

```

220 RSCODE    =#12    ;BILDSCHIRMCODE VON "R"
225 ISCODE   =#09    ;BILDSCHIRMCODE VON "I"
230 NORMIRQ  =#EA31  ;EINSPRUNGADRESSE FUER NOR-
235 :        ;MALE INTERRUPTROUTINE
240 IRQVCT   =#314   ;ADRESSE DES INTERRUPTVECTOR
245 COLOR    =#01    ;FARBE DER INDIKATORBUCH-
250 :        ;STABEN (WEISS)
255 SCREEN   =#427   ;SPEICHERPLATZ IM BILD-
260 :        ;SCHIRMSPEICHER
265 CLRTBL   =#D827  ;SPEICHERPLATZ IN DER
270 :        ;COLORTABELLE
275 ;
280 ;
285 *= $C000
290 ;
295 :        SEI      ;INTERRUPT VERHINDERN
300 :        LDA #<START ;STARTADRESSE DER NEU-
305 :        LDX #>START ;EN ROUTINE LADEN
310 :        STA IRQVCT ;IRQ-VEKTOR AUF NEUE
315 :        STX IRQVCT+1 ;ROUTINE STELLEN
320 :        CLI      ;INTERRUPT ERLAUBEN
325 :        LDA #DELAY ;VERZOEGERUNGS-
330 :        STA COUNT ;ZAEHLER SETZEN
335 :        RTS     ;ZURUECK INS BASIC
340 ;
345 START    DEC COUNT ;VERZ. ZAEHLER-1
350 :        BEQ CHANGE ;GENUEGEND UEBERSPR.
355 :        JMP NORMIRQ ;NEIN, NORMALER IRQ
360 CHANGE   LDA #DELAY ;JA,NEUE ROUTINE
365 :        STA COUNT ;ZAEHLER HOCHSETZEN
370 :        LDA #00
375 :        STA FLAG  ;FLAGGE LOESCHEN
380 :        LDA QUOTE ;EDITOR IM QUOTEMODE
385 :        BEQ NOTQUOTE ;NEIN, UEBERSPRINGEN
390 :        LDA #FF
395 :        STA FLAG  ;FLAGGE SETZEN
400 :        LDA #QSCODE ;LADE "Q"
405 :        STA SCREEN ;BRINGE AUF BILDSCHIRM
410 :        LDA #COLOR ;SETZE FARBE INS
415 :        STA CLRTBL ;FARBAM
420 :        JMP REVERS ;NAECHSTE ABFRAGE
425 NOTQUOTE LDA #" " ;LEERZEICHEN AN STELLE
430 :        STA SCREEN ;DES "Q" SCHREIBEN
435 REVERS   LDA INVERS ;EDITOR IN REVERSMODE
440 :        BEQ NOREVERS ;NEIN, UEBERSPRINGE
445 :        LDA #RSCODE ;LADE "R", PRINTE EINE
450 :        STA SCREEN-1 ;STELLE LINKS VON "Q"
455 :        LDA #COLOR ;FARBE
460 :        STA CLRTBL-1 ;SETZEN
465 :        JMP INSERT ;NAECHSTE ABFRAGE
470 NOREVERS LDA #" " ;LEERZEICHEN AN STELLE
475 :        STA SCREEN-1 ;DES "R" SCHREIBEN
480 INSERT   LDA NINSERT ;WURDE 'INSERT'
485 :        ;GEDRUECKT
490 :        BEQ NOINSERT ;NEIN, UEBERSPRINGE
495 :        LDA #FF
500 :        STA FLAG  ;FLAGGE SETZEN
505 :        LDA #ISCODE ;"I" LINKS VON "R"
510 :        STA SCREEN-2 ;HINSCHREIBEN
515 :        LDA #COLOR ;FARBE
520 :        STA CLRTBL-2 ;SETZEN
525 :        JMP KEYTEST ;ZUR ABFRAGE VON 'F7'
530 NOINSERT LDA #" " ;BLANK AN STELLE
535 :        STA SCREEN-2 ;DES "I" SCHREIBEN
540 KEYTEST  LDA LASTKEY ;LETZTE GEDRUECKTE
545 :        CMP #F7 ;TASTE GLEICH 'F7'
550 :        BNE END   ;NEIN, ZUM ENDE
555 :        LDX #00 ;JA
560 :        STX QUOTE ;QUOTE-MODE LOESCHEN
565 :        STX INVERS ;REVERS-FLAGGE AUS
570 :        STX NINSERT ;INSERT-MODUS AUS
575 :        LDA FLAG  ;WENN FLAGGE GLEICH
580 :        BEQ END   ;NULL, DANN ZEICHEN
585 :        ;NICHT LOESCHEN
590 :        INX     ;JETZT I IM X-REG
595 :        STX NOKEYS ;TASTATURBUFFER ENT-
600 :        ;HAELT EIN ZEICHEN
605 :        LDY BSPALTE ;CURSOR EINE STELLE
610 :        DEY     ;ZURUECKBEWEGEN
615 :        LDA CHCURSOR ;CHAR. UNTER CURSOR
620 :        STA (ZEILE),Y ;AUF BILDSCHIRM
625 :        LDA #9D    ;ASCII-CODE FUER
630 :        ;CURSOR ZURUECK
635 :        STA KBBUFFER ;IN TASTATURPUFFER
640 END      JMP NORMIRQ ;ENDE, GEHE ZUR NOR-
645 :        ;MALEN IRQ-ROUTINE
READY.

