

Die große Software-Vielfalt der CBMs für den C 64 ausnützen

Den meisten Anwendern wird sich schon einmal das Problem gestellt haben, wie sie Programme von anderen Computersystemen auf ihr eigenes übertragen können. Dieser Artikel sollte es jedem ermöglichen, auch komplexe Basicprogramme der weitverbreiteten Commodore-Systeme CBM 30XX, CBM 40XX (hier kurz als CBMs bezeichnet) und C 64 füreinander umzuschreiben.

Hat man das Programm fertig eingegeben und auf Kassette oder Diskette abgespeichert, so ist das Laden von CBM-Programmen in den C 64 kein Problem: Disketten- und Kassettenformat sind identisch (zumindest bei den Doppelaufwerken 2031 und 4040 mit dem Einzelaufwerk VC 1541). Der Basic-Benutzerspeicher beginnt beim C 64 mit der Speicherstelle 2048 und endet bei 40959. Ein spezielles »Verschiebeprogramm« im C 64 sorgt dafür, daß CBM-Programme in den richtigen Speicherbereich geladen werden. Wer jedoch schon einmal versucht hat, C 64-Programme in einen CBM zu laden, wird damit keinen Erfolg gehabt haben, denn die CBMs besitzen kein solches Programm. Dort beginnt der Speicher für Basic-Programme bei 1024 und endet, je nach RAM-Ausbaustufe (8 KByte, 16 KByte oder 32 KByte) bei spätestens 32767. In einen CBM geladene C 64-Programme sind ohne Verbiegung von Zeigern weder zu listen noch zu starten. Man könnte bei den CBMs nach dem Laden den Zeiger für den Programmanfang verändern (durch den Befehl »POKE 40,0:POKE 41,8«), aber man müßte dies jedesmal nach dem Laden tun, ein auf die Dauer sehr umständliches Verfahren. Ein entsprechendes »Verschiebeprogramm«, welches diesen Aufwand vermeidet, finden Sie unter der Überschrift »Von den Kleinen auf die Großen« in dieser Ausgabe. Nun zur eigentlichen Arbeit, dem Umschreiben: Alle Computer haben die gleichen Basic-Anweisungen (bis auf die CBMs der Serie 4000, welche zusätzlich komfortable Diskettenbefehle besitzen). Sie unterscheiden sich im wesentlichen nur durch die Speicherbelegung und die Farbe, die Sprites und die hochauflösende Grafik des C 64.

Der Bildschirmbereich:

Die meisten Basicprogramme erstellen bewegte Grafiken mit Hilfe der Befehle PEEK und POKE. Im Bildschirmbereich wird durch den Befehl »POKE X, Y« an der Stelle mit der Adresse X das Zeichen mit dem Bildschirm-Code Y gesetzt. Ge-

löscht wird es dann durch den Befehl »POKE X, 32«, denn die Zahl 32 entspricht dem Leerzeichen (Blank). Durch PRINT PEEK(X) kann man abfragen, welches Zeichen sich an der Stelle X befindet. Der Bildschirmcode ist bei allen drei Systemen identisch. Jedoch liegt der Bildschirmspeicher an unterschiedlichen Stellen: Bei den CBMs zwischen 32768 und 33767, beim C 64 zwischen 1024 und 2023. Wenn man ein CBM-Programm auf den C 64 überträgt, muß man also von allen Bildschirmadressen 31744 subtrahieren, umgekehrt 31744 addieren.

Jedoch kommt da noch ein kleines Problem hinzu: Der C 64 kann 16 Farben darstellen. Dazu besitzt er einen Farbenspeicher, welcher ganz analog zum Bildschirmspeicher zu behandeln ist. Er beginnt bei 55296 und endet bei 56295; die Differenz zwischen Bildschirmadresse und Farbadresse beträgt 54272. Wenn man also ein CBM-Programm für den C 64 um-

