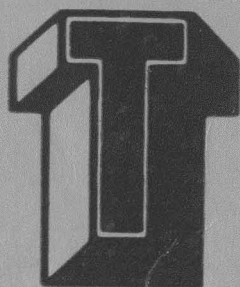


# Kleinstrechner

---

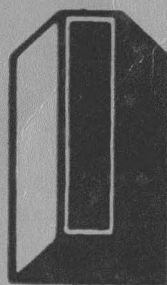
Tendenzen  
und Theorien



Spracheingabe

---

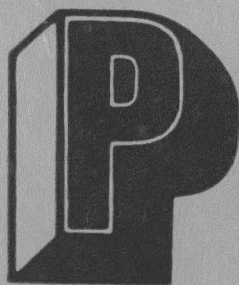
Informationen  
und Ideen



Mikrorechner

---

Programme  
und Projekte



Vokabeln lernen

---

Spaß  
und Spiel



S

---



---

# Kleinstrechner-TIPS

**Heft 5**

Mit 28 Bildern

Herausgegeben von

Prof. Dr.-Ing. Hans Kreul

Doz. Dr.-Ing. Wilhelm Leupold

Doz. Dr. sc. techn. Thomas Horn



**VEB Fachbuchverlag Leipzig**

---

---

## Inhalt

*Girlich*: Iterationen und der Feigenbaum 4

*Hübner*: Spracheingabe für Mikrocomputer 11

*Schönfelder*: Homecomputer – ein neues Gebiet für den Amateur (Teil 3) 21

*Fischer*: Primfaktorenzerlegung 36

*Schönfelder*: Computerspiele – mehr als eine Spielerei 41

*Girlich/Girlich*: Vokabeln lernen – leicht gemacht 56

Rechentechnische Begriffe für den Laien erklärt 10, 20, 35, 43 und 64

ISBN 3-343-00129-5

---



Wie bei den ersten Heften der Broschürenreihe »Kleinstrechner-TIPS«, bestimmte auch bei diesem Heft das Ziel der Herausgeber und des Verlages, möglichst vielen Interessenten der Mikrorechentechnik durch eine breit gefächerte Thematik, die vielfältigen Probleme der Anwendung von Mikrorechnern und vor allem des Kleincomputers in einer dem technisch-interessierten Laien verständlichen Form darzulegen, wiederum den Inhalt des vorliegenden Heftes 5. Wir sind besonders darüber erfreut, daß die ersten erschienenen Hefte auf ein großes Echo bei unseren Lesern gestoßen sind, die uns auch eine Reihe von Beiträgen angeboten haben. Die ersten Leserbeiträge konnten bereits in dieses Heft aufgenommen werden. Allen unseren Lesern sei an dieser Stelle für die zahlreichen Zuschriften und Veröffentlichungsangebote herzlich gedankt.

Im ersten Artikel dieses Heftes beschäftigt sich GIRLICH mit mathematischen Experimenten auf dem Kleincomputer. Dabei wird vor allem das in der numerischen Mathematik häufig angewendete Iterationsverfahren näher analysiert.

HÜBNER, der bereits in Heft 3 über Anwendung und Erweiterungsmöglichkeiten des Polycomputers 880 schrieb, gibt eine Anleitung zur Erweiterung eines Mikrorechners durch ein einfaches Spracherkennungssystem. Die

Ausführungen können leicht auf einen anderen Rechnertyp mit dem Mikroprozessor U 880 übertragen werden. Die für die Eingabe einer begrenzten Zahl von Worten vorgeschlagene Art der Sprachanalyse kann als Grundlage für ein weiterführendes eigenes Experimentieren dienen.

Die in Heft 1 begonnene Artikelserie von SCHÖNFELDER zum Thema Kleincomputer wird in diesem Heft mit der Beschreibung eines Minibetriebssystems und der Erweiterung der Konfiguration durch ein Kassettengerät abgeschlossen. Mit diesem Minibetriebssystem kann auch das Spielprogramm »Turm von Hanoi« lauffähig gemacht werden, das vom gleichen Autor den Interessenten an Computerspielen angeboten wird.

Insbesondere bei Mikrorechner-Arbeitsgemeinschaften und in Schulen dürfte der Artikel von FISCHER zur Primfaktorenbestimmung auf starkes Interesse stoßen.

Die Möglichkeit des Einsatzes eines Mikrorechners für das Vokabeltraining wird von INGE und UWE GIRLICH gezeigt. Die wichtigsten Informationen über das BASIC-Programm, mit deren Hilfe für einen bereits etwas geübten Anwender die eigene Programmentwicklung möglich sein dürfte, werden von den Autoren gegeben.

*Thomas Horn*

# Iterationen und der Feigenbaum

(Mathematische Experimente mit dem Kleincomputer)



In Heft 1 der Kleinstrechner-TIPS wird gezeigt, wie mit Hilfe eines Taschenrechners quadratische Gleichungen iterativ gelöst werden können (vgl. [3]). Wir werden zunächst eine spezielle kubische Gleichung in analoger Weise lösen. Dazu wird eine Zahlenfolge rekursiv erzeugt, die mit wachsender Gliederzahl die gesuchte Lösung immer besser annähert. Solch ein Iterationsverfahren braucht allerdings nicht in jedem Falle zu einer Lösung zu führen. Um sich Informationen über die Eigenschaften von Iterationen zu verschaffen, bedient man sich seit jeher gewisser Testrechnungen. Der Kleincomputer eröffnet nun die Möglichkeit, derartige Experimente im großen Stil bei kleinem Aufwand durchzuführen. Auch diese »mathematischen« Experimente führen mitunter zu ganz überraschenden Ergebnissen, die unerwartete Strukturen offenbaren, wie z. B. den »Feigenbaum«, auf dessen Bedeutung für die moderne Physik und die Technik hier nicht eingegangen werden kann (vgl. [2]).

## Kubikwurzelziehen

Wir betrachten die einfache kubische Gleichung

$$G(x) = x^3 - 7 = 0, \quad (1)$$

deren reelle Lösung  $x_0 = \sqrt[3]{7}$  näherungs-

weise berechnet werden soll. Dazu verwenden wir die Rekursion

$$a_{n+1} = \frac{2}{3} \left( a_n + \frac{7}{2a_n^2} \right) \quad (2)$$

mit dem Anfangswert  $a_0 = 0,5$  und erhalten – schon mit dem Taschenrechner –

$a_1 = 9,666666\ 6$	$a_6 = 1,967905\ 9$
$a_2 = 6,469414\ 7$	$a_7 = 1,914452\ 6$
$a_3 = 4,368693\ 4$	$a_8 = 1,912932\ 4$
$a_4 = 3,034719\ 2$	$a_9 = 1,912931\ 2$
$a_5 = 2,276507\ 1$	$a_{10} = 1,912931\ 2$

Nach neun Iterationen ist die Kubikwurzel auf 8 Stellen genau ermittelt. Die erforderliche Anzahl der Iterationen hängt natürlich vom Anfangswert ab. Wenn wir im folgenden diesen Wert (0,5) nicht verändern wollen, so sollten wir nach einer günstigeren Rekursionsbeziehung fragen. Nun ist  $1/a_{10} \approx 0,52$ , so daß es nahe liegt, nach einer Nähe-

rung für  $1/x_0 = 1/\sqrt[3]{7}$  zu suchen. Dafür eignet sich gerade – wie wir gleich sehen, doch später erst begründen werden – die Rekursion

$$a_{n+1} = \frac{4}{3} \left( a_n - \frac{7}{4} a_n^4 \right). \quad (3)$$

Mit unserem Anfangswert  $a_0 = 0,5$  bekommen wir

$a_1 = 0,520833\ 33$
$a_2 = 0,522743\ 82$
$a_3 = 0,522757\ 96$
$a_4 = 0,522757\ 96.$

Bereits nach drei Iterationen ist  $\frac{1}{\sqrt[3]{7}}$  auf 8 Stellen genau berechnet. Hieraus folgt  $\frac{1}{a_4} = 1,9129312$ , die gesuchte Kubikwurzel.

### Newton'sches Iterationsverfahren

Die Rekursion (2) ergibt sich unmittelbar aus der NEWTONSchen Formel (vgl. [1] 7.1.2.3.)

$$a_{n+1} = a_n - \frac{G(a_n)}{G'(a_n)}. \quad (4)$$

Dabei gilt nach (1) für die Ableitung:  $G'(x) = 3x^2$ . Wir werden versuchen, für einen beliebigen Parameterwert  $y$  die Gleichung

$$G(x) = x^3 - y = 0$$

zu lösen, und zwar mittels

$$a_{n+1} = a_n - \frac{a_n^3 - y}{3a_n^2} = \frac{2}{3} \left( a_n + \frac{y}{2a_n^2} \right). \quad (5)$$

Das gelingt im Intervall  $(0,10)$  mit dem Anfangswert  $x_0 = 0,5$  bei einer Lösungsgenauigkeit von acht Stellen mit weniger als zehn Iterationen. Für  $n \geq 10$  und  $0 \leq y \leq 10$  erhalten wir

$$a_n = g(y) = \sqrt[3]{y}.$$

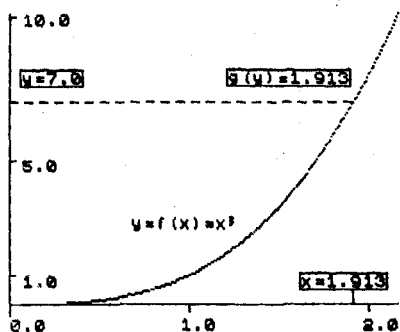


Bild 1. Die Umkehrfunktion der Kubikwurzel

Diesen Sachverhalt stellen wir in Bild 1 dar, indem wir jedem Punkt auf der vertikalen  $y$ -Achse (z. B. ist dort  $y = 7$  hervorgehoben) einen Punkt  $(g(y), y)$  auf der kubischen Parabel  $y = f(x) = x^3$  zuordnen. Bekanntlich ist  $f$  die Umkehrfunktion von  $g$ .

Die Rekursion (3) läßt sich ebenfalls über die NEWTONSche Formel (4) gewinnen, wobei  $G(x) = \frac{1}{x^3} - y$  ist. Wir bekommen

$$\begin{aligned} a_{n+1} &= a_n - \frac{a_n^{-3} - y}{(-3a_n^{-4})} \\ &= \frac{4}{3} a_n \left( 1 - \frac{a_n^3}{4} y \right). \end{aligned} \quad (6)$$

### Ein spezielles Iterationsverfahren

Wir untersuchen nun eine Rekursionsformel, die sich äußerlich wenig von (6) unterscheidet, jedoch zu Iterationen führt, die in gewissen Parameterbereichen sich gänzlich anders verhalten, als wir das entsprechend Bild 1 erwarten.

Für  $2 \leq y \leq 4$  und  $a_0 = 0,5$  betrachten wir

$$a_{n+1} = a_n (1 - a_n) y. \quad (7)$$

Wie wir uns leicht überzeugen können, gilt für jedes natürliche  $n$  stets

$$0 \leq a_n \leq 1.$$

Wenn wir zunächst einmal annehmen, daß die Folge  $\{a_n\}$  für ein festes  $y$  gegen einen Grenzwert  $a$  strebt, so folgt aus (7)

$$a = \frac{y-1}{y} = 1 - \frac{1}{y}. \quad (8)$$

Für  $y = 2$  bekommen wir

$$a_1 = 0,5 \cdot 0,5 \cdot 2 = 0,5 = a_0,$$

und damit existiert der Grenzwert, und es ist  $a = 0,5$ .

Für  $y = 2,5$  finden wir

$$\begin{array}{ll} a_1 = 0,625 & a_{22} = 0,599\ 999\ 99 \\ a_2 = 0,585\ 937\ 5 & a_{23} = 0,600\ 000\ 01 \\ a_3 = 0,606\ 536\ 87 & a_{24} = 0,6 \\ a_4 = 0,596\ 624\ 74 & a_{25} = 0,6 \end{array}$$

Der Grenzwert (8) wird nach 24 Iterationen erreicht.

Für  $y = 3$  wird es bereits problematisch, wir erhalten

$$\begin{array}{ll} a_1 = 0,75 & a_2 = 0,5625 \\ a_{101} \approx 0,6884 & a_{102} \approx 0,6435 \\ a_{1001} \approx 0,6739 & a_{1002} \approx 0,6592 \\ a_{10001} \approx 0,6690 & a_{10002} \approx 0,6643 \\ a_{100001} \approx 0,6674 & a_{100002} \approx 0,6659 \end{array}$$

Mit 100000 Iterationen haben wir immer noch nicht feststellen können, ob die Folge gegen  $a = 0,666\ 666\ 67$  strebt oder nicht.

Für  $y = 3,1$  wird das Verhalten wieder überschaubar:

$$\begin{array}{ll} a_1 = 0,775 & a_2 = 0,540\ 562\ 5 \\ a_3 = 0,769\ 899\ 52 & a_4 = 0,549\ 178\ 17 \\ a_{55} = 0,764\ 566\ 53 & a_{56} = 0,558\ 014\ 12 \\ a_{57} = 0,764\ 566\ 52 & a_{58} = 0,558\ 014\ 12 \\ a_{59} = 0,764\ 566\ 52 & a_{60} = 0,558\ 014\ 12 \end{array}$$

Nach 60 Iterationen erkennen wir, daß die Zahlenfolge  $a = 0,677\ 419\ 35$  nicht erreicht, sondern ein periodisches Verhalten zeigt, wobei die Glieder zwischen zwei Werten hin- und herpendeln. Damit existiert kein Grenzwert, sondern es gibt zwei Häufungspunkte. Wir haben zunächst nur 4 Parameterwerte untersucht. Was können wir daraus schon für das Verhalten der Iterationen im Intervall  $(2, 4)$  schließen? Leider nur sehr wenig. Für kleinere  $y$  wird wohl jeweils ein Grenzwert  $g(y)$  existieren, für größere  $y$  kann ein periodisches Verhalten beobachtet werden, vielleicht auch noch etwa ganz anderes.

## Der Feigenbaum

Wir stehen vor der Aufgabe, eine äußerst umfangreiche Menge von Daten zu erzeugen und auszuwerten, um das Grenzverhalten der Folgen  $(a_n(y))$  gemäß (7) für  $2 \leq y \leq 4$  zu erfassen. Für einen qualitativen Überblick reicht es aus, im Intervall  $(2, 4)$  etwa 700 gleichabständige  $y$ -Werte auszuwählen. Des weiteren müßte es genügen, zur Auswertung der Zahlen  $a_n(y)$  einen Fehler von höchstens  $5 \cdot 10^{-3}$  zuzulassen. Damit brauchen wir nur etwa  $n = 500$  Werte pro Folge zu berechnen, das wären insgesamt etwa 350000 Werte. Diese ungeheueren Datenmenge kann in verblüffend einfacher Weise von einem Kleincomputer verarbeitet werden, indem man nach einer gewissen Bereinigung die Werte nicht ziffernmäßig ausdrückt, sondern plottet, d.h. jeden Wert durch einen Punkt auf einem Bildschirm darstellt und das entstandene Gebilde nebst einem Achsenkreuz ausdrucken läßt (vgl. Bild 2).

Wir nennen das Gebilde »Feigenbaum«, weil aus einem hohen Stamm zwei Äste hervorgehen, die sich wiederum verzweigen und in ein Blätterdach münden. Ob allerdings ein Botaniker in diesem Gebilde einen Feigenbaum (*Ficus*) erkennt, ist wohl zweifelhaft, doch der Mathematiker verbindet mit diesem Begriff sofort eine allgemeine Gesetzmäßigkeit der Bifurkationstheorie, die vor knapp zehn Jahren Herr M. J. FEIGENBAUM – ausgehend von der Rekursion (7) – entdeckt hat.

### Das BASIC-Programm »Feigenbaum«

Der Kleincomputer KC 85/2 aus Mühlhausen erlaubt,  $320 \times 256$  Bildpunkte einzeln zu programmieren. Uns stand allerdings nur ein Gerät zur Verfügung, das  $256 \times 176$  Bildpunkte ansprechen kann. Wir haben deshalb den  $y$ -Bereich



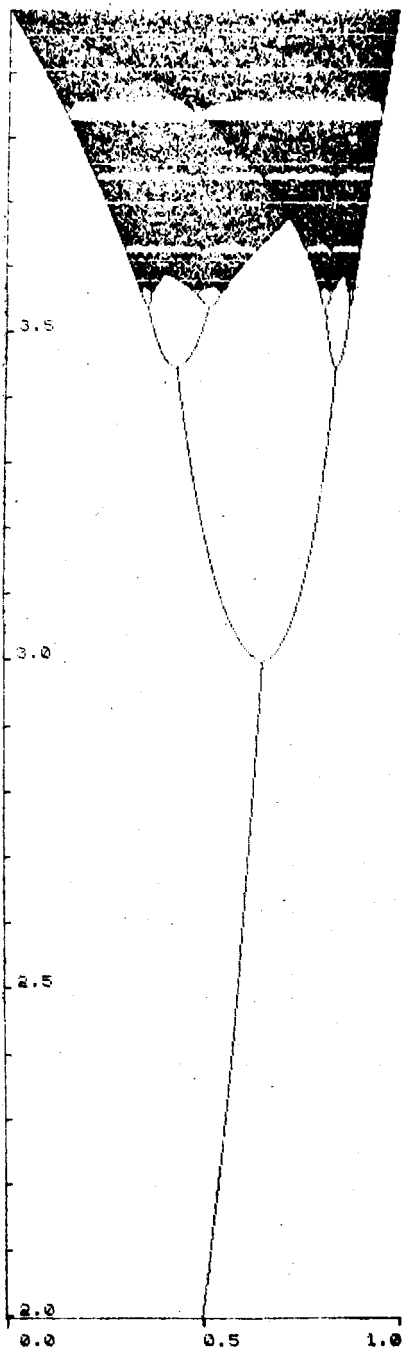


Bild 2. Der Feigenbaum

(2, 4) in vier Teile zerlegt: (2, 2,5), (2,5, 3), (3, 3,5), (3,5, 4), um über vier Bildschirme  $256 \times 704$  Bildpunkte parat zu haben. Mit der Zählvariable  $d$  wird der Durchlauf des auf einen Teil bezogenen Programms angezeigt (vgl. Bild 3, Zeile 20). Die Zählvariable  $v$  reguliert die Veränderung der  $y$ -Werte in einem Bereich der Länge 0,5, wobei 176 äquidistante Punkte verfügbar

```

0>REM Feigenbaum
   © 1985 by U. Girlich
10 LET y=4+0.5/176
20 FOR d=1 TO 4
30 REM Plotten und Beschriften
   der y-Achse
40 PLOT 0,0
50 DRAW 0,175
60 FOR s=0 TO 140 STEP 35
70 PLOT 0,s
80 DRAW 4,0
90 NEXT s
100 PRINT AT 21,1;4-0.5*d
110 IF INT (4-0.5*d)=4-0.5*d
   THEN PRINT AT 21,2;"0"
120 REM Hauptteil (Iteration)
130 FOR v=175 TO 0 STEP -1
140 LET y=y-0.5/176
150 LET a=0.5
160 FOR n=1 TO 325
170 LET a=a*y*(1-a)
180 NEXT n
190 FOR n=325 TO 500
200 LET a=a*y*(1-a)
210 PLOT 255*a,v
220 NEXT n
230 NEXT v
240 REM Ausdruck
250 COPY
260 CLS
270 NEXT d
280 REM Plotten und Beschriften
   der x-Achse
290 PLOT 0,175
300 DRAW 255,0
310 PRINT AT 1,1;"0.0"
320 PRINT AT 1,15;"0.5"
330 PRINT AT 1,29;"1.0"
340 FOR s=0 TO 255 STEP 127.5
350 PLOT s,175
360 DRAW 0,-4
370 NEXT s
380 COPY
390 STOP

```

Bild 3. Ausdruck des BASIC-Programms »Feigenbaum«

sind. Der Abstand aufeinanderfolgender  $y$ -Werte beträgt daher  $0,5/176$  (vgl. die Zeilen 10, 130, 140).

Das Herz des Programms besteht aus der Rekursion (7) (vgl. die Zeilen 170 und 200), wobei die Iterationen von

der Variablen  $n$  gezählt werden. Zunächst läuft  $n$  von 1 bis 325, damit wird  $a_{325}$  berechnet. Das Ziel unserer Untersuchung ist es, das Grenzverhalten der Folge  $\{a_n(y)\}$  zu erkennen. Wie unsere obigen Beispiele zeigen, findet zunächst eine Einschwingphase statt, die aber nach etwa 300 Iterationen im wesentlichen abgeschlossen ist. Die Daten der Einschwingphase und die das Grenzverhalten charakterisierenden Daten müssen getrennt werden, damit das Grenzverhalten durch die Einschwingphasen nicht vollständig überdeckt wird. Deshalb werden 325 Iterationen durchgeführt, allein um den Startwert  $a_{325}$  zu erhalten, mit dem dann die für das Grenzverhalten wesentlichen Daten  $a_{326}, a_{327}, \dots, a_{500}$  berechnet werden. Anschließend werden die Punkte  $(a_n(y), y)$  geplottet (vgl. Zeile 210). Nachdem die  $y$ -Achse und die  $x$ -Achse gezeichnet, mit einer Skala versehen und beschriftet worden sind, wird der gesamte Bildschirminhalt mittels eines Druckers ausgedruckt (Zeile 250) und das nächste Bild aufgebaut (Zeile 270).

Der Feigenbaum kann schneller »wachsen«, wenn nur der erste Durchlauf gemäß obigem Programm verläuft. Für den Parameterbereich  $(2,0, 3,5)$  reichen 100 Voriterationen und 10 Plottiterationen für jedes  $y$ .

### Auswertung der Experimente

Der aus 704 Experimenten hervorgegangene Feigenbaum gibt uns Auskunft über das Grenzverhalten der Folgen  $\{a_n(y)\}$  in Abhängigkeit von  $y$ . Wir lesen von Bild 2 ab: Im Bereich  $2,0 \leq y < 3,0$  konvergiert das Iterationsverfahren. Nach (8) gilt dort

$$a_n(y) \rightarrow g(y) := \frac{y-1}{y}.$$

Für  $3,0 < y < 3,44$  springt die Folge

nach der Einschwingphase zwischen zwei Werten hin und her, im Bereich  $3,45 < y < 3,54$  zwischen vier Werten, etwa bis 3,56 zwischen acht Werten, danach zwischen 16 Werten usw.

