

**H**aben Sie schon einmal einen Ball länger als ein paar Minuten auf derselben Stelle liegen sehen? Ich noch nicht, denn der nächste Vorbeikommer kickt ihn — irgendwohin. Sie doch auch, oder? Dasselbe Phänomen können Sie überall dort beobachten, wo ein Heimcomputer steht. Eingeschaltet oder nicht — jeder, der vorbeikommt, drückt auf eine Taste. Und wenn das Ding gar reagiert — mit einem Buchstaben auf dem Bildschirm — dann bleibt selbst die Oma stehen und tippt nochmal und nochmal.

### Die Tasten wirken auf die Menschen wie das Licht auf die Motten

Wenn vom VC 20 oder Commodore 64 die Rede ist, kommt diese sehr rasch auf die Tastatur. Ihre Schreibmaschinen-Qualität und die klaren Steuer- und Funktionstasten heben sie aus der Schar der Konkurrenten heraus. Sie ist sehr benutzerfreundlich, besonders für Anfänger, während der ersten Tast(en)versuche.

Aber rasch wird offensichtlich, daß die Tasten auch ihre Geheimnisse haben, und zwar frühestens, wenn die vier Funktionstasten keine Reaktionen, zumindest keine sichtbaren, hervorrufen und spätestens bei dem Versuch, in einem Programmablauf bestimmte Tastenfunktionen ausführen zu lassen, wie zum Beispiel »Cursor Down« oder »Bildschirm löschen«.

Ich will Ihnen deshalb in diesem Aufsatz die Grundlagen der Tastaturabfrage, das Geheimnis der Funktionstasten, die CHR\$-Anhängsel von PRINT-Befehlen, die mysteriösen »negativen« Zeichen in Anführungsstrichen und gleichzeitig gedrückte Tasten erklären und mit Kochrezepten ausschmücken.

Übrigens: das hier Gesagte gilt sowohl für den VC 20 als auch für den Commodore 64. Nur die Programmbeispiele nicht, die sind auf den kleinen VC 20 zugeschnitten. Die 64er mögen mir das verzeihen. Ich scheue mich aber, Kochrezepte abzugeben, die ich nicht selbst ausprobiert habe. Und ich habe leider (noch) keinen C 64. Falls Unterschiede bei den Adressen auftreten, gebe ich den Wert für den C 64 in Klammern an.

Als allererstes möchte ich folgende Behauptung aufstellen:

