



lerdings nur den Zeichensatz 2, der Groß- und Kleinbuchstaben, Ziffern und einige Grafikzeichen enthält (normal und invers), aber nur 2 KByte RAM beansprucht.

»DEUZEI« benutzt den RAM-Speicher wie folgt: Zeichengenerator: Page 16 bis 23; Bildschirmspeicher: Page 24 und 25; Basicprogramme ab Page 26 (Bild 2). Eine Page entspricht einem 256-Byte-Block. Der Bildschirmspeicher befindet sich beim VC 20 ab 8 KByte Erweiterung normalerweise in Page 16 und 17 (4096 bis 4607), muß bei »DEUZEI« allerdings »umziehen«; dies erfährt das Betriebssystem durch entsprechende Codierung der Speicherstellen 648 (siehe Zeile 130) und 36869 (zugleich Pointer für den Zeichengenerator).

Das eigentliche Programm »DEUZEI« beginnt bei der Speicherstelle 6656 (Page 26). Da man es nach dem Generieren des neuen Zeichensatzes ja nicht mehr benötigt, wird es beim Einladen eines neuen Programms ab Page 26 gelöscht. Es fehlen dem neuen Programm also nur genau 2 KByte RAM-Speicherplatz, den der neue Zeichensatz beansprucht.

#### »DEUZEI« intern

Zeile 110-120:	Ab Speicherstelle 700 wird das Maschinenprogramm (35 Byte) abgelegt (unbenutzter RAM-Bereich).
Zeile 130:	Pointer für neuen Zeichengenerator-Bereich und Bildschirm-Bereich.
Zeile 140:	Zeichen-Set 2 wird durch Starten des Maschinenprogramms ab Page 16 abgelegt.
Zeile 210-270:	Bitmuster für deutsche Umlaute wird in neuen Zeichensatz gePOKEd.
Zeile 300-360:	Bitmuster für Umlaute (invers)
Zeile 410:	Umschaltperre zum anderen Zeichen-Set (nur ein Zeichen-Set existiert im neuen Zeichengenerator-Bereich).
Zeile 420-450:	Bildschirm-Display (neue Zeichen erscheinen)
Zeile 510-530:	Data für Maschinenprogramm
Zeile 540-600:	Data für Umlaute Ä,Ö,Ü,ä,ö,ü,ß
Zeile 700-760:	Data für inverse Umlaute Ä,Ö,Ü,ä,ö,ü,ß

#### CHR\$-Codes:

17 =	Cursor nach unten
18 =	Inverse Zeichen an
28 =	rot
31 =	blau
147 =	clr home
156 =	purpur

## Eingabe von »DEUZEI«

Wenn man nun das Programm »DEUZEI« in den Computer eingibt und übereifrig mit RUN startet, wird man eine böse Überraschung erleben: Das Programm stürzt sofort ab und ist zudem auch noch gelöscht!

Des Rätsels Lösung: Normalerweise werden Basicprogramme ja ab Page 18 (Speicherstelle 4608 = 18 \* 256) abgelegt. In diesen Bereich transformiert allerdings das Maschinenprogramm den neuen Zeichensatz und »zerstört« frühzeitig das Basicprogramm. Man muß folglich Sorge tragen, daß das Programm ab Page 26 im Speicher steht. Hierzu sollte man folgendermaßen vorgehen:

1. »DEUZEI« eingeben und auf Band oder Floppy abspeichern (kein RUN!)
2. Programm mit NEW löschen und folgende Basiczeile eingeben:  
1 POKE 44,26:POKE 6656,0:PRINT CHR\$(3):RUN
3. RETURN-Taste drücken, aber nicht RUN!
4. POKE 44,26:POKE 6656,0 eintippen, dann RETURN-Taste drücken
5. Eingabe von NEW; RETURN-Taste
6. Einladen des abgespeicherten »DEUZEI«