Um die Verzweigungspunkte zu lokalisieren, ist es notwendig, die Umgebungen der jeweils abgelesenen Verzweigungspunkte zu untersuchen, das heißt, weitere Experimente anzustellen und die entsprechenden Werte auszudrucken.

Für  $y = 2,95$  liegen noch normale Verhältnisse vor:

$$\begin{aligned} a_{320} &= 0,661\ 016\ 94 \\ a_{321} &= 0,661\ 016\ 95 = a_{322} = a_{323} \\ &= g(2,95). \end{aligned}$$

Für  $y = 2,99$  zeigt sich eine scheinbare Periodizität:

$$\begin{aligned} a_{1458} &= 0,665\ 551\ 82 \\ a_{1459} &= 0,665\ 551\ 85 = a_{1461} = a_{1463} \\ a_{1460} &= 0,665\ 551\ 83 = a_{1462} = a_{1464}, \end{aligned}$$

die durch Rundung bedingt ist. Für das arithmetische Mittel gilt

$$\frac{a_{1459} + a_{1460}}{2} = g(2,99).$$

Rundungseffekte beobachten wir auch am nächsten Verzweigungspunkt, der bei  $y = 1 + \sqrt{6} = 3,4494897$  liegt.

Für  $y = 3,44$  erhalten wir

$$\begin{aligned} a_{568} &= 0,442\ 192\ 99 \\ a_{569} &= 0,848\ 504\ 72 = a_{573} \\ a_{570} &= 0,442\ 192\ 94 = a_{574} \\ a_{571} &= 0,848\ 504\ 70 = a_{575} \\ a_{572} &= 0,442\ 192\ 98 = a_{576} \end{aligned}$$

und über das arithmetische Mittel die beiden Grenzlagen

$$\frac{a_{569} + a_{571}}{2} = 0,848\ 504\ 71$$

$$\frac{a_{570} + a_{572}}{2} = 0,442\ 192\ 96.$$

Für  $y = 3,4495$  bekommen wir

$$\begin{aligned}
 a_{71114} &= 0,439\ 099\ 41 \\
 a_{71115} &= 0,849\ 581\ 21 = a_{711,9} = a_{99999} \\
 a_{71116} &= 0,440\ 821\ 87 = a_{71120} = a_{100000} \\
 a_{71117} &= 0,850\ 294\ 67 = a_{71121} = a_{100001} \\
 a_{71118} &= 0,439\ 099\ 42 = a_{71122} = a_{100002}
 \end{aligned}$$

Für  $y = 3,45$  werden die vier Grenzlagen eher erreicht:

$$\begin{aligned}
 a_{3832} &= 0,445\ 967\ 85 \\
 a_{3833} &= 0,852\ 427\ 82 = a_{3837} = a_{50001} \\
 a_{3834} &= 0,433\ 991\ 49 = a_{3838} = a_{50002} \\
 a_{3835} &= 0,847\ 467\ 92 = a_{3839} = a_{50003} \\
 a_{3836} &= 0,445\ 967\ 84 = a_{3840} = a_{50004}
 \end{aligned}$$

In der Arbeit [2] wird »verwähnt, daß ab  $y = 3,56998 \dots$  im allgemeinen chaotisches Verhalten auftritt«. Das bedeutet, die vorher beobachtete Periodizität der Folge wird hier abgelöst durch ein nicht vorhersagbares unregelmäßiges Verändern der Glieder der Folge. Aus Bild 2 entnehmen wir, daß für  $y = 3,57$  der Rand des »Blätterdaches« des Feigenbaums erreicht ist. Damit wir diesen Rand genauer charakterisieren können, sind weitere Experimente erforderlich. Wir berechnen für  $y = 3,57$  die Werte  $a_1, a_2, \dots, a_{255}$  und plotten den Wert  $a_n$  an die Stelle  $(n, 176 \cdot a_n)$ . Damit ist auf jeder der 256 Bildschirmspalten genau ein Punkt erschienen. Lassen wir den Bildschirm

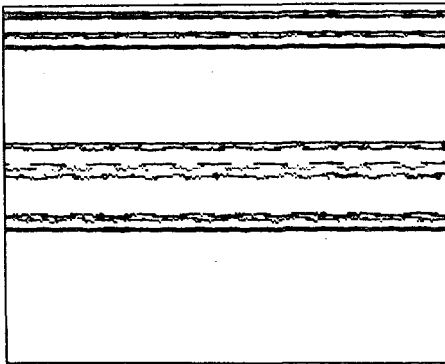


Bild 4.  $y = 3,57$

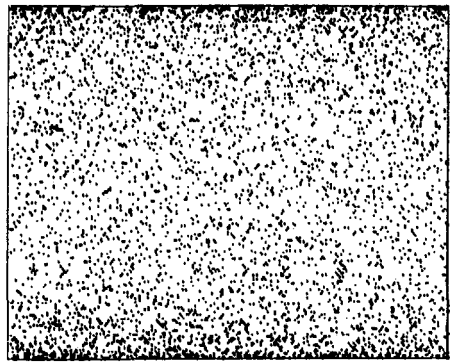


Bild 5. Das Chaos

20mal durchlaufen, so erhalten wir eine Verteilung der 5120 geplotteten Bildpunkte gemäß Bild 4.

Die untere Linie liegt bei 0,341, die obere bei 0,893. Man könnte hierbei von kleinen Schwankungen um 16 Werte sprechen. Der Ausdruck »chaotisches Verhalten« ist noch nicht sachgemäß. Anders dagegen liegen die Verhältnisse bei  $y = 3,99$ . Die 5120 Bildpunkte, die in Bild 4 im wesentlichen in Parallelen zur  $x$ -Achse angeordnet sind, verteilen sich in Bild 5 völlig chaotisch über den Bildschirm.

Es ist äußerst merkwürdig, daß die Rekursion (7) für  $y = 2,5$  eine konvergente Folge erzeugt, dagegen für  $y = 3,99$  eine Folge im Intervall  $(0, 1)$  nahezu gleichmäßig verteilter Zufallszahlen. Die Art der Verteilung bedarf allerdings noch einer Präzisierung und einer statistischen Prüfung.

### Ausblick

Wir haben hier das Grenzverhalten des speziellen Iterationsverfahrens (7) studiert. Die dabei praktizierte Datenverdichtung auf einem Bildschirm sollte gerade für die Entwicklung und Erprobung neuer Algorithmen ausgiebig genutzt werden.

## Literatur

- [1] BRONSTEIN, I. N.; SEMENDJAJEW, K. A.: Taschenbuch der Mathematik. – Leipzig: B. G. Teubner Verlagsgesellschaft, 1979
- [2] HENNIG, K.: Chaotisch-stochastische Systeme in der Technik. – In: spectrum 15 (1984) H. 10. – S. 1–4
- [3] KREUL, H.: Quadratische Gleichungen – einmal anders gelöst. – In: Kleinstrechner-TIPS, H. 1. – S. 41–43

Autor:

Prof. Dr. sc. nat. Hans-Joachim Girlich  
Karl-Marx-Universität Leipzig  
Sektion Mathematik

---

## Rechentechnische Begriffe für den Laien erklärt

- Cursor** Kleine Lichtmarke, die auf dem Bildschirm (Display, Monitor) anzeigt, an welcher Stelle die nächste Eingabe erscheinen wird. Der Cursor kann in allen Richtungen auf dem Bildschirm bewegt werden. Mit seiner Hilfe wird es damit auf einfache Weise möglich, fehlerhafte Eingaben nachträglich zu korrigieren, Informationen zu löschen, zusätzliche Informationen zu dem bereits vorhandenen Text hinzuzufügen usw.
- Chip** In der modernen Mikroelektronik wird mit Chip ein Halbleiterkristallplättchen (z. B. aus Silizium oder Galliumarsenid) bezeichnet, das in der Regel eine vollständig integrierte Schaltung aus vielen elementaren Bauelementen, wie Transistoren, Dioden, Widerständen, Kondensatoren, enthält.
- SSI** Kleinintegration (Small Scale Integration). Integration von bis zu 100 Bauelementen auf einem Chip.
- MSI** Mittelintegration (Medium Scale Integration). Integration von bis zu 1000 Bauelementen auf einem Chip.
- LSI** Großintegration (Large Scale Integration). Integration von bis zu 10000 Bauelementen auf einem Chip.
- Betriebssystem<sup>1)</sup>** Das Betriebssystem ist ein Paket von Grundprogrammen, das den Computer erst funktionsfähig macht. Bei Kleincomputern ist dieses Programmpaket im ROM gespeichert und tritt nach dem Einschalten sofort in Aktion. Es realisiert die Tastaturabfrage, die Bildschirmsteuerung und die Bedienung peripherer Einheiten, wie z. B. den Kasettenrecorder oder den Drucker.
- Display<sup>1)</sup>** Ein Display ist eine Anzeigeeinheit zur optischen Darstellung von Informationen. Dies kann z. B. in einem einfachen Fall die digitale Anzeige eines Taschenrechners sein. Bei Kleincomputern wird aus ökonomischen Gründen das meistens bereits vorhandene Fernsehgerät als Display genutzt.

---

<sup>1)</sup> Dieser Begriff wurde BÜCKNER, U.: Kleincomputer leichtverständlich. – Leipzig, 1986 entnommen



Die »Verständigung« des Menschen mit einem Rechner erfolgt heute zum überwiegenden Teil mit Hilfe von Tastaturen und optischen Anzeigen (z. B. Bildschirm). Hier soll nun gezeigt werden, daß das nicht immer so bleiben muß, weil es mit den entsprechenden technischen Vorkehrungen heute möglich geworden ist, sich mit einem Rechner auch mittels der menschlichen Sprache akustisch zu verständigen. Die Spracheingabe (Sprachanalyse) ist dabei die technisch schwierigere Seite. Wie die nachfolgend beschriebene praktische Realisierung eines Versuchsaufbaus zeigt, ist es aber trotzdem mit relativ niedrigem Aufwand möglich, einfache Spracherkennungssysteme zu bauen. Damit können dann erste Erfahrungen auf diesem interessanten Gebiet gesammelt werden.

Anwendungen der Spracheingabe unterstützen Tätigkeiten des Menschen, bei denen er keine Hand frei hat, aber trotzdem technische Einrichtungen bedienen möchte. Beispiele dafür sind die Bedienung von Meß- und Prüfplätzen, Krananlagen, Sortiereinrichtungen und vieles andere mehr.

## *Arbeitsweise*

Zur Realisierung einer Spracheingabe könnte man zunächst den niederfrequenten Schwingungszug eines gesprochenen Wortes als Muster regi-

strieren. Wenn das mit einem Rechner erfolgen soll, muß das analoge Sprachsignal in eine digitale Darstellung übergeführt werden. Aus der Fernsprechtechnik sind dafür Verfahren unter dem Namen Pulskodemodulation (PCM) bekannt. Für diese Verfahren ist charakteristisch, daß das niederfrequente Signal mit einem Analog/Digital-Wandler in ein digitales Datenwort von üblicherweise 8 Bit Länge (d. h. einem Byte) übergeführt wird. Da sich der Momentanwert der NF-Spannung laufend ändert, muß er alle 125 µs (d. h. 8000mal je Sekunde) ausgewertet werden; dabei wird von der beim Telefon üblichen Sprachqualität mit einer oberen Grenzfrequenz unter 4 kHz ausgegangen. Man ersieht daraus, daß 8000 Byte Speicherplatz (RAM) benötigt werden, um eine Sekunde Sprache abzuspeichern.

Im Speicher des Rechners selbst seien die digitalisierten Schwingungszüge aller zu erkennenden Worte gespeichert. Ein dem Erkennungssystem zugeführtes Wort wird nun ebenfalls digitalisiert und die entstehende Datenfolge mit allen Mustern verglichen. Bei Übereinstimmung hat der Rechner herausgefunden, welches Wort gesprochen wurde.

Ganz so einfach wie gerade beschrieben funktioniert es leider nicht. Zunächst einmal gelingt es keinem Menschen,

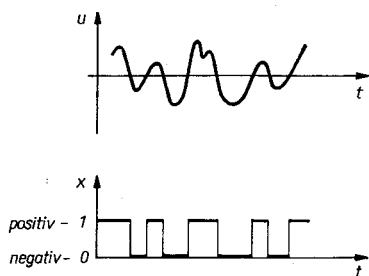


Bild 1. Vorzeichenfunktion des Sprachsignals

die gleichen Worte immer so auszusprechen, daß sich immer genau der gleiche Schwingungszug ergibt; noch ungünstiger sind die Verhältnisse bei mehreren Stimmen. Es muß also aus der Schwingungsform ein charakteristisches Merkmal gewonnen werden, das nicht so starken individuellen Schwankungen unterliegt. Ein solches Merkmal kann z. B. die Frequenz sein; weil in der Sprache aber immer mehrere Frequenzen gleichzeitig auftreten, spricht man besser von einem Frequenzspektrum. Da sich das Frequenzspektrum über die Dauer eines Wortes ändert, ist es jeweils für kleine Zeitabschnitte (einige Millisekunden) getrennt zu bestimmen.

Die Spektralanalyse ist von der technischen Realisierung her recht aufwendig, hier soll deshalb ein einfacherer (aber auch weniger leistungsfähiger) Weg beschrieben werden. Man geht dabei davon aus, daß das Vorzeichen des NF-Signals sehr einfach durch einen Schwellwertschalter (Komparator) zu ermitteln ist (Bild 1). Die Häufigkeit der Nulldurchgänge bzw. die Abstände zwischen ihnen sind ebenfalls ein Maß für die im Sprachsignal enthaltenen Frequenzen und damit charakteristische Merkmale des gesprochenen Wortes.

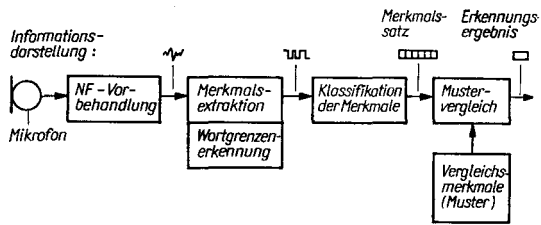
Ein weiterer Nachteil der ursprünglich diskutierten Variante (Abspeicherung der digitalisierten Schwingungsform)

war der sehr hohe Speicheraufwand für die Muster. Dieser reduziert sich bei der Auswertung des Vorzeichens sehr stark, wenn man Klassen der Nulldurchgangsabstände einführt. Es wird also nicht mehr jeder einzelne Abstand registriert, sondern nur noch, in welche der festgelegten Bereichsklassen er fällt. Als Ergebnis erhält man beispielsweise, daß  $n_1$  Abstände im Bereich  $< 0,3$  ms (Klasse 1),  $n_2$  Abstände im Bereich von  $0,3 \cdots 0,6$  ms,  $n_3$  Abstände im Bereich  $0,6 \cdots 1$  ms und  $n_4$  Abstände im Bereich  $> 1$  ms liegen; die hier angegebenen vier Klassen entsprechen denen der praktischen Realisierung.

Es wurde schon darauf hingewiesen, daß sich das Spektrum während eines Wortes ändert. Das kann auch bei der Nulldurchgangsanalyse ausgenutzt werden, indem das Wort in eine Anzahl Zeitabschnitte aufgeteilt wird, für die die Häufigkeiten der einzelnen Klassen getrennt ermittelt werden. Die Einteilung der gesprochenen Worte in eine feste Zahl von Abschnitten hat gegenüber der Auswertung nach festen Zeitabschnitten den Vorteil, daß Variationen in der Sprechgeschwindigkeit das Resultat weniger beeinflussen. Bei vier Klassen und vier Zeitabschnitten je Wort erhält man 16 Merkmalswerte (in unserem Fall sind das 16 Byte Speicherplatz).

Der Vergleich der nach dem beschriebenen Verfahren gewonnenen Merkmale mit den Mustermerkmalen wird in den seltensten Fällen völlige Übereinstimmung ergeben. Es muß also ein Maß für die »größte Ähnlichkeit« gefunden werden. Ein solches Maß, das sehr einfach zu ermitteln ist, erhält man durch Bilden der Summe der Differenzbeträge der einzelnen Merkmale, d. h., beim Vergleich der Werte  $x_1 \cdots x_{16}$  (Eingabemerkmale) mit  $y_1 \cdots y_{16}$  (Muster) wird

Bild 2. Blockschaltbild eines Sprach-eingabesystems



$d =$

$$|x_1 - y_1| + |x_2 - y_2| + \dots + |x_{16} - y_{16}|$$

gebildet. Das Muster, bei dem der kleinste Wert  $d$  ( $d_{\min}$ ) auftritt, wird als zutreffend für das eingegebene Wort gewertet.

Nun kann es noch vorkommen, daß ein Wort oder Geräusch zur Auswertung gelangt, von dem überhaupt kein Muster vorhanden ist. In einem solchen Fall muß die Erkennung abgewiesen werden. Der Fall kann daran erkannt werden, daß auch die kleinste Differenz  $d_{\min}$  eine vorgegebene Abwieschwelle  $s$  übersteigt.

Aus dieser Beschreibung der Funktionsweise ergibt sich das Blockschaltbild einer Spracherkennungseinrichtung nach Bild 2.

### Hardware

Der praktische Aufbau einer Spracherkennungseinrichtung nach den erläuterten Prinzipien erfordert einen Mikrorechner auf U-880-Basis, der um einen Zusatz zum Anschluß des Mikrofons erweitert wird. Das System wurde praktisch erprobt mit einem POLY-COMPUTER 880, es ist aber mit geringen Modifikationen auch auf andere Rechner übertragbar.

Die Speicher- und E/A-Adressen sind dann zu korrigieren, evtl. ist auch die veränderte Taktfrequenz zu berücksichtigen (Klassengrenzen).

Der Stromlaufplan des Mikrofonanschlusses ist in Bild 3 dargestellt. Als Mikrofon sind beispielsweise die dyna-

mischen Mikrofone von Hör-Sprech-Kombinationen geeignet. A1 verstärkt das NF-Signal. Der nachfolgende Tiefpaß mit einer Grenzfrequenz von etwa 4 kHz trennt höherfrequente Sprachanteile ab, die für die Erkennung nicht relevant sind. Mit dem Operationsverstärker A2 wird das Vorzeichen des Sprachsignals ermittelt und das binäre Resultat einem PIO-Anschluß zugeführt. Der Schwellwertschalter A3 dient zusammen mit der davor befindlichen Gleichrichterschaltung der Wortgrenzenerkennung. Das Resultat wird ebenfalls einem PIO-Anschluß zugeführt (1-Ruhe).

R1 dient zur Anpassung an Mikrofone mit unterschiedlicher Empfindlichkeit, R2 zur Einstellung der Amplitudeschwelle für die Wortgrenzenerkennung. Mit einer Leuchtdiode kann die Funktion dieser Anordnung kontrolliert werden. Wenn die Leuchtdiode während eines Wortes verlischt, ist die Ansprechschwelle zu hoch; umgekehrt spricht die Anordnung bei zu niedriger Schwelle auf Störgeräusche aus der Umgebung an.

Die angegebene Steckerbelegung gilt für den Peripherieanschluß des POLY-COMPUTER 880, es werden die Bits 0 und 1 von Port B der eingebauten PIO genutzt.

### Software

Das Programm des Mikrorechners, dessen Ablaufplan in Bild 4 dargestellt ist, muß zwei Aufgaben realisieren:





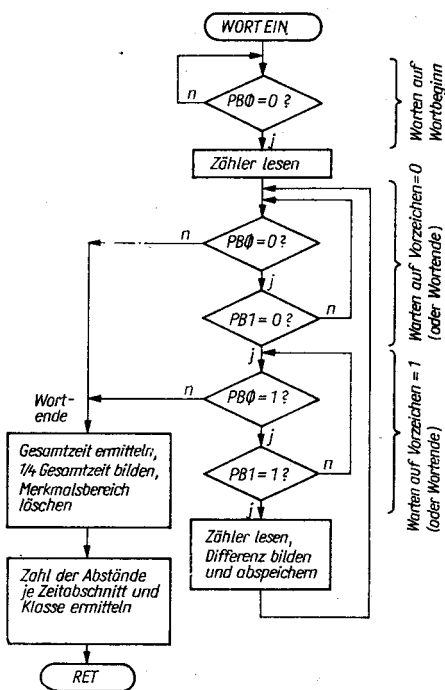


Bild 5. Ablaufplan des Unterprogramms »Worteingabe«

modus erfolgt. Wenn eine der 16 Hexadezimaltasten betätigt wird, erfolgt ein Übergang in den Lernmodus. Das nächste gesprochene Wort wird als Vergleichsmuster »0...F« abgespeichert; es können folglich bis zu 16 Worte »angelernt« werden. Man muß sich allerdings merken, welches Wort welcher Hexadezimalziffer entspricht, wenn man nicht gerade die gesprochenen Hexadezimalziffern selbst benutzt.

Bei Betätigung der Taste EXEC wird in den Erkennungsmodus übergegangen. Erkennungsergebnisse werden in Form einer Hexadezimalziffer in der ersten Stelle der Siebensegmentanzeige dargestellt. Falls keine hinreichende Ähnlichkeit mit einem der gespeicherten Muster besteht (Abweisung), wird ein waagerechter Strich ausgegeben.

Die Ausgabe erfolgt nur für etwa eine Sekunde, dann ist das System wieder bereit zur Erkennung des nächsten Wortes. Ein Programmabbruch kann nur über die MON-Taste erfolgen.

### Erfahrungen

Der vorgestellte Versuchsaufbau realisiert die Erkennung von bis zu 16 Worten, wobei die Wortzahl auf Kosten der Speicherkapazität (16 Byte je Wort) erhöht werden kann (bis etwa 100). Alle Worte müssen vorher angelernt werden, das Verfahren ist also sprecherabhängig. Ein Versuch mit verschiedenen Sprechern in der Anlern- und Erkennungsphase ist nicht prinzipiell aussichtslos, bringt aber eine stark erhöhte Fehlerrate. Die Worte müssen isoliert, d.h. mit deutlichen Pausen dazwischen, gesprochen werden. Bei stärkeren Umgebungsgeräuschen ist die Anordnung nicht brauchbar.