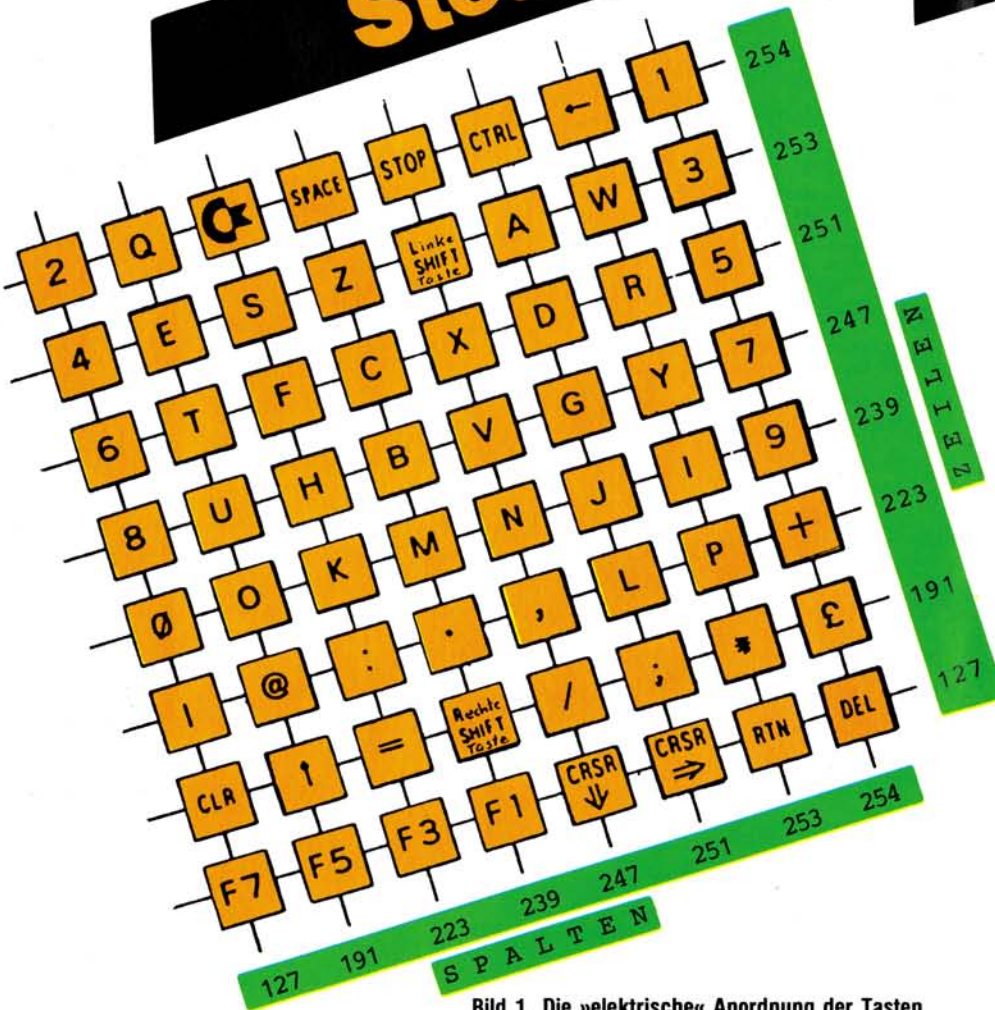


Bild 1. Die »elektrische« Anordnung der Tasten

### Dem Computer sind alle Tasten gleich

Und er behandelt sie auch alle gleich. 60 mal in jeder Sekunde unterbricht der Computer was er auch immer gerade ausführt, merkt sich, wobei und wo er gerade unterbrochen hat, und schaut nach, ob eine der Tasten gedrückt worden ist.

Keine Regel ohne Ausnahme, auch beim Computern nicht: Die RESTORE-Taste und die SHIFT-LOCK-Taste fallen völlig aus dem

Rahmen und haben mit den kommenden Erklärungen nichts zu tun. Ich werde sie deshalb auch nicht mehr erwähnen. Der Vollständigkeit halber sei hier nur kurz gesagt, daß die RESTORE-Taste den Computer bei jeglicher Arbeit unterbricht und ihn mit READY und blinkendem Cursor in den Anfangszustand zurücksetzt. Die SHIFT-LOCK-Taste ist (wie bei der Schreibmaschine) eine mechanische Arretierung der SHIFT-Taste.

Es bleiben uns immerhin 64 Tasten, die vom Computer während seiner Verschnaufpause inspiziert werden. Die Funktionstasten sind

(Teil 1)

Die gute und übersichtliche Tastatur der Computer

von Commodore bietet gerade dem Anfänger sehr leichte Einstiegsmöglichkeiten.

Wer aber tiefer in die Computerei eindringt, stößt beim VC 20 und

Commodore 64 schnell auf eine verwirrende Vielfalt von Codes und Steuerzeichen, deren Kenntnis und Beherrschung jedoch auch dem

Anfänger ein weites Anwendungsfeld eröffnen kann.