02BC	A9 00	LDA #\$00	Low-byte Zeichenbereich laden
02BE	85 FB	STA \$FB	speichern in dez. 251
02C0	A9 88	LDA #\$88	High-byte Zeichenbereich laden
02C2	85 FC	STA \$FC	speichern in dez. 252
02C4	A9 00	LDA #\$00	Low-byte neuer Z.-Bereich laden
02C6	85 FD	STA \$FD	speichern in dez. 253
02C8	A9 10	LDA #\$10	High-byte neuer Z.-Bereich lad.
02CA	85 FE	STA \$FE	speichern in dez. 254
02CC	A2 08	LDX #\$08	X-Register: Zahl der Blöcke
02CE	A0 00	LDY #\$00	Y-Register Null setzen (Zähler)
02D0	B1 FB	NEXT LDA(\$FB),Y	Laden von Koppeladr. (i.indiz.)
02D2	91 FD	STA(\$FD),Y	Speichern in Koppeladresse
02D4	88	DEY	Zähler -1
02D5	D0 F9	BNE NEXT	Y nicht 0, dann Sprung zu NEXT
02D7	E6 FC	INC \$FC	High-byte Z.-Bereich +1
02D9	E6 FE	INC \$FE	High-byte neuer Z.-Bereich +1
02DB	CA	DEX	Block-Zähler -1
02DC	D0 F2	BNE NEXT	X nicht 0, dann Sprung zu NEXT
02DE	60	RTS	zurück zu Basic

**Listing 2. Assembler-Listing zu »DEUZEI«. Das Assemblerprogramm ist im Basiclader nach Listing 1 bereits enthalten und braucht nicht zusätzlich eingegeben zu werden.**

7. POKE 44,18 eintippen und unter neuem Namen abspeichern.

Die kleine Mühe hat sich sicherlich gelohnt, denn das neu abgespeicherte Programm ist nun absolut »wartungsfrei«. Es kann ganz normal eingeladen werden, setzt automatisch die Basicuntergrenze auf Page 26, generiert den neuen Zeichensatz in 2 bis 3 Sekunden und erlaubt das Einladen beliebiger Basicprogramme ohne Zusatz-POKES.

Erklärung für obige Prozedur: Die erste Basiczeile muß am Anfang von Page 18 liegen, der Interpreter erwartet das so. Startet man das »umgebaute« Programm, so wird in dieser Zeile der neue Basicbeginn mitgeteilt, und zwar mit POKE 44,26.

Dieses Miniprogramm wird mit PRINT CHR\$(3) abgebrochen und mit RUN in der gleichen Zeile das eigentliche »DEUZEI« ab Page 26 gestartet. Dies sitzt bereits richtig, da man es (siehe unter 4) gleich in Page 26 geladen hat. Danach wurde der Basicanfang wieder durch POKE 44,18 herabgesetzt und somit das funktionsfähige Programm abgespeichert.

Dieses Vorgehen bedeutet zwar, daß beim Abspeichern fast 2 KByte unbrauchbares Material mit abgespeichert wird, aber der Vorteil der besonderen Bedienungsfreundlichkeit überwiegt doch eindeutig.

## Arbeiten mit »DEUZEI«

Nach dem Einladen des — wie oben beschrieben bearbeiteten — Programms »DEUZEI« erscheinen die neuen Sonderzeichen auf dem Bildschirm normal/invers, und beliebige Programme können geladen werden.

Die neuen Zeichen und ihr CHR\$- beziehungsweise POKE-Code:

	ä	ö	ü	Ä	Ö	Ü	ß
CHR\$:	91	92	93	123	124	125	94
POKE:	27	28	29	91	92	93	30
statt:	[	£	]	SHIFT	[	SHIFT £	SHIFT ]
							1

Dieser Zeichensatz bleibt erhalten, solange der VC 20 eingeschaltet ist, wird allerdings durch Betätigen der RUN STOP/RESTORE-Tasten abgeschaltet; es wird wieder der Pointer auf den normalen Zeichensatz gesetzt. Durch Eingabe von: POKE 36869,236:POKE 648,24:CLR kann man aber wieder den neuen Zeichensatz verfügbar machen. Leider sind sich offensichtlich nicht alle Produzenten von Drucker-Interfaces bezüglich der CHR\$-Codes für die deutschen Um-

laute einig. So kann es vorkommen, daß bei bestimmten Druckern das »ß« nicht gedruckt wird, wenn es auch noch so deutlich am Bildschirm prangt. Das ist allerdings kein Beinbruch, man muß nur herausfinden, welchen CHR\$-Code dieses Zeichen vom Hersteller erhalten hat. Eine Anpassung von »DEUZEI« ist dann mit Hilfe der Tabellen des VC 20-Handbuchs einfach:

