

mindestens eine Zeile enthält! Der Grund liegt darin, daß das System erst das Ende einer Zeile, gekennzeichnet durch eine 0, gefunden haben muß, bevor der Sprung in die nächste Zeile erfolgt.

Vorgehen: Man gibt im Direktmodus ein:  
 fori=1025to1111:pokei,peek(i+1024):next:poke43,peek  
 (43)-4

Der Wert 1111 aus mnemonischen und Sicherheitsgründen (1111 - 1024 > 80, Anzahl der Zeichen in einer Basic-Zeile). Die »Verschiebung« dauert etwa eine Sekunde.

Man listet einige Zeilen des Programms, setzt den Cursor auf die erste Zeile und übernimmt sie mit RETURN. Dadurch werden auch alle anderen Zeiger richtig gesetzt, wobei das Betriebssystem gleichzeitig die einzelnen Zeilen lückenlos in steigenden Nummern aneinanderfügt. (Das ist übrigens auch der Grund, warum es zwar prinzipiell möglich, jedoch nicht einfach ist, etwa relokatable Maschinenprogramme zwischen den Basic-Zeilen zu verstecken. Über die Möglichkeit, Textteile in Basic-Zeilen zu verstecken, werden wir bei anderer Gelegenheit reden). Das Programm ist jetzt auch auf CBM-Geräten lauffähig, von systemgebundenen Befehlen natürlich abgesehen.

Aber auch diese Prozedur ist zeitraubend, wenn mehrere Programme von einem System auf das andere übertragen werden müssen. Das kleine Basic-Programm (siehe Listing) namens »transfer« wäre in diesem Fall eine bessere Alternative: Es erzeugt ein Maschinenprogramm ab Adresse 960 und löscht sich anschließend selbst. Das Maschinenprogramm bleibt, so lange der Computer nicht ausgeschaltet wird, in diesem Bereich resistent, verschiebt beim Abruf mit »sys 960« einen einzigen Block des C 64-Programms ab Adresse 2049 auf Adresse 1025 aufwärts und setzt auch die Zeiger für den Beginn der Variablen richtig.

Vorgehen: Das Basicprogramm »transfer« laden und mit RUN starten. Anschließend das zu verschiebende Programm laden, »sys 960« eintippen und mit RETURN abschließen. Dadurch werden die ersten 256 Byte des Basicprogramms »herunter« geholt. Das verschobene Programm läßt sich jetzt listen. Cursor auf die erste Zeile setzen und mit RETURN übernehmen. Gegebenenfalls eine eigene REM-Zeile als Zeile mit der niedrigsten Nummer übernehmen!

(K. Petanides/rg)

## User-Port-Display

**Mit dem User-Port-Display lassen sich viele Steuerungsprobleme lösen. Dabei ist es manchmal wichtig zu erkennen, ob die gewünschten Daten anliegen. Dieses Problem löst das folgende Programm.**

Nachdem ich mir im Sommer '83 einen C 64 zugelegt hatte, begann ich mich als vorheriger Nur-Elektroniker sofort für die Hardware und den User-Port des Computers zu interessieren.

Während einiger Steuerexperimente kam mir die Idee zu diesem Programm.

Es eignet sich gut zum Austesten eigener Ein/Ausgabeoperationen. Möchte man zum Beispiel fremde Drucker anschließen, mit anderen Computern (zum Beispiel anderen C 64) kommunizieren oder irgendwelche Steuerungsprobleme lösen, so kann man mit Hilfe dieses Programmes jederzeit (auch während der Basic-Programmbearbeitung) erkennen, ob am User-Port die gewünschten Daten anliegen.

Die Routine ist in Maschinensprache geschrieben, das Listing zeigt den Basic-Lader. Nach dem Starten des Programms wird in der rechten oberen Ecke des Bildschirms nach einmaligem Aufruf ständig binär der momentane Zustand des Datenregisters angezeigt.

Mit SYS 49309 setzt das Programm den IRQ-Vector auf die eigentliche Anzeigenroutine.

Mit RUN/STOP-RESTORE wird der Zeiger wieder zurückgesetzt.

(Jan Legenhausen/gk)

```

10 rem *****
15 rem ***   userport-display   ***
20 rem ***                                     ***
30 rem *** (c) by jan legenhausen ***
40 rem ***                                     ***
42 rem ***           nocken 11           ***
44 rem ***                                     ***
46 rem ***           56 wuppertal 11       ***
48 rem ***                                     ***
50 rem ***           fuer den c-64        ***
55 rem *****
60 rem *** start mit sys (49309) ***
70 rem *****
100 data 169, 48, 157, 30, 4, 169, 0, 15
7, 30, 216, 173, 1
110 data 221, 234, 96, 169, 192, 141, 1,
192, 173, 1, 221, 162
120 data 0, 160, 46, 140, 0, 192, 41, 12
8, 208, 98, 32, 9
130 data 192, 232, 160, 59, 140, 0, 192,
41, 64, 208, 85, 32
140 data 9, 192, 232, 160, 72, 140, 0, 1
92, 41, 32, 208, 72
150 data 32, 9, 192, 232, 160, 85, 140,
0, 192, 41, 16, 208
160 data 59, 32, 9, 192, 232, 160, 98, 1
40, 0, 192, 41, 8
170 data 208, 46, 32, 9, 192, 232, 160,
111, 140, 0, 192, 41
180 data 4, 208, 33, 32, 9, 192, 232, 16
0, 124, 140, 0, 192
190 data 41, 2, 208, 20, 32, 9, 192, 232
, 160, 137, 140, 0
200 data 192, 41, 1, 208, 7, 32, 9, 192
210 data 76, 49, 234, 234, 169, 49, 157,
30, 4, 169, 0, 157, 30, 216, 173, 1
220 data 221, 108, 0, 192, 169, 24, 141,
20, 3, 169, 192, 141
230 data 21, 3, 96
2000 for=49161to49152+167:reada:zz=zz+a
:pokes,a:next
2100 ifzz<>17028thenprint"fehler in data
-zeilen!":stop
2110 sys49309
ready.

```

**Der Basic-Lader des Programms  
 »User-Port-Display«**