gedrückte Taste	CBM 30XX	CBM 40XX	Commodore 64
—	75	95	57
1	26	49	56
2	18	50	59
3	25	51	8
4	42	52	11
5	34	53	16
6	41	54	19
7	58	55	24
8	50	56	27
9	57	57	32
0	10	48	35
+	17	43	40
—	9	45	43
£	--	--	48
HOME	7	19	51
INS/DEL	?	?	0
Q	64	81	62
W	56	87	9
E	63	69	14
R	55	82	17
T	62	84	22
Y	54	89	25
U	23	86	30
I	53	73	33
O	60	79	38
p	52	80	41
@	15	64	46
*	33	42	49
!	59	94	54
A	48	65	10
S	40	83	13
D	47	68	18
F	39	70	21
G	46	71	26
H	38	72	29
J	45	74	34
K	37	75	37
L	44	76	42
=	1	61	53
RETURN	27	13	1
Z	32	90	12
X	24	88	23
C	31	67	20
V	23	86	31
B	30	66	28
N	22	78	39
M	29	77	36
,	70	44	47
.	2	46	44
/	49	47	55
CRSI	?	?	7
CRS—	?	?	2
SPACE	6	32	60

Tabelle 1. Vergleich der Tastaturcodes

schreibt, muß man hinter jedem POKE-Befehl für den Bildschirm einen entsprechenden für die Farbe anhängen, also zum Beispiel hinter POKE 1024,64 anfügen: POKE 1024 + 54272,X ,wobei X eine Zahl zwischen 0 und 15 ist, welche dann die Farbe des Zeichens bestimmt. Dieser Befehl kann weggelassen werden, wenn ein Zeichen gelöscht wird, also im Befehl der Bildschirmcode 32 (= Leerzeichen) eingesetzt wird. Die Farbcodes haben folgende Bedeutung:

0:schwarz	8:orange
1:weiß	9:braun
2:rot	10:hellrot
3:türkis	11:grau 1
4:violett	12:grau2
5:grün	13:hellgrün
6:blau	14:hellblau
7:gelb	15:grau 3

Umgekehrt müssen die POKE-Befehle für den Farbenspeicher bei einer Übertragung auf die CBMs gestrichen werden. Sprites und hochauflösende Grafik können nicht auf die CBM-Serie übertragen werden.

Tastatur und Joystick

Viele Programme fragen durch PEEK die gerade gedrückte Taste ab. Dazu existiert in der Zero-Page eine Speicherstelle, in welcher sich die Matrixkoordinate der gerade gedrückten Taste befindet. Bei den CBMs ist dies die 151, beim C 64 die 203. Die Wirkungsweise kann man sich leicht durch folgendes kleines Programm verdeutlichen:

```
10 PRINT PEEK (203)
20 GOTO 10
```

(Bei den CBMs muß die 203 in Zeile 10 durch 151 ersetzt werden). Nach dem Starten durch RUN erscheint eine Reihe von Zahlen, beim C 64 die 64, bei den CBMs die 255. Drückt man nun irgendeine Taste, so erscheint eine gerade Zahl, je nachdem, welche Taste gerade gedrückt ist. In Tabelle 1 ist ein Vergleich der Tastaturcodes dargestellt.

Es gilt, einige besondere Adressen zu beachten:
CBM: In 152 steht 1 (anstatt 0), wenn die Taste SHIFT gedrückt ist.

C 64: Adresse 653:

- 0: wenn nichts gedrückt
- 1: SHIFT gedrückt
- 2: Commodore-Taste gedrückt
- 4: CTRL gedrückt

Dazu noch ein Tip für eigene Programme: Meistens benötigt man für ein Spiel eine Unterscheidung zwischen verschiedenen Tastendrücken (zum Beispiel bis zu neun verschiedene Tasten für die Richtungssteuerung eines Raumschiffes). Am

```
1 REM KANONE
5 ZA = 1984:ZE = 2023:OP = 2004
10 DIM A%(255)
20 A%(56) = -1:A%(59) = 1
:
1000 REM ABFRAGE — ZEICHEN SETZEN, SUBROUTINE
1010 NP = OP + A%(PEEK(203)):REM NEUE POSITION
1020 IF NP > ZE OR NP < ZA THEN NP = OP:REM ZEILENBEGRENZUNG
1030 POKE OP, 32:REM ALTE POSITION LÖSCHEN
1040 POKE NP, 30:REM NEUE POSITION SETZEN
1050 POKE NP+54272,1:REM FARBE SETZEN
1060 OP = NP
1070 RETURN
```

Listing 1. Ein kleines Beispielprogramm für den C 64.

schnellsten funktioniert die Abfrage, wenn man ein Feld mit 256 (Ganzzahl-)Variablen definiert (zum Beispiel A%(255)), und die Richtung als Wert des entsprechenden Indexes darin speichert.