Bis zur Beherrschung der Eingabe fortlaufend gesprochener Sprache mit dem vollen natürlichen Vokabular sind noch wesentliche Verbesserungen und auch neue Prinziplösungen notwendig. Es ist auch nicht zu erwarten, daß der »hörende und sprechende Rechner« die heute üblichen Bedien- und Anzeigeeinrichtungen völlig verdrängen wird, vor allem wenn es um Exaktheit und große Informationsmengen geht.

An der beschriebenen einfachen Realisierung können noch folgende Verbesserungen vorgenommen werden:

- Umstellung der Sprachdatenaufnahme auf Unterbrechungsbetrieb. Damit ist dann eine kontinuierliche Anzeige möglich.
- Bilden der Mittelwerte aus mehreren Anlernversuchen für jedes Wort
- Optimierung der Werte für Klassengrenzen, Abweisschwelle, Klassen- und Zeitabschnittszahl.

ADR	OBJ CODE	NR	SPRACHEINGABE QUELLANWEISUNG	SEITE
		1	;;;	
		2	;	
		3	; Software zum Spracheingabe-Versuch	
		4	;	
		5	;;;	
2000		6	ORG 2000H ;ROM-Bereich	
		7	START:	
		8	;Programmieren eines CTC-Kanals als	
		9	;durchlaufender Zaehler mit der	
		10	;Zeiteinheit 16us (1 MHz-Takt)	
2000	3E07	11	LD A,07H	
2002	D38B	12	OUT (CTC3),A	
2004	3E00	13	LD A,0	
2006	D38B	14	OUT (CTC3),A	
		15	;Programmieren einer PIO auf Bit-Mode/Ein-	
		16	;gabe (nur Bit 0,1 sind benutzt)	
2008	3EFF	17	LD A,OFFH	
200A	D387	18	OUT (PBC),A	
200C	D387	19	OUT (PBC),A	
		20	;Aurufschleife fuer Tastaturprogramm aus	
		21	;PC880-Monitor, liefert Kode in A,	
		22	;Hexadezimalziffern in C	
		23	TASFRA:	
200E	CD4B01	24	CALL TASTU	
2011	28FB	25	JR Z,TASFRA ;warten auf Taste	
2013	FE41	26	CF 41H ;EXEC-Taste ?	
2015	281E	27	JR Z,ERKEN	
2017	CB47	28	BIT 0,A ;Hex-Taste ?	
2019	20F3	29	JR NZ,TASFRA ;andere Taste	
		30	;Lernmodus, Wortindex in C	
		31	;Abspeicheradresse fuer Merkmale dieses	
		32	;Wortes bestimmen: 16*Index + Basis	
201B	69	33	LD L,C	
201C	2600	34	LD H,0	
201E	29	35	ADD HL,HL	
201F	29	36	ADD HL,HL	
2020	29	37	ADD HL,HL	
2021	29	38	ADD HL,HL	
2022	110042	39	LD DE,KLBER ;Basis	
2025	19	40	ADD HL,DE	
2026	E5	41	PUSH HL	
2027	CD8D20	42	CALL WORTEIN ;Wortauswertung	
202A	D1	43	POP DE ;Ziel	
202B	210043	44	LD HL,AKTWER ;Quelle	
202E	011000	45	LD BC,16 ;Laenge	
2031	EDB0	46	LDIR	
2033	18D9	47	JR TASFRA ;fertig	
		48	;	
		49	;Erkennungsphase	
		50	ERKEN:	
2035	CD8D20	51	CALL WORTEIN ;Wortauswertung	
		52	;Kleinste Differenzbetrags-Summe ermitteln	
		53	; D-Voreinstellung fuer kleinsten Wert=FF	
		54	; E-Zaehler fuer die 16 Vergleichsmuster	
		55	; B-Zaehler fuer die 16 Merkmale je Wort	
		56	; C-Differenzbetrags-Summe	
2038	210042	57	LD HL,KLBER ;Musterbereich	
203B	1110FF	58	LD DE,OFF10H	
		59	NEUGL:	
203E	D5	60	PUSH DE	
203F	110043	61	LD DE,AKTWER	

```

2042 010010      62      LD      BC,1000H
                63      DIFBIL:
2045 1A          64      LD      A,(DE)
2046 96          65      SUB     (HL)
2047 280A        66      JR      Z,IDENT ;Uebereinstimmung
2049 3002        67      JR      NC,POSIDIF ;posit. Differenz
204B ED44        68      NEG     ;Betragsbildung
                69      POSIDIF:
204D 81          70      ADD     A,C
204E 4F          71      LD      C,A
204F 3002        72      JR      NC,IDENT ;kein Ueberlauf
2051 0EFF        73      LD      C,OFFH ;Maximalwert bei CY
                74      IDENT:
2053 23          75      INC     HL ;weiter zum
2054 13          76      INC     DE ;naechsten Merkmal
2055 10EE        77      DJNZ   DIFBIL
                78      ;Testen, ob gerade ermittelter Wert (in C)
                79      ;neues Minimum ist (wenn ja, abspeichern)
2057 D1          80      POP     DE
2058 79          81      LD      A,C
2059 BA          82      CP      D
205A 3005        83      JR      NC,NOTMAX
205C 51          84      LD      D,C ;neues Minimum
205D ED531043    85      LD      (RESULT),DE ;merken
                86      NOTMAX:
2061 1D          87      DEC     E ;naechstes Wort
2062 20DA        88      JR      NZ,NEUGL
                89      ;Vergleich beendet, Abweisung pruefen
2064 ED5B1043    90      LD      DE,(RESULT)
2068 3E10        91      LD      A,16
206A 93          92      SUB     E
206B 5F          93      LD      E,A ;zutreffender Index
206C 7A          94      LD      A,D
206D FE80        95      CP      ABWEIS
206F 3807        96      JR      C,GUELT ;wenn d < ABWEIS
                97      ;keine Aehnlichkeit, "-" ausgeben
2071 3E10        98      LD      A,10H
2073 32E343     99      LD      (ANZBER+7),A
2076 180A       100     JR      ANZEIG
                101     ;Wort gefunden (in E), Anzeige konvertierung
                102     GUELT:
2078 211003    103     LD      HL,ANZDEC ;7-Seg.-Tabelle
207B 1600      104     LD      D,0
207D 19        105     ADD     HL,DE
207E 7E        106     LD      A,(HL) ;7-Seg.-Kode
207F 32E343    107     LD      (ANZBER+7),A
                108     ;Anzeige fuer etwa eine Sekunde
                109     ANZEIG:
2082 06C8     110     LD      B,200
                111     ANZ1S:
2084 C5        112     PUSH    BC
2085 CD4B01    113     CALL   TASTU ;Routine des Monitors
2088 C1        114     POP     BC
2089 10F9     115     DJNZ   ANZ1S
208B 18AB     116     JR      ERKEN ;fertig
                117     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                118     ; Unterprogramm "Worteingabe"
                119     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                120     WORTEIN:
                121     ;warten auf Wortanfang
208D DB86     122     IN      A,(PBD)
208F 1F       123     RRA ;liefert CY, wenn PBD=1
2090 38FB     124     JR      C,WORTEIN
                125     ;Datenerfassung, warten auf Wortende

```

```

2092 110040 126 LD DE,WBER ;Abspeicherbereich
2095 DBBB 127 IN A,(CTC3)
2097 47 128 LD B,A ;alter Zaehlerstand
129 ;warten auf Vorzeichenfunktion (PB1)=0
130 WART0:
2098 DBB6 131 IN A,(PBD)
209A 1F 132 RRA
209B 3B1B 133 JR C,WOEND
209D 1F 134 RRA
209E 3BF8 135 JR C,WART0
136 ;warten auf Vorzeichenfunktion (PB1)=1
137 WART1:
20A0 DBB6 138 IN A,(PBD)
20A2 1F 139 RRA
20A3 3B13 140 JR C,WOEND
20A5 1F 141 RRA
20A6 30FB 142 JR NC,WART1
143 ;Abstand (Zaehlerdifferenz)- registrieren
20AB DBBB 144 IN A,(CTC3)
20AA 4F 145 LD C,A
20AB 78 146 LD A,B
20AC 91 147 SUB C ;ergibt Differenz
20AD 41 148 LD B,C
20AE 12 149 LD (DE),A
20AF 13 150 INC DE
151 ;Ueberlaufsaecherung fuer Wertebereich
20B0 210042 152 LD HL,WBER+200H
20B3 A7 153 AND A
20B4 ED52 154 SBC HL,DE
20B6 3BE0 155 JR N C,WART0
156 ;Abbruch bei Ueberlauf
157 ;Wartende - Gesamtzeit bestimmen
158 WOEND:
20B8 EB 159 EX DE,HL
20B9 110040 160 LD DE,WBER
20BC A7 161 AND A
20BD ED52 162 SBC HL,DE ;HL-Werteanzahl
20BF EB 163 EX DE,HL
20C0 DD210000 164 LD IX,0 ;Zeit
165 ZSUMM:
20C4 4E 166 LD C,(HL) ;Wert
20C5 0600 167 LD B,0
20C7 DD09 168 ADD IX,BC
20C9 23 169 INC HL
20CA 1B 170 DEC DE ;weiter zum
20CB 7A 171 LD A,D ;naechsten Wert
20CC B3 172 OR E
20CD 20F5 173 JR NZ,ZSUMM
174 ;1/4 der Gesamtzeit bilden
20CF DDE5 175 PUSH IX
20D1 E1 176 POP HL
20D2 CB3C 177 SRL H
20D4 CB1D 178 RR L
20D6 CB3C 179 SRL H
20D8 CB1D 180 RR L
20DA E5 181 PUSH HL
20DB DDE1 182 POP IX ; IX := T/4
20DD E5 183 PUSH HL
20DE FDE1 184 POP IY ; IY := T/4
185 ;Auswertung nach den vier Klassen
186 ;vorher Ergebnisbereich loeschen
187 ;IY - Laenge des aktuellen Zeitabschnitts
188 ; evtl. mit Rest des vorherigen Abschnitts
20E0 0610 189 LD B,16

```

```

20E2 210043 190 LD HL,AKTWER
191 AKTLOE: LD (HL),0
20E5 3600 192 INC HL
20E7 23 193 DJNZ AKTLOE
20EB 10FB 194 LD B,4 ;Zeitabschnitte
20EA 0604 195 LD HL,AKTWER
20EC 210043 196 NEZEIT: PUSH BC
197 NEWER:
20EF C5 198 PUSH HL
199 LD A,(DE)
20F0 E5 200 INC DE
20F1 1A 201 INC KL1
20F2 13 202 CP C,KLA1
20F3 FE13 203 CP KL2
20F5 380B 204 CP C,KLA2
20F7 FE26 205 CP KL3
20F9 3806 206 JR C,KLA3
20FB FE3E 207 ;Rest ist Klasse 4
20FD 3801 208 ;Weiterzaehlen in der ermittelten Klasse
209
20FF 23 210 INC HL
2100 23 211 KLA3 INC HL
2101 23 212 KLA2 INC HL
2102 34 213 KLA1 INC (HL)
214 ;Pruefen, ob Zeitabschnitt zu Ende
215 LD C,A
2103 4F 216 LD B,0
2104 0600 217 PUSH IY
2106 FDE5 218 POP HL
2108 E1 219 AND A
2109 A7 220 SBC HL,BC
210A ED42 221 PUSH HL
210C E5 222 POP IY
210D FDE1 223 POP HL
210F E1 224 POP HL
2110 30DE 225 JR NC,NEWER
226 ;naechster Zeitabschnitt
2112 23 227 INC HL
2113 23 228 INC HL
2114 23 229 INC HL
2115 23 230 INC HL
2116 DDE5 231 PUSH IX
2118 C1 232 POP BC
2119 FD09 233 ADD IY,BC ;Rest + 1/4 Gesamtzeit
211B C1 234 POP BC
211C 10D1 235 DJNZ NEZEIT ;naechster Zeitabschn.
211E C9 236 RET ;fertig
237 ;
238 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
239 ; RAM-Bereich fuer Spracherkennung
240 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4000 241 ORG 4000H
4000 242 WBER DEFS 200H ;Abstaende
4200 243 KLBER DEFS 100H ;16 Muster a 16 Byte
4300 244 AKTWER DEFS 16 ;ermittelte Merkmale
4310 245 RESULT DEFS 2 ;minimale Differenz
246 ;und zugehoeriges Muster
247 ;
248 ;Portadressen
249 CTC3 EQU 8BH ;CTC-Kanal
250 PBD EQU 86H ;PIO-Daten
251 PBC EQU 87H ;PIO-Steuerung
252 ;
253 ;Klassengrenzen und Abweisschwelle

```

254	KL1	EQU	13H
255	KL2	EQU	26H
256	KL3	EQU	3EH
257	ARWEIS	EQU	80H
258	;		
259	;	Adressen des POLYCOMPUTER-Monitorprogramms:	
260	;	Tastatur- und Anzeigeroutine des Monitors	
261	TASTU	EQU	14BH
262	;	Hex-zu-Siebensegment-Umrechnungstabelle	
263	ANZDEC	EQU	310H ;im Monitor
264	;	Anzeigeabbild fuer TASTU	
265	ANZBER	EQU	43DCH

Die Erkennungssicherheit ist stark von einer gleichmäßigen Mikrofonhaltung (Abstand) abhängig; größere Abstände als einige Zentimeter sind wegen der Störgeräuschbeeinflussung kaum möglich. Gute Erfahrungen wurden mit Hör-Sprech-Garnituren und Studio-Nahbesprechungsmikrofonen gemacht. Bei geeigneter Auswahl der Worte lassen sich Erkennungssicherheiten von über 95<sup>0</sup>/<sub>0</sub> erreichen. Sehr ähnlich klingende Worte können auch von der automatischen Spracherkennung nur schlecht unterschieden werden, z. B. »drei«- »zwei« (besser: »zwo«).

Es soll an dieser Stelle betont werden, daß dieser Beitrag kein »Kochrezept« darstellen soll, sondern eine Einführung in die Problematik und eine Anregung zu eigenen Experimenten.

Autor:

*Dr. sc. techn. Uwe Hübner*

Wissenschaftlicher Mitarbeiter  
im Institut für Nachrichtentechnik  
Forschungszentrum des VEB Kombinat  
Nachrichtenelektronik

---

## Rechentechnische Begriffe für den Laien erklärt

**Firmware<sup>1)</sup>** Programme, wie z. B. das Betriebssystem oder ein Interpreter, die fest im ROM gespeichert sind, bezeichnet man als Firmware. Dazu gehören bei Kleincomputern zumindest die Routinen zur Tastaturabfrage und zur Bildschirmsteuerung. Firmware ist also hardwaremäßig installierte Software.

**Maus<sup>1)</sup>** Die Maus ist eine recht junge, erst 1983 international populär gewordene Ergänzung zum Kleincomputersystem. Hinter diesem vertrauten Begriff verbirgt sich ein kleines handliches Kästchen mit mehreren Schaltern auf der Oberseite und einer Rollkugel auf der Unterseite. Wird die Maus nun auf einer glatten Oberfläche hin- und herbewegt, so wird diese Bewegung auf dem Bildschirm durch einen Cursor nachvollzogen. Auf diese Weise können Kommandos, Worte, Zeichen usw. leicht auf dem Bildschirm ausgewählt und durch Betätigung eines Schalters bestimmte Funktionen ausgelöst werden. Dabei ersetzt die Maus selbstverständlich keine Tastatur, sondern stellt als Eingabehilfe eine sinnvolle Ergänzung dar.

---

<sup>1)</sup> Dieser Begriff wurde BÜCKNER, U.: Kleincomputer leichtverständlich. - Leipzig, 1986 entnommen

---

# Homecomputer – ein neues Gebiet für den Amateur (Teil 3)



Mit diesem Teil soll die Vorstellung der Grundkonfiguration für einen Kleincomputer abgeschlossen werden. Um das in Teil 1 und 2 (Heft 1 und 2) vorgestellte System nutzbar zu machen, ist ein Betriebssystem erforderlich, welches die Voraussetzungen schafft, weitere Entwicklungsarbeiten mit dem Rechner zu realisieren. Außerdem wird ein Kassetteninterface vorgestellt, welches der Programm- und Datenspeicherung dient und bereits höheren Anforderungen Rechnung trägt.

## Das Mini-Betriebssystem

Der Kleincomputer erfordert für seine Nutzung ein Programm, mit dessen Hilfe der Nutzer in der Lage ist, dem Rechner Programme einzugeben, diese zu starten und zu korrigieren. Als Minimum für ein solches System sind folgende Funktionen anzusehen:

- a) Initialisieren des Rechners, d.h. Setzen verschiedener Register im Prozessor, Bildschirm Löschen, Definieren von Arbeitszellen im Speicher usw.
- b) Anzeige des Speicherinhaltes
- c) Eingabe in den Speicher
- d) Starten eines eingegebenen Programms ab einer definierbaren Adresse.

Ergänzt werden können diese Kommandos in der nächsten Stufe durch Befehle zur Eingabe und Ausgabe von Programmen auf bzw. von einem Datenträger.

Um diese Aufgabe mit einem möglichst geringen Speicherbedarf zu realisieren, bietet es sich an, mehrere allgemein nutzbare Module zu schaffen. Sie können später oft auch aus dem Anwenderprogramm heraus genutzt werden. Diese Modulsammlung läßt sich in mehreren Stufen aufbauen. Für das Minimum sind erforderlich:

- a) Eingabe von 4 Zeichen Hex (Hexadezimal)
- b) Ausgabe von 4 Zeichen Hex
- c) a) und b) für 2 Zeichen
- d) Wandlung ASCII-Code in Hexadezimaldarstellung und umgekehrt

Ergänzungen dafür wären:

- e) Wandlung von Hexwerten in Dezimalwerte und umgekehrt
- f) Ausgabe von Textketten.

Bei dem folgenden Programm wird davon ausgegangen, daß die Tastatur ASCII-Zeichen liefert. Von den nicht im Bildschirmprogramm angegebenen Zeichen des ASCII-Satzes wird noch das Zeichen ETX (= CNTRL-C=03H) und eine als SYS definierte Taste genutzt, welche als Endekennung verwendet werden.

Nach einer Anfangsmeldung reagiert das Betriebssystem auf die nachfolgenden Befehle. Die Bereitschaft des Systems zur Befehlseingabe wird durch die Ausgabe eines \* gekennzeichnet.

#### V (Video)

In dieser Schleife werden nur die eingegebenen Zeichen auf dem Bildschirm dargestellt. Er kann zum Testen verwendet werden und wird durch ETX verlassen.

#### R XXXX (Read)

Lesen des Speicherbereiches ab Adresse XXXX bis Ende durch ETX. Mit Bedienung einer Taste ungleich ETX wird eine Zeile zu 8 Byte angezeigt.

#### W XXXX (Write)

Schreiben in den Speicher ab Adresse XXXX. Bei Eingabe von Leerzeichen bleibt der angezeigte Inhalt erhalten, bei Werteingabe wird er überschrieben. Mit der Taste »Cursor hoch« wird die Adresse um 4 Plätze zurückgestellt. Beendet wird mit ETX.

#### G XXXX (Go)

Start eines Programms ab Adresse XXXX. Mit dem Befehl »RET« bzw. »RST 38H« im Programm wird in das Betriebssystem zurückgekehrt.

#### E X (Echo Tastatur)

Eingabe des Geschwindigkeitsfaktors für die Echofunktion der Tastatur. ( $X = 0 \dots F$ )

#### L (Load)

Laden eines Programms vom Magnetband.

#### S XXXX YYYY (Save)

Aufzeichnung des Bereiches von Adresse XXXX bis einschließlich Adresse YYYY auf Magnetband.

#### CL XXXX YYYY (Clear)

Löschen des Speicherbereiches. Aus Sicherheitsgründen werden 2 Buchstaben bei der Eingabe gefordert, wodurch ein versehentliches Löschen ausgeschlossen wird. Gelöscht wird von Adresse XXXX bis einschließlich Adresse YYYY. Soll nur ein Byte gelöscht werden, so ist  $XXXX = YYYY$  zu setzen.

Zusätzlich wird für die Tastaturroutine ein Vorprogramm geschrieben, welches auf eine vom Nutzer zu definierende Taste von jeder beliebigen Stelle aus eine Rückkehr in die Systemgrundschleife erzwingt. Sie trägt den Namen SYSTS. Das gleiche kann durch ein entsprechendes Programm zum Bildschirmtreiber erfolgen (SYSVD). Damit ist es möglich, den Rechner bei jeder Ein- oder Ausgabe in den Grundzustand zurückzuholen, falls sich ein Programm »verlaufen« hat.

Das Betriebssystem arbeitet ohne Interrupt und initialisiert auch die dazu nötigen Register nicht. Es läßt sich sicher noch an einigen Stellen optimieren, worauf zugunsten der Übersichtlichkeit verzichtet wurde.

