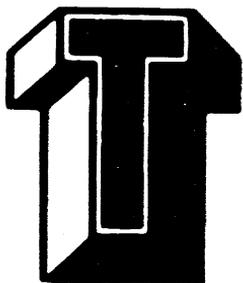


# Kleinstrechner



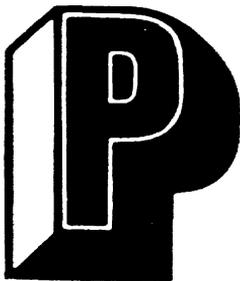
**Tendenzen  
und Theorien**

**Taschenrechner-  
sortiment  
in der DDR**



**Informationen  
und Ideen**

**Polycomputer 880**



**Programme  
und Projekte**

**Programmwurf**



**Spaß  
und Spiel**

**3**



---

# Kleinstrechner-TIPS

**Heft 3**

Mit 40 Bildern

Herausgegeben von

Prof. Dr.-Ing. Hans Kreul

Dr.-Ing. Wilhelm Leupold

Doz. Dr. sc. techn. Thomas Horn

**f**  
**v**

**VEB Fachbuchverlag Leipzig**

---

---

# Inhalt

*Ehram:* Die Taschenrechnerproduktion in der DDR 4

*Hübner:* Polycomputer 880 – Anwendung und Erweiterungsmöglichkeiten 20

Spielereien mit dem Taschenrechner 31

*Horn:* Wie kann ein Programm systematisch entworfen werden? 32

*Gutzer:* Eine Mondlandung mit dem K 1003 49

Der Taschenrechner hilft, mathematische Gesetzmäßigkeiten zu entdecken 64



Die Absicht der Herausgeber und des Verlages, möglichst vielen Interessenten der Kleinstrechentechnik – vom Taschenrechner bis zum Mikrorechner – durch eine breit gefächerte Thematik Anregungen für eine eigene aktive Beschäftigung zu geben, bestimmte den Inhalt auch des vorliegenden Heftes 3 der »Kleinstrechner-TIPS«.

Einen Überblick über die derzeit in der DDR produzierten Taschenrechner und deren Gebrauchswerteigenschaften gibt EHRMANN im ersten Artikel. Aus der Beschreibung des Aufbaus dürfte mancher Nutzer ihm bisher noch nicht Bekanntes erfahren, ebenso aus den Hinweisen des Autors für den Umgang mit Taschenrechnern.

Die in Heft 1 begonnene Artikelserie von SCHÖNFELDER zum Thema Heimcomputer wird in einem folgenden Heft mit der Beschreibung eines Mini-Betriebssystems und der Erweiterung der Konfiguration durch ein Kassettengerät abgeschlossen. Mit diesem Minibetriebssystem kann auch das Spielprogramm des sog. »Turmes von Hanoi« lauffähig gemacht werden, das in einem Beitrag des gleichen Autors den Interessenten an Computerspielen später angeboten wird.

Die Konzeption der Kleinstrechner-TIPS sieht vor, auch Informationen über die in der DDR entwickelte oder eingesetzte Rechentechnik und Programmsysteme zu bringen, sofern sie für einen größeren, nichtkommerziellen Anwenderkreis verfügbar sind. Als Veröffentlichung in dieser Richtung ist die Vorstellung des Mikrorechnersystems Polycomputer 880 durch HÜBNER in diesem Heft anzusehen.

Im Beitrag über den systematischen Programmwurf gibt HORX eine Einführung in die Technik der Struktogramme, mit deren Hilfe Programmsysteme mit einer komplizierten inneren Logik strukturiert entworfen werden können. Der strukturierte Programmwurf ist die Voraussetzung für eine strukturierte Programmierung, die zu leicht wartbaren und zuverlässigen Programmen führt.

GUTZER bringt wieder ein Spiel: eine Mondlandung mit dem K 1003.