Beispiel (hier für C 64): Wir wollen für ein Spielprogramm eine Kanone (1 haben, die man am unteren Bildschirmrand mit Taste "1" nach links und mit Taste "2" nach rechts steuern kann. In unserem Beispiel würde das Programm entsprechend Listing 1 aussehen.

A% (255) ist das Variablenfeld. Der Tastaturcode von "1" ist beim C 64 die 56, von "2" die 59. Die Tastaturadresse ist 203.

Variablen:

OP die alte Position der Kanone

NP die neue Position der Kanone

ZE Bildschirmadresse am Ende der Zeile, in der sich die Kanone bewegt

ZA Bildschirmadresse am Zeilenanfang

CBM-Besitzer können nach Studium des Artikels an diesem Beispielprogramm ihr neugelerntes Wissen ausprobieren. Die Lösung steht am Ende des Artikels (aber nicht vorher nachschauen!).

Diese Methode ist auch bei der Verwendung eines Joysticks nützlich, denn dieser muß ähnlich wie die Tastatur mit PEEK abgefragt werden. Seine Adresse ist 56321 (Port 1) beziehungsweise 56320 (Port 2), wobei Port 1 weniger geeignet ist, da der Inhalt seines Speichers auch von der Tastatur beeinflusst wird. Die Codierung ist der Tabelle 2 zu entnehmen.

Bewegung:	Port 1:	Port 2:
Feuer-Taste	239	111
nach oben	254	126
nach unten	253	125
nach rechts	247	119
nach links	251	123
nach links oben	250	122
nach rechts unten	245	117
nach rechts oben	246	118
nach links unten	249	121
keine Taste gedrückt	255	127

Tabelle 2. Die Codierung der Joystick-Ports

Die CBMs besitzen keinen Anschluß für Joysticks, es gibt jedoch diverse Konstruktionen zum Anschluß an den User-Port. Daher empfiehlt es sich, beim Übertragen von C 64-Programmen auf die CBMs auf den Joystick zu verzichten und diese Programmteile durch Tastaturabfragen zu ersetzen.

PRINT-Befehle

Die ASCII-Codes sind bei allen drei Systemen größtenteils identisch, sie unterscheiden sich praktisch nur in den Farbsteuerzeichen und der Umschaltung von Grafik auf Groß/Kleinschreibung. Die Umschaltung auf Groß/Kleinschreibung erfolgt bei den CBMs durch den Befehl "POKE 59468,14" und zurück auf die Blockgrafik durch "POKE 59468,12". C 64-Besitzer benutzen hier den Befehl "PRINT CHR\$(14)" (Kleinschreibung) beziehungsweise den Befehl "PRINT CHR\$(142)" (Grafik). Beim Übertragen von Programmen auf den C 64 kann man nach Belieben Farbsteuerzeichen einfügen, umgekehrt sind diese zu löschen.

Der Ton

Wichtig für das Übertragen von Programmen mit Ton ist vor allem die Tonhöhe und ihre Adresse. Beim C 64 stehen drei Tongeneratoren zur Verfügung. Ihre Tonhöhe wird dabei durch je zwei Adressen bestimmt, wobei die erste Adresse (das Low-

Byte) die Tonhöhe nur sehr wenig, die zweite Adresse (das High-Byte) die Tonhöhe hingegen wesentlich stärker (nämlich genau 255 mal so stark wie das Low-Byte) beeinflusst. In der Tabelle 3 sind die wichtigsten Tonadressen des Commodore 64 zusammengefaßt.