### Das Kassetteninterface

Die Anschlußsteuerung für ein Kassettengerät (oder auch Spulentonbandgerät) ist Bild I zu entnehmen. Die Schaltung arbeitet nach dem Phasenmodulationsprinzip, d.h., jede Periode der aufgezeichneten Schwingung trägt ein Informationsbit. Die Grundidee zu dieser Schaltung wurde aus [1] entnommen. In der hier angegebenen modifizierten Schaltung läuft sie sehr zuverlässig und eignet sich dadurch sowohl für asynchrone als auch für synchrone Aufzeichnungen. Tests haben ergeben, daß Aufzeichnungen mit 4800 Bit/s auf Kassettengeräten und



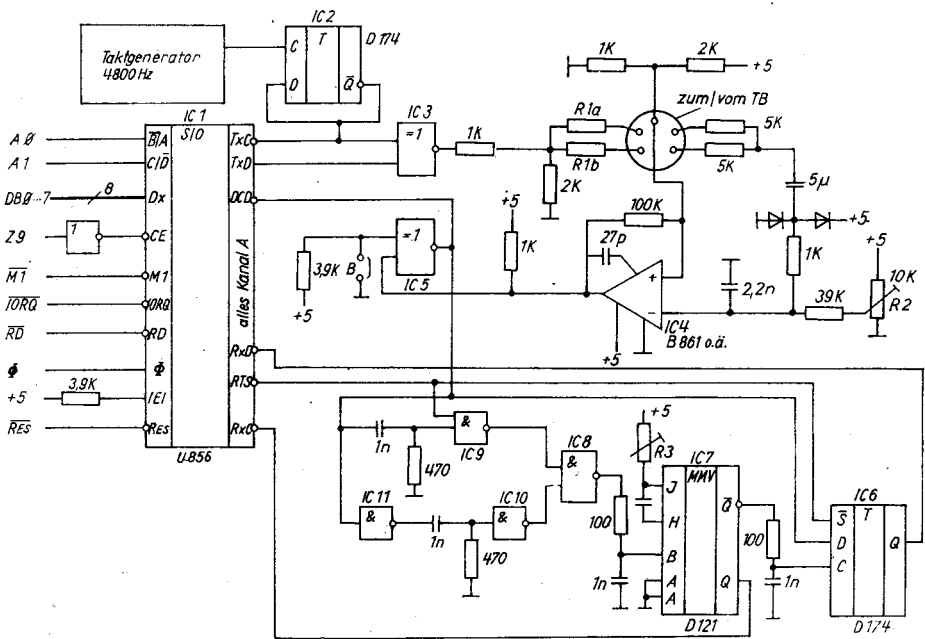


Bild 1. Anschlußsteuerung für ein Kassettengerät

9600 Bit/s auf Spulengeräten noch gut möglich sind. Allerdings sollten bei diesen Speicherdichten Stereogeräte verwendet werden. Dadurch sinkt die Wahrscheinlichkeit, Datenfehler durch Drop-outs zu bekommen, da eine Spur für diesen Moment zumeist unbeschädigt ist. Einen optimalen Wert für Aufzeichnungsdichte und Datensicherheit stellt die Aufzeichnungsrate von 2400 Bit/s dar, was bei Asynchronbetrieb 200 Byte je Sekunde Nutzinformation ergibt.

### Senderteil

Der Senderteil besteht lediglich aus dem Generator und den Schaltkreisen 2 und 3. Als Generator kann eine beliebige, genügend stabile Schaltung verwendet werden. Es bieten sich dazu Schaltungen mit dem A301 oder einem CTC (U857) an. Die absolute Taktfrequenz ist dabei unkritisch, solange

die Kassetten nicht auf mehreren Geräten einsetzbar sein sollen (z. B. beim Austausch von Kassetten). Der Generator schwingt auf der doppelten Frequenz des Datentaktes, was im konkreten Fall 4800 Hz sind. Das Flip-Flop sichert ein exaktes Tastverhältnis von 1:1.

Durch das Gatter 3 werden in den Grundtakt Phasensprünge in Abhängigkeit vom Datenstrom eingebracht, welche aufzuzeichnen sind. Die Widerstände R1 sind so einzumessen, daß das verwendete Tonbandgerät noch nicht übersteuert. Bei Verwendung von Geräten mit automatischer Aussteuerung sollte der Pegel gewählt werden, der den maximalen Wiedergabepiegel bringt.

### Empfänger

Der Empfänger besteht aus dem als Trigger geschalteten Operationsver-

stärker IC4, der Flankenerkennung mit den Gattern 8 bis 11 und der Abtastung. Da der OPV mit nur einer Betriebsspannung arbeitet, wurde er eingangsseitig schwimmend geschaltet. Aus diesem Grund darf das Tonbandgerät außer über das Diodenkabel nicht mit dem Rechner masseseitig verbunden sein (auch nicht über Schutzkontakt). Am Ausgang von IC4 steht ein dem Bandsignal äquivalentes Rechtecksignal zur Verfügung. Das exakte Tastverhältnis ist mit dem Regler R2 einzustellen, wobei dazu ein Signal vom Magnetband zu verwenden ist, das keine Phasensprünge enthält (Dauer-ton). Mit der Brücke am Gatter 5 läßt sich ein eventuelles Invertieren des Signals durch das Bandgerät rückgängig machen. Gatter 5 kann auch durch einen Negator ersetzt werden, der im Bedarfsfall überbrückt wird. Der DCD-Eingang wird in dem anschließend vorgestellten Treiberprogramm nicht genutzt. An ihm läßt sich über Software das Laufen des Bandes nachweisen und auswerten.

Durch die Flankenerkennung wird gesichert, daß trotz der Phasensprünge je Periode ein Taktimpuls an den Monoflop gegeben wird. Durch das Sperren von Gatter 9 reagiert die Schaltung anfangs nur auf fallende Flanken, wodurch die Abtastung bei H-Pegel erfolgt, was dem Bandvorspann (Ruhezustand bei Asynchronbetrieb) entspricht. Durch diese Maßnahme treten keine Fehler beim Zuschalten des Bandes auf.

Die Abtastung der Daten erfolgt mit dem Monoflop IC7. Es ist in seiner Haltezeit auf 75% der Periodendauer des Sendetaktes einzustellen. Damit werden die Daten in der Mitte der Zeit zwischen den Flanken in IC6 übernommen.

Alle Einstellvorgänge sind sehr sorgfältig auszuführen, da sie wesentlich

die Datensicherheit mitbestimmen. Die SIO sollte dazu aus der Schaltung entfernt werden.

## **Treiberprogramme Magnetbandanschluß**

### **Initialisierung**

Durch den Baustein U856 ist eine Initialisierung der Schaltung erforderlich. Die exakten Daten sind der Tabelle T1 im Programm beziehungsweise [2] zu entnehmen. Es wurde folgender Modus im Kanal eingestellt:

- Asynchron ohne Interrupt
- 8 Bit Daten, Parität gerade, 2 Stoppbit
- Takt 1:1

Es ergeben sich damit auf dem Magnetband 12 Bit für ein aufzuzeichnendes Datenbyte.

Kanal B verbleibt im Einschaltreset.

### **Schreibmodul**

Das Schreibprogramm arbeitet im Abfragebetrieb. Es beginnt mit der Abfrage der Aufnahmebereitschaft des Bausteines für Daten. Ist diese vorhanden, wird das Zeichen geladen. Damit stehen etwa 5 ms (bei 2400 Bit/s) bis zum nächstmöglichen Laden zur Verfügung, in denen das folgende Zeichen aufbereitet werden kann.

### **Lesemodul**

Mit dem ersten Leseruf wird die SIO hardwaremäßig freigegeben (/RTS = High). Anschließend wird auf das ankommende Zeichen gewartet. Parallel zur Zeichenabfrage erfolgt die Auswertung des Fehlerstatus. Das Programm kann sich auf dem Bildschirm mit 3 Fehlerarten melden:

- P für Paritätsfehler
- O für Überlauf (wenn mehr als 3 Zeichen nicht abgeholt wurden)

R für Rahmenfehler (der Baustein vermißt die Stopppbit am Byteende). Der Fehlerstatus wird anschließend im Baustein gelöscht. Durch Fehler im einzelnen Byte ergeben sich normalerweise Fehler im gesamten Datenblock. Bei derartigen Blockprüfsummenfehlern erscheint (durch das Ladeprogramm, nicht durch den Treiber) auf dem Bildschirm die Endadresse des fehlerhaften Blockes. Der Fehler befindet sich dann innerhalb des davor liegenden Blockes, welcher beim verwendeten Lader 16 Byte groß ist.

### Die Betriebssystemfunktionen S und L

Die Aufzeichnung erfolgt in einem Format, das den Maschinencodeloehbändern für den K1520 ähnlich ist. Es werden aus dem zugewiesenen Speicherbereich Blöcke gebildet, welche folgenden Aufbau haben:

5 × 0FFH	Vorspann
00H	Start
BB	Blocklänge (ab Daten)
LL	Anfangsadresse <L>
HH	Anfangsadresse <H>
DD	Daten
...	16 ×
DD	
PS	Prüfsumme

Die Prüfsumme ergibt sich aus dem Komplement der Modulo-256-Addition der Werte von BB bis Datenende, d. h., die Summe aller Bytes ab BB bis einschließlich PS muß Null ergeben.

Alle Daten werden unverändert als 8-Bit-Werte aufgezeichnet. Der Endblock besitzt die Länge BB = 0. Der Standardblock hat eine Länge von 16 Datenbyte. Die Adresse im Endblock ist die Adresse des ersten freien Speicherplatzes nach dem Programm. Sie wird allerdings vom Leseprogramm nicht ausgewertet.

Die Magnetbandausgabe und -eingabe sind im Gegensatz zu allen anderen Pro-

grammen nicht abbrechbar. Deshalb bricht das Ladeprogramm ab, wenn es nach 48 empfangenen Zeichen noch keinen Blockkopf gefunden hat. Das Schreibprogramm kehrt von selbst zurück, wenn es die Aufzeichnung beendet hat.

Die Aufzeichnung von 1 KByte dauert etwa 10 Sekunden. Das Betriebssystem wurde getestet.

Entsprechende Programme finden Sie auf den Seiten 26 bis 35.

### Literatur

- [1] KLEIN, R.: Mikrocomputer Hard- und Softwarepraxis. - München, 1981. - S.86
- [2] Mikrorechnersystem der II. Leistungsklasse/SIO U 856. Technische Beschreibung. - Funkwerk Erfurt

Autor:

*Dr.-Ing. Gert Schönfelder*

Problemanalytiker im Rechenzentrum der Sektion Informationsverarbeitung der Ingenieurhochschule Dresden

```

0001          PN   SYS
0002          ;
0003          ;
0004          ; MINIBETRIEBSSYSTEM
0005          ;
0006          ;
0007          ;
0008          ;     ORG   00
0009          ;
0010 0000 C38602 START: JMP   ANF          ; SPRUNG UEBER RST-ADR.
0011          ;
0012          ;
0013          ;
0014          ; PLATZ FUER RESTART-RUFE
0015          ;
0016          ;
0017          ;-----
0018          ;
0019          ; RST 38 = SYSTEMRUECKKEHR
0020          ;
0021          ;     ORG   38H
0022 0038 C39102 JMP   GRUND
0023          ;
0024          ; PLATZ FUER ABSPRUNG NMI
0025          ;
0026          ;
0027          ;-----
0028          ;
0029          ; PLATZ FUER TREIBER VON TAST. UND VIDEO
0030          ;
0031          ;     ORG   280H
0032          ;
0033          ;-----
0034          ;
0035          ; ADRESSENVERBINDUNGEN ZU TAST UND VIDEO
0036          ;
0037 0280 C36D00 VIDEO: JMP   6DH
0038 0283 C36A00 TAST:  JMP   6AH
0039          ;
0040          ;
0041          ;
0042          ; S Y S T E M B E G I N N *****
0043          ;
0044 0286 316BFC ANF:   LD    SP,STAK          ; SP LADEN
0045 0289 3E08      LD    A,8
0046 028B 3203FC      LD    (F7),A          ; ECHO TASTATUR
0047 028E CDC803      CALL INIT          ; INIT RECHNER
0048          ;
0049          ; RUECKKEHRPUNKT
0050 0291 316BFC GRUND: LD    SP,STAK
0051 0294 3E1E      LD    A,1EH
0052 0296 CD8002      CALL VIDEO          ; NL
0053 0299 3E2A      LD    A,'*'
0054 029B CD8002      CALL VIDEO
0055          ; SYSTEMMELDUNG MIT *
0056 029E CD8302      CALL TAST
0057 02A1 F5          PUSH AF          ; RETTEN A
0058 02A2 CD8002      CALL VIDEO          ; ECHO BS
0059 02A5 3E20      LD    A,20H
0060 02A7 CD8002      CALL VIDEO          ; SPACE
0061 02AA F1          POP  AF          ; ALTER A
0062          ; AUSWERTUNG BEFEHLE
0063 02AB FE40      CMP   40H
0064 02AD 38E2      JRC  GRUND-#          ; KEIN BUCHST.
0065 02AF CBAF      RES  5,A
0066          ; GROSSBUCHST. = KLEINBU.
0067 02B1 FE52      CMP   'R'
0068 02B3 CAF702      JPZ  GRA          ; READ
0069 02B6 FE57      CMP   'W'
0070 02B8 CA3D03      JPZ  GRB          ; WRITE
0071 02BB FE47      CMP   'G'

```

```

0072 02B0 CAEF02 JPZ GRC ; GO
0073 02C0 FE45 CMP 'E'
0074 02C2 CAE102 JPZ GRD ; ECHO
0075 02C5 FE4C CMP 'L'
0076 02C7 CAAB04 JPZ GRE ; LOAD
0077 02CA FE53 CMP 'S'
0078 02CC CA1005 JPZ GRF ; SAVE
0079 02CF FE43 CMP 'C'
0080 02D1 CA1003 JPZ GRG ; CLEAR
0081 02D4 FE56 CMP 'U' ; VIDEO
0082 02D6 C29102 JPNZ GRUND
0083 ;
0084 ; TESTMODUS -----
0085 ;
0086 02D9 CDE103 GRH: CALL SYSTS
0087 02DC CD8002 CALL VIDEO
0088 02DF 18F8 JR GRH-#
0089 ;
0090 ; ECHO TASTATUR SETZEN -----
0091 ;
0092 02E1 CDEF03 GRD: CALL HTAST ; HEX-ZEI. HOLEN
0093 02E4 3203FC LD (F7),A
0094 02E7 38F8 JRC GRD-# ; FEHLER
0095 02E9 CD7C03 CALL ASCII
0096 02EC C39102 JMP GRUND
0097 ;
0098 ; START PROGRAMM (GO) -----
0099 ;
0100 02EF CD8C03 GRC: CALL ADRIN ; ADR HOLEN
0101 02F2 119102 LD IE,GRUND
0102 02F5 D5 PUSH DE
0103 02F6 E9 JMP (HL)
0104 ; ERGIBT CALL (HL)
0105 ; DIE RUECKKEHR ERFOLGT MIT .RET. ODER .RST 30H.
0106 ; UND ENDET AUF DER ADRESSE GRUND
0107 ;
0108 ;
0109 ;
0110 ;
0111 ; SPEICHER LESEN -----
0112 ;
0113 02F7 CD8C03 GRA: CALL ADRIN ; ANF.-ADR.
0114 02FA 0608 LD B,8
0115 02FC 3E1E LD A,1EH
0116 02FE CD8002 CALL VIDEO ; NL
0117 0301 CDA503 CALL VIDAD ; ADRESSE
0118 0304 2B DEC HL
0119 ;
0120 0305 CD0E04 GRAB: CALL BYINC ; BYTE AUSGEBEN
0121 0308 10FB DJNZ GRAB-# ; 8 MAL
0122 030A 23 INC HL
0123 030B CDE103 CALL SYSTS ; ENDE ??
0124 030E 18EA JR GRAA-# ; NAECHSTEN 8 BYTE
0125 ;
0126 ; SPEICHERLOESCHEN -----
0127 ;
0128 0310 CD8302 GRG: CALL TAST
0129 0313 FE4C CMP 'L'
0130 0315 C29102 JPNZ GRUND ; SICHERHEIT
0131 ; PRUEFUNG AUF .CL. , SONST ABGANG
0132 0318 CD8002 CALL VIDEO
0133 031B 3E20 LD A,20H
0134 031D CD8002 CALL VIDEO
0135 0320 CD8C03 CALL ADRIN
0136 0323 CD8002 CALL VIDEO
0137 0326 EB EX DE,HL ; <DE>=ANF
0138 0327 CD8C03 CALL ADRIN ; <HL>=END
0139 032A AF XOR A
0140 032B ED52 SBC HL,DE
0141 032D DA1804 JPC ERROR ; ANF > END
0142 0330 EB EX DE,HL ; <HL>=ENF
0143 0331 13 INC DE ; <DE>=ANZAHL

```

```

0144
0145 0332 3600 GRGA: LD M,0
0146 0334 23 INC HL
0147 0335 1B DEC DE
0148 0336 7A LD A,D
0149 0337 B3 OR E
0150 0338 20F8 JRNZ GRGA-#
0151 033A C39102 JMP GRUND
;
; SPEICHER SCHREIBEN -----
0154
0155 033D CD8C03 GRB: CALL ADRIN ; ANF.-ADR HOLEN
0156 0340 3E1E GRBA: LD A,1EH
0157 0342 CD8002 CALL VIDEO ; NL
0158 0345 CDA503 CALL UIDAD ; ADRESSE AUSG.
0159 0348 0604 LD B,4
;
0160
0161 034A CD0104 GRBB: CALL BYOUT ; ALTES BYTE
0162 034D 3E2D LD A,'-'
0163 034F CD8002 CALL VIDEO
;
0164
0165 0352 CDE103 CALL SYSTS ; ENDE ??
0166 0355 FE20 CMP 20H
0167 0357 280F JRZ GRBC-# ; UEBERGEHEN
0168 0359 FE11 CMP 11H
0169 035B 2819 JRZ GRBF-# ; 4 BYTE ZURUECK
0170 035D 0E00 LD C,0
0171 035F CD2F04 CALL BYINB ; REST EINGEBEN
0172 0362 CD8002 CALL VIDEO ; SPACE
0173 0365 71 LD M,C
0174 0366 1809 JR GRBD-#
;
0175
0176 0368 CD8002 GRBC: CALL VIDEO
0177 036B CD8002 CALL VIDEO
0178 036E CD8002 CALL VIDEO ; 3 X SPACE
0179 0371 23 GRBD: INC HL
0180 0372 10D6 DJNZ GRBB-#
0181 0374 18CA JR GRBA-#
;
0182
0183 0376 2B GRBF: DEC HL
0184 0377 2B DEC HL
0185 0378 2B DEC HL
0186 0379 2B DEC HL
0187 037A 18C4 JR GRBA-#
;
; -----
0188
0189
0190 ; WANDLUNG HEX IN ASCII UND AUSGABE
0191 ; <A> BLEIBT HEX
0192 ;
0193 037C F5 ASCII: PUSH AF
0194 037D E60F AND 0FH
0195 037F FE0A JMP 0AH
0196 0381 3802 JRC ASC-# ; 0...9
0197 0383 C607 ADD 7
0198 0385 C630 ASC: ADD 30H
0199 0387 CD8002 CALL VIDEO
0200 038A F1 POP AF
0201 038B C9 RET
;
; -----
0202
0203
0204 ; EINGABE ADRESSE IN <HL>
0205 ; WERT WIRD DURCHGESCHOBEN, D.H. BEI
0206 ; FEHLER 4 NULLEN EINGEBEN UND NEU BE-
0207 ; GINNEN
0208 ; ENDE DURCH SPACE,NL,CR,CU-BEWEGUNGEN
0209 ;
0210 038C 210000 ADRIN: LD HL,0
0211 038F CDE103 ADA: CALL SYSTS
0212 0392 FE21 CMP 21H
0213 0394 D8 RC ; ENDE
0214 ;
0215 0395 CDF203 CALL HEX

```

```

0216 0398 38F5 JRC ADA-# ; FEHLER
0217 039A 29 ADD HL,HL
0218 039B 29 ADD HL,HL
0219 039C 29 ADD HL,HL
0220 039D 29 ADD HL,HL
0221 039E 85 ADD L
0222 039F 6F LD L,A
0223 03A0 CD7C03 CALL ASCII ; ECHO AUF BS
0224 03A3 18EA JR ADA-#
0225
0226
-----
0227 ; ADRESSE UND 2X SPACE AUSGEBEN
0228 ; WERT IN HL
0229
VIDAD: LD (HILF),HL
0230 03A5 226DFC PUSH HL
0231 03A8 E5 HL
0232 03A9 216EFC LD HL,HILF+1
0233 03AC ED6F RLD
0234 03AE CD7C03 CALL ASCII
0235 03B1 ED6F RLD
0236 03B3 CD7C03 CALL ASCII
0237 03B6 2B DEC HL
0238 03B7 ED6F RLD
0239 03B9 CD7C03 CALL ASCII
0240 03BC ED6F RLD
0241 03BE CD7C03 CALL ASCII
0242 03C1 E1 POP HL
0243 03C2 3E20 LD A,20H
0244 03C4 CD8002 CALL VIDEO
0245 03C7 CD8002 CALL VIDEO
0246 03CA C9 RET
0247
; INITIALISIEREN -----
0248
INIT: LD A,0EH ; LOESCHEN BS
0250 03CB 3E0E CALL VIDEO
0251 03CD CD8002 LD HL,TXA ; SYS-MELDUNG
0252 03D0 218705 CALL TEXT
0253 03D3 CDD703
0254
0255 03D6 C9 RET
0256
; -----
0257 ; TEXTAUSGABE AB ADRESSE IN HL BIS ETX (03)
0258
TEXT: LD A,M
0260 03D7 7E CMP 3
0261 03D8 FE03 RZ ; ENDE
0262 03DA C8 CALL VIDEO
0263 03DB CD8002 INC HL
0264 03DE 23 JR TEXT-#
0265 03DF 18F6
0266
; -----
0267 ; TASTATURPROGRAMM MIT SYSTEMRUECKKEHR
0268 ; DURCH .SYS. ODER .ETX. -TASTEN
0269
SYSTS: CALL TAST
0270 03E1 CD8302 CMP 3
0271 03E4 FE03 JPZ GRUND ; ETX
0272 03E6 CA9102 CMP SYS
0273 03E9 FE8D RNZ ; KEIN ENDE
0274 03EB C0 JMP GRUND ; SYS-TASTE
0275 03EC C39102
0276
; SIEHE BEMERKUNGEN BEI SYSUD
0277
; -----
0278 ; WANDLUNG /1 TASTE ASCII IN HEX
0279 ; ( EINGABE HEXWERTE MIT KONVERTIERUNG)
0280
HTAST: CALL SYSTS
0281 03EF CDE103 ; --> NUR WANDLUNG
0282
0283
0284 03F2 E67F HEX: AND 7FH
0285
0286

