

Hier haben wir zwei Leckerbissen für Besitzer des C 128: Ein Programm, das die Benutzung aller C 128-Tasten im C 64-Modus gestattet und ein Programm, das den Commodore 128 im C 64-Modus um 35 Prozent beschleunigt.

Will man einen Commodore 128 im C 64-Modus betreiben, muß man auf vieles verzichten, was im C 128-Modus bereitsteht. Die Zehnertastatur läßt sich nicht benutzen und der FAST-Befehl, der den Commodore 128 um 100 Prozent beschleunigt, ist nur auf dem 80-Zeichen-Monitor anwendbar, der vom C 64 nicht unterstützt wird.

Zwei Programme schaffen hier Abhilfe. Das erste namens »key 128« (Listing 1) gestattet die Benutzung sämtlicher C 128-Tasten im C 64-Modus, bis auf die Taste ASCII/

DIN, deren Zustand in Speicherzelle 1 steht und die Taste 40/80 DISPLAY, die im C 64-Modus nicht abgefragt werden kann.

»key128« wird als Basic-Lader und nicht als MSE-Listing abgedruckt, was den Vorteil hat, daß es sich an jede beliebige Speicherstelle legen läßt. Nach dem Programm-lauf werden SYS-Befehle für Ein- und Ausschalten der C 128 Tastatur angegeben. Die Tasten geben, mit

»GET A\$« abgefragt, folgende ASCII-Codes zurück:

Taste:	Code:	Shift:
ESC:	chr\$(27),	chr\$(27)
TAB:	chr\$(9),	chr\$(24)
ALT:	chr\$(14),	chr\$(142)
HELP:	chr\$(8),	chr\$(9)
LINE FEED:	chr\$(10),	chr\$(10)
NO SCROLL:	chr\$(3),	chr\$(3)