(Adresse = 54272 + Register)

Bei den CBMs steht jedoch nur ein Tongenerator zur Verfügung, auch können dort weder Wellenform, noch Lautstärke, noch Hüllkurve programmiert werden. Daher ist es bei der Übertragung von C 64-Programmen auf die CBMs bei komple-

xen Klängen und Geräuschen meist am einfachsten, den Ton selbst neu zu gestalten. Wenn man den Ton beim Übertragen auf den C 64 direkt übernehmen will, ist das Umschreiben nicht schwierig. Bei den CBMs gibt es drei Tonadressen, welche wie üblich durch PEEK und POKE angesprochen werden: 59464, 59466 und 59467. Der Tongenerator wird dort eingeschaltet durch POKE 59466,16:POKE 59467,20: POKE 59464,0. Nun kann man die Tonhöhe einfach festsetzen, indem man eine Zahl zwischen 0 und 255 in die Adresse 59646 schreibt (durch POKE 59464,X, wobei X eine Zahl zwischen 0 und 255 ist). Dabei ist 255 ein relativ tiefer Ton und 0 ein sehr hoher Ton (Ultraschall, daher unhörbar und mit »Ton ausgeschaltet« zu vergleichen). Die Tonhöhe kann man für den C 64 als High-Byte übernehmen, jedoch bedeutet dort 0 einen sehr tiefen Ton und 255 einen sehr hohen, also genau umgekehrt. Daher muß man wie folgt vorgehen. Wenn im CBM-Programm steht: POKE TA, TH, wobei TA die Adresse der Tonhöhe 59464 und TH die Tonhöhe ist, muß es nun im C 64-

Stimme	REGISTER			INHALT	
	2	2	3		
0	7	14	Frequenz Low-Byte (0-255)		
1	8	15	Frequenz High-Byte (0-255)		
2	9	16	Tastverhältnis Low-Byte (0-255) bei Rechteck		
3	10	17	Tastverhältnis High-Byte (0-15)		
4	11	18	Wellenform: Rauschen = 129 Rechteck = 65 Sägezahn = 33 Dreieck = 17		
5	12	19	Anschlag 0 (hart) — 15*16 (weich) + Abschwollen 0 (hart) — 15 (weich)		
6	13	20	Halten 0 (stumm)— 15*16 (laut) + Ausklingen 0 (schnell)— 15 (langsam)		
24	24	24	Lautstärke: 0 (stumm) — 15 (volle Lautstärke)		

Tabelle 3. Die wichtigsten Tonadressen von C 64.

Doppelfloppy 4040 (die neuen 4er Befehle stehen in Klammern)	Monofloppy 1541
<u>Diskette neu formatieren:</u> OPEN 1,8,15,"NO:Name,XX" (HEADER"Name",IXX I = Laufwerksnummer)	OPEN 1,8,15,"N:Name,XX"
<u>Inhaltsverzeichnis lesen:</u> LOAD"\$1",8 bzw. LOAD"\$",8 (DIRECTORY)	LOAD"\$",8
<u>Beliebiges Programm laden:</u> LOAD"Name",8 bzw. LOAD"Name1",8 (DLOAD"Name" bzw. DLOAD"Name"D1)	LOAD"Name",8
<u>Beliebiges Programm sichern:</u> SAVE"Name",8 bzw. SAVE"Name1",8 (DSAVE"Name" bzw. DSAVE"Name",D1)	SAVE"Name",8
<u>Datei löschen:</u> OPEN 1,8,15,"S0:Name" bzw. OPEN 1,8,15,"S1:Name" (SCRATCH"Name",D0 bzw. SCRATCH"Name",D1)	OPEN 1,8,15,"S:Name"
<u>Datei umbenennen:</u> OPEN 15,8,15: PRINT15,"R0:Neu=0:Alt" bzw. OPEN 15,8,15:PRINT15,"R1:Neu=1:Alt" (RENAME D1, "Alt"TO"Neu" bzw. RENAME D0,"Alt"TO"Neu")	OPEN 15,8,15: PRINT15,"R:Neu=Alt"