```

```

0287 03F4 D630 SUB 30H
0288 03F6 D8 RC ; FEHLER CY=1
0289 03F7 FE0A CMP 0AH
0290 03F9 3F CCF ; 0...9 / CY=0
0291 03FA D0 RMC
0292 03FB D607 SUB 7
0293 03FD FE10 CMP 10H
0294 03FF 3F CCF
0295 0400 C9 RET
0296 ; CV=1 FEHLERHAFTER WERT
0297 ; CV=0 RICHTIGER WERT
0298 ;
0299 ;
0300 ; BYTE AUS SPEICHERZELLE <<HL>> AUSGEBEN
0301 ;
0302 0401 7E BYOUT: LD A,M
0303 0402 07 RLCA
0304 0403 07 RLCA
0305 0404 07 RLCA
0306 0405 07 RLCA
0307 0406 CD7C03 CALL ASCII
0308 0409 7E LD A,M
0309 040A CD7C03 CALL ASCII
0310 040D C9 RET
0311 ;
0312 ;
0313 ; BYTEAUSGABE MIT ERHOEHEN HL UND SPACE
0314 ;
0315 040E 23 BYINC: INC HL
0316 040F CD0104 BYSP: CALL BYOUT
0317 0412 3E20 LD A,20H
0318 0414 CD8002 CALL VIDEO
0319 0417 C9 RET
0320 ;
0321 ;
0322 ; FEHLERAUSGABE
0323 ;
0324 0418 3E3F ERROR: LD A,'?'
0325 041A CD8002 CALL VIDEO
0326 041D CD8002 CALL VIDEO
0327 0420 E1 POP HL
0328 0421 CDA503 CALL VIDAD
0329 0424 C39102 JMP GRUND
0330 ; ES WIRD DIE ADRESSE NACH DEM FEHLERAB-
0331 ; SPRUNG AUSGEBEN
0332 ;
0333 ;
0334 ; BYTE EINGEBEN (DURCHSCHIEBEPINZIP)
0335 ;
0336 0427 0E00 BYIN: LD C,0
0337 0429 CDE103 BYINA: CALL SYSTS
0338 042C FE20 CMP 20H
0339 042E C8 RZ ; ENDE
0340 042F CDF203 BYINB: CALL HEX
0341 0432 38F5 JRC BYINA-# ; FEHLER
0342 0434 CB21 SLA C
0343 0436 CB21 SLA C
0344 0438 CB21 SLA C
0345 043A CB21 SLA C
0346 043C B1 OR C
0347 043D 4F LD C,A
0348 043E CD7C03 CALL ASCII ; ECHO BS
0349 0441 18E6 JR BYINA-#
0350 ;
0351 ;
0352 ;
0353 ; INIT SIO FUER KASSETTENANSCHLUSS
0354 ; IO-KANAL 90H = DAT A
0355 ; 91H = CMD A
0356 ; 92H = DAT B
0357 ; 93H = CMD B

```



```

0358 ;
0359 ; INIT NUR KANAL A
0360 ;
0361 0443 214F04 ;STOIN: LD HL,T1
0362 ;
0363 ; PGM ZUR IO-PORT-INITIALISIERUNG
0364 0446 7E ;
0365 0447 B7 ;
0366 0448 C8 ; ENDE
0367 0449 4F ; TORADR. IN <C>
0368 044A 23 ;
0369 044B EDA3 ; INHALT AUSGEBEN
0370 044D 18F7 ;
0371 ;
0372 ; INIT-TABELLE
0373 044F 91 ;
0374 0450 18 ; T1: DB 91H ; REG 0
0375 0451 91 ; DB 18H ; KANAL-RESET
0376 0452 04 ; DB 91H
0377 04 7 91 ; DB 4 ; REG 4
0378 0454 0F ; DB 91H
0379 0455 91 ; DB 0FH ; 2 STOPBIT/PARITY EVEN
0380 0456 05 ; DB 91H
0381 0457 91 ; DB 5 ; REG 5
0382 0458 6A ; DB 91H
0383 ; SENDER FREIGEBEN / EMPFANGEN-HW RUECKSETZEN
0384 0459 91 ; DB 6AH ; 8 BIT SENDEN
0385 045A 03 ; DB 91H
0386 045B 91 ; DB 3 ; REG 3
0387 045C C0 ; DB 91H
0388 045D 91 ; DB 0C0H ; 8 BIT EMPFANGEN
0389 045E 30 ; DB 91H
0390 045F 0000 ; DB 30H ; ERROR-RESET
0391 ; DA 0 ; ENDE TABELLE
0392 ;
0393 ;
0394 ;
0395 ; BANDLESEN - WERT IN <A>
0396 ;
0397 MLES: LD A,3
0398 0463 D391 OUT 91H
0399 0465 3EC1 LD A,0C1H
0400 0467 D391 OUT 91H
0401 0469 3E05 LD A,5
0402 046B D391 OUT 91H
0403 046D 3E62 LD A,62H
0404 046F D391 OUT 91H
0405 ; FREIGABE HW NACH ZWANGSSYNCRONISATION
0406 ;
0407 0471 DB91 MLES1: IN 91H
0408 0473 E601 AND 1
0409 0475 28FA JRZ MLES1-# ; WARTEN AUF RDY
0410 ;
0411 0477 3E01 ; LD A,1
0412 0479 D391 OUT 91H
0413 047B DB91 IN 91H ; FEHLERSTATUS LESEN
0414 047D E670 AND 70H
0415 047F 2003 JRNZ MLES2-# ; FEHLER ->
0416 0481 DB90 IN 90H ; ZEICHEN LESEN
0417 0483 C9 RET
0418 ;
0419 ; FEHLERAUSWERTUNG
0420 0484 CB67 MLES2: BIT 4,A
0421 0486 2010 JRNZ MLES4-#
0422 0488 CB6F BIT 5,A
0423 048A 2010 JRNZ MLES5-#
0424 048C 3E52 LD A,'R' ; RAHMENFEHLER
0425 048E CD8002 MLES3: CALL VIDEO
0426 0491 3E30 LD A,30H
0427 0493 D391 OUT 91H ; FEHLER-RESET
0428 0495 DB90 IN 90H ; WERT LESEN

```

```

0429      0497 C9                RET
0430
0431      0498 3E50             MLES4: LD  A,'P'                ; PARITAETS-FEHLER
0432      049A 18F2             JR      MLES3-#
0433
0434      049C 3E4F             MLES5: LD  A,'0'                ; OVERFLOW
0435      049E 18EE             JR      MLES3-#
0436
0437
0438
0439
0440
0441
0442
-----
0443      04A0 F5                MWRIT: PUSH AF
0444      04A1 DB91             MWR:   IN   91H
0445      04A3 E604             AND   4
0446      04A5 28FA             JRZ   MWR-#                ; SIO NICHT FREI
0447      04A7 F1                POP  AF
0448      04A8 D390             OUT  90H                ; ZEICHEN RAUS
0449      04AA C9                RET
0450
-----
0451
0452
0453
0454      04AB CD4304           GRE:   CALL SIOIN                ; INIT SIO
0455      04AE 219805           LD    HL, TXB
0456      04B1 CDD703           CALL  TEXT
0457
0458      04B4 CDE103           GREA:  CALL SVSTS                ; STOP ?
0459      04B7 FE4A             CMP   'J'
0460      04B9 20F9             JRNZ  GREA-#                ; KEIN JA
0461      04BB CD8002           CALL  VIDEO
0462
0463      04BE 0E30             GREE:  LD    C, 30H
0464      04C0 0605             GREB:  LD    B, 5
0465      04C2 0D                DEC   C
0466      04C3 CA1804           JPZ   ERROR                ; KEIN KOPF DA
0467
0468      04C6 CD6104           GREC:  CALL MLES
0469      04C9 FEFF             CMP   0FFH
0470      04CB 20F3             JRNZ  GREB-#                ; KEINE 5X FFH
0471      04CD 10F7             DJNZ  GREC-#
0472      04CF CD6104           CALL  MLES
0473      04D2 FE00             CMP   0
0474      04D4 20EA             JRNZ  GREB-#                ; KEINE 0
0475
0476      04D6 CD6104           ; KOPF ERKANNT
0477      04D9 47                CALL  MLES
0478      04DA 4F                LD    B,A
0479      04DB B1                LD    C,A
0480      04DC CA9102           OR    C
0481
0482      04DF CD6104           JPZ   GRUND                ; ENDE (LAENGE =0)
0483
0484      04E2 6F                CALL  MLES
0485      04E3 81                LD    L,A
0486      04E4 4F                ADD  C
0487      04E5 CD6104           LD    C,A
0488      04E8 67                CALL  MLES
0489      04E9 81                LD    H,A
0490      04EA 4F                ADD  C
0491
0492      04EB CD6104           LD    C,A
0493
0494      04EE 77                ; EINLESEN DER EIGENTLICHEN DATEN
0495      04EF 81                GRED:  CALL MLES
0496      04F0 4F                LD    M,A
0497      04F1 23                ADD  C
0498      04F2 10F7             LD    C,A
0499      04F4 CD6104           INC  HL
0500      04F5 4F                DJNZ  GRED-#                ; N BYTE EINLESEN
0501
0502      04F7 CD6104           CALL  MLES

```

```

0500      04F7  81          ADD  C
0501      04F8  28C4       JRZ  GREE-#          ; PSUM O.K.
0502
0503      ; PRUEFSUMMENFEHLER
0504      ; ADRESSE NACH FEHLERHAFTEM BLOCK ANZEIGEN
0505      04FA  CDA503     CALL UIDAD
0506      04FD  3E1E       LD   A,1EH
0507      04FF  CD0002    CALL VIDEO
0508      0502  18BA       JR   GREE-#
0509
0510
0511      ; -----
0512      ; ABRUCHABFRAGE BEI BS-AUSGABE
0513
0513      0504  F5          SYSVD: PUSH AF
0514      0505  DB0F       IN   0FH          ; TASTATUR LESEN
0515      0507  FEFD       CMP  ABR
0516      0509  CA9102     JPZ  GRUND          ; SYS-TASTE
0517      050C  F1         POP  AF
0518      050D  C30002    JMP  VIDEO
0519
0520      ABR:  EQU  0FDH
0521
0522      ; ABR = WERT, DER AM TASTENPORT ANLIEGT, WENN
0523      ; SYS-TASTE GEDRUECKT WIRD (EINSCHLIESS-
0524      ; LICH STEUERBITS ).
0525
0526      SYS:  EQU  8DH
0527
0528      ; SYS = WERT, DER SICH NACH DEM RUF DES TASTATUR-
0529      ; TREIBERS ERGIEBT, WENN DIE SYS-TASTE GE-
0530      ; DRUECKT WURDE.
0531
0532
0533      ; !! CODIERUNG MUSS ENTSPRECHEND DER TASTENBELEGUNG
0534      ; GEANDERT WERDEN !!
0535      ; EINGETRAGEN IST DIE TASTE MIT DER MATRIXPOSITION 3DH.
0536
0537
0538      ; -----
0539      ; PGM AUF MAGNETBAND AUSGEBEN
0540
0540      0510  CD4304     GRF:  CALL SIOIN          ; INIT SIO
0541      0513  CD8C03     CALL ADRIN          ; VON ADR.
0542      0516  3E20       LD   A,20H
0543      0518  CD8002    CALL VIDEO
0544      051B  E5         PUSH HL
0545      051C  CD8C03     CALL ADRIN          ; BIS ADR.
0546      051F  E5         PUSH HL
0547      0520  219805    LD   HL,7XB
0548      0523  CDD703    CALL TEXT
0549
0550      ; MARTEN AUF START BAND
0550      0526  CDE103     GRFA: CALL SYSYS          ; ENDE ?
0551      0529  FE4A       CMP  'J'
0552      052B  20F9       JRNZ GRFA-#
0553      052D  CD8002    CALL VIDEO
0554
0555      ; BAND LAEUFT !
0556      0530  E1         POP  HL          ; BIS
0557      0531  D1         POP  DE          ; VON
0558      0532  AF         XOR  A
0559      0533  ED52       SBC  HL,DE
0560      0535  23         INC  HL          ; DE=ANZAHL
0561      0536  EB         EX   DE,HL          ; HL=VON
0562
0563      ; UORSPANN AUSGEBEN
0564      0537  3EFF       GRFB: LD   A,0FFH
0565      0539  0605       LD   B,5
0566      053B  CDA004     GRFC: CALL MWRIT
0567      053E  10FB       DJNZ GRFC-#          ; 5X 0FFH
0568      0540  AF         XOR  A
0569      0541  CDA004     CALL MWRIT          ; NULL
0570

```

```

0571      ; KOPFDATEN AUSGEBEN
0572      LD      A,0
0573      CMP      D
0574      JRNZ     GRFD-#      ; >256 BYTE
0575      LD      A,E
0576      CMP      10H
0577      JRNC     GRFD-#      ; >16 BYTE
0578      LD      B,A          ; LAENGE = REST
0579      JR      GRFF-#
0580      LD      B,10H        ; LAENGE = 16 BYTE
0581      GRFF:   LD      A,B
0582      OR      B
0583      JRZ     GRFE-#      ; ENDEBLOCK
0584      CALL    MWRIT        ; LAENGE
0585      LD      A,L
0586      LD      C,B
0587      CALL    MWRIT
0588      ADD     C
0589      LD      C,A
0590      LD      A,H
0591      CALL    MWRIT
0592      ADD     C
0593      LD      C,A          ; ADRESSE GESENDET
0594
0595      ; DATENBLOCK AUFZEICHNEN
0596      GRFG:   LD      A,M
0597      CALL    MWRIT        ; BYTE RAUS
0598      ADD     C
0599      LD      C,A          ; PSUM
0600      INC     HL
0601      DEC     DE
0602      DJNZ    GRFG-#
0603      ; PSUM AUFZEICHNEN
0604      XOR     A
0605      SUB     C
0606      CALL    MWRIT
0607      JMP     GRFB
0608
0609      ; ENDEBLOCK
0610      GRFE:   CALL    MWRIT        ; LAENGE =0
0611      LD      A,L
0612      CALL    MWRIT        ; L
0613      LD      A,H
0614      CALL    MWRIT        ; H
0615      JMP     GRUND
0616
0617
0618      ; -----
0619      ; TEXTE
0620
0621      TXA:    DB      360,360,360
0622      058A    DB      'MINI-SYS-1'
0623      058E    2D535953
0624      0592    2D31
0625      0594    1E1E1E      DB      360,360,360
0626      0597    03          DB      3
0627
0628      TXB:    DB      0EH,360
0629      0598    DB      'BAND S T A R T E N ! !'
0630      059A    42414E44
0631      059E    20205320
0632      05A2    54204120
0633      05A6    52205420
0634      05AA    45204E20
0635      05AE    212021
0636      05B1    1E1E      DB      360,360
0637      05B3    4C414555    DB      'LAEUFT BAND ? (JA?) : '
0638      05B7    46542042
0639      05BB    414E4420
0640      05BF    3F20284A
0641      05C3    413F2920
0642      05C7    3A

```

```

0630      05C8 03          ;          DB      3
0631          ;
0632          ;
0633          ; -----
0634          ;
0635          ;          R A M
0636          ;
0637          ;          SYSTEMSPEICHER AB ADRESSE FC00H
0638          ;
0639          ;
0640          ;          ORG      0FC00H
0641          ;
0642          ;          ZELLEN VON BILDSCHIRM
0643      FC00      CUS0:   BER      2
0644      FC02      BLI:   BER      1
0645          ;
0646          ;          ZELLEN VON TASTATUR
0647      FC03      F7:   BER      1
0648      FC04      F4:   BER      1
0649      FC05      ZEIT1: BER      2
0650          ;
0651          ;          ZELLEN VON B-SYS
0652      FC07      BER      100
0653      FC6B      STAK:  BER      2          ; STACK
0654      FC6D      HILF:  BER      2          ; HILFSZELLE (UIDAD)
0655          ;
0656          ;
0657          ;
0658          ;
0659          ;          END
0660      KEINE SYNTAX-FEHLER CRAS 4200-K1520

```

## Rechentechische Begriffe für den Laien erklärt

**Monitor** Mit Monitor werden in der Computertechnik zwei verschiedene Dinge bezeichnet. Einmal handelt es sich dabei um eine Anzeigeeinheit (Display), einen Bildschirm, für ein Computersystem. Die zweite Bedeutung des Begriffs wird insbesondere durch den Begriff Monitorprogramm deutlich. Dabei wird mit Monitor ein Dienstprogramm bezeichnet, das den Mensch-Maschine-Dialog über Tastatur und Bildschirm organisiert und in der Regel Bestandteil des Betriebssystems ist.

**Plotter<sup>1)</sup>** Ein Plotter ist ein spezielles Ausgabegerät für Grafiken, die mit einem Computer erstellt wurden. Die Aufzeichnung erfolgt dabei durch einen Stift, der durch Schrittmotoren über ein Papier gesteuert wird. Bei mehrfarbigen Plottern werden nicht nur ein, sondern mehrere Farbstifte angesteuert.

<sup>1)</sup> Dieser Begriff wurde BÜCKNER, U.: Kleincomputer leichtverständlich. - Leipzig, 1986 entnommen



Am Beispiel der Zerlegung einer Zahl in ihre Primfaktoren wird gezeigt, wie ein einmal erarbeitetes und getestetes Programm durch zusätzliche Überlegungen verbessert werden kann.

Dieser Beitrag soll auch als Anregung für eigenes Programmieren dienen.

Um zu erreichen, daß die Programme auf Computern mit unterschiedlichen BASIC-Dialekten ohne wesentliche Änderungen laufen, werden nur solche Anweisungen benutzt, die in fast allen BASIC-Dialekten vorkommen. Weil z.B. nicht immer über die komplette Anweisung `IF ... THEN ... ELSE` verfügt werden kann, wird `ELSE` nicht benutzt. Argumente von Funktionen werden in Klammern gesetzt, obwohl das nicht überall erforderlich ist.

Außerdem können die Programme auch für programmierbare Taschen- oder Tischrechner, die nicht unter BASIC arbeiten, umgeschrieben werden, wenn man sich dabei an den Struktogrammen orientiert.

## Grundlagen

Zunächst ist zu klären, was wir unter einer Primzahl verstehen:

Eine Primzahl ist eine Zahl  $Z > 1$ , die nur durch 1 und durch sich selbst ohne Rest teilbar ist.

Jede natürliche Zahl  $Z > 1$  ist entweder eine Primzahl, oder sie kann so

in Primfaktoren zerlegt werden, daß sich als Produkt aller Primfaktoren wieder die Zahl  $Z$  ergibt.

Beispielsweise kann man die natürliche Zahl 252 in Faktoren zerlegen:

$$252 = 4 \cdot 9 \cdot 7$$

Während 7 bereits eine Primzahl ist, sind die Faktoren 4 und 9 noch keine Primzahlen, denn beide können in weitere Faktoren zerlegt werden. Diese Zerlegung führt zu

$$252 = 2 \cdot 2 \cdot 3 \cdot 3 \cdot 7$$

$$252 = 2^2 \cdot 3^2 \cdot 7$$

Die Faktoren 2, 3 und 7 sind Primzahlen, und gleichzeitig haben wir die Zerlegung der Zahl 252 in ihre Primfaktoren ermittelt.

Bei der natürlichen Zahl 127 ist eine derartige Zerlegung nicht möglich. Deshalb ist 127 eine Primzahl.

## Aufgabe

Es ist ein Programm zu entwickeln, welches für eine gegebene Zahl  $Z$  feststellt, ob diese Zahl eine Primzahl ist. Für den Fall, daß  $Z$  keine Primzahl ist, sind ihre Primfaktoren anzugeben. Ist jedoch  $Z$  selbst eine Primzahl, dann soll  $Z$  durch »IST EINE PRIMZAHL« kenntlich gemacht werden.

## Lösungsalgorithmus

Am einfachsten ist es, die Zahl  $Z$  der Reihe nach durch alle natürlichen Zah-

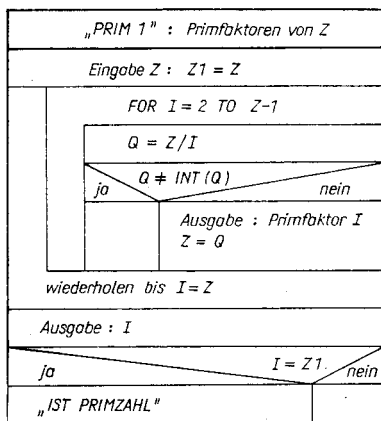


Bild 1. Struktogramm zur ersten Lösung

len  $I = 2, 3, \dots, Z - 1$  zu dividieren und jedesmal zu prüfen, ob solche Divisionen ohne Rest möglich sind. Existiert keine Zahl  $I$ , für die der Quotient

$$Q = \frac{Z}{I}$$

eine ganze Zahl ist, dann muß  $Z$  eine Primzahl sein.

Gibt es jedoch einen ganzzahligen Quotienten  $Q$ , dann ist  $Z = Q \times I$ , und  $I$  ist der erste Primfaktor. Weitere Primfaktoren findet man, falls sie existieren, wenn jetzt anstelle von  $Z$  die Zahl  $Q$  auf Primfaktoren untersucht

```

1 "PRIM 1"
10 PRINT "PRIMFAKTOREN VON Z"
20 INPUT "Z = "; Z: Z1=Z
30 PRINT "Z = "; Z
40 PRINT "Z = ";
50 FOR I=2 TO Z-1
60 Q=Z/I
70 IF Q<>INT(Q) THEN 100
80 PRINT I;"*";
90 Z=Q: GOTO 50
100 NEXT I
110 IF I=Z THEN PRINT I;
120 IF I=Z1 THEN PRINT " IST EINE PRIMZAHL"
130 END

```

wird. Setzt man dieses Verfahren so lange fort, bis der letzte Quotient  $Q$  eine Primzahl ist, dann hat man die Primfaktoren von  $Z$  ermittelt.

Weil  $I$  jeweils um 1 zunehmen soll, bietet sich auf den ersten Blick eine FOR-NEXT-Schleife mit der Schrittweite 1 an. Das folgende Programm soll so aufgebaut werden. Ob diese Lösung noch zu verbessern ist, wird später untersucht.

### Das erste Programm

Der geplante Ablauf wird aus dem Struktogramm des Bildes 1 deutlich.

Sobald ein Primfaktor  $I$  gefunden ist, wird die FOR-NEXT-Schleife abgebrochen und so lange mit der neuen Zahl  $Z = Q$  wieder aufgebaut, bis  $I = Z$  ist. Um zu vermeiden, daß bei einer Primfaktorenzerlegung der letzte Faktor als Primzahl gekennzeichnet wird, muß am Ende  $I$  noch einmal mit der ursprünglich eingegebenen Zahl  $Z$ , die jetzt  $Z1$  heißt, verglichen werden. In Bild 2 ist das Listing dieses Programms wiedergegeben. Nach dem Start des Programms mit RUN ENTER bzw. mit RUN RETURN erscheint auf dem Bildschirm

PRIMFAKTOREN VON Z  
Z = ?

Nach der Eingabe einer Zahl  $Z$  mit RETURN bzw. ENTER erscheint die Lösung.