Mit Hilfe des zweiten Programms »fast64« (Listing 2) läuft ein Commodore 128 im C 64-Modus um 35 Pro-

```

10 REM KEY128 V19-10-85
20 REM VON ANDREAS ZELLER, HANAU
30 :
40 REM AUFGABE: 128ER-TASTEN IM
50 REM 64ER-MODUS BEDIENEN.
60 :
70 REM DURCH ÄENDERN DER ZEILE 170
80 REM KANN DAS PROGRAMM IN BELIEBIGE
90 REM BEREICHE GELEGT WERDEN.
100 :
110 REM WENN DAS PROGRAMM ZUSAMMEN MIT
120 REM "FAST64" LAUFEN SOLL, MUSS
130 REM 395 POKE CODE+74,44
140 REM EINGEFUEGT WERDEN.
150 :
160 :
170 CODE=49152:REM STARTADRESSE
180 :
190 POKE 53265,11:POKE 53296,1
200 :
210 DIM P(12):FOR J=0 TO 12:READ P:P(J)=CO
DE+P:NEXT
220 :
230 DATA 0,74,110,171,195,213,232,233,234,
235,236,224,225
240 :
250 FOR I=0 TO 284:READ X$
260 :
270 REM UMWANDLUNG ADRESSEN
280 A$=RIGHT$(X$,1):B$=LEFT$(X$,1)
290 IF A$="+" THEN P=P(ASC(B$)-65):X=P/256:
GOTO 360
300 IF A$="-" THEN P=P(ASC(B$)-65):X=(P/256
-INT(P/256))*256:GOTO 360
310 :
320 REM UMWANDLUNG HEX => DEC
330 :
340 X=ASC(A$)+(A$>"e")*55+(A$<"")*48
350 X=X+(ASC(B$)+(B$>"e")*55+(B$<"")*48)*
16
360 POKE CODE+I,X:NEXT
370 :
380 POKE 53296,0:POKE 53265,27
390 :
400 PRINT">DAS PROGRAMM BENUTZT DEN BEREIC
H"
410 PRINT" VON"CODE"- "CODE+284". "
420 PRINT
<051>
<234>
<006>
<196>
<171>
<036>
<203>
<193>
<111>
<076>
<247>
<156>
<247>
<036>
<126>
<136>
<026>
<156>
<106>
<176>
<171>
<196>
<250>
<216>
<138>
<238>
<236>
<124>
<072>
<220>
<032>
<036>
<052>
<048>
<107>
<255>
<092>
<005>
<112>
<244>
<012>
<012>
430 PRINT">TABELLE NORMAL AB:"CODE+237";
440 PRINT" {6SPACE}MIT SHIFT AB:"CODE+261".
450 PRINT
460 PRINT">TASTEN AN MIT: SYS"CODE";"
470 PRINT" {7SPACE}AUS MIT: SYS"CODE+80". "
480 :
490 :
500 END
510 :
520 :
530 :
1000 DATA 08,48,78,AD,14,03,C9,C-
1010 DATA AD,15,03,E9,C+,F0,0C,AD
1020 DATA 14,03,8D,G-,G+,AD,15,03
1030 DATA 8D,H-,H+,AD,02,03,C9,B-
1040 DATA AD,03,03,E9,B+,F0,0C,AD
1050 DATA 02,03,8D,I-,I+,AD,03,03
1060 DATA 8D,J-,J+,A9,C-,8D,14,03
1070 DATA A9,C+,8D,15,03,A9,B-,8D
1080 DATA 02,03,A9,B+,8D,03,03,68
1090 DATA 28,60,20,A-,A+,6C,I-,I+
1100 DATA 08,48,78,AD,G-,G+,8D,14
1110 DATA 03,AD,H-,H+,8D,15,03,AD
1120 DATA I-,I+,8D,02,03,AD,J-,J+
1130 DATA 8D,03,03,68,28,60,A9,40
1140 DATA 85,CB,29,00,8D,2F,D0,09
1150 DATA FF,8D,00,DC,CD,01,DC,F0
1160 DATA 42,A0,59,A9,FB,8D,2F,D0
1170 DATA 48,AD,01,DC,CD,01,DC,D0
1180 DATA F8,A2,08,0A,80,02,84,CB
1190 DATA 88,C0,41,90,07,CA,D0,F3
1200 DATA 68,4A,10,E1,68,AD,8D,02
1210 DATA 0A,C9,08,90,02,A9,06,AA
1220 DATA BD,L-,L+,85,F5,8D,M-,M+
1230 DATA 85,F6,20,F-,F+,20,E0,EA
1240 DATA 20,F-,F+,A9,FF,8D,2F,D0
1250 DATA 29,7F,8D,00,DC,A4,CB,8C
1260 DATA K-,K+,6C,G-,G+,A5,C5,AE
1270 DATA K-,K+,8D,K-,K+,86,C5,60
1280 DATA D-,D+,E-,E+,E-,E+,D-,D+
1290 DATA 31,EA,83,A4,40
1300 DATA 08,38,35,09,32,34,37,31
1310 DATA 18,2B,2D,0A,0D,36,39,33
1320 DATA 0E,30,2E,91,11,9D,1D,03
1330 DATA 09,38,35,18,32,34,37,31
1340 DATA 18,2B,2D,0A,0D,36,39,33
1350 DATA 8E,30,2E,91,11,9D,1D,03
<217>
<043>
<042>
<185>
<173>
<202>
<212>
<248>
<232>
<244>
<254>
<216>
<051>
<102>
<227>
<146>
<222>
<200>
<057>
<004>
<011>
<189>
<116>
<054>
<193>
<189>
<159>
<130>
<133>
<005>
<160>
<182>
<101>
<173>
<048>
<079>
<164>
<112>
<114>
<160>
<252>
<201>
<163>
<020>
<231>
<193>
<114>