```

Listing 1. Das Assemblerlisting zum Programm »Escape-Taste« (Schluß)

werden, aber ein Druck auf »F7« genügt, und das Problem ist behoben.

Das Programm wird dann mit SYS49152 aufgerufen, und ab sofort ist die Routine »Escape-Taste« in den Systeminterrupt eingebunden. Das allerdings nur solange bis RUN/STOP-RESTORE gedrückt wird, dann wird der Interruptvektor wieder direkt auf die normale Interruptroutine gelenkt und die neue Routine wird so lange übergangen, bis das Programm neu gestartet wird.

Da diese Routine für den Einsatz beim Programmieren gedacht ist, sollten dabei keine Probleme entstehen.

Als einzige Einschränkung muß gesagt werden, daß die Benutzer von Kassettengeräten bei einem Zugriff auf die Datensette die Routine mit RUN/STOP-RESTORE ausschalten sollten, da Kassettensoperationen den Interruptvektor auf spezielle Kassettensroutinen umschreiben und sich ein schon verstellter IRQ-Vektor negativ auswirkt. (Christian Spörri/aw)

## Daten komprimieren — Sparen Sie Speicherplatz und Ladezeit!

Schreiben Sie öfter Programme, bei denen Sie größere Datenmengen verwalten müssen? Mit dem »Daten-Packer« wird der Aufwand an Ladezeit und Speicherplatz auf Diskette um 25 Prozent reduziert.

Dieses Programm kürzt Strings auf  $\frac{3}{4}$  der normalen Länge. Wenn man zum Beispiel einen String mit 12 Zeichen Inhalt hat, und den »Daten-Packer« darauf losläßt, ist der String nur noch neun Zeichen lang, bei gleichem Inhalt!

Um ihn sich ansehen zu können, muß er natürlich erst wieder in eine lesbare Form gebracht, also »entpackt« werden.

Bei selbstgeschriebenen Adressenverwaltungen läßt sich das gut anwenden:

Bevor man die Adressen, die ja meist in Strings enthalten sind, auf der Diskette abspeichert, »packt« man sie mit dem Daten-Packer, und speichert sie dann. So spart man eine Menge wertvollen Diskettenplatz und Ladezeit.

Nach dem Einladen müssen die Adressen wieder »entpackt« werden, bevor man mit ihnen weiterarbeiten kann.

Das Programm (siehe Listing) wird mit dem MSE eingegeben und ist dann sofort einsatzbereit.

### Umgang mit dem Daten-Packer

Packen: SYS 49152, A\$, B\$

In A\$ muß der zu packende String stehen. Danach enthält B\$ die gepackte Version.

Entpacken: SYS 49339, C\$, D\$

In C\$ muß der gepackte String stehen. In D\$ ist dann der entpackte String zur Weiterverarbeitung bereit.