Je nachdem, wie groß die Zahl ist und ob sie durch eine Zerlegung rasch auf kleinere Zahlen reduziert werden kann, ist die Lösung nach einigen Sekunden oder erst nach mehreren Minuten gefunden.

Die folgenden zwei Beispiele zeigen den Bildschirminhalt nach der Lösung.

Bild 2. Programm »PRIM 1«

## 1. BEISPIEL: $Z = 54321$

Bildschirminhalt:

PRIMFAKTOREN VON  $Z$

$Z = ? 54321$

$Z = 54321$

$Z = 3 \times 19 \times 953$

## 2. BEISPIEL: $Z = 9721$

Bildschirminhalt:

PRIMFAKTOREN VON  $Z$

$Z = ? 9721$

$Z = 9721$

$Z = 9721$  IST EINE PRIMZAHL

Weitere Testbeispiele, die der Leser auch für den Test eigener Programme verwenden kann, und deren Primfaktorenzerlegungen enthält Tabelle 1 (s. S. 40).

### Wir machen das Programm schneller

Vor dem Bildschirm oder dem Drucker zu sitzen und zu warten, bis die Lösung endlich erscheint, kann recht lästig sein, wenn minutenlang nichts geschieht. Das Gefühl der Befriedigung über die ersten richtigen Lösungen und den korrekten Lauf des Programms verliert sich dabei bald. Man möchte den Ablauf beschleunigen.

Überlegen wir deshalb, ob unser Programm noch zu viele überflüssige Berechnungen ausführt. Vielleicht gelingt es auch, die relativ langsame FOR-NEXT-Schleife zu vermeiden.

Im Programm »PRIM 1« wird für alle  $I < Z$  die Teilbarkeit untersucht. Das ist eigentlich nicht notwendig. Nehmen wir an, die Zahl  $Z$  läßt sich in wenigstens zwei Faktoren zerlegen, die mit  $I_1$  und  $I_2$  bezeichnet werden, dann gilt:

$$Z = I_1 \times I_2.$$

Von diesen beiden Faktoren wird einer kleiner als der andere sein. Im äußersten Falle sind beide gleich.

Wenn jedoch

$$I_1 = I_2 = I$$

ist, gilt

$$Z = I \times I = I^2$$

und

$$I = \sqrt{Z}.$$

Das heißt, einer der Faktoren, in die sich  $Z$  zerlegen läßt, ist höchstens gleich  $\sqrt{Z}$ . Er kann natürlich auch kleiner sein. Existiert jedoch kein Primfaktor  $I \leq \sqrt{Z}$ , dann kann auch kein Primfaktor  $> \sqrt{Z}$  existieren, und  $Z$  ist eine Primzahl.

Wir müssen also gar nicht alle Zahlen von 2 bis  $Z - 1$  daraufhin prüfen, ob sie Teiler von  $Z$  sind. Es genügt, anstelle der oberen Grenze  $Z - 1$  die obere Grenze  $\sqrt{Z}$  zu setzen.

Im 2. Beispiel mußte das Programm »PRIM 1« fast 10000 Quotienten bilden und prüfen, ob sie ganzzahlig sind. Verwenden wir  $\sqrt{Z}$  als obere Grenze, dann sind nur noch etwa 100 Rechnungen erforderlich. Wir sparen fast 99% der Rechenzeit ein!

Für ein solches Programm müßten auch die Tests in den Zeilen 110 und 120 verändert werden. Dem Leser, der sich im Programmieren üben will, sei dieser Weg empfohlen.

Allerdings ist dieser Weg nicht ganz so problemlos, wie es den Anschein hat. Es gibt Gefahren. Der Computer liefert für die Quadratwurzel in der Regel keine genauen Werte. Testen Sie z. B.

PRINT SQR (9721 \* 9721) — 9721, so erhalten Sie nicht den richtigen Wert Null, sondern eine, wenn auch kleine, von Null verschiedene Zahl. Das kann zu Fehlern im Programmlauf führen.

Wir umgehen die durch den Näherungswert für die Wurzel entstandene Fehlerquelle ganz einfach, wenn wir nicht mit



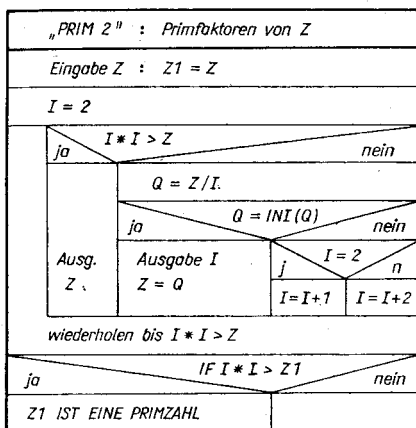


Bild 3. Struktogramm der verbesserten Lösung

der Bedingung  $I \leq \sqrt{Z}$  arbeiten, sondern diese Bedingung quadrieren:

$$I * I \leq Z$$

Der Computer multipliziert ganze Zahlen exakt ohne Näherungsfehler.

Zur Verkürzung der Rechenzeit bietet sich noch eine weitere Überlegung an: Primfaktoren größer als 2 sind immer ungerade Zahlen. Wir dividieren deshalb zunächst die Zahl Z oder Q so lange durch 2, bis feststeht, daß die verbleibende Zahl Q nur noch ungerade sein kann. Dann wird I von 2 auf 3 um 1 erhöht, und die Zahl 3 wird daraufhin geprüft, ob sie Primfaktor ist. Alle weiteren Rechnungen werden

```

1 "PRIM 2"
10 PRINT "PRIMFAKTOREN VON Z"
20 INPUT "Z = "; Z; Z1=Z: PRINT "Z = ";
30 I=2
40 IF I*I>Z THEN 90
50 Q=Z/I
60 IF Q=INT(Q) THEN PRINT I;"*"; Z=Q: GOTO 40
70 IF I=2 THEN LET I=I+1: GOTO 40
80 I=I+2: GOTO 40
90 PRINT Z;
100 IF I*I>Z1 THEN PRINT " IST EINE PRIMZAHL"
110 END

```

Bild 4. Programm »PRIM 2«

mit der Schrittweite 2 ausgeführt, indem I immer auf I + 2 erhöht wird. Die Rechnung ist abgeschlossen, sobald  $I * I > Z$  ist.

Einen Überblick über dieses Vorgehen zeigt das Struktogramm von Bild 3.

Das Listing des neuen Primfaktorenprogramms »PRIM 2« gibt das Bild 4 wieder.

Ein Test des Programms mit den Werten der beiden Beispiele zeigt deutlich, wieviel schneller die neue Programmvariante ist.

Nachfolgend wird der Bildschirminhalt für die beiden Beispiele wiedergegeben:

1. BEISPIEL: Z = 54321  
PRIMFAKTOREN VON Z  
Z = ? 54321  
Z = 3 \* 19 \* 953
2. BEISPIEL: Z = 9721  
PRIMFAKTOREN VON Z  
Z = ? 9721  
Z = 9721 IST EINE PRIMZAHL

Schon bei diesen Beispielen merkt man, daß sich unsere Überlegungen gelohnt haben; das Programm »PRIM 2« ist deutlich schneller als das Programm »PRIM 1«.

In der Tabelle 1 sind für einige Beispiele die Rechenzeiten und die erzielten Einsparungen angegeben.

Bei verschiedenen Computern können die Laufzeiten von den in der Tabelle angegebenen Zeiten abweichen. Einige Computer sind schneller, andere langsamer. Die prozentuale Einsparung an Rechenzeit wird jedoch unabhängig vom Rechner immer in den angegebenen Größenordnungen liegen.

Tabelle 1. Beispiele mit Lösungen und Vergleich der Rechenzeiten

Beispiel und Ergebnis	Rechenzeit in Sekunden für SANYO VZ 100		Einsparung in %
	PRIM 1	PRIM 2	
Z = 123456 Z = 2×2×2×2×2×2×2×3×643	8,8	0,7	92
Z = 1234567 Z = 127×9721	122	1,9	98
Z = 12345678 Z = 2×3×3×47×14593	180	2,0	99
Z = 654321 Z = 3×218107	2660	5,6	99,8
Z = 54321 Z = 3×19×953	14,2	0,8	94
Z = 127 IST EINE PRIMZAHL	1,9	0,2	89
Z = 9721 IST EINE PRIMZAHL	119,6	1,4	98,8

Es lohnt sich stets, und zwar nicht nur bei Primfaktoren, langsam laufende Programme daraufhin zu untersuchen, ob sie nicht schneller zu machen sind.

### Grenzen der Möglichkeiten

Mit den hier benutzten Algorithmen kann man natürlich die Genauigkeit des Computers nicht erhöhen. Im Normalfall können Zahlen in ihre Primfaktoren zerlegt werden, die einige Stellen mehr haben, als die Anzahl der angezeigten Stellen beträgt. Zeigt Ihr Computer z. B. maximal sechs Ziffern an, dann werden auch noch Primzahlen mit acht oder neun Ziffern erkannt.

Gibt man jedoch zu große Zahlen ein, dann rundet der Computer bereits unmittelbar nach der Eingabe die Zahl Z, und anstelle der eingegebenen Zahl wird die gerundete Zahl in Primfaktoren zerlegt. Man erhält falsche Ergebnisse. Besteht die Möglichkeit, Zahlen mit doppelter Genauigkeit zu definieren, dann läßt sich die Anwendungsgrenze für beide Programme nach oben verschieben.

Autor:  
*Dr.-Ing. Peter Fischer*

Fachgruppenleiter an der  
Ingenieurschule für Schwermaschinenbau  
Roßwein

---

# Computerspiele – mehr als eine Spielerei



Durch die Möglichkeiten der Mikroelektronik haben Fernsehspiele eine große Verbreitung gefunden. Dabei muß der Begriff »Fernsehspiele« bereits wieder mit Vorsicht eingesetzt werden, da schon Ausführungen in Taschenrechnerformat mit eigenem LCD-Display existieren.

Die Fernsehspiele, genauer eigentlich, die auf dem Fernschirmschirm dargestellten Spiele, sollte man nochmals in zwei Gruppen aufteilen. Die eine Gruppe, die ich weiterhin Fernsehspiele nennen möchte, stellen eine mehr oder weniger komplizierte Zeitablaufsteuerung dar (wie z. B. Tennis). Es ergeben sich damit im wesentlichen Geschicklichkeitsspiele, welche den Nachteil haben, daß sie ihren Reiz verlieren, wenn man die nötigen Reaktionen beherrscht. Die andere Gruppe, die Computerspiele, besitzen als Kernstück einen Mikrorechner, Taschenrechner oder spezielle Prozessoren. Es ergibt sich damit die Möglichkeit, den Spieler gegen die »Intelligenz« des Rechners antreten zu lassen. Durch die Leistungen des Rechners, beginnend mit der Bildung von Zufallszahlen, steigt die Vielfalt der möglichen Spielverläufe ins Unermeßliche, wodurch man bei den wenigsten Spielen behaupten kann, sie zu beherrschen. Vertreter dieser Gruppe erstrecken sich vom Zahlenratespiel bis hin zum Schachcomputer. Es ist

auch denkbar, daß der Computer als Spielmeister für eine Gruppe fungiert, z. B. in einem Roulettespiel.

Bevor ein Spiel aus dieser zweiten Gruppe vorgestellt wird, noch einige allgemeine Gedanken zu diesen. Die Programmierung derartiger Spiele hat für den Profi wie für den Amateur einige Vorteile. So kann man mit Hilfe von Spielprogrammen sehr schnell eine Programmiersprache erlernen. Das liegt daran, daß Spiele meist sehr ausgabeintensiv sind. Somit ist es möglich, an Hand der stattgefundenen oder fehlenden Ausgaben einen Fehler im Programm sehr genau einzukreisen. Diese Methode der Fehlersuche erfordert auch kaum Hilfsmittel, die vor allem dem Amateur meist nicht zur Verfügung stehen. Ein weiterer Gesichtspunkt ist das Training im logischen Denken, was sowohl beim Programmieren als auch im Spiel von Nöten ist; und nicht zuletzt kann man Spiele umschreiben oder neue entwickeln, wenn man die alten nicht mehr spannend genug findet.

Auf S. 44 bis 55 zu diesem Thema ein konkretes Beispiel, den TURM VON HANOI. Es handelt sich dabei um ein Kombinatorikspiel, das sehr verbreitet ist. Das Spielbild, welches der Rechner aufbaut, ist in Bild 1 zu sehen. Die Spielregeln sind sehr einfach und werden vom Rechner streng überwacht.

Die Pyramide, die auf Platz 1 steht, ist auf Platz 3 umzusetzen. Dabei kann Platz 2 als Zwischenlager verwendet werden. Die einzelnen Ebenen können beliebig oft zwischen den 3 Plätzen hin und her transportiert werden. Es bestehen dabei keine Richtungseinschränkungen. Ziel ist es, möglichst wenig Transporte durchzuführen. Die minimal mögliche Anzahl berechnet sich nach der Gleichung:

$$N(\min) = (2 \text{ hoch Ebenenanzahl}) - 1$$

Zwei Bedingungen existieren allerdings:

- Man kann je Transport nur eine Ebene, und zwar jeweils die oberste, bewegen, und
- man kann auf eine schmale Ebene keine breitere ablegen, d. h., man kann die Pyramide z. B. nicht auf Platz 2 mit der Spitze nach unten aufbauen.

In der Punktbewertung zählen die fehlerhaften Transporte mit.

TURM VON HANOI  
 TRANSPORT VON:  
 NACH:  
 ANZAHL DER TRANSPORTE: 0001

1	2	3
*		
***		
*****		
*****		
*****		
*****	.....	.....

Bild 1 Spielbild

Die Punkte im Spielbild erscheinen nicht auf dem Bildschirm. Sie sollen die beiden anderen Standorte markieren.

Der Rechner setzt die Schreibmarke hinter die Ausschrift »TRANSPORT VON« und erwartet die Eingabe der Platznummer, von wo aus der Transport erfolgen soll. Anschließend fordert er die Eingabe der Zielplatznummer.

Kernstück des Programms bildet eine Tabelle, in der die einzelnen dargestellten Ebenen als Zeichenkettenworte angegeben sind. Die in der Bildtabelle (ab Zeile 270) zusammengestellten Worte ergeben bei ihrer Ausgabe als Textkette das Spielbild. Dabei existiert als »Baugrund« noch eine sechste und siebente Ebene, die nicht mit bewegt werden. Sie dienen als Begrenzung bei der Bildrechnung. Damit ist es möglich, die gewünschten Transporte auf dem Bildschirm durch Manipulation der Werte in der Bildtabelle auszuführen und diese anschließend auszugeben. Jede Zeile in der Tabelle stellt dabei eine Bildschirmzeile dar. Da die drei Worte je Zeile 33 Zeichen ergeben, also einen Zeilenüberlauf erzeugen, wird die Zeile nicht mit NL beendet, sondern der Cursor durch CR nur an den Zeilenanfang (der neuen Zeile) zurückgeführt.

Im folgenden noch einige Hinweise zum Aufbau des Programms. Nach dem Löschen des Bildschirms wird die Bildtabelle in ihren Arbeitsbereich geladen (Zeile 10 bis 13), wodurch der Ausgangszustand jederzeit zur Verfügung steht. Nach der Ausgabe des Textkopfes und des Spielbildes (UP BAUS, ab Zeile 222) wird der erste Tastenwert angefordert. Dabei nimmt das Programm nur die Werte von eins bis drei an. Anschließend erfolgt das gleiche für den Wert des Zielplatzes (Zeile 24-40). Zu Beginn der Auswertung steht in B die Quellplatznummer und in C die Zielplatznummer. Aus diesen Werten wird die absolute Adresse in der obersten Zeile der Bildtabelle bestimmt (Zeile 47 bis 61). Zwischen Zeile 63 und 78 wird die oberste Ebene der Quellposition gesucht, wobei durch den Sprung auf Zeile 72 gesichert wird, daß der »Baugrund« nicht transportiert wird, falls keine Ebene sich mehr auf diesem

Platz befindet. Am Ende des Abschnitts steht die Adresse der Ebene (d.h. des Wortes in der Bildtabelle) in DE und die Größe der Ebene in H. Der gleiche Vorgang findet anschließend für die Zielposition statt, wobei die Adresse in BC steht und die Größe in L. Durch Vergleich der Größenwerte wird ermittelt, ob vom Spieler die Regeln eingehalten wurden. War das der Fall, findet der Transport der Ebenen statt (Zeile 97 bis 100). Bei der Kontrollsummenbildung (Zeile 103 bis 118) werden die Ebenengrößen addiert, die sich auf den Plätzen eins und zwei befinden. Sind beide Plätze leer, ergibt sich durch den »Baugrund« der Wert zwölf, der zur Endauswertung und Kommentierung der Versuchsanzahl führt (ab Zeile 126).

Das Programm ist auf dem in diesem Heft vorgestellten Betriebssystem für den Kleincomputer lauffähig. Soll es auf einem anderen Rechner genutzt

werden, so sind in der Anschlußtabelle (ab Zeile 207) die Adressen der entsprechenden Routinen einzutragen.

Die Anschlußbedingungen sind dem Betriebssystem zu entnehmen.

Das Programm enthält zwei allgemein nutzbare Unterprogramme, die auch ins Betriebssystem übernommen werden können. Es handelt sich dabei um die Hexadezimal-Dezimal-Wandlung (HEDE) und das Programm zur Ausgabe von Textketten (RTEX), welche kein Laden einer Adresse erfordern und nur das A-Register zerstören.

Das Programm wurde auf dem angegebenen Betriebssystem getestet.

Autor:

*Dr.-Ing. Gert Schönfelder*

Problemanalytiker im Rechenzentrum der Sektion Informationsverarbeitung der Ingenieurhochschule Dresden

---

## Rechentechnische Begriffe für den Laien erklärt

**Drucker<sup>1)</sup>** Zur Programmdokumentation sowie für Rechnungsbelege und Korrespondenzen stellt ein Drucker eine notwendige Ergänzung des Kleincomputersystems dar. Es besteht die Möglichkeit, geeignete elektronische Schreibmaschinen (z. B. S 6010 oder S 6005) oder Drucker über ein entsprechendes Interface mit dem System zu koppeln.

**Modul<sup>1)</sup>** Dieses Wort wird in der Elektronik im weitesten Sinne als Synonym für Baustein benutzt. Bei Kleincomputern werden in erster Linie die kleinen kompakten Ergänzungseinheiten, wie z. B. die RAM-Erweiterungen, als Modul bezeichnet.

---

<sup>1)</sup> Dieser Begriff wurde BÜCKNER, U.: Kleincomputer leichtverständlich. – Leipzig, 1986 entnommen

```

0001 PN TURM
0002 ;
0003 ; 30.01.84
0004 ;
0005 ORG 1000H
0006 ;
0007 ANF: LD A,0EH
0008 CALL VIDEO
0009 ;
0010 LD BC,40
0011 LD DE,BTA
0012 LD HL,BTA2
0013 LDIR
0014 ;
0015 ;
0016 LD BC,0
0017 LD (HA),BC
0018 ;
0019 H0: CALL BAUS
0020 LD B,2
0021 LD C,19
0022 CALL SETCU
0023 ;
0024 M1: CALL HTAST
0025 CMP 4
0026 JRNC M1-#
0027 OR A
0028 JRZ M1-#
0029 CALL ASCII
0030 PUSH AF
0031 ;
0032 LD B,3
0033 LD C,19
0034 CALL SETCU
0035 CALL HTAST
0036 CMP 4
0037 JRNC M2-#
0038 OR A
0039 JRZ M2-#

1000 3E 0E
1002 CD 40 12
; BILD LOESCHEN

1005 01 28 00
1008 11 E2 13
100B 21 3F 13
100E ED 80
; AUFBAU BILDTABELLE

1010 01 00 00
1013 ED 43 DC 13
; VERSUCHE = 0

1017 CD 52 12
101A 06 02
101C 0E 13
101E CD 3D 12
; BILD AUFBAUEN

1021 CD 4C 12
1024 FE 04
1026 30 F9
1028 B7
1029 28 F6
102B CD 4F 12
102E F5
; CURSOR AUF 2/19
; =VON= EINGEBEN
; WAR >3
; WAR 0
; ECHO
; RETTEN

102F 06 03
1031 0E 13
1033 CD 3D 12
1036 CD 4C 12
1039 FE 04
103B 30 F9
103D B7
103E 28 F6
; CURSOR AUF 3/19
; =NACH= EINGEBEN
; WAR >3
; WAR 0

```

1040	CD	4F	12	0040	CALL ASCII	; ECHO
1043	4F			0041	LD C,A	; ZIEL IN C
1044	F1			0042	POP AF	
1045	47			0043	LD B,A	; QUELLE IN B
				0044		
				0045		
				0046	AUSWERTUNG TRANSPORT	
1046	ED	5B	DC 13	0047	LD DE,(AR)	
104A	13			0048	INC DE	
104B	ED	53	DC 13	0049	LD (AR),DE	; VERSUCHSZAHL +1
				0050		
104F	11	00	00	0051	LD DE,0	
1052	21	E1	13	0052	LD HL,BTA-1	
1055	58			0053	LD E,B	
1056	19			0054	ADD HL,DE	; <DE>=SPALTE IN BILDTAB.
1057	EB			0055	EX DE,HL	; ENTSPRECHEND QUELLE
				0056		
1058	21	E1	13	0057	LD HL,BTA-1	
105B	06	00		0058	LD B,0	
105D	09			0059	ADD HL,BC	
105E	44			0060	LD B,H	; <BC>=SPALTE IN BILDTAB.
105F	40			0061	LD C,L	; ENTSPRECHEND ZIEL
				0062		
1060	13			0063	INC DE	
1061	13			0064	INC DE	
1062	13			0065	INC DE	
1063	13			0066	INC DE	
1064	1A			0067	LD A,(DE)	; ZEILE QUELLHAUFEN +1
1065	E6	07		0068	AND 7	
1067	FE	00		0069	CMP 0	; MASKIEREN AUF 0...7
1069	28	F5		0070	JRZ M3-#	; KEINE SCHICHT GEFUNDEN
106B	FE	06		0071	CMP 6	
106D	20	06		0072	JRNZ M4R-#	; KEIN BAUGRUND
106F	18			0073	DEC DE	
1070	1B			0074	DEC DE	
1071	1B			0075	DEC DE	
1072	1B			0076	DEC DE	; KORR. ZEIGER, WENN LEER
1073	3E	00		0077	LD A,0	
1075	67			0078	M4R: LD H,A	; H=SCHICHTBREITE QUELLE
				0079		
1076	03			0080	INC BC	

