man neue Zeichensätze entwerfen kann. Allerdings sollte das Programm einen Zusatz enthalten, damit man diese Daten leicht mit anderen Programmen verbinden kann.

Zuletzt noch zwei Anregungen. Das Register 7 des VIC bestimmt die Vertikal-Synchron-Position des Controllers. Den Originalwert können Sie mit PRINT PEEK%(7) abfragen (Wert = 32). Falls Ihr Bildschirm etwas nach unten verschoben ist, können Sie dies einfach mit POKE% 7,33 korrigieren. Bei kleineren Werten rutscht der Bildschirm entsprechend nach unten.

Versuchen Sie auch einmal POKE% 9,8 (Originalwert = 7).

(Heino Velder/dm)

```
10 REM ********************************
20 REM *                              *
30 REM * PEEK & POKE ZUM VIDEO-CHIP   *
40 REM *                              *
70 REM ********************************
80 :
85 REM PROGRAMM-AUFRUF: SYS 4864
90 :
100 READ D$  : AD=DEC(D$)     : ZL=1001
110 FOR I=1 TO 10 : READ D$   : D=DEC(D$)
120 SU=SU+D  : POKE AD,D      : AD=AD+1
130 NEXT     : READ P$        : PR=DEC(P$)
140 D=PR-32768 : IF D=>0 THEN PR=D
150 IF PR<>SU THEN PRINT "FEHLER IN",ZL: STOP
160 ZL=ZL+1: SU=0: IF D<0 THEN 110 : ELSE END
170 :
180 :
190 :
1000 DATA  1300
1001 DATA A9,1F,8D,04,03,A9,13,8D,05,03,  02AD
1002 DATA A9,95,8D,08,03,A9,13,8D,09,03,  032B
1003 DATA A9,4D,8D,0A,03,A9,13,8D,0B,03,  02E7
1004 DATA 60,20,0D,43,A2,FF,E8,BD,00,02,  0418
1005 DATA F0,22,C9,20,F0,F6,C9,30,90,04,  056E
1006 DATA C9,3A,90,16,A2,FF,E8,BD,00,02,  04F1
1007 DATA D0,FA,BD,00,02,9D,01,02,CA,10,  0403
1008 DATA F7,A9,3A,8D,00,02,60,20,29,14,  0326
1009 DATA C9,C2,F0,03,4C,DA,78,85,FD,20,  05BE
1010 DATA 0E,14,D0,F6,B0,02,85,FD,C8,20,  0504
1011 DATA 34,14,A5,17,48,A5,16,48,20,D7,  0346
1012 DATA 77,20,15,88,A5,FE,48,A5,FC,48,  0508
1013 DATA A5,FD,85,FE,A9,00,85,FC,20,BE,  062D
1014 DATA 13,A8,68,85,FC,68,85,FE,68,85,  057C
1015 DATA 16,68,85,17,4C,D4,84,68,AA,20,  03F0
1016 DATA 29,14,C9,97,F0,03,4C,A2,4A,85,  044D
1017 DATA FE,85,FC,20,0E,14,D0,F4,B0,02,  0537
1018 DATA 85,FE,8A,48,C8,20,34,14,20,D7,  047C
1019 DATA 77,20,15,88,20,5C,79,20,F4,87,  03C4
1020 DATA A9,00,AC,00,FF,8D,00,FF,24,FE,  0502
1021 DATA 10,0F,A5,67,A6,17,D0,06,A6,16,  037A
1022 DATA E0,20,90,1D,4C,28,7D,A2,12,A5,  03F7
1023 DATA 17,20,CC,CD,E8,A5,16,20,CC,CD,  052C
1024 DATA A2,1F,24,FC,10,0B,A5,67,20,CA,  03F2
1025 DATA CD,A2,12,24,FC,30,07,20,DA,CD,  049F
1026 DATA 8C,00,FF,60,20,CC,CD,8C,00,FF,  052F
1027 DATA A0,FF,C6,3E,20,34,14,4C,93,13,  03FD
1028 DATA C8,20,2B,14,C9,20,F0,F8,C9,25,  04E6
1029 DATA F0,05,C9,23,D0,01,18,60,E6,3D,  044D
1030 DATA D0,02,E6,3E,A0,00,2C,A0,01,8D,  03F0
1031 DATA 01,FF,B1,3D,8D,03,FF,60,18,98,  048D
1032 DATA 65,3D,85,3D,90,02,E6,3E,60,00,  837A
```