Die Herausgeber hoffen, auch mit den Beiträgen dieses Heftes eine gute Resonanz bei den Lesern zu finden.

*Wilhelm Leupold*

---

# Die Taschenrechnerproduktion in der DDR



## 1. Einleitung

Die Taschenrechner erfreuen sich international und natürlich auch in der DDR großer Beliebtheit.

Dieses Konsumgut der Mikroelektronik hat sehr schnell große Verbreitung gefunden. Es zeigt anschaulich, welches Leistungsvermögen dieser jüngste Zweig der modernen Elektronik besitzt. Bedenkt man, daß der erste elektronische Rechner im Jahr 1946 die Größe eines Wohnhauses besaß, so wird jedermann verständlich, welche rasante technische und technologische Entwicklung sich auf dem Gebiet der Elektronik in den letzten Jahren vollzogen hat.

In der DDR werden seit 1973 Taschenrechner in verschiedenen Versionen produziert. Einziger Hersteller ist der VEB Mikroelektronik »Wilhelm Pieck« Mühlhausen (ehemals VEB Röhrenwerk Mühlhausen), ein Betrieb des VEB Kombinat Mikroelektronik.

Das bisher produzierte Gerätesortiment umfaßt die Geräte der »minirex«- und der »konkret«-Serie und bestand aus insgesamt 7 verschiedenen Typen, die auch heute noch genutzt, aber nicht mehr produziert werden. Ab 1978 begann die schrittweise Ablösung durch eine moderne Gerätegeneration, die zum gegenwärtigen Zeitpunkt 6 verschiedene Geräte um-

faßt. Sie ist gekennzeichnet durch den Einsatz modernster, in den Abmessungen stark minimierter und mit geringem Energieverbrauch auskommender elektronischer Bauelemente.

Verbunden mit einer neuen konstruktiven Gestaltung und neuen Technologien, wurde eine Gerätegeneration geschaffen, die hohen Ansprüchen genügt. Auf diese Taschenrechnergeneration wird näher eingegangen.

Die Bestrebungen der Taschenrechnerproduzenten gehen weltweit dahin, daß jeder Käufer eines Gerätes »seinen Typ« findet, also ein Gerät, das für den von ihm vorgesehenen Anwendungsfall die besten Voraussetzungen mitbringt. Durch die große Anwendungsbreite des Taschenrechners ist es daher auch erforderlich, eine umfassende Gerätepalette anzubieten. Wir wollen uns nun den verschiedenen Gerätetypen der DDR-Produktion und ihren unterschiedlichen Gebrauchswerteigenschaften zuwenden.

## 2. Die verschiedenen Taschenrechner- typen und ihre Gebrauchswerteigenschaften

Das internationale Taschenrechner-sortiment wird im allgemeinen in die nachfolgenden Kategorien eingeteilt:

- einfache 4-Spezies-Rechner
- 4-Spezies-Rechner mit Zusatzfunktionen und Speicher

