {J} im Listing (Bild 4) zu erzeugen.

Breitschrift - Schmalschrift

Während {N} beim Bildschirmbetrieb auf Kleinbuchstaben umschaltet, zeigt der Drucker eine andere Wirkung. Er wird veranlaßt, alle nachfolgenden Zeichen in doppelt breiter Schrift auszugeben. Es ist also bei der Anwendung von {N} Vorsicht geboten, da ein für den Bildschirm konzipiertes Programm beim Druckerlisting unerwünschte Steuerungen zur Folge haben kann. In diesen Fällen sollte man {N} wieder durch den konventionellen Befehl CHR\$ ersetzen.

Wollen Sie jedoch die Breitschrift zur übersichtlichen Gestaltung eines Listings nutzen, so können Sie ruhig das {N} hier und da im Programm verstecken. Es bleibt demjenigen, der das Programm lesen soll, sowieso verborgen. Wenn nur eine Hervorhebung einzelner Worte, Befehle oder Zeilen gewünscht wird, muß die Breitschriftphase entsprechend mit {O} beendet werden. Im Beispielprogramm in Bild 6 sehen Sie dieses Vorhaben realisiert. Hier bewirkt das {N} vor dem Wort »Test« die Umschaltung auf Breitschrift, während ein {O} vor »Lauf« diesen Modus beendet.

Die Druckerstartpositionierung – vergleichbar mit TAB – wird normalerweise mit CHR\$(16) und nachgestellter Startadresse vorgenommen. Also etwa:

```
PRINT #4,CHR$(16) "10 TEST"
```

Diese Zeile druckt beginnend in Spalte 10 nur das Wort »Test« aus. Die im String vorangestellte Zahl dient dabei als Startadresse. Das kürzere Äquivalent sieht nun wie folgt aus:

```
PRINT #4, "{P} 10 TEST"
```

Doch Vorsicht! Obwohl diese Anweisung während des Programmablaufs korrekt ausgeführt wird, verschluckt sie der Drucker im Listing vollständig. Das Beispiel in Bild 7 zeigt, daß von {P} 10 zwischen »TEST« und »LAUF« nicht mehr zu bemerken ist.

Damit verbietet sich die Anwendung des synthetischen {P} bei Programmen, die zum Beispiel zur Veröffentlichung vom Drucker dokumentiert werden sollen.

Für die Zeichen {Z} Zeichenwiederholung und {} (Punktadresse für Druckerstart) haben sich bislang noch keine sinnvollen Einsatzmöglichkeiten ergeben. Sie sollen daher jetzt nicht weiter diskutiert werden. Auch auf die Beschreibung von {Q}, {R}, {SHIFT Q} und {SHIFT R} möchte ich verzichten, da sie keine synthetischen Zeichen sind und sich damit »ganz normal« verhalten.

Ich persönlich benutze die synthetischen Steuerzeichen gern zur Gestaltung von Listings. Wie so etwas aussehen kann, habe ich Ihnen im abschließenden Beispiel (Bild 8) zusammengestellt.

Vielleicht versuchen Sie einmal, die versteckten und unsichtbaren Steuerzeichen herauszufinden.

Die Suche nach den Synthetischen

Das folgende Programm ermöglicht die systematische Suche nach allen vom Betriebssystem unterstützten Steuerzeichen (Listing 1). Man ist nun nicht mehr angewiesen auf zum Teil lückenhafte Tabellen synthetischer Steuerzeichen, sondern kann sich statt dessen selbst auf die Suche begeben.

Das Programm-Listing zur Steuerzeichensuche (Listing 1) gibt nach Eingabe des gewünschten ASCII-Wertes alle Tastenkombinationen aus, deren Betätigung die Tastaturdecodieroutine dazu veranlaßt, den entsprechenden ASCII-Wert in den Tastaturpuffer zu schreiben.

0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	16
0	0	0	0	0	0	0	0	0	0	0	32
0	0	0	0	0	0	0	0	0	0	0	64
											+128
136	156	148	148	148	148	148	156	136	128	128	

Bild 3. Vergrößertes Grafiksymbol aus dem Beispiel in Bild 1

TEST=LAUF

```
10 REM GRAFIK-MODUS
20 REM IM LISTING
30 OPEN4,4
40 PRINT#4,"TEST=LAUF"
50 PRINT#4:CLOSE4

READY.
```

Bild 2. Beispiel zum Grafik-Modus

TEST LAUF

```
10 REM ZEILENVORSCHUB
20 REM IM LISTING
30 OPEN4,4
40 PRINT#4,"TEST
LAUF"
50 PRINT#4:CLOSE4

READY.
```

Bild 4. Beispiel-Programm zum Zeilenvorschub im Wort

TESTLAUF

```
10 REM ZEILENVORSCHUB
20 REM IM LISTING
30 OPEN4,4
40 PRINT#4,"TESTLAUF"

50 PRINT#4:CLOSE4

READY.
```

Bild 5. Beispiel-Programm: Zeilenvorschub am Wortende

TESTLAUF

```
10 REM BREITSCHRIFT
20 REM IM LISTING
30 OPEN4,4
40 PRINT#4,"TESTLAUF"
50 PRINT#4:CLOSE4

READY.
```

Bild 6. Beispiel zum Umschalten von Schmal- auf Breitschrift

TEST LAUF

```
10 REM DRUCKSTARTPOSITION
15 REM IST IM LISTING
20 REM NICHT SICHTBAR
30 OPEN4,4
40 PRINT#4,"TESTLAUF"
50 PRINT#4:CLOSE4

READY.
```