# Der C 64 lernt sprechen

**Dieses Programm ist wohl die billigste Sprachein-/ausgabe, die es gibt. Man kann sie vielseitig einsetzen, zum Beispiel bei Spielen, sprechenden Menüs oder einfach als Spielerei.**

Außer dem C 64 ist für dieses Programm eine Datensette erforderlich. Nachdem das Listing abgetippt und das Programm mit SYS 49231 gestartet wurde, ändern sich die Bildschirmfarben, und eine Einschaltmeldung erscheint. Jetzt stehen Ihnen 2045 Byte für Basic-Programme zur Verfügung. Der Grund dafür ist, daß die Aufzeichnung von Sprache äußerst speicherintensiv ist. Selbst der Bereich unter dem Basic-ROM (\$A000 bis \$BFFF) wird ausgenutzt. Die Tonwiedergabe hat zwar keine Hi-Fi-Qualität, wenn man aber laut und deutlich spricht, kann man sich gut wiedererkennen. Während der Tonein-/ausgabe wird der Bildschirm abgeschaltet, damit der Prozessor nicht ständig vom Video-Controller unterbrochen wird.

Das Arbeiten mit diesem Programm ist denkbar einfach. Es stehen fünf neue Basic-Befehle zur Verfügung. Der Speicher des C 64 wird in verschiedene Abschnitte aufgeteilt (Bild 1), in denen Wörter gespeichert werden. Hinter allen Befehlen (außer -R und -G) ist anzugeben, in welchen Bereich gespeichert werden soll.

x = 1 bis 16: Diese Abschnitte sind für ein- bis zweisilbige Wörter reserviert. Hier können zum Beispiel Zahlen oder kurze Wörter abgelegt werden.

x = 17: Dies ist ein Abschnitt für 4 Sekunden Sprechzeit.

Hier kann ein kleiner Satz abgelegt werden.

x = 255: Hier wird der gesamte Speicher beschrieben, also auch die Abschnitte 1 bis 17. Es stehen etwa 13 Sekunden Sprechzeit zur Verfügung.

Alle anderen Zahlen werden als 17 interpretiert.

## Befehle

Doch nun zu den Befehlen:

-Lx: »learn« — Nach Eingabe dieses Befehls wird der Bildschirm abgeschaltet. Die Tonsignale werden digitalisiert und im Speicher abgelegt. x entscheidet, in welchen Bereich die Sprache gespeichert wird.

-Sx: »say« — Bei diesem Befehl wird ebenfalls der Bildschirm gelöscht. Aus dem Lautsprecher des Fernsehgerätes ertönt nun Ihre eigene Stimme. Im Gegensatz zu dem Befehl »learn« benötigt dieser Befehl keinerlei Hardware. Er kann also ohne irgendwelchen Umbau benutzt werden. x entscheidet, aus welchem Bereich die Sprache entnommen wird.

-P"Filename"x: »put« — Mit diesem Befehl kann die Sprache, die sich im Speicher befindet, auf Diskette gespeichert werden. Da das Speichern von Sprache sehr aufwendig ist, belegen 13 Sekunden Sprechzeit 178 Blöcke. x gibt an, in welchem Bereich die Sprache gespeichert wird.

-G"Filename": »get« — Dieser Befehl wird zum Laden der Sprache benutzt.

-R: »rate« — Mit diesem Befehl läßt sich die Sprechgeschwindigkeit einstellen. Nach dem Befehl muß ein numerischer Aus-

Natürlich kann an Stelle von A\$, B\$, und D\$ jede andere Stringvariable stehen.

Das geht doch nicht mit rechten Dingen zu, wird der eine oder andere Leser jetzt vielleicht denken.

Nun, dieses Programm kürzt 4 Byte auf 3 Byte. Ein Byte hat bekannterweise 8 Bit, gebraucht werden aber nur 6 Bit, um die ASCII-Werte darzustellen, aus denen Datenstrings ja bestehen.

Mit diesen 6 Bit kann man alle alphanumerischen Zeichen und einige Sonderzeichen darstellen. Der Algorithmus des Programms besteht nun darin, vom ASCII-Wert des zu packenden Byte 32 abzuziehen, um eine Darstellung mit 6 Bit zu ermöglichen. Dann wird jedes vierte Byte auf die vorhergehenden 3 Byte verteilt, bei denen die letzten 2 Bit nach unserer Subtraktion nun nicht mehr gebraucht werden. »Überhang-Bytes«, also die letzten Zeichen eines Strings, die keine vollen 4 Byte mehr ergeben, werden nicht gepackt, sondern nur an den gepackten String angehängt.