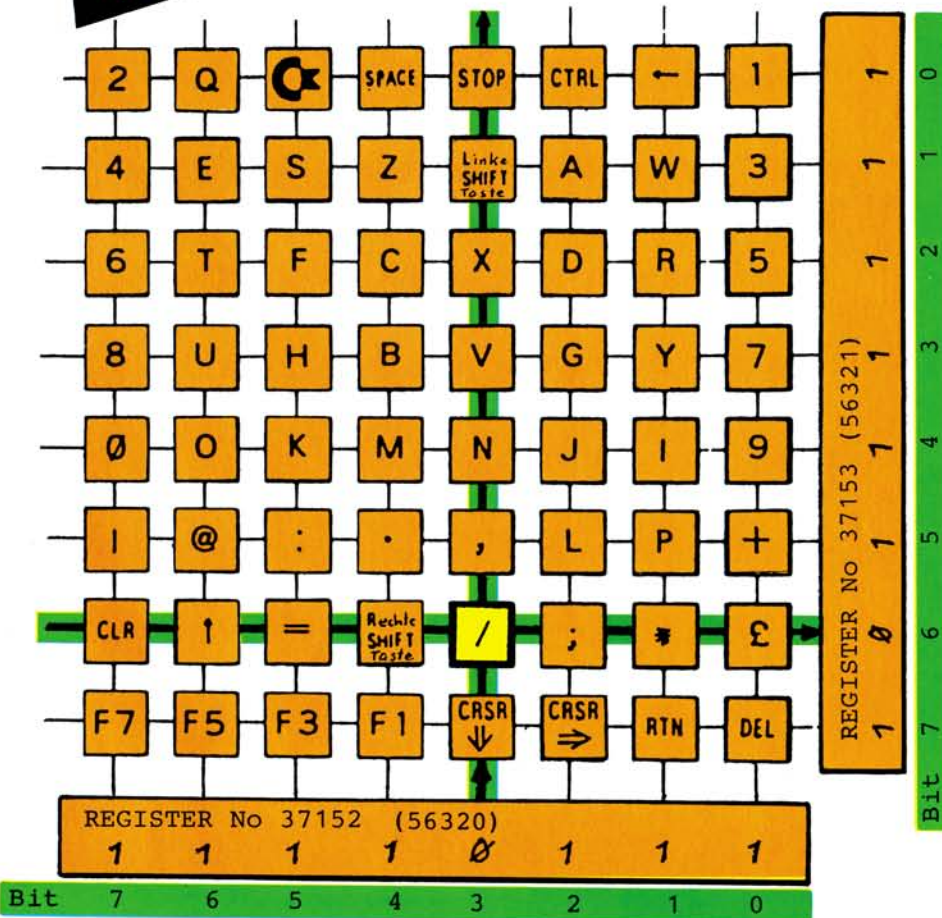


Bild 2. Die Tasten-Matrix mit ihren Registern

immer dabei! Diese Inspektion — Tastaturabfrage genannt — wollen wir uns näher anschauen.

Der Computer erhält nicht, wie es eigentlich logisch wäre, von jeder gedrückten Taste ein spezielles Code-Signal. Das wäre für einen

Homecomputer zu aufwendig und zu teuer. Das Betriebssystem des Computers veranstaltet vielmehr eine Befragung seines Tastenvolkes, nach dessen Stimmenabgabe er dann entscheidet, welche Taste nun eigentlich gedrückt worden ist.

Schauen Sie sich bitte das Bild 1 genauer an. Es zeigt Ihnen die 64

Tasten in einer 8 x 8-Matrix. Diese Anordnung entspricht der elektrischen Verbindung der Tasten. Die Abfrage selbst ist sehr einfach. Der Computer ruft alle senkrechten Spalten einzeln auf, indem er die Zahl, die an der jeweiligen Spalte in Bild 1 steht, in eine spezielle Speicherzelle hineinschiebt. Beim VC 20 ist das 37152, beim Commodore 64 die 56320.

Falls die gedrückte Taste in dieser Spalte liegt, meldet sie sich und die Zahl der waagrechten Zeile wird in die Adresse 37153 (56321) geschrieben. Wenn überhaupt keine Taste gedrückt war, erscheint in 37153 (56321) eine 255.

### Theorie und Praxis: Eine untrennbare Einheit

10 IF Sie jetzt fragen, warum die Zahlen so willkürlich und außerdem paarweise gleich sind, THEN lesen Sie weiter: GOTO nächsten Absatz.

20 REM ein bißchen Theorie würde nicht schaden: GOTO übernächsten Absatz.

Die Zahlen sind natürlich nicht willkürlich gewählt. Es sind vielmehr die Dezimalwerte von Dualzahlen, die beim Anwählen einer Spalte beziehungsweise durch Drücken einer Taste in den schon genannten Speicherzellen 37152 (56320) und 37153 (56321) entstehen. Bild 2 zeigt den Zusammenhang.

Am Rande der Matrix sehen Sie

# Neu von Sybex:

## Farbspiele mit dem COMMODORE 64

Dieses Buch zeigt Ihnen, daß Sie Ihren Commodore 64 auch für Ihr Freizeit-Vergnügen nutzen können: Wolfgang Black und Matthias Richter haben 20 herrliche Spiele für Sie zusammengestellt. Jedes Spiel wird zunächst beschrieben, es folgen ausführlich dokumentierte Programmlisten. Bildschirm-Abbildungen im Buch machen den typischen Spielverlauf für Sie anschaulich. Farbgraphiken und Synthesizer-Töne Ihres VC 64 kommen bei diesen Spielen voll zur Geltung.