Bild 7. Beispiel: Ab Spalte 10 wird gedruckt. Die Druckeranweisung wird nicht gelistet.

```
100 REM DEMO ZUR OPTISCHEN STRUKTURIERUNG
110 REM VON DRUCKER-LISTINGS MIT
120 REM SYNTHETISCHEN
130 REM STEUERZEICHEN
200 REM * HAUPTPROGRAMM *
210 FORI=1TO64
220 PRINT"64 ER-MAGAZIN"
230 GOSUB300:NEXT I
240 END
300 REM * UNTERPROGRAMM *
310 FORJ=1TO50:NEXT
320 RETURN:"
400 REM"ORIGINAL-LISTING!"

READY.
```

Bild 8. Beispiel zur Gestaltung von Listings durch synthetische Steuerzeichen

Es werden allerdings auch solche Kombinationen aufgeführt, deren »ASCII-Werte« zwar in den Decodiertabellen (die ab \$EB81 im Betriebssystem beginnen) an entsprechender Stelle aufgeführt sind, jedoch NICHT im Tastaturpuffer erscheinen (zum Beispiel die »2« bei Betätigung der SHIFT-Taste). Der Leser wird diese jedoch schnell von der ersten Gruppe unterscheiden können.

Zum Schluß noch zwei Bemerkungen:

1) Bei gleichzeitigem Erscheinen von zwei oder mehreren Tastenkombinationen für einen bestimmten ASCII-Code kann oft – jedoch nicht immer – eine Kombination gegen eine andere ausgetauscht werden, ohne das Endresultat zu verändern (Beispiel für eine Ausnahme: Sowohl die RUN/STOP-Taste wie auch die CTRL C-Kombination belegen gemäß der Decodiertabellen den ASCII-Wert »3«, jedoch kann mit »CTRL-C« kein Programm gestoppt werden, da für die Abfrage der RUN/STOP-Taste eine gesonderte Routine zuständig ist, die nur diese Taste erkennt).

2) Alle »synthetischen Steuerzeichen«, für deren Erzeugung der Basic-Interpreter zuständig ist, kann dieses Programm nicht erkennen, da es lediglich auf die Tastaturdecodiertabellen im Kernel Bezug nimmt.

Da das Programm lediglich die Tastaturdecodiertabellen des Betriebssystems benötigt (und die dort aufgeführten ASCII-Werte in ein ARRAY einliest) kann es leicht durch entsprechende Abänderung der in Zeile 20 enthaltenen Anfangsadressen in eine auch auf dem VC 20 laufende Version umgeschrieben werden.

(Jürgen Wagner/Engin Gülen/cg)

```
1 REM * STEUERZEICHEN *
2 REM * FUER DEN CBM-64 *
3 REM * ENGIN GULEN *
4 REM * POSTWEG 2 *
5 REM * 4192 KALKAR 1 *
9 POKE53280,0:POKE53281,0
10 DIMAS(3,63)
18 REM ANFANGSADRESSEEN DER
19 REM TASTATURDEKODIERTABELLEN
20 DATA60289,60354,60419,60536
30 FORI=0TO3:READA(I):NEXT
40 FORI=0TO3:FORJ=0TO63
50 AS(I,J)=PEEK(A(I)+J)
60 NEXTJ,I
159 REM TASTATURMATRIX
160 DATAEL,RETURN,CRSR(RIGHT),F7,F1,F3,F5,CRSR(DOWN)
161 DATA3,W,A,4,Z,S,E,SHIFT(L)
162 DATA5,R,D,6,C,F,T,X
163 DATA7,Y,G,8,B,H,U,V
164 DATA9,I,J,0,M,K,O,N
165 DATA+,P,L,-,.,DOPPELPUNKT,@,KOMMA
166 DATAE,*,;,HOME,SHIFT(R),=,↑,/,
167 DATA1,+,CTRL,2,SPACE,C=,Q,RUN/STOP
170 DIMTA(64):FORI=0TO63:READTA(I):NEXTI
180 DATA-----,SHIFT,C=,CTRL,
190 FORI=0TO3:READA(I):NEXT
200 INPUT"VORGEBEN ASCII-KODE ";AS%
210 IFAS%<0ORAS%>255THEN200
220 PRINT:PRINT:FORI=0TO3:FORJ=0TO63
230 IFAS%=AS(I,J)THENPRINT"■AS(I),:GOTO250
240 GOTO270
250 IFLEFT$(TA$(J),2)=LEFT$(AS(I),2)THENPRINT:GOTO270
260 PRINT"■TA$(J)"
270 NEXTJ,I
280 PRINTSPC(193)"■TASTE "
290 GETA$:IFA$=""THEN290
300 GOTO200

READY.
```

Listing 1. Steuerzeichensuche