```

© 64'er

Listing 1. Mit diesem Listing können Sie im C 64-Modus des C 128 alle Tasten abfragen (auch zum Beispiel die Zehnertastatur).

```

10 REM FAST64 V19-10-85                <209>
20 REM VON ANDREAS ZELLER, HANAU        <234>
30 :                                   <006>
40 REM AUFGABE: 128ER IM 64ER-MODUS    <144>
50 REM                                UM 35% BESCHLEUNIGEN. <062>
60 :                                   <036>
70 REM DURCH AENDERN DER ZEILE 170     <203>
80 REM KANN DAS PROGRAMM IN BELIEBIGE <193>
90 REM BEREICHE GELEGT WERDEN.         <111>
100 :                                  <076>
110 REM WENN DAS PROGRAMM ZUSAMMEN MIT <247>
120 REM "KEYS128" LAUFEN SOLL, MUSS    <250>
130 REM FOLGENDE ZEILE EINGEFUEGT WERDEN: <003>
140 REM 395 POKE CODE+110,44          <038>
150 :                                  <126>
160 :                                  <136>
170 CODE=49440:REM STARTADRESSE        <012>
180 :                                  <156>
190 POKE 53265,11:POKE 53296,1        <106>
200 :                                  <176>
210 DIM P(6):FOR J=0 TO 6:READ P:P(J)=CODE <103>
    +P:NEXT                             <196>
220 :                                  <245>
230 DATA 0,110,159,216,217,218,219   <216>
240 :                                  <141>
250 FOR I=0 TO 219:READ X$             <238>
260 :                                  <236>
270 REM UMWANDLUNG ADRESSEN            <124>
280 A$=RIGHT$(X$,1):B$=LEFT$(X$,1)    <072>
290 IF A$="+" THEN P=P(ASC(B$)-65):X=P/256: <220>
    GOTO 360                             <032>
300 IF A$="-" THEN P=P(ASC(B$)-65):X=(P/256 <036>
    -INT(P/256))*256:GOTO 360          <052>
310 :                                  <048>
320 REM UMWANDLUNG HEX => DEC         <107>
330 :                                  <255>
340 X=ASC(A$)+(A$>"@")*55+(A$<"")*48 <092>
350 X=X+(ASC(B$)+(B$>"@")*55+(B$<"")*48)* <005>
    16                                   <112>
360 POKE CODE+I,X:NEXT                <244>
370 :                                  <108>
380 POKE 53296,0:POKE 53265,27        <012>
390 :                                  <085>

400 PRINT">DAS PROGRAMM BENUTZT DEN BEREIC <244>
    H"                                   <108>
410 PRINT " VON"CODE"-"CODE+219"."    <012>
420 PRINT                                <085>
430 PRINT">SCHNELL MIT: SYS"CODE";"   <160>
440 PRINT"{2SPACE}NORMAL MIT: SYS"CODE+116 <172>
    ". "                                <208>
450 :                                  <192>
460 END                                <202>
470 :                                  <212>
480 :                                  <216>
490 :                                  <051>
1000 DATA 08,48,78,AD,14,03,C9,C-   <181>
1010 DATA AD,15,03,E9,C+,F0,0C,AD    <028>
1020 DATA 14,03,8D,D-,D+,AD,15,03   <146>
1030 DATA 8D,E-,E+,AD,02,03,C9,B-   <045>
1040 DATA AD,03,03,E9,B+,F0,0C,AD    <001>
1050 DATA 02,03,8D,F-,F+,AD,03,03   <057>
1060 DATA 8D,G-,G+,A9,C-,8D,14,03   <017>
1070 DATA A9,C+,8D,15,03,A9,B-,8D   <058>
1080 DATA 02,03,A9,B+,8D,03,03,A9   <078>
1090 DATA 00,8D,30,D0,AD,12,D0,D0   <251>
1100 DATA FB,A9,31,8D,12,D0,AD,11   <225>
1110 DATA D0,29,7F,8D,11,D0,A9,01   <007>
1120 DATA 8D,30,D0,AD,1A,D0,09,01   <104>
1130 DATA 8D,1A,D0,68,28,60,20,A-   <112>
1140 DATA A+,6C,F-,F+,08,48,78,AD   <127>
1150 DATA 1A,D0,29,FE,8D,1A,D0,A9   <202>
1160 DATA 00,8D,30,D0,AD,D-,D+,8D   <052>
1170 DATA 14,03,AD,E-,E+,8D,15,03   <047>
1180 DATA AD,F-,F+,8D,02,03,AD,G-   <042>
1190 DATA G+,8D,03,03,68,28,60,AC   <011>
1200 DATA 19,D0,30,07,AD,0D,DC,58   <053>
1210 DATA 6C,D-,D+,98,4A,90,F9,8C   <118>
1220 DATA 19,D0,AD,30,D0,49,01,8D   <251>
1230 DATA 30,D0,4A,B0,03,A9,FA,2C   <154>
1240 DATA A9,31,8D,12,D0,68,A8,BA   <221>
1250 DATA BD,04,01,D0,03,DE,05,01   <145>
1260 DATA DE,04,01,68,AA,68,28,60
1270 DATA 31,EA,83,A4