Listing 3. Eine Befehlserweiterung, die den Video-Chip unterstützt

Tips & Tricks für Einsteiger

Wie lese ich innerhalb eines Basic-Programms das Diskettenінhaltsverzeichnis ein? Wie berechne ich mit dem C 64 Primzahlen? Wie rechne ich mit hexadezimalen Zahlen? Auf diese und einige weitere Fragen sollen Sie hier Antwort bekommen.

Häufig bekommen wir von unseren Lesern Anfragen, wie man von einem Basic-Programm aus das Inhaltsverzeichnis einer Diskette einlesen kann. Die wirklich einfachste Lösung soll hier einmal vorgestellt werden. Sie kann problemlos in eigene Basic-Programme eingebaut und mit »GOSUB 1000« aufgerufen werden:

```
1000 OPEN 1,8,0,"$" : GET#1,A$,A$
1010 GET#1,A$,A$ : IF ST=64 THEN CLOSE 1 : RETURN
1020 GET#1,A$,B$ : PRINT ASC(A$+CHR$(0))+256*ASC(B$+
CHR$(0));
1030 GET#1,A$ : PRINT A$; : IF A$< >" " THEN 1030
1040 PRINT : GOTO 1010
```

Wem es gelingt, eine bessere Version zu entwerfen, soll uns sofort schreiben! (tr)

Ein paar kleine Effekte

1. Diese Routine gibt einen vorher festgelegten Namen in dreieckiger Form auf dem Bildschirm aus.

```
0 A=53270:A$="PETER":L=LEN(A$)
1 FOR R=1 TO L : PRINT TAB(20)MID$(A$,R,L):POKE A,R:NEXT R
```

Der Ausdruck (Bildschirm) sieht folgendermaßen aus:

```
PETER
ETER
TER
ER
R
```

Der Name ist also horizontal und vertikal lesbar.

2. Mit dieser Routine wird folgendes erreicht:

a) Diskname und ID werden extern eingegeben, das heißt beide können fest im Programm gespeichert werden.
b) Diskname und ID erscheinen beim Listen des Directory untereinander!

```
10  INPUT "NAME";N$
20  INPUT "ID"; I$
30  OPEN 1,8,15,"N:□"+N$+","+I$:CLOSE 1
```

Das Zeichen »□« erhält man folgendermaßen:
Geben Sie ein: OPEN 1,8,15,"N:"
Zurückfahren auf das zweite Anführungszeichen (Cursor nach links) und <CTRL+9, SHIFT+M, CTRL+0> drücken.
3. Dieser Trick erlaubt das Anhängen von Buchstaben an Directory-Namen.

```
OPEN 1,8,15,"R:Prgname{shift-space} Buchstaben = Prgname"
```

Prgname = Programmname

Natürlich funktioniert das auch beim Speichern:

```
SAVE "Programmname {shift+space} Buchstaben",8,1
```

Beim Laden braucht nur der Programmname angegeben zu werden.

(Stefan Wichmann/tr)

Input-Zeichen selbst wählen

Wenn man bei INPUT ein anderes Zeichen als das »?« haben möchte, muß man vor dem INPUT-Befehl folgende POKEs eingeben:

```
POKE 631,20:POKE 632,20:POKE 633,ASC("gewünschtes Zei-
chen"): POKE 634,32:POKE 198,4
```

(Axel Stämmler/tr)

Schnelles Primzahlenprogramm in Basic

Das Programm (Listing 1) arbeitet nach dem Prinzip des Eratosthenes. Hierbei werden die geraden Zahlen gar nicht erst gestrichen, sondern durch einen STEP-Befehl übersprungen. Nun werden alle Vielfachen der Zahlen gestrichen. Alle Zahlen, die übrigbleiben, sind Primzahlen.