Beim Entpacken läuft der Vorgang in umgekehrter Reihenfolge ab. (Joachim Matzner/gk)

```

PROGRAMM : PACKROUTOBJ      C000 C1EB
-----
C000 : 20 9C C0 20 6D C1 A0 00 0A
C008 : A2 00 B1 22 38 E9 20 0A C2
C010 : 0A 9D EE C1 E8 C8 E0 03 3B
C018 : D0 F0 B1 22 38 E9 20 8D 7F
C020 : F1 C1 AC F3 C1 18 6E F1 16
C028 : C1 6E EE C1 6E F1 C1 6E 6E
C030 : EE C1 6E F1 C1 6E EF C1 AB
C038 : 6E F1 C1 6E EF C1 6E F1 88
C040 : C1 6E F0 C1 6E F1 C1 6E 07
C048 : F0 C1 AD EE C1 99 F5 C1 A6
C050 : C8 AD EF C1 99 F5 C1 C8 05
C058 : AD F0 C1 99 F5 C1 C8 8C CA
C060 : F3 C1 AD F4 C1 18 69 04 C8
C068 : 8D F4 C1 CD F2 C1 F0 28 EB
C070 : 18 69 04 CD F2 C1 F0 02 FD
C078 : B0 06 AC F4 C1 4C 08 C0 15
C080 : AE F3 C1 AC F4 C1 B1 22 96
C088 : 38 E9 20 9D F5 C1 C8 E8 D3
C090 : CC F2 C1 D0 F1 8E F3 C1 47
C098 : 20 B8 C1 60 20 FD AE 20 FF
C0A0 : 88 B0 85 64 84 65 20 A3 AC
C0A8 : B6 A0 00 B1 47 8D F2 C1 15
C0B0 : C8 B1 47 85 22 C8 B1 47 91
C0B8 : 85 23 60 20 9C C0 20 6D 16
C0C0 : C1 A8 A2 00 B1 22 9D EE FE
C0C8 : C1 C8 E8 E0 03 D0 F5 20 12
C0D0 : 76 C1 AC F3 C1 A2 00 BD 7D
C0D8 : EE C1 99 F5 C1 E8 C8 E0 14
C0E0 : 04 D0 F4 8C F3 C1 AD F4 09
C0E8 : C1 18 69 03 8D F4 C1 CD 93
C0F0 : F2 C1 F0 52 69 03 CD F2 15
C0F8 : C1 F0 23 B0 06 AC F4 C1 2D
C100 : 4C C2 C0 AC F4 C1 AE F3 73
C108 : C1 B1 22 18 69 20 9D F5 27
C110 : C1 C8 E8 CC F2 C1 D0 F1 6D
C118 : 8E F3 C1 4C 69 C1 A2 00 C9
C120 : AC F4 C1 B1 22 9D EE C1 38
C128 : C8 E8 E0 03 D0 F5 2C EE 48
C130 : C1 30 19 70 17 2C EF C1 74
C138 : 30 12 70 10 2C F0 C1 30 41
C140 : 08 70 09 4C 03 C1 AD F3 2C
C148 : C1 4C 69 C1 20 76 C1 AC D8
C150 : F3 C1 A2 00 BD EE C1 99 5A
C158 : F5 C1 E8 C8 E0 04 D0 F4 DC
C160 : AD F3 C1 18 69 04 8D F3 4F
C168 : C1 20 B8 C1 60 A9 00 8D CF
C170 : F4 C1 8D F3 C1 60 A9 00 EC
C178 : 8D F1 C1 AA 18 2E F0 C1 FE
C180 : 2E F1 C1 2E F0 C1 2E F1 97
C188 : C1 2E EF C1 2E F1 C1 2E 6A
C190 : EF C1 2E F1 C1 2E EE C1 F6
C198 : 2E F1 C1 2E EE C1 2E F1 8F
C1A0 : C1 BD EE C1 4A 4A 18 69 5E
C1A8 : 20 9D EE C1 E8 E0 03 D0 CE
C1B0 : F0 AD F1 C1 18 69 20 8D 94
C1B8 : F1 C1 60 20 FD AE 20 8B 93
C1C0 : B0 85 49 84 4A 85 64 84 81
C1C8 : 65 20 A3 B6 AD F3 C1 20 BF
C1D0 : 75 B4 A0 00 B9 61 00 91 91
C1D8 : 49 C8 C0 03 D0 F6 A0 00 5D
C1E0 : B9 F5 C1 91 62 C8 C4 61 79
C1E8 : D0 F6 60 A5 B6 D0 25 33 ED
    
```

Das Listing zum »Datenpacker«, Eingabe mit dem MSE. Beachten Sie bitte die Eingabehinweise auf Seite 8