1077	03	0081	INC BC	
1078	03	0082	INC BC	
1079	03	0083	INC BC	; ZEILE QUELLHAUFEN +1
107A	0A	0084	LD A, (BC)	
107B	E6 07	0085	AND 7	; MASKIEREN AUF 0...7
107D	FE 00	0086	CMP 0	
107F	28 F5	0087	JRZ M4-#	; KEINE SCHICHT GEFUNDEN
1081	6F	0088	LD L, A	; L-BREITE GEFUNDENE ZIELSCHICHT
1082	08	0089	DEC BC	
1083	08	0090	DEC BC	
1084	08	0091	DEC BC	
1085	08	0092	DEC BC	
1086	7C	0093	LD A, H	; ZEIGER AUF ZEILE VOR ZIELSCH.
1087	BD	0094	CMP L	; VERGLEICH SCHICHTBREITEN
1088	D2 14 12	0095	JFNC M5	; FEHLER
		0096	QUELLE > ZIEL	
108B	F6 A0	0097	OR 0A0H	
108D	02	0098	LD (BC), A	; QUELLSCH. AM ZIEL EINTRAGEN
108E	3E A0	0099	LD A, 0A0H	
1090	12	0100	LD (DE), A	; SCHICHT IN QUELLHAUFEN LOESCHEN
		0101		
		0102	KONTROLLSUMME BILDEN	
1091	0E 00	0103	LD C, 0	
1093	06 08	0104	LD B, 8	
1095	21 E2 13	0105	LD HL, BTA	
1098	7E	0106	LD A, M	
1099	E6 07	0107	AND 7	
109B	81	0108	ADD C	; < SUMMIEREN DER UNTEREN 3 BIT
109C	4F	0109	LD C, A	; < VON HAUFEN 1 UND 2 IN DER BILDTRAB.
109D	23	0110	INC HL	; <
109E	7E	0111	LD A, M	; <
109F	E6 07	0112	AND 7	; < LEERE HAUFEN ERGEBEN DEN WERT
10A1	81	0113	ADD C	; < 12 DURCH 2 MAL BAUGRUND.
10A2	4F	0114	LD C, A	
10A3	23	0115	INC HL	
10A4	23	0116	INC HL	
10A5	23	0117	INC HL	
10A6	10 F0	0118	DJNZ M6-#	
		0119		
10A8	79	0120	ENDETEST	
		0121	LD A, C	



10A9	FE 0C	CMP 12	; TEST AUF LEER
10B8	C2 17 10	JPNZ 00	; KEIN ENDE
10B9			
10B9	CD 52 12	AUSMERTUNG PUNKTZAHL	
10B1	3E 1C	CALL BRAUS	
10B3	CD 40 12	LD R,1CH	
10B6	ED 4B DC 13	CALL VIDEO	
10B9	79	LD BC,(HR)	
10B8	FE 21	LD R,0	
10B0	DA 92 11	CMP 33	
10C0	FE 24	JPC M7	
10C2	DA 76 11	CMP 36	
10C3	FE 2A	JPC M8	%
10C7	DA 48 11	CMP 42	
10CA	FE 32	JPC M9	
10CC	DA 26 11	CMP 50	
10CF	FE 40	JPC M10	
10D1	DA 00 11	CMP 64	
		JPC M11	
10D4	CD CE 13	CALL RTEK	
10D7		DB 'HOFFNUNGSLOS !'	
10E3	1E	DB 360	
10E6		DB 'SIE BRAUCHEN JA EWIG !'	
10FC	03	DB 3	
10FD	C3 C1 11	JMP M12	
1100	CD CE 13	M11: CALL RTEK	
1103		DB 'WEITERMACHEN !'	
1111	1E	DB 360	
1112		DB 'NICHT AUFGEBEN !'	
1122	03	DB 3	
1123	C3 C1 11	JMP M12	
1126	CD CE 13	M10: CALL RTEK	
1129		DB 'SIE LERNEN DAS SCHON NOCH.'	
1143	1E	DB 360	
1144	03	DB 3	
1145	C3 C1 11	JMP M12	
1148	CD CE 13	M9: CALL RTEK	

```

114B 0157 DB 'SIE MUESSEN SICH MEHR KON-
1165 0157 DB 360
1166 0158 DB 'ZENTRIEREN .'
1172 0158 DB 3
1173 0159 JMP M12
0160 ;
1176 0161 CALL RTEK
1179 0162 DB 'SEHR GUT GEMACHT !!!'
1180 0162 DB 360
118E 0162 DB 3
118F 0163 JMP M12
0164 ;
1192 0165 CALL RTEK
1195 0166 DB 'HABEN SIE AUCH NICHT GE-'
11AD 0166 DB 360
11AE 0167 DB 'MOGELT ????'
11BD 0167 DB 3
11BE 0168 JMP M12
0169 ;
11C1 0170 ;
11C4 0171 CALL RTEK ; SAMMELPUNKT BEWERTUNG
11C5 0172 DB 360
11C6 0172 DB 360
11E1 0173 DB 'WOLLEN SIE WEITERSPIELEN? '
11E2 0173 DB 3
11E5 0174 CALL TAST ; EING. ENTSCHEIDUNG
11E8 0175 CALL VIDEO
11EA 0176 CMP 'N'
0177 RZ ; ENDE BEI NEIN
0178 ;
11EB 0178 LD A,(HA)
11EE 0179 CMP 50
11F0 0181 JPC ANF ; NEUSTART
11F3 0182 CALL RTEK ; RUECKFRAGE BEI >49 VERSUCHEN
11F6 0183 DB 360
11F7 0183 DB 360
11F8 0184 DB 'WIRKLICH ??? ; '
1207 0184 DB 3
1208 0185 CALL TAST
120B 0186 CALL VIDEO
120E 0187 CMP 'N' ; ENDE

```

1210	C8	*		RZ		ENDE
1211	C3	00 10		JMP ANF		NEUSTART
1214	06	0F				FEHLERHAFT EINGABE ==> FEHLERMELDUNG
1216	0E	00		LD B,15		
1218	CD	3D 12		LD C,0		
121B	CD	CE 13		CALL SETCU		
121E				CALL RTEK		
1231	03			DB /FALSCH E OPERATION !		
				DB 3		
1232	3E	F0		LD A,0F0H		
1234	CD	3A 12		CALL LOOP		CA. 1 S WARTEN
1237	C3	17 10		JMP H0		FORTSETZEN
123A	C3	00 00				VERBINDUNGEN INS BETRIEBSSYSTEM
123D	3E	1F		LOOP: JMP 0		ZEITSCHLEIFE
123F	00			SETCU: LD A,1FH		CURSOR SETZEN
1240	C3	00 00		NOP		
1243	C3	00 00		VIDEO: JMP 0		BS-AUSGABE
1246	C3	00 00		VIDAD: JMP 0		AUSGABE ADRESSE
1249	C3	00 00		TEXT: JMP 0		AUSGABE TEXT
124C	C3	00 00		TAST: JMP 0		EINGABE 1 ALPHAZEI.
124F	C3	00 00		HTAST: JMP 0		EING. 1 HEXZEICHEN
				ASCII: JMP 0		AUSG. 1 HEXZ. AUF BS
1252	3E	0E				BILDAUFBAU
1254	CD	40 12		BAUS: LD A,0EH		
1257	21	7A 12		CALL VIDEO		LOESCHEN BILD
125A	CD	46 12		LD HL,T1		
125D	ED	4B DC 13		CALL TEXT		KOPF SCHREIBEN
1261	CD	66 13		LD BC,(HA)		
				CALL HEDE		WANDELN HEX->DEC

1264	C5	0228	PUSH BC			
1265	E1	0229	POP HL			
1266	CD 43 12	0230	CALL VIDAD			; EINTRAGEN VERSUCHSZAHL
1269	21 D4 12	0231	LD HL, T2			
126C	CD 46 12	0232	CALL TEXT			; FORTS. KOPF
126F	21 E6 13	0234	LD HL, BTA+4			
1272	CD A4 13	0235	CALL WTEX			; AUSGABE PYRAMIEDEN
1275	C9	0236	RET			
		0237				
		0238				
1276		0239	BER 4			
127A		0240	DB / TURM VON HANOI			
1299	1E	0241	DB 360			
129A	1E	0242	DB 360			
129B	1E	0243	DB / TRANSPORT VON :			
12AE	1E	0244	DB 360			
12AF	09	0244	DB 110			
12B0		0245	DB / NACH :			
12BB	1E	0246	DB 360			
12BC	1E	0246	DB 360			
12BD		0247	DB / ANZAHL DER VERSUCHE :			
12D3	03	0247	DB 3			
		0248				
12D4	1E	0249	DB 360			
12D5	1E	0249	DB 360			
12D6		0250	DB / 1		2	3
12F1	1E	0251	DB 360			
12F2	03	0251	DB 3			
		0252				
		0253				
		0254				
		0255				
12F3	00	0256	WRTABE FUEER BILDDARSTELLUNG			
12F4	03	0256	WRTAB: DB 0			
12F5		0257	DB 3			
1300	03	0258	DB 3			
1301		0259	DB / *			
130C	03	0259	DB 3			
130D		0260	DB / ***			
1318	03	0260	DB 3			

1319		0261	DB	/	*****	/
1324	03	0261	DB	3		
1325		0262	DB	/	*****	/
1330	03	0262	DB	3		
1331		0263	DB	/	*****	/
133C	03	0263	DB	3		
133D	00	0264	DB	0		
133E	03	0264	DB	3		
		0265				
		0266				
		0267				
		0268				
		0269				
		0270	DB	0A0H		
1340	A0	0270	DB	0A0H		
1341	A0	0270	DB	0A0H		
1342	0D	0270	DB	0DH		
1343	A0	0271	DB	0A0H		
1344	A0	0271	DB	0A0H		
1345	A0	0271	DB	0A0H		
1346	0D	0271	DB	0DH		
1347	A1	0272	DB	0A1H		
1348	A0	0272	DB	0A0H		
1349	A0	0272	DB	0A0H		
134A	0D	0272	DB	0DH		
134B	A2	0273	DB	0A2H		
134C	A0	0273	DB	0A0H		
134D	A0	0273	DB	0A0H		
134E	0D	0273	DB	0DH		
134F	A3	0274	DB	0A3H		
1350	A0	0274	DB	0A0H		
1351	A0	0274	DB	0A0H		
1352	0D	0274	DB	0DH		
1353	A4	0275	DB	0A4H		
1354	A0	0275	DB	0A0H		
1355	A0	0275	DB	0A0H		
1356	0D	0275	DB	0DH		
1357	A5	0276	DB	0A5H		
1358	A0	0276	DB	0A0H		
1359	A0	0276	DB	0A0H		
135A	0D	0276	DB	0DH		

BILDTABELLE

BTR2: DB 0A0H

135B	R6	0277	DB	0A6H
135C	R6	0277	DB	0A6H
135D	R6	0277	DB	0A6H
135E	0D	0277	DB	0DH
135F	R6	0278	DB	0A6H
1360	R6	0278	DB	0A6H
1361	R6	0278	DB	0A6H
1362	0D	0278	DB	0DH
1363	03	0279	DB	3
1364	03	0279	DB	3
1365	03	0279	DB	3
1366	D5	0280		
1367	E5	0281		
1368	C5	0282		
1369	E1	0283		
136A	01 00 00	0284		
136D	11 E8 03	0285		
1370	AF	0286		
1371	04	0287	HEDE:	PUSH DE
1372	ED 52	0288		PUSH HL
1374	30 FB	0289		PUSH BC
1375	19	0290		POP HL
1377	05	0291		LD BC,0
1378	CB 20	0292		LD DE,1000
137A	CB 20	0293		XOR A
137C	CB 20	0294	HEDE1:	INC B
137E	CB 20	0295		SBC HL,DE
1380	11 64 00	0296		JRNC HEDE1-#
1383	AF	0297		ADD HL,DE
1384	04	0298		DEC B
1385	ED 52	0299		SLA B
1387	30 FB	0300		SLA B
1389	19	0301		SLA B
138A	05	0302		SLA B
		0303		LD DE,100
		0304		XOR A
		0305	HEDE2:	INC B
		0306		SBC HL,DE
		0307	JRNC	HEDE2-#
		0308		ADD HL,DE
		0309		DEC B

WANDLUNG DURCH SUBTRAKTION  
HEX----> DEZ IN BC



```

13AD 30 09      JRNC WTEXB-#      ; WORT
0351
0352      CALL VIDEO      ; NORMALES ZEICHEN
13AF CD 40 12  INC HL
13B2 23        JR WTEXC-#
13B3 18 F1
0356      WTEXA:POP AF      ; ENDE
0357      WTEXB:POP BC
13B6 C1      RET
13B7 C9
0360      WTEXB:PUSH HL      ; RETTEN TEXTZEIGER
13B8 E5      SUB SFH
13B9 D6 9F    LD B,A
13BB 47      LD HL,WTAB      ; B=NUMMER DES WORTES IN WTAB
13BC 21 F3 12 WTEXD:LD A,M
13BF 7E      INC HL
13C0 23      CMP 3
13C1 FE 03   JRNZ WTEXD-#      ; SUCHEN ETX (=3)
13C3 20 FA   DJNZ WTEXD-#      ; SUCHEN N-TES ETX
13C5 10 F8
0370      CALL WTEX
13C7 CD A4 13 REKURSIVER AUFRUF ZUR AUSGABE DER ZEICHENKETTE AUS WTAB
0371
0372
0373      POP HL
13CA E1      INC HL
13CB 23      JR WTEXC-#      ; WEITER MIT ZEICHENAUSGABE
13CC 18 D8
0378
0379
0380      AUFBAU DER WORTTABELLE
0381
0382      WTAB: DB 0,0,3      ; WORT 0
0383          DB 'PAUL',3      ; WORT A0
0384          DB 'OTTO',3      ; WORT A1
0385          DB '0A0H',-1,0A1H,3 ; RUF VON 'PAUL-OTTO'
0386
0387
0388
0389
0390          DB 3,3      ; ENDE
0391

```



```

0392 ; RELATIV-TEXTAUSGABE
0393 ;
0394 RTEX: EX (SP),HL
0395 RTEXR:LD R,M
0396 INC HL
0397 CMP 3
0398 JRZ RTEXB-#
0399 CALL VIDEO
0400 JR RTEXA-#
0401 RTEXB:EX (SP),HL
0402 RET
0403 ;
0404 ;
0405 ; R A M
0406 ;
0407 HA: BER 6
0408 BTA: BER 40
0409 BER 0
0410 ;
0411 ;
0412 ;
0413 ;
PROGRAMM ENTHRELT 00000 FEHLER
END
130E E3
130F 7E
1300 23
1301 FE 03
1303 28 05
1305 CD 40 12
1308 18 F5
130A E3
130B C9
; RETTEN HL / HOLEN ADR.
; TEST AUF ENDE
; ENDE
; AUSGEBEN
; HERSTELLEN HL
; PUNKTZAEHLER
; ARBEITSTABELLE BILD
130C
13E2
140A
140A

```



## 1. Vokabeltraining mit dem Kleincomputer

Das Abfragen von Vokabeln einer Lektion im Fremdsprachenunterricht erfordert einen »Lehrer«. Falls aber kein geeigneter Partner, dagegen ein Kleincomputer zur Verfügung steht, so sollte letzterer dessen Funktion übernehmen. Über den Bildschirm erhält der Schüler alle die Informationen, die ihm sonst der Lehrer gibt. Zunächst wird ein Wort erscheinen, das zu übersetzen ist. Hat der Schüler über die Tastatur des Rechners eine Übersetzung geliefert, so erwartet er eine Auskunft über die Richtigkeit seiner Antwort. Diese vermag der Rechner zu erteilen, indem er das gerade eingetippte Wort mit dem vergleicht, das mit der Abfrage ein Wortpaar (Korrespondenz) bildet und im Arbeitsspeicher steht. Dabei wird eine richtige Antwort bestätigt und bei einer falschen Antwort entweder die Möglichkeit einer zweiten Antwort eingeräumt oder gleich die richtige Übersetzung angezeigt. Die Leistung des Schülers wird durch seine Fehlerquote bewertet, das heißt durch den prozentualen Anteil der fehlerhaften Antworten. Die Aktionen des Computers bestehen also aus Zählen und Vergleichen sowie dem Auswählen von Wörtern aus einer eingegebenen Vokabelliste, wobei die Auswahl nicht nach der

Reihenfolge der Eingabe, sondern zufällig erfolgen sollte.

Im folgenden wird ein BASIC-Programm entwickelt, das die genannten Operationen ausführt und den Kleincomputer zum Vokabeltrainer werden läßt. Dieses Programm wurde auf der Messe der Meister von morgen der EOS »Georgi Dimitroff« der Stadt Leipzig im April 1985 ausgestellt.

Aus Platzgründen verzichten wir auf den Abdruck des vollständigen Quellprogramms, das immerhin 20 KByte umfaßt. Allerdings enthält dieses verschiedene Extras, die für den eigentlichen Trainer entbehrlich sind, so z. B. einen graphisch gestalteten und mit Musik unterlegten Vorspann (vgl. Bild 1), der während des Ladens den

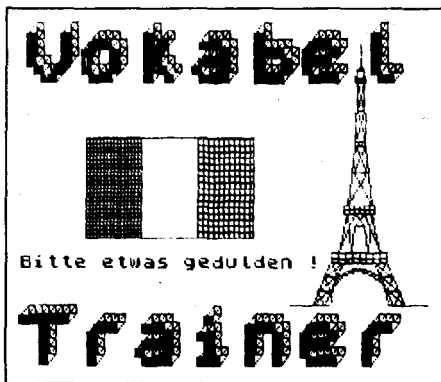


Bild 1

Bildschirm füllt, um die Wartezeit zu überbrücken, sowie das Programmende (Bild 13).

## 2. Struktur des Programms

Der Ladevorgang des Vokabeltrainers Französisch endet im Autostart mit einem Menü (vgl. Bild 2), das die aufrufbaren Programmteile auflistet.

Die ÜBERSICHT spricht für sich. Details werden im folgenden aufgeführt.

### ÜBERSICHT

Wählen Sie durch Druck der entsprechenden Zifferntaste eine der folgenden Möglichkeiten aus:

- 1 Allgemeine Instruktionen
- 2 Vokabeln eingeben
- 3 Vokabeltraining (dt.-frz.)
- 4 Vokabeltraining (frz.-dt.)
- 5 Lektion von Kassette laden
- 6 Lektion auf Kassette ablegen
- 7 Vokabeln korrigieren
- 8 Programmende

Bild 2

## 3. Allgemeine Instruktionen

Der Nutzer muß am Anfang über Grundsätzliches informiert werden, das ihn erst in die Lage versetzt, mit dem Vokabeltraining beginnen zu können. Da er über die Tastatur mit dem Rechner in den Dialog eintritt, sind vor allem Angaben erforderlich, wie sämtliche Buchstaben der Alphabete der beiden Sprachen realisiert werden. Die Schwierigkeit setzt bereits mit dem deutschen Alphabet ein, weil die üblichen Rechner Tastaturen nicht über die Umlaute ä, ö, ü und das ß verfügen. Natürlich kann man hierbei auf ae, oe, ue und ss ausweichen, doch ein Sprachlehrer würde allein schon diese Umschreibung zum Anlaß nehmen, einen

derartig beschränkten Vokabeltrainer abzulehnen. Für das französische Alphabet benötigen wir weiterhin â, à, è, é, ê, î, ô, û, ç, œ.

Das sind immerhin insgesamt 17 Sonderzeichen für einen Französisch-Trainer. Für einen Russisch-Trainer würden sogar  $7 + 38 = 45$  Sonderzeichen gebraucht.

Jedes Sonderzeichen wird nun grafisch über  $8 \times 8 = 64$  Pixel definiert und über ein Unterprogramm eine modifizierte Tastenbelegung erklärt.

```

9030 DATA "a",0,68,56,4,60,68,60
,0: REM a
9040 DATA "b",0,0,126,90,94,88,1
,0: REM b
9050 DATA "c",0,0,28,32,32,32,28
,48: REM c
9060 DATA "d",64,32,56,4,60,68,6
,0: REM d
9070 DATA "e",64,32,56,68,120,64
,60: REM e
9080 DATA "f",16,40,56,4,60,68,6
,0: REM f
9090 DATA "g",16,40,0,56,68,68,5
,0: REM g
9100 DATA "h",16,40,0,68,68,68,5
,0: REM h
9110 DATA "i",16,40,0,48,16,16,5
,0: REM i
9120 DATA "j",0,68,0,68,68,68,56
,0: REM j
9130 DATA "l",0,68,56,68,68,68,5
,0: REM l
9140 DATA "o",68,60,66,66,66,66,
,60: REM o
9150 DATA "q",68,60,66,66,126,66
,66: REM q
9160 DATA "r",16,40,56,68,120,64
,60: REM r
9170 DATA "s",0,56,68,72,68,88,6
,4: REM s
9180 DATA "t",4,8,56,68,120,64,6
,0: REM t
9190 DATA "u",66,0,66,66,66,66,6
,0: REM u

```

Bild 3

Damit der ungeübte Nutzer bei der Eingabe schnell die erforderlichen Buchstaben findet, werden die Sonderzeichen auf einer Liste beigelegt (Bild 4).