Die Primzahl 2 wird nur deshalb ohne Rechnung ausgegeben (Zeile 30), da es sich um die einzige gerade Primzahl handelt. Man müßte sonst in der Ausgabeschleife mit einer Verzögerung rechnen.

Übrigens ist in der IF-Abfrage in Zeile 30 der Punkt hinter dem Gleichheitszeichen kein Fehler. Der Computer rechnet hier mit dem Punkt schneller als mit einer Null. Das Pro-

gramm benötigt für die Primzahlen bis 1000 weniger als 9 Sekunden. Da mit Feldvariablen gearbeitet wird, können nun Primzahlen bis 19300 ausgerechnet werden.

(Marcus Werner/tr)

```
10 INPUT"{CLR}PRIMZAHLEN BIS";H:Z=3:T=INT(
   SQR(H)+1):W=(T-1)/2:G=T*T:DIM P%(G)      <042>
20 FOR X=1 TO W:FOR Y=Z*Z TO G STEP Z*2:P%
   (Y)=1:NEXT:Z=Z+2:NEXT                   <131>
30 PRINT 2;:FOR X=3 TO H STEP 2:IF P%(X)=.
   THEN PRINT X;                           <049>
40 NEXT                                    <050>

64'er
```

Listing 1. Die schnellste Primzahlenberechnung in Basic

Die Multifunktions-Taste

Bei der Auswahl aus einem Menü störte mich immer die umständliche Sucherei nach der richtigen Taste. Ich habe mir deshalb eine kleine Routine ausgedacht, die einer Taste mehrere Funktionen zuweist und diese Taste zur »Auswahltaste« macht. Je nachdem, ob die »Auswahltaste« (hier: < ↑ >) einmal oder kurz hintereinander zwei-, drei-, viermal und so weiter gedrückt wird, werden die verschiedenen Menüpunkte angewählt. Bei bis zu sechs Wahlmöglichkeiten ist ein durchaus komfortables Arbeiten möglich.

Programmbeschreibung:

Im Prinzip besteht das Programm (Listing 2) aus zwei hintereinandergeschalteten GET-Befehlen (Zeilen 40 und 60). Mittels einer entsprechend dimensionierten FOR(K)-NEXT-Schleife verweilt das Programm beim zweiten GET aber nur sehr kurz. (Dauer einstellbar in Zeile 50; die Zahl 35 hat sich als ideal erwiesen.)

Je nachdem wieviel Funktionen zugewiesen wurden (Variable x in Zeile 6), so oft wird das zweite GET mit seiner »Verweilschleife« mittels einer weiteren FOR(A)-NEXT-Schleife durchlaufen — das aber nur, wenn jeweils rechtzeitig die »Auswahltaste« (festgelegt in Zeile 5) gedrückt wird. Ansonsten dringt das Programm nach Zeile 90 vor. Dort verzweigt es sich, je nachdem, wie oft die FOR(A)-NEXT-Schleife durchlaufen wurde (= wie oft die »Auswahltaste« gedrückt wurde).

Durch die Zeilen 70 und 30 ist das Programm gegen Fehlbedienung geschützt.

(Harald Poxrucker/tr)

Tip zum MSE

In der Adresse 3586 steht die Gerätenummer der Datasette. Wenn Sie zum Beispiel zwei Diskettenlaufwerke mit den Geräteadressen 8 und 9 haben, lohnt sich ein POKE 3586,9. Dadurch wird beim Speichern und Laden auf »TAPE« das Laufwerk 9 angesprochen. (tr)

Vereinfachte Joystick-Abfrage

Mit »SYS 49152, Richtung, Programmzeile« wird die Joystick-Stellung abgefragt. Wenn »Stellung« mit »Richtung« übereinstimmt, verzweigt der Befehl zur angegebenen Zeilen-Nummer. Für die Richtung gilt:

```
 1 = oben
 2 = unten
 4 = links
 8 = rechts
16 = Feuer
 5 = links oben
 6 = links unten
 9 = rechts oben
10 = rechts unten
```