Tabelle 4. Vergleich der Befehle für die Laufwerke 4040 und 1541

CBM		C 64		Bedeutung
Hex	Dezimal	Hex	Dezimal	
0028-0029	40-41	002B-002C	43-44	Zeiger auf Basic-Anfang
002A-002B	42-43	002D-002E	45-46	Zeiger auf Variablen Anfang
002C-002D	44-45	s.u.		Zeiger auf Ende der Variablen-tabelle
s.o.		002F-0030	47-48	Zeiger auf Beginn der Felder
002E-002F	46-47	0031-0032	49-50	Zeiger auf Ende der Felder
0030-0031	48-49	0033-0034	51-52	Zeiger auf Beginn der Strings (rückwärts)
0032-0033	50-51	0037-0038	55-56	Zeiger auf Ende der Strings (= Speichergrenze) laufende Zeilennummer
0036-0037	54-55	0039-003A	57-58	vorhergehende Zeilennummer nächster Befehl (für CONT)
0038-0039	56-57	003B-003C	59-60	aktuelle DATA-Zeile
003A-0038	58-59	003D-003E	61-62	aktuelles DATA-Element (Adresse)
003C-003D	60-61	003F-0040	63-64	aktueller Variablenname
003E-003F	62-63	0041-0042	65-66	Zeiger auf aktuelle Variable
0042-0043	66-67	0045-0046	69-70	Variablenzeiger für FOR...NEXT
0044-0045	68-69	0047-0048	71-72	diese Routine holt nächstes Basic-Zeichen
0046-0047	70-71	0049-004A	73-74	Zufallszahl
0070-0087	112-135	0073-008A	115-138	interne Uhr
0088-008C	136-140	008B-008F	139-143	gedrückte Taste
008D-008F	141-143	00A0-00A2	160-162	Cursor an/aus (0=an,1=aus)
0097	151	00CB	203	Spaltenposition des Eingabecursors
00A7	167	00CC	204	Zeilenposition des Eingabecursors
00C6	198	00CA	202	freier Platz für Zeiger in Page 0
00D8	216	00C9	201	Tastaturpuffer
00FD-00FF	253-255	00FB-00FE	251-254	
026F-0278	623-632	0277-0280	631-640	

Tabelle 5. Die wichtigsten Zero-Page-Adressen

Programm heißen: POKE FH, 255 — TH. Es kommt aber dazu, daß der Tonumfang des C 64 etwas kleiner ist als der der CBMs. Daher empfiehlt es sich, den Wert für die Tonhöhe mit 2 oder 3 zu multiplizieren, wobei darauf zu achten ist, daß der Bereich von 0 bis 255 nicht überschritten wird. Der Befehl muß also lauten: POKE FH, 255 — 2 * TH.

Diskettenbefehle

Die komfortablen Diskettenbefehle des CBM 40 XX können nicht übernommen werden, jedoch kann man die überall möglichen Kommando-String-Befehle verwenden. Allerdings muß dafür für die Monofloppy 1541 (beziehungsweise 1540) die Laufwerksnummer gestrichen werden. Tabelle 4 stellt eine Vergleichsliste der beiden Laufwerke dar.

Zum Schluß noch einige wichtige Zero-Page-Adressen zum Vergleich in Tabelle 5 (aus urheberrechtlichen Gründen dürfen die vollständigen Listen leider nicht abgedruckt werden).

Lösung des Beispielprogramms »Kanone« für CBM:

```
1 REM KANONE
5 ZA = 33728:ZE = 33767:OP = 33748
10 DIM A%(255)
20 A%(26) = -1:A%(18) = 1:REM (Für CBM 30XX)
oder
20 A%(49) = 1:A%(50) = 1:REM (Für CBM 40XX)
1000 REM Abfrage und Zeichen setzen, Subroutine
1010 NP = OP + A%(PEEK(151)):REM neue Position
1020 IF NP > ZE OR NP < ZA THEN NP = OP:REM Zeilenbegrenzung
1030 POKE OP, 32:REM alte Position löschen
1040 POKE NP,30:REM neue Position setzen
1050:
1060 OP = NP
1070 RETURN
```

Und nun viel Erfolg beim Umschreiben!

(M. und J. Heinz/rg)

Tips & Tricks

MERGE für C 64 / VC 20

Hier ist eine einfache MERGE-Routine zum Verbinden zweier Basic-Programme. Sie kann sowohl für die Floppy als auch für die Datasette (auch mit Turbo-Tape) verwendet werden. Einzige Voraussetzung: Das zweite Programm muß höhere Zeilennummern haben als das erste.