```

© 64'er

Listing 2. Mit »fast 64« läuft der C128 im C64-Modus um 35 Prozent schneller.

zent schneller. Dies geschieht mit Hilfe der Taktumschaltung von 1 MHz auf 2 MHz. Normalerweise erscheint bei der Umschaltung auf 2 MHz »Speichermüll« auf dem Bildschirm, da der Video-Chip nicht mehr auf den Speicher zugreifen kann. Deswegen wird beim Commodore 128 beim FAST-Befehl der 40-Zeichen-Videochip auch einfach abgeschaltet. Unser Programm umgeht diese Einschränkung so, daß der 2 MHz-Modus nur eingeschaltet wird, wenn der Videochip den Rahmen oben und unten aufbaut und ohnehin nicht auf den Speicher zugreift.

Während die Beschleunigung wirksam ist, können Peripheriegeräte nicht korrekt benutzt werden (das hängt mit der IRQ-Routine zusammen). Vor Laden/Speichern auf Diskette oder Cassette sowie Drucken sollte das Programm deshalb wieder abgeschaltet werden (Listing 2, Zeile 440)!

Auch dieses Programm wird als Basic-Lader veröffentlicht und läßt

sich somit in jeden beliebigen Speicherbereich legen. Beide Programme sind gegen RUN/STOP-RESTORE geschützt, das heißt, daß sie auch nach dem Drücken dieser Tastenkombination noch aktiviert sind. Sie laufen über den IRQ-Vektor. Wenn beide Programme gleichzeitig benutzt werden, sollten Sie die im Listing in den REM-Zeilen angegebenen Änderungen vornehmen. Diese Zeilen müssen Sie natürlich nicht mit abtippen. »fast 64« muß zuletzt aktiviert werden.

Wenn Sie diese Hilfen in Ihren eigenen Programmen benutzen wollen, sollte es keinerlei Anpassungsprobleme geben; es sei denn, Sie arbeiten mit Interrupt-Steuerung. In diesem Fall sollten Sie die Programme erst aktivieren, nachdem Ihr Interrupt eingerichtet ist. Dasselbe gilt natürlich auch für Spiele und kommerzielle Programme. Hier wird man in den meisten Fällen nicht umhin kommen, das jeweilige Programm mit Hilfe eines Monitors auf Veränderungen des IRQ-Vektors zu untersuchen.

Noch ein Hinweis: Beide Programme verwenden indirekte Sprungbefehle, bei denen der 6510/8510-Prozessor unter gewissen Umständen (beim Verwenden von kritischen Adressen, zum Beispiel JMP(\$20FF) oder JMP(\$21FF) etc.) Fehler macht. Der 6502-Befehl JMP(addr) funktioniert nämlich nur so lange korrekt, wie das Lo-Byte von »addr« ungleich \$FF ist. Ansonsten wird die Adresse nicht aus »addr/addr+1« geholt, sondern aus »addr/addr-\$FF«!

Beispiel: JMP(\$0103) springt zur Adresse, die in (\$0103/\$0104) steht, aber JMP(\$01FF) springt zur Adresse, die in (\$01FF/\$0100) steht! Sollte ein Programm nicht laufen, erhöhen Sie die Startadresse um 1. Die im Listing angegebenen Startadressen sind selbstverständlich geprüft.

Anmerkung: Die Beschleunigung um 35 Prozent läßt sich auch im C128-Modus programmieren. Wann erscheint das erste Listing dazu? Profis mögen uns schreiben! (Andreas Zeller/tr)