Nachfolgend der Basic-Lader:

```
10 FOR A = 0 TO 41: READ B: POKE 49152+A,B: NEXT
20 DATA 32,115,0,32,138,173,32,247,183,173,0,220,37,
   20,234,208,3,76,30,192
30 DATA 32,115,0,32,138,173,32,247,183,96,32,115,0,32,
   138,173,32,247,183,76
40 DATA 163,168
```

(Ralf Stumm/tr)

SMON auf Tastendruck

Wenn man die <RUN/STOP + RESTORE>-Tasten betätigt, so führt der Computer einen Interrupt (NMI) aus und springt zu einem Maschinenprogramm, dessen Startadresse in den Speicherzellen 792 und 793 gespeichert wurde.

Anwendung:

Man kann die Startadresse von einem eigenen Maschinenprogramm in 792 und 793 speichern und damit das Programm mit Drücken der ⟨RESTORE⟩-Taste starten. In der Speicherzelle 792 muß das Low-Byte und in 793 das High-Byte der Startadresse stehen.

Beispiel:

`POKE 792,0:POKE 793,192` startet SMON C000 (das Programm muß vorher geladen werden).

`POKE 792,226:POKE 793,252` Es wird ein Reset ausgeführt.

(Jörg Ch. Ewert/tr)

Das Hexadezimal-System

1. Hexadezimalzahl in Dezimalzahl umformen

```
1 z=0:fors=1to4:a=asc(mid$(h$,s,1)):a=a-48+(a>64)*7:z=z+a*
16↑(4-s):next
```

In der Stringvariablen h$ muß zunächst die vierstellige (!) Hexadezimalzahl abgelegt werden, also zum Beispiel C000, ABCD, 0001 etc. Die Dezimalzahl befindet sich dann in der Variablen z.

2. Dezimalzahl in Hexadezimalzahl umformen

```
1 h$=" ":fors=1to4:m=16↑(4-s):a=int(z/m):z=z-
m*a:h$=h$+chr$(a+48-(a>9)*7):next
```

Hier muß zunächst eine Dezimalzahl in der Variablen z abgelegt werden. Die Hexadezimalzahl liegt danach in h$.

(Knut Smoczyk/tr)

```
 5 T=ASC("↑"):REM HIER AUSWAHLTASTE EINSETZ
   EN                                      <221>
 6 X=6:REM ANZAHL DER FUNKTIONEN           <004>
10 PRINT"{CLR}DEMO-PROGRAMM FUER 'MULITFUN
   KTIONSTASTE'":PRINT:PRINT                <126>
20 FOR Z=1 TO X:PRINT Z"MAL <"CHR$(T)"> DR
   UECKEN = FUNKTION "Z:PRINT:NEXT Z:GOTO
   40                                       <164>
30 FOR K1=1 TO 50:FOR K2=640 TO 631 STEP-1
   :POKE K2,0:NEXT K2:NEXT K1               <250>
40 GET T$:IF T$<>CHR$(T)THEN 40             <049>
50 FOR A=1 TO X:FOR K=1 TO 35               <244>
60 GET T$:IF T$=CHR$(T)THEN NEXT A          <183>
70 IF A>X THEN 30                           <086>
80 NEXT K                                   <180>
90 ON A GOTO 1000,2000,3000,4000,5000,6000 <188>
100 :                                       <076>
1000 PRINT"{CLR}FUNKTION 1":END             <163>
2000 PRINT"{CLR}FUNKTION 2":END             <179>
3000 PRINT"{CLR}FUNKTION 3":END             <195>
4000 PRINT"{CLR}FUNKTION 4":END             <211>
5000 PRINT"{CLR}FUNKTION 5":END             <227>
6000 PRINT"{CLR}FUNKTION 6":END             <243>

64'er
```

Listing 2. Auswahlmenü mit einer einzigen Taste