Da bei den einzelnen Spielen jede Programmzeile kommentiert wird, können Sie bald eigene Spiele entwickeln. Nutzen Sie diese nicht formale Einführung in die Spielprogrammierung.



Wolfgang Black  
Matthias Richter  
**Farbspiele mit dem  
COMMODORE 64**  
ca. 200 Seiten, illust.  
Ref.-Nr. 3044 (1984)  
DM 34,-

Ihr  
**COMMODORE 64**  
spricht BASIC!  
Sprechen Sie  
seine Sprache!

Sybex-Bücher sind erhältlich bei Ihrem Fachhändler. Fragen Sie danach!

### Verlagsauslieferung:

Österreich: Fachbuch-Center  
ERB, Amerlingstr. 1, 1061 Wien  
Schweiz: Versandbuchhandlung  
Thali AG, Industriest. 2,  
6285 Hitzkirch

Direktbestellungen beim Verlag  
gegen Verrechnungsscheck  
(+DM 2,50 Versandkostenanteil)

Fordern Sie ein  
Gesamt-Buch-Verzeichnis an.



**SYBEX-VERLAG** GM  
BH

Abt. CJ 284 Postfach 30 09 61  
4000 DÜSSELDORF 30  
Tel. 0211-626441, Telex: 8588163

## Alle Codes

## C 64/VC 20

jetzt nicht die Zahlen, sondern die beiden Speicherzellen, im folgenden »Register« genannt, und ihre bitweise Verbindung mit Spalten und Zeilen der Tasten. Prinzipiell gilt, daß diejenige Spalte angewählt ist, an deren Stelle eine »Null« im Register 37152 (56320) steht. Das ergibt im Register eine Dualzahl, die entsprechend der Bit-Numerierung zu lesen ist. Eine gedrückte Taste ihrerseits erzeugt eine Null im Register 37153 (56321) an der Stelle, wo ihre Zeile angeschlossen ist. Ich habe in Bild 2 ein Beispiel (Taste mit dem Zeichen »/«) eingezeichnet, und wenn Sie die Dualzahlen kennen, werden Sie beim Vergleich mit den Zahlen in Bild 1 die Übereinstimmung erkennen.

Wie bitte? Sie sagen, daß es aber doch möglich sein müßte, weit mehr als die in Bild 1 gezeigten acht Zahlen im Register 37153 (56321) zu erzeugen, indem man mehr als eine Taste gleichzeitig drückt? Richtig gesehen, das geht in der Tat, und das wollen wir uns auch gleich einmal ansehen. Dazu brauchen wir ein Kochrezept. Ich schlage vor, das Rezept — und auch die noch folgenden — gleich auszuprobieren. Nehmen Sie diesen Artikel und verfolgen Sie die weiteren Zeilen meiner Beschreibung abwechselnd lesend und eintippend.

## Die Abfrage der Tastatur wird auf dem Bildschirm dargestellt

Als erstes wollen wir, so wie der Computer es macht, eine Spalte auswählen, zum Beispiel wie in Bild 2, die vierte von rechts. Das ergibt die Dualzahl 11110111, in dezimal die Zahl 247.

```
100 POKE 37152,247
200 PRINT PEEK(37152),
    PEEK (37153)
300 GOTO 100
```

Zeile 200 druckt uns sowohl den von uns gePOKEten (verzeihen Sie mir das schreckliche Computerdeutsch) Wert 247 als auch den in Register 37153 (56321) stehenden Wert aus, der dort erscheint, sobald wir eine Taste in der Spalte 247 drücken. Der Rücksprung in Zeile 300 erzeugt auf dem Bildschirm zwei senkrechte, durchlaufende Zahlenstreifen.

Nach RUN läuft rechts die Zahl 255 (dual = lauter ler, das heißt keine Taste gedrückt). Drücken Sie nun die »/«-Taste, und es erscheint

die 191. Die X-Taste erzeugt 251, die links angeordnete SHIFT-Taste eine 253. Die rechte SHIFT-Taste dagegen zeigt keine Reaktion — ist auch klar, denn sie liegt ja in einer anderen Spalte.