Als weitere Instruktionen kommen Angaben über die Schwachstellen des Trainers in Betracht. So muß z. B. der Nutzer darauf hingewiesen werden, daß unser Trainer gegenüber Synonymen hilflos ist und einen Fehler meldet,

## SONDERZEICHEN

Die deutschen und französischen Buchstaben, die nicht auf der Tastatur zu finden sind, stellt man auf folgende Weise dar:

- 1.) Man drückt "CAPS SHIFT" und "9" gleichzeitig.
- 2.) Es erscheint ein blinkendes "G".
- 3.) Man sucht in der untenstehenden Liste das Sonderzeichen und findet die entsprechende Taste.
- 4.) Man drückt diese Taste.
- 5.) Um wieder Normalbuchstaben eingeben zu können, ist wiederum "CAPS SHIFT" und "9" zu betätigen.

deutsche Sonderzeichen	Taste
Ä	--> [Symbol] 9
Ö	--> [Symbol] 0
Ü	--> [Symbol] 1
ß	--> [Symbol] 2
ä	--> [Symbol] 3
ö	--> [Symbol] 4
ü	--> [Symbol] 5

französische Sonderzeichen	Taste
À	--> [Symbol] 6
Á	--> [Symbol] 7
Â	--> [Symbol] 8
Ã	--> [Symbol] 9
Ä	--> [Symbol] 0
Å	--> [Symbol] 1
Æ	--> [Symbol] 2
Ç	--> [Symbol] 3
È	--> [Symbol] 4
É	--> [Symbol] 5
Ê	--> [Symbol] 6
Ë	--> [Symbol] 7
Ë	--> [Symbol] 8
Ï	--> [Symbol] 9

Um "1" oder "\*" oder "?" darzustellen, verfährt man wie folgt:

"1"	-->	"SYMBOL SHIFT" + "j"
"*"	-->	"SYMBOL SHIFT" + "7"
"?"	-->	"SYMBOL SHIFT" + "c"

Bild 4

wenn der Schüler eine durchaus richtige (zweite) Übersetzung verwendet, die aber nicht zusätzlich vorher gespeichert war.

### 4. Vokabeln eingeben

Das Vokabellernen beginnt mit dem Abschreiben – etwa von der im Lehrbuch zusammengestellten Vokabelliste einer Lektion – auf der Tastatur des Rechners und der anschließenden Ein-

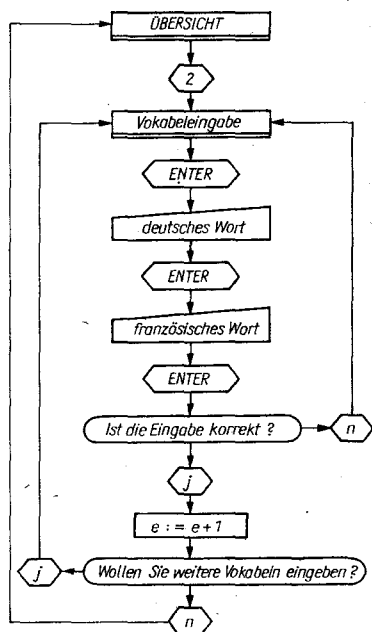


Bild 5

gabe. Dieser Vorgang wird wie folgt organisiert:

Entsprechend der TGL 22451 verwenden wir für einen Tastendruck das Sinnbild und für die Worteingabe .

Eine Bildschirminformation, die durch einen speziellen Tastendruck verschwindet, ist doppelt unterstrichen. Die Zählvariable *e* gibt die um 2 vergrößerte

Vokabeleingabe

Anzahl der eingegebenen Vokabeln beträgt 50

dt. Wort:

frz. Wort:

dt. Wort: die Bibliothek

Bild 6a

Vokabeleingabe

Anzahl der eingegebenen Vokabeln beträgt 50

dt. Wort:

frz. Wort:

frz. Wort: ■

Bild 6b

Vokabeleingabe

Anzahl der eingegebenen Vokabeln beträgt 50

dt. Wort:

frz. Wort:

Ist die Eingabe korrekt? (j/n)

Bild 6c

Anzahl der eingegebenen Korrespondenzen an:

Auf Bild 6a, b, c sind drei Bildschirmphasen bei der Vokabeleingabe dargestellt. Das deutsche Wort ist auf Bild 6a abgeschrieben, auf Bild 6b aber erst – mit ENTER – eingegeben. Die vollständige Korrespondenz zeigt Bild 6c an.

### 5. Vokabeltraining

Von der ÜBERSICHT gelangt man über Taste 3 zu den Instruktionen über die Organisation der Abfrage: deutsch-französisch (vgl. Bild 7).

Drei Bildschirmphasen der Abfrage sind auf Bild 8 wiedergegeben.

Die Fehlerquote wird erst bei der Wahl des nächsten deutschen Wortes verändert. Da das Vokabeltraining das Herzstück des Programms darstellt, wird auf Bild 9 der entsprechende Ausschnitt des Quellprogramms zusam-

ABFRAGE: deutsch  
französisch  
-----

Der Computer gibt Ihnen ein dt. Wort vor und Sie können nun die französische Übersetzung eingeben. Dazu ist folgendes zu beachten:

- 1) Sie haben zwei Versuche, um die richtige Übersetzung zu finden.
- 2) Wenn Ihnen die entsprechende Vokabel völlig entfallen ist, tippen Sie "?" ein.
- 3) Nach "?" bzw. nach zwei falschen Eingaben, erscheint die richtige Übersetzung.
- 4) Ihre Vokabelkenntnis wird durch die Anzahl der Fehler bezogen auf die Anzahl der Versuche -Fehlerquote- bewertet.
- 5) Um das Training zu beenden, drücken Sie die Tasten "z" und "ENTER".

Beginnen Sie das Training mit "ENTER"!

Bild 7

ABFRAGE: deutsch  
französisch  
-----

(Die Fehlerquote beträgt 0%.)

dt. Wort:

frz. Wort:

Geben Sie die Übersetzung des Wortes ein.

■

Bild 8a

ABFRAGE: deutsch  
französisch  
-----

(Die Fehlerquote beträgt 0%.)

dt. Wort:

frz. Wort:

Das ist leider eine falsche Übersetzung. Sie haben noch eine Möglichkeit. Drücken Sie dazu eine beliebige Taste.

Bild 8b

(Die Fehlerquote beträgt 0%.)

dt. Wort: der Flughafen

frz. Wort: L'aéroport

ÜBERSETZUNG RICHTIG

Bild 8c

men mit den erforderlichen Unterprogrammen ausgedruckt.  
 Zum Verständnis wird zunächst die Variablenliste beigefügt:

- a legt in der Ausgabe die Sprache und die Lage auf dem Bildschirm fest;  $a = 1$  entspricht dt.,  $a = 2$  frz.
- b Testergebnis des eingegebenen Wortes auf zulässige Länge;  $b = 0$  bedeutet zulässige,  $b = 1$  unzulässige Länge.
- c Sprache des vom Rechner vorgegebenen Wortes
- d Sprache des einzugebenden Wortes
- e Anzahl der eingegebenen Vokabelpaare + 2
- i Zählvariable in Schleifen
- j Anzahl der falsch übersetzten Vokabeln
- k Fehlerquote (100  $j/l$ )
- l Anzahl der abgefragten Vokabeln
- m Nummer einer zufällig ausgewählten Null im Feld  $w$
- n Lage von  $m$  im Feld  $w$
- o Zählvariable für Abfragen (Versuche)
- p Maximalzahl der eingebbaren Vokabelpaare
- w eindimensionales Feld der Länge  $p + 1$  zur Bewertung der Übersetzung:  $w(n) = 1$ , falls Vokabel sofort richtig übersetzt wurde,  $w(n) = 0$  sonst

```

7010 GO SUB 7900
7020 IF (L=0 THEN LET K=0: GO TO 7040
7030 LET K=(100*J)/I
7040 PRINT AT 5,0;"(Die Fehlerqu
ote beträgt ",INT K,"%.)"
7050 IF Z=2 THEN PRINT AT 10,0
;"Sie haben alle ",Z," Vokabeln
richtig bewältigt." LIM WIP+1
); LET Z=0: GO TO 9900
7060 LET M=INT (RND*(E-2-Z)+1):
LET N=1
7070 FOR I=1 TO M
7080 LET N=N+1
7090 IF W(N)=1 THEN GO TO 7080
7100 NEXT I
7110 LET O=0
7120 LET C$=A$(C,N): LET A=C: GO
SUB 9700
7130 LET C$=A$(2,1): LET A=D: GO
SUB 9700
7140 PRINT AT 18,0;"Geben Sie di
e Übersetzung des Wortes ein."
INPUT LINE B$: GO SUB 9600
7150 GO SUB 9800: IF B=1 THEN GO
TO 7140
7160 IF B$="" THEN GO TO 7220
7170 LET D$=B$: FOR I=LEN B$ TO
19: LET D$=D$+":": NEXT I
7180 LET C$=D$: LET A=D: GO SUB
9700
7190 IF A$(D,N)=D$ THEN PRINT AT
18,7; FLASH 1;"ÜBERSETZUNG RICHTIG":
BEEP .5,0: BEEP .5,4: BEEP
.5,7: BEEP .7,12: LET W(N)=(O=0
); LET Z=Z+(O=0): GO TO 7240
7200 LET O=O+1
7210 IF O=1 THEN PRINT AT 18,0;"
Das ist leider eine falsche
Übersetzung. Sie haben noch eine
Möglichkeit. Drücken Sie dazu
eine beliebige Taste.": PAUSE 0
); GO SUB 9600: LET L=L+1: LET J=
J+1: GO TO 7130
7220 PRINT AT 18,0;"Dies ist die
richtige Korrespondenz"
Z
7230 LET C$=A$(D,N): LET A=D: GO
SUB 9700: LET J=J+1: FOR I=1 TO
O STEP -1: BEEP .01 I: NEXT I
7240 LET L=L+1: GO TO 7010
7300 CLS: PRINT AT 0,6;"ABFRAGE
";:E$(C): AT 1,15;E$(D): AT 2,6;"
-----": RETURN
  
```

9599 REM

Subroutines

```

9600 REM Löschen des unteren
Bildes
9610 PRINT AT 18,0;"
  
```

```

9620 RETURN
9700 REM Ausgabe
9710 IF A=1 THEN PRINT AT 10,0;"
dt. Wort: "; GO TO 9730
9720 PRINT AT 14,0;"frz. Wort: "
9730 PRINT INK 0; PAPER 7: AT 4+
+6,11; C$
9740 RETURN
9800 REM Eingabetest
9810 IF LEN B$ < 1 OR LEN B$ > 20 TH
EN LET B=1: RETURN
9820 IF B$="Z" THEN LET B=0: GO
TO 20
9830 LET B=0: RETURN
9900 REM Rücksprung in die
Übersicht (Zeile 20)
9910 FOR I=0 TO 300: IF I=200;"
Z": THEN GO TO 20
9920 NEXT I: GO TO 20
  
```

Bild 9

- z Anzahl der abgefragten Vokabeln, die sofort richtig übersetzt wurden
- a\$ zweidimensionales Feld (Sprache: 2, Länge:  $p + 1$ ), jedes Element besitzt 20 Zeichen;
- a\$ (2,1) = {20 × □}
- b\$ eingetipptes Wort
- c\$ auszugebendes Wort
- d\$ durch Freizeichen auf 20 Zeichen aufgefülltes Wort
- e\$ eindimensionales Feld der Länge 2, jedes Element besteht aus 11 Zeichen; e\$ (1) = »deutsch«, e\$ (2) = »französisch«

*Dokumentation des Hauptteils der Abfrage*

- 7010 Löschen des Bildschirms, Ausgabe einer Überschrift
- 7020—7040 Ausgabe der Fehlerquote.
- 7050 Bei richtiger Übersetzung aller eingegebenen Vokabeln wird Training beendet (Sprung zur ÜBERSICHT).
- 7060—7100 Zufällige Auswahl einer noch nicht (richtig) übersetzten Vokabel, deren Lage durch  $n$  charakterisiert wird.
- 7120—7140 Aufbau des unteren Teils von Bild 8a. Nach Eingabe Löschen der Anweisung.
- 7150 Eingabetest, der bei unzulässiger Wortlänge auf erneute Eingabe führt.
- 7160 Falls »?« eingegeben (vgl. Bild 7, Punkt 3), erfolgt über 7220, 7230 die Ausgabe der richtigen Übersetzung.
- 7170, 7180 Auffüllen des eingegebenen Wortes durch Freizeichen auf 20 Zeichen und Ausgabe dieses Wortes.
- 7190 Falls die gespeicherte Übersetzung mit dem eingegebenen Wort übereinstimmt, wird dies optisch (vgl. Bild 8c) und akustisch ange-

zeigt, und es erfolgt Sprung nach 7240.

Handelt es sich hierbei um eine unmittelbar richtige Übersetzung, so wird die Bewertung  $w$  der Vokabel (der Lage)  $n$  von 0 auf 1 gestellt und  $z$  um 1 erhöht.

- 7210 Falls es sich noch um den ersten Versuch handelt, erscheint Text des unteren Teils von Bild 8b, und es wird auf einen Tastendruck gewartet. Darauf wird Anweisung gelöscht und  $l$  sowie  $j$  um 1 erhöht. Anschließend wird erneuter Versuch gestartet.
- 7220, 7230 Ausgabe der richtigen Übersetzung mit abwärtsleitender Tonfolge.
- 7240 Anzahl der abgefragten Vokabeln wird um 1 erhöht und Abfrage neu begonnen.
- 7900 Unterprogramm: Löschen des Bildschirms, Ausgabe der Überschrift.

Die Verwendung von Variablen zur Kennzeichnung der Sprache ( $a, c, d$ ) gestattet uns, dasselbe Programm für die Abfrage: französisch-deutsch (vgl. Bild 10) zu benutzen.

ABFRAGE:   französisch  
                  deutsch  
-----

(Die Fehlerquote beträgt 0%.)

dt. Wort:

frz. Wort:

Geben Sie die Übersetzung des Wortes ein.

■  
Bild 10

### 6. Vokabeln korrigieren

Manchmal wird erst beim Vokabeltraining bemerkt, daß ein früher eingegebenes Vokabelpaar keine wahre

### Eingabekorrektur

Zur Korrektur einer bestimmten Korrespondenz geben Sie bitte das deutsche Wort ein, das zu dem falschen Vokabelpaar gehört.

Eingabe: ■

Bild 11

Korrespondenz bildet. Dann ist eine nachträgliche Korrektur erforderlich. Über Taste 7 erhält man die Instruktionen von Bild 11.

Nach der Eingabe des deutschen Wortes des falschen Vokabelpaars sucht der Rechner die eingegebene Korrespondenz und zeigt diese gemäß Bild 12a an, wobei er zunächst auffordert, das richtige deutsche Wort (falls es bereits richtig war, so ist nur ENTER zu drücken) bzw. das richtige französische Wort einzugeben.

Nach erfolgter Eingabe wird durch Auswechseln der Wörter korrigiert.

### Eingabekorrektur

dt. Wort:

frz. Wort:

Geben Sie das richtige frz. Wort ein:

le bousier ■

Bild 12a

### Eingabekorrektur

dt. Wort:

frz. Wort:

Ist die Korrespondenz jetzt richtig? (j/n)

Bild 12b

Das Ergebnis gibt in unserem Beispiel Bild 12b an.

## 7. Vokabeln auf Kassette sichern

Hat der Nutzer die Vokabeln einer Lektion eingegeben und geübt, so ist es mitunter ratsam, das Training am nächsten Tag fortzusetzen. Dazu müssen die im RAM gespeicherten Vokabeln auf einen externen Speicher gebracht und gesichert werden. Das Speichern eines Feldes auf eine Kassette bereitet prinzipiell keine Schwierigkeiten. Für ein schnelles Überspielen (in beiden Richtungen) erweist es sich als günstig, die das Feld  $a\$$  charakterisierenden Werte der Variablen  $e$  und  $p$  als String in  $a\$(1,1)$  und  $a\$(2,1)$  abzuspeichern. Für den Nutzer ist es weiterhin angenehm, wenn er von Systemmeldungen verschont wird. Das gelingt durch Anlegen von Vorder- und Hintergrundfarben.

## 8. Vokabeln von Kassette laden und Ausgabe einer Vokabelliste

Mittels des Befehls LOAD »Vokabeln«: DATA  $a\$( )$  werden die auf einer Kassette gemäß Abschnitt 7. abgespeicherten Vokabeln in den Rechner gebracht. Über den Befehl VAL erhalten wir die



Bild 13



aktuellen Werte der Variablen  $e$  und  $p$ , und ein früher abgebrochenes Training kann weitergeführt werden. Natürlich besteht auch die Möglichkeit, einem Schüler Vokabeln zum Üben anzubieten, die er selbst nicht eingegeben hat. In diesem Fall sollte er sich die Vokabelliste bei Bedarf (und wegen des Synonymproblems) auf dem Bildschirm ansehen können.

Das Programmende wird über die Taste 8 erreicht und besteht aus dem Schlußbild (vgl. Bild 13) und einer Abschlußmelodie.

## 9. Ausblick

Unseren Vokabeltrainer können wir als *Wortschatztrainer* verwenden, wenn wir z. B. den Grundwortschatz von etwa 1500 Wörtern, in Lektionen aufgeteilt, auf eine Kassette bringen und jeweils eine Lektion entsprechend Abschnitt 8. laden. Danach werden die Korrespondenzen auf dem Bildschirm gezeigt, und der Schüler lernt in der Ausgangsphase. Anschließend werden gemäß Abschnitt 5. französische Vokabeln aus dieser Lektion zufällig ausgewählt und vom Schüler ins Deutsche übersetzt (passive Phase). Schließlich erfolgt in der (aktiven) Schlußphase die Abfrage: deutsch-französisch.

Der Vokabeltrainer läßt sich zu einem *Sprachtrainer* ausbauen, wenn man statt der Wortlänge von 20 Zeichen eine Satzlänge von mindestens 60 Zeichen vorsieht. Dann können einfache Sätze, gängige Redewendungen, aber auch Grammatik-Übungssätze abgefragt werden, wenn man ein dazu erforderliches Sprachmaterial auf eine Kassette gebracht hat. Natürlich wird hierzu mehr Speicherplatz benötigt. Außerdem muß man für den Vergleich, den der Rechner nach jeder eingegebenen Übersetzung anstellen muß, eine größere Rechenzeit in Kauf neh-

men, so daß der Übergang zu einem Maschinenprogramm bald nötig wird. Das Grammatikwissen kann auch mit weniger Aufwand trainiert werden, wenn man mit *Lückentexten* arbeitet. Hierbei wird neben dem vollständigen deutschen Satz ein unvollständiger französischer Satz angeboten. Das bedeutet, in der französischen Übersetzung sind einige Wörter oder auch nur Endungen weggelassen, die buchstabenweise durch den Schüler zu ergänzen sind.

Jüngere Schüler werden im Sprachunterricht stärker gefesselt, wenn die oben angedeutete Methodik mit einem Spiel verknüpft wird, das sich auf dem Computer realisieren läßt. Dazu eignet sich z. B. das bekannte »Galgenraten« oder auch Schreibspiele, bei denen möglichst viele Wörter eines speziellen Sachgebiets in vorgegebener Zeit zu notieren sind. In letzterem Falle kann der Computer auch als *elektronisches Wörterbuch* dienen, der alle eingegebenen Wörter orthographisch überprüft und die Fehlerquote feststellt.

Autoren:

*Dipl.-Math. Inge Girlich; Uwe Girlich*  
Leipzig

# Rechentechnische Begriffe für den Laien erklärt

- Sensor<sup>1)</sup>** Ein Sensor ist ein Meßfühler. In der Elektronik wird ein Baustein, der nichtelektrische Größen in elektrische Signale umwandelt, als Sensor bezeichnet. Diese Signale können dann einem elektronischen System, z. B. einem Kleincomputer, zur Auswertung bereitgestellt werden.
- Wort<sup>1)</sup>** Als Wort wird in der Computerfachsprache eine Folge von so vielen Bits, wie der Prozessor, also die CPU, auf einmal verarbeiten kann, bezeichnet. Bei den in Kleincomputern eingesetzten Mikroprozessoren handelt es sich meistens wie beim U 880 um 8-Bit-Prozessoren, deren Wortbreite, wie der Name bereits sagt, 8 Bit beträgt.
- VLSI** Höchstintegration (**V**ery **L**arge **S**cale **I**ntegration). Integration von bis zu 100000 Bauelementen auf einem Chip.
- ULSI** Ultrahöchstintegration (**U**ltra **L**arge **S**cale **I**ntegration). Integration von mehr als 100000 Bauelementen auf einem Chip.

---

<sup>1)</sup> Dieser Begriff wurde BÜCKNER, U.: Kleincomputer leichtverständlich. – Leipzig, 1986 entnommen

---

ISBN 3-343-00129-5

© VEB Fachbuchverlag Leipzig 1986

1. Auflage

Lizenznummer 114-210/1/86

LSV 1083

Verlagslektor: Helga Fago

Gestaltung: Lothar Gabler

Printed in GDR

Satz und Druck:

Messdruck Leipzig, Bereich Borsdorf

III-18-328

Redaktionsschluß: 15. 3. 1986

Bestellnummer: 547 131 6

00780

Kleinstrechner-TIPS/Hrsg. von

Hans Kréul u. a. – Leipzig: Fachbuchverl.,

H. 5. – 1. Aufl. – 1986. – 64 S.: 28 Bild.

Anschrift des Verlages:

VEB Fachbuchverlag

PSF 67

DDR – Leipzig

7031

---

# Vorschau auf die nächsten Hefte

*Schilling*: Ein Formalismus zur Beschreibung von Problemlösungen

*Michel*: Wieso rechnet (m)ein Taschenrechner  $\sqrt[3]{8} \cdot \sqrt[3]{8} = 8$ ?

*Kühnel*: Kalenderfunktionen

*Lorenz/Schulze*: Simulation auf Mikrorechnern: Spiele und Experimente (Teil I)

*Schönfelder*: Master Mind – gegen den Rechner gespielt

*Hähnel/Kühnel*: Programm zur Berechnung von Kennwerten einfacher Verstärkerstufen mit bipolaren Transistoren

*Hähnel/Kühnel*: »Intelligente« Eingaberoutine für Programme zur Realisierung einer komplexen Arithmetik

---

Die Broschürenreihe

## KLEINSTRECHNER-TIPS

behandelt

- Tendenzen und Theorien
- Informationen und Ideen
- Programme und Projekte
- Spaß und Spiel

und stellt sich das Ziel

- den Nutzer der Mikrorechentechnik aus allen Bereichen der Volkswirtschaft und dem Bildungswesen bei der Einarbeitung in die Informatik und Computertechnik zu unterstützen
- Entwicklungstendenzen der Informatik und Computertechnik vorzustellen und zur Erweiterung des Grundwissens beizutragen
- Anregungen für den Computereinsatz zu geben und Beispielprogramme für Kleincomputer zu veröffentlichen

um somit einem großen Kreis von Freunden der Informatik und Computertechnik zu helfen, sich moderner Hilfsmittel und Methoden zu bedienen.