Und so wird's gemacht:

1. Sie laden das erste Programm. Dann geben Sie im Direktmodus ein: PRINT PEEK(43), PEEK(44) Diese beide Zahlen schreiben Sie sich auf.

2. Sie geben ein: POKE 43, (PEEK(45) + 256 * PEEK(46) - 2) AND 255 (Return) POKE 44, (PEEK(45) + 256 * PEEK(46) - 2) / 256 (Return)

Laden Sie nun das zweite Programm. Danach geben Sie ein: POKE 43, (erste Zahl) : POKE 44, (zweite Zahl) (Return) Nun befinden sich beide Programme hintereinander im Speicher.

(Michael Keukert)

Funktionstastenbelegung

Simons Basic bietet ja bekanntlich die Möglichkeit, die Funktionstasten mit beliebigen Zeichenketten zu belegen. Um nun

die Funktionstasten nicht jedesmal nach dem Einschalten neu belegen zu müssen, wäre es sinnvoll, die Belegung auf Floppy abspeichern zu können.

Die Funktionstastenbelegung ist bei Simons Basic in dem von Basic nicht erreichbaren Speicherbereich \$C64D bis \$C74B (50765 bis 51019 dezimal) abgelegt. Mit dem folgenden kleinen Programm wird dieser Speicherbereich als Maschinenprogramm abgespeichert:

```
10 INPUT »Filename«; X$
20 OPEN 5, 8, 5, X$ + »,PW«
30 A = 50765 : E = 51019
40 H = INT(A / 256) : L = A AND 255
50 PRINT # 5, CHR$(L); CHR$(H);
60 FOR I = A TO E
70 PRINT # 5, CHR$(PEEK(I));
80 NEXT I : CLOSE 5
```

Mit LOAD "Name",8,1 kann die Funktionstastenbelegung nun jederzeit geladen werden, ohne ein eventuell vorhandenes Basicprogramm zu zerstören. (Uwe Schwarz)

Hilfe für »Turbo Tape«

Das Programm »Turbo Tape« ist ja ein Segen für alle diejenigen, die sich keine Floppy leisten können oder wollen. Es gibt allerdings einige Maschinenprogramme, die nach dem Gebrauch von »Turbo Tape« abstürzen.

Abhilfe: Nach dem Laden das Programm LISTen und den SYS-Befehl zu Anfang notieren. Nun SYS 64738 und danach den notierten SYS-Befehl eingeben — und schon läuft das Programm. (Andreas Klofanda)

Basicprogramme retten

Ein durch NEW oder durch einen RESET gelöscht Basicprogramm kann durch Eingabe folgender Zeilen im Direktmodus wieder zurückgeholt werden:

```
POKE 46, PEEK(56) - 1 : POKE 45, PEEK(55) + 247 : CLR (Return)
POKE PEEK(44) * 256 + PEEK(43) + 1, PEEK(44) (Return)
63999 (Return)
FOR I = PEEK(44) * 256 + PEEK(43) TO PEEK(46) * 256 + PEEK(45) : IF PEEK(I) OR PEEK(I + 1) OR PEEK(I + 2) THEN NEXT (Return)
POKE 45, (I + 3) AND 255 : POKE 46, (I + 3) / 256 : CLR (Return)
```

Diese »Rettungsmaßnahme« funktioniert sowohl beim VC 20 wie auch beim C 64. (Ralf Berle)

Cursor steuern

Das Betriebssystem des C 64 enthält eine Routine, die man benutzen kann, um den Cursor an eine beliebige Stelle zu setzen. Geben Sie doch mal folgendes ein:

```
POKE 214, (Zeile) : POKE 211, (Spalte) : SYS 58640 : PRINT "TEXT"
```

(Michael Keukert)

Und noch ein Tip

Der FORMULAR TOO COMPLEX - Error ist sehr unangenehm, da sich das Programm danach oft nicht mehr listen läßt. Nach Eingabe von POKE 24,0 verhält sich der Computer aber wieder normal. (Roger Limberg)