Aber es ist interessant, sich zu merken, daß bei der Tastaturabfrage die beiden SHIFT-Tasten völlig eigenständig behandelt werden, im Gegensatz zu Ihrer Funktion. Sie sehen, ich habe recht gehabt: Alle Tasten sind gleich.

Sie können auch die RUN-STOP-Taste drücken, aber bitte nur ganz, ganz kurz antippen, denn das Programm bleibt natürlich entsprechend ihrer ungeSHIFTeten Funktion sofort stehen. Wenn das Betätigen der Taste aber kurz genug war, dann steht als letzte Zahl die 254 da.

## Gleichzeitige Abfrage von zwei Tasten

Jetzt kommt das große Experiment. Nach neuem RUN drücken Sie »Cursor down« und »/« gleichzeitig — es erscheint 63. Der Blick auf Bild 2 zeigt uns für beide Tasten das Bitmuster 00111111 und das ist 63. Es geht also. Probieren Sie ruhig alle Kombinationen aus. Wenn Sie Zeit und die Gelenkigkeit eines Konzertpianisten haben, können Sie alle Zahlen bis 254 erzeugen, auch die Null. Dazu sind alle acht Tasten gleichzeitig zu drücken.

Erwähnenswert ist, daß jetzt das Programm nicht stehen bleibt, obwohl die STOP-Taste gedrückt ist. Aber die SHIFT-Taste ist auch gedrückt, und das ist laut Handbuch die LOAD-RUN-Funktion, die jetzt nicht zum Tragen kommt.

Sie wollen sicher endlich auch einmal die Funktionstasten zur Geltung bringen. Dazu müssen wir aber eine andere Spalte auswählen. Ändern Sie bitte die Zeile 100 ab:  
100 POKE 37152, 239

In der Spalte 239 liegt die f-1-Taste. Nach RUN passiert aber was Komisches: Trotz des POKENS von 239 läuft links wieder die 247, genauso wie vorhin. Und außerdem reagieren die Tasten der Spalte 239 nicht, nur die der Spalte 247.

Was passiert da? Es liegt daran, daß nur beim Eintippen eines Programms die Tastaturabfrage genau so funktioniert wie beschrieben. Wenn aber ein Programm abläuft, dann ist der Computer nur an der STOP-Taste interessiert und schiebt deshalb immer wieder eine 247 in das Register 37152 (56320).

Eine Ausnahme gibt es auch hier: INPUT- und GET-Befehle fragen auch die anderen Tasten ab.

Diese erzwungene Vorfahrt der Spalte 247 können wir durch unsere Zeile 100 nicht ändern, denn wir sind ja nach RUN in einem Programmablauf. Die einzige Möglichkeit, andere Spalten POKEN zu können, ergibt sich für uns durch Programmierung in Maschinensprache. Aber darauf will ich jetzt nicht eingehen, sondern erst am Schluß des Aufsatzes.

Wir haben also bei unserem Versuch, das Ergebnis der Befragung schon bei der Stimmenabgabe — sozusagen im Wahllokal — auszukundschaften, Pech gehabt. Das macht aber nichts, denn irgendwann wird ja ein Wahlergebnis offiziell bekanntgegeben. Bei der Tastaturabfrage ist das auch so. Vorher aber wollen wir wenigstens aus dem Teilergebnis, das wir ausspioniert haben, Kapital schlagen und die mehrfache Tastenabfrage der Spalte 247 an einem klaren Beispiel demonstrieren.

## Programmsteuerung mit zwei unabhängigen Tasten

Das Programm soll auf dem Bildschirm in den Spalten 6 und 15 zwei senkrechte Bänder mit Sternen darstellen, deren Farbe mit der »X«-Taste und der »/«-Taste unabhängig voneinander, auch gleichzeitig, verändert werden kann.

```
100 PRINT CHR$(147)
```

Diese Zeile löscht den Bildschirm. Diese Methode (ASCII-Code) kennen Sie sicher aus dem Programmier-Handbuch. Ich werde sie aber noch genau behandeln.

```
110 BS=PEEK(648)*256
```

```
120 FS=4*(PEEK(36866)
```

```
AND 128) + 37888
```

Zeile 110 und 120 machen das Programm unabhängig von Speichererweiterungen. BS und FS sind Variable für die Anfangsadressen des Bildschirm- beziehungsweise des Farbspeichers.

```
130 F=2:G=2
```