1. In CHR\$-Tabelle Zeichen suchen, das der Code-Nummer entspricht.
2. In POKE-Tabelle den POKE-Wert für dieses Zeichen ablesen.
3. POKE-Wert\*8+4096 ergibt erste Speicherstelle für neues Zeichen; diesen Wert im Listing ersetzen.

Beispiel: Buchstabe »ß« ist als CHR\$(94) vorgesehen.

Normales Zeichen (Pfeil nach oben) entspricht POKE 30.  $30 * 8 + 4096 = 4336$ . In Zeile 270 des Listings ist dieser Wert zu finden. Ist nun der CHR\$-Code beispielsweise für »ß« = 126 (entspricht dem Zeichen π), so ergibt sich aus der POKE-Tabelle:  $94 * 8 + 4096 = 4848$  als neuer Wert im Listing.

Um die entsprechenden Speicherplatzdaten für die inversen Zeichen zu finden, muß man nur zu den berechneten Wer-

ten noch 1024 addieren (vergleiche DATA-Zeilen 700 bis 760 in Listing 1).

## »DEUZEI«-Ausbau

Wem die deutschen Sonderzeichen nicht ausreichen für seine verschiedenen Vorhaben, der kann natürlich recht einfach in das Programm noch zusätzliche »Phantasiezeichen« einbauen. Das Auffinden der gewünschten Speicherstelle und die Berechnung der 8 Byte-Werte für das Bitmuster der 8x8-Zeichenmatrix sollten ja keine Schwierigkeiten mehr bereiten.

Als Beispiel sei der Ersatz des »Klammeraffen« (CHR\$:64; POKE:0) durch eine »Brezel« aufgezeigt, die ab Speicherstelle  $0 * 8 + 4096 = 4096$  codiert wird (Bild 3). Diese beiden Programmzeilen wären hier einzufügen:

```
361 forbr=0to7:readda%:poke4096+br,da%:next
770 data0,60,66,165,153,153,102,0
```

In analoger Weise kann man sich auch leicht beliebige andere Zeichen definieren.

(Peter Wülknitz/ev)

Speicher:	27	26	25	24	23	22	21	20	Dez.-Wert
34824									0
34825									0
34826			1	1	1				56
34827						1			4
34828			1	1	1	1			60
34829	1					1			68
34830			1	1	1		1		58
34831									0

Bild 1. Die 8 x 8-Matrix des Buchstabens »a«

Speicher:	27	26	25	24	23	22	21	20	Dez.-Wert
4096									0
4097			1	1	1	1			60
4098		1					1		66
4099	1		1			1		1	165
4100	1			1	1			1	153
4101	1			1	1			1	153
4102		1	1			1	1		102
4103									0

Bild 3. Die 8 x 8-Matrix des Grafiksymbols »Brezel«

Zeichen-	ROM		Zeichen-	ROM	
Set 2	136-143	34816-36863	Set 2	136-143	34816-36863
Set 1	128-135	32768-34815	Set 1	128-135	32768-34815
Basic-RAM	18-127	4608-32767	Basic-RAM	26-127	6656-32767
TV-Bereich	16 + 17	4096-4607	TV-Bereich	24 + 25	6144-6655
3 KByte Erweit.	4-15	1024-4095	Zeichen-Set (neu)	16-23	4096-6143
1 KByte RAM (benutzt)	0-3 Page	0-1023	3 KByte Erweit.	4-15	1024-4095
			1 KByte RAM (benutzt)	0-3 Page	0-1023

Bild 2. VC 20-Speicher ab 8 KByte Erweiterung