Das ist die Anfangsfarbe »rot« für beide Sternreihen.

```
200 FOR Z=0 TO 22
```

Mit Z werden die 23 Zeilen abgezählt.

```
300 POKE BS+5+Z*22,42
```

```
310 POKE FS+5+Z*22,F
```

Keine Angst, ich mute Ihnen keine höhere Mathematik zu. Ab Zeile 300 wird der Bildschirm-Code (42)

für den Stern (auch das behandle ich später noch) in Spalte 6 (BS+5) für jede Zeile (Z\*22) untereinander gePOKEd, dazu die Farbe F in den gleichen Platz des Farbspeichers. Dasselbe gilt in Zeile 400 und 410 für die Spalte 15.

```
400 POKE BS+15+Z*22,42
```

```
410 POKE FS+15+Z*22,G
```

```
500 AA = PEEK(37153)
```

```
510 IF AA = 191 OR AA = 187 THEN F=F+1
```

```
520 IF AA = 251 OR AA = 187 THEN G=G+1
```

Ab Zeile 500 werden die »X«-Taste (191) und die »/«-Taste (251) abgefragt. Wenn eine davon gedrückt ist, wird die Farbzahl F beziehungsweise G um 1 erhöht. Die OR-Funktion, mit welcher der Wert 187 abgefragt wird, erlaubt ein gleichzeitiges Drücken beider Tasten, und es werden sowohl F als auch G erhöht.

Um die Farben zwischen 2 (rot) und 7 (gelb) zu begrenzen, verwenden wir:

```
600 IF F=8 THEN F=2
```

```
610 IF G=8 THEN G=2
```

Zum Schluß wird der Zeilenzähler Z weitersetzt. Wenn er 22 (das heißt die 23. Zeile) erreicht hat, springt das Programm an den Anfang zurück.

```
700 NEXT Z
```

```
800 GOTO 200
```

Nun geben Sie RUN ein. Wenn Sie eine der Tasten zu lange, das heißt länger als einen Z-Zyklus, drücken, springt die Farbe weiter. Um das etwas zu erleichtern, können Sie noch eine Verzögerung einbauen:

```
220 FOR T=1 TO 50: NEXT T
```

Ich gebe ja zu, das ist kein gewaltiges Programm. Aber es zeigt Ihnen wenigstens, wie auch in Basic eine mehrfache Tastenabfrage möglich ist.

```
100 PRINT CHR$(147)
```

```
110 BS=PEEK(648)*256
```

```
120 FS=4*(PEEK(36866)
```

```
AND 128) + 37888
```

```
130 F=2:G=2
```

```
200 FOR Z=0 TO 22
```

```
220 FOR T=1 TO 50: NEXT T
```

```
300 POKE BS+5+Z*22,42
```

```
310 POKE FS+5+Z*22,F
```

```
400 POKE BS+15+Z*22,42
```

```
410 POKE FS+15+Z*22,G
```

```
500 AA = PEEK(37153)
```

```
510 IF AA = 191 OR AA = 187 THEN F=F+1
```

```
520 IF AA = 251 OR AA = 187 THEN G=G+1
```

```
600 IF F=8 THEN F=2
```

```
610 IF G=8 THEN G=2
```

```
700 NEXT Z
```

```
800 GOTO 200
```

Weiter geht's, und zwar mit dem schon erwähnten Bekanntgeben des Wahlergebnisses. In anderen Worten: Wie wertet der Computer die Tastenabfrage über die Register 37152 (56320) und 37153 (56321) weiter aus?

Sobald der Computer merkt, daß eine Taste gedrückt ist, nimmt er die beiden Zahlen, die in den Registern 37152 (56320) und 37153 (56321) stehen, und wandelt sie in eine Code-Zahl um, die er in der Speicherzelle 203 ablegt. Die Code-Zahl steht auch in der Speicherzelle 197. Mit dem Grund für diese Verdopplung muß ich mich aber erst noch beschäftigen.

Wir bleiben bei Adresse 203. Wie bei der Abfrage der Tastatur-Matrix wollen wir uns den Inhalt dieser Speicherzelle ansehen. Löschen Sie bitte das alte Programm und geben Sie ein:

```
100 PRINT PEEK(203)
```

```
200 GOTO 100
```

Nach RUN sehen wir wieder das laufende Zahlenband, jetzt aber mit 64. Das ist die Code-Zahl für »keine Taste gedrückt«. Die »/«-Taste ergibt jetzt 30, die X-Taste 26 und so weiter. Und endlich ist es soweit! Die Funktionstasten reagieren und geben ihre Code-Zahl preis.

## Die Funktionstasten reagieren doch!

Probieren Sie alle Tasten durch und schreiben Sie die Code-Zahlen auf die Tasten von Bild 1 oder 2. Jetzt sehen Sie auch die Gesetzmäßigkeit, nach der der Computer die Spalten- und Zeilenzahl der Register ummodellt. Schreiben Sie sich am besten eine komplette Liste der Code-Zahlen für die weitere Verwendung. Die RUN-STOP-Taste läßt sich hier leichter als beim ersten Mal überlisten, natürlich nur mit gleichzeitigem SHIFT.

A propos »gleichzeitig«! Wiederholen Sie das Experiment von vorher. Hier erleiden wir unseren zweiten Fehlschlag: Mehrfach Tasten geben keinen Sinn, denn die Umcodierung verwehrt es uns. Wie gut, daß wir die Methode der Matrix-Abfrage haben, auch wenn sie in voller Eleganz nur in Maschinensprache möglich ist. Doch wie gesagt, davon später.

Zurück zu den einzelnen Tasten. In der Liste der Code-Zahlen fehlen die Tasten RESTORE, SHIFT, C=, CTRL. Sie haben das nicht bemerkt? Dann haben Sie auch noch

nicht die von mir vorgeschlagene Liste gemacht!

Um auch diese Zahlen auf den Bildschirm zu bringen, ergänzen Sie bitte die Zeile 100 in:

```
100 PRINT PEEK (203), PEEK
    (653)
200 GOTO 100
```

Jetzt sehen Sie zwei Zahlenreihen laufen. Zu der bekannten Reihe ist auf der zweiten Hälfte des Bildschirms (bedingt durch das Komma zwischen den PEEKs) eine 0-Reihe gekommen. Drücken Sie jetzt die SHIFT-Taste: Rechts läuft eine 1 — die Code-Zahl dieser Taste.

### Die Steuertasten haben ihre Code-Zahl in der Speicherzelle 653

Die C=-Taste erzeugt eine 2, die CTRL-Taste eine 4 (und die natürlich ganz langsam). Drücken Sie mal SHIFT und C= gleichzeitig. Siehe da, bei der Speicherzelle 653 und ihren Steuertasten gibt das einen Sinn. Die Tabelle aller Kombinationen sieht so aus:

keine Taste	0
SHIFT	1
C=	2
SHIFT u. C=	3
CTRL	4
SHIFT u. CTRL	5
CTRL u. C=	6
alle 3 Tasten	7

Während wir das ausprobieren haben, läuft links unbeirrt die 64. Und in der Tat, durch das Aufspalten und Abspeichern der Code-Zahlen in zwei getrennte Speicherzellen können wir beide Tastenarten, nämlich Zeichentasten und Steuertasten, unabhängig voneinander und/oder gleichzeitig abfragen:

Das nutzen wir zum Beispiel bei den Funktionstasten aus. Jede von ihnen hat ihre eigene Code-Zahl in Adresse 203. Aber das gäbe uns nur vier Möglichkeiten, entsprechend der Aufschrift die ungeraden f-Zahlen. Um auch f-2 bis f-8 zu erhalten, kombinieren wir die vier Zahlen in 203 einfach mit SHIFT-Taste gedrückt oder nicht (1 oder 0 in 653).

Aber Sie sehen schon, wie willkürlich das ist, denn wir könnten f-6 auch definieren als Kombination von der dritten Funktionstaste und CTRL (also 55 und 4).

Überhaupt, wir sind gar nicht auf acht Funktionstasten beschränkt, wie es uns durch den Ausdruck eingeredet wird. Die vier Funk-

tionstasten ergeben zusammen mit den acht Kombinationen der Steuertasten 32 mögliche Funktionen. Natürlich gilt das für alle Tasten der Tastatur. Der Computer selbst nutzt allerdings nur wenige Kombinationen aus. SHIFT und C= (3) schaltet alle Buchstaben in Groß-

### Eine »Heimorgel« ganz besonderer Art

Kleinschrift um, die CTRL-Taste mit den Zahlentasten erzeugt die Vordergrund-Farben. Sie haben also viel Raum für phantasievolle Abfragekombinationen. Die Abfrage selbst und ihre Verwendung in einem Programm will ich abschließend mit den Funktionstasten demonstrieren.

Wie man mit dem VC 20 Töne erzeugt, wissen Sie. In Zeile 10 definieren wir das Sopranregister und geben ihm den Namen Z, in Zeile 20 schalten wir die Lautstärke ein. Lautstärke des Fernsehers nicht vergessen!

```
10 Z=36876
20 POKE 36878,10
```

Ab Zeile 40 bis 110 wird jede einzelne Kombination der Code-Zahlen von Funktions- und Steuertasten abgefragt. Sobald sie auftritt, wird ein entsprechender Ton der Tonleiter gePOKEd.

```
30 A= PEEK(203): B=
   PEEK(653)
40 IF A=39 AND B=0 THEN
   POKE Z,131
50 IF A=47 AND B=0 THEN
   POKE Z,145
60 IF A=55 AND B=0 THEN
   POKE Z,157
70 IF A=63 AND B=0 THEN
   POKE Z,162
80 IF A=39 AND B=1 THEN
   POKE Z,172
90 IF A=47 AND B=2 THEN
   POKE Z,181
100 IF A=55 AND B=4 THEN
   POKE Z,189
110 IF A=63 AND B=7 THEN
   POKE Z,193
```

Ein Rücksprung in die Zeile 30 verleiht dem Ton auch die Dauer.

```
120 GOTO 30
```

Damit ein Ton aber nur solange klingt, wie eine Tastenkombination gedrückt ist, schieben wir noch Zei-

le 35 ein, die das Sopranregister auf 0 (Stille) setzt, sobald »keine Taste« gedrückt ist.

```
35 IF A=64 AND B=0 THEN
   POKE Z,0
```

Jetzt will ich Ihnen natürlich noch die von mir gewählten Tastenkombinationen verraten, damit Sie gezielt »Alle meine Entchen« spielen können. Es gilt der Reihe nach:

Zeile 40	: f-1
Zeile 50	: f-3
Zeile 60	: f-5
Zeile 70	: f-7
Zeile 80	: f-1 u. SHIFT
Zeile 90	: f-3 u. C=
Zeile 100	: f-5 u. CTRL
Zeile 110	: f-7 u. CTRL u. C= u. SHIFT

Ich gebe zu, daß diese Tastenauswahl nicht gerade eine bequeme Klaviatur ergibt.

Ihrem Ehrgeiz ist es überlassen, eine Orgel zu programmieren, die zwar immer noch einstimmig ist, aber eine »normale« Klaviatur hat und auch den vollen Tonumfang des VC 20 ausnutzt. Stellen Sie einfach die Code-Zahlen der Zeichentasten so zusammen, daß eine Tastenreihe die »weißen« Tasten, die darüberliegende Reihe die »schwarzen« Tasten darstellt.

### Tastenabfrage und kein Ende: was es sonst noch alles gibt

Mit den Steuertasten können Sie die Oktaven umschalten, mit den Funktionstasten verschiedene Lautstärken. Die Zahlen für die Abfrage entnehmen Sie Ihrer Liste (jetzt wird's aber höchste Zeit, sie zu schreiben).

Statt Töne zu POKEN, können Sie natürlich mit dieser Abfragetechnik der Funktionstasten (und auch der anderen Tasten) alles mögliche per Programm steuern: Raumschiffe abschießen, Textseiten weiter-schalten, den Hund rauslassen oder Toast rösten.

Es gibt noch andere Methoden der Tastenabfrage, doch diese will ich Ihnen das nächste Mal erklären.

(Dr. Helmut Hauck)

Literatur

- (1) VIC Revealed von Nick Hampshire, Computabits Ltd., 1981
- (2) M. Bassman, S. Lederman in COMPUTE!'s first Book of VIC. Compute! Books Publication, 1982
- (3) A. Grant in VIC COMPUTING, Dezember 1982
- (4) A. Dripke, VC 20 Spiele-Buch 1, Computer Life Verlag, 1983