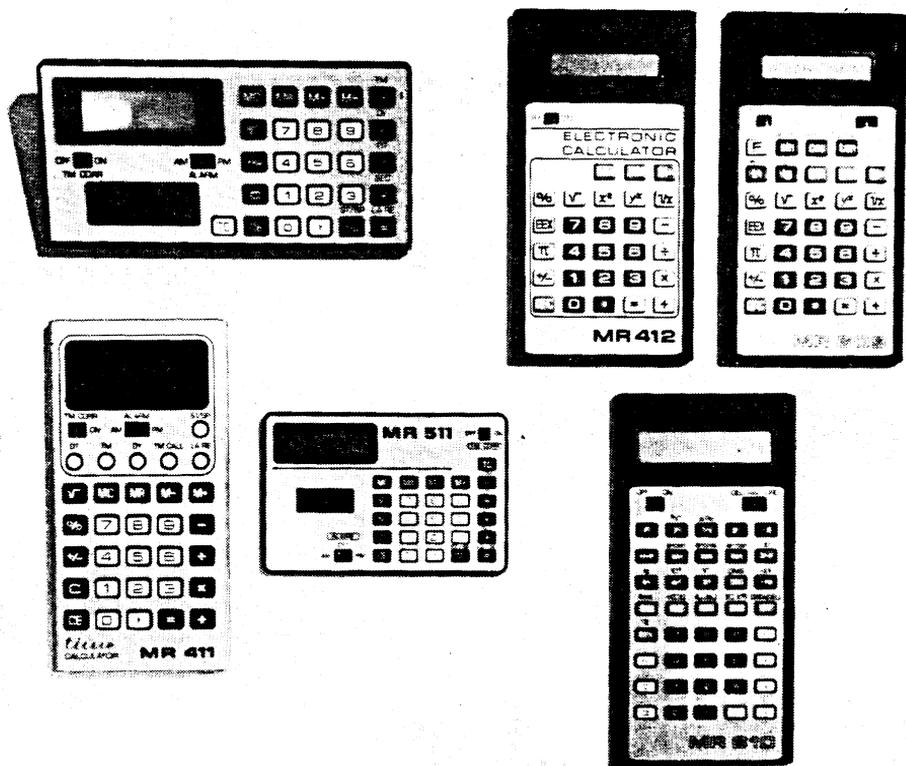


Bild 1. Die Familie der in Produktion befindlichen Taschenrechner

- wissenschaftliche Taschenrechner
- programmierbare Taschenrechner

Die Gruppe der einfachen 4-Spezies-Rechner wird kaum noch hergestellt, da sie nicht mehr den gestiegenen Anforderungen entspricht und weil es die moderne Mikroelektronik leicht ermöglicht, die Gebrauchswerteigenschaften der nächsten Gruppe ohne großen Mehraufwand zu erreichen.

Wie bereits dargelegt, umfaßt das derzeitige Produktionsortiment in der DDR 6 verschiedene Gerätetypen, die in Bild 1 dargestellt sind und nun näher erläutert werden sollen.

Bevor wir uns die detaillierten technischen Kennwerte der Geräte ansehen,

soll zwecks einer besseren Übersichtlichkeit die Zuordnung zu einer der obigen Gebrauchswertgruppen dargelegt werden.

Hergestellt werden die Typen:

MR 412 4-Spezies-Rechner mit 5 mathematischen Zusatzfunktionen und Speicher

Gruppe der Kombinationsgeräte  
Taschenrechner

MR 411

MR 511 mit Uhr;

MR 4110 4-Spezies-Rechner mit 2 mathematischen Zusatzfunktionen und saldierendem Speicher; Digitaluhr mit Anzeige des Datums, des Wochentages

**Tabelle 1** Technische Daten der Taschenrechner der MK-Serie

Kennwert	Typ					
	MR 412	MR 411	MR 511	MR 4110	MR 609	MR 610
Abmessungen in mm	134 × 70 × 8,5	135 × 68 × 9	95 × 62 × 5,7	138 × 70 × 12/38	134 × 70 × 8,5	144 × 70 × 8,5
Masse in g	ca. 85	ca. 85	ca. 45	ca. 120	ca. 85	ca. 90
Rechen- arten	4 Grund- rechenarten, Konstanten- rechnung	wie MR 412, aber Kon- stantenrech- nung nur bei Multiplikation und Division	wie MR 411	wie MR 411	wie MR 412	wie MR 412
Funktionen	%; $\sqrt{\quad}$ ; $x^2$ , $y^x$ ; $1/x$	%, $\sqrt{\quad}$	%, $\sqrt{\quad}$	%, $\sqrt{\quad}$	wie MR 412; zusätzlich sin, cos, tan, arcsin, arccos, arctan (Winkel in Grad, Rad und Gon), ln, lg, $e^x$ , $10^x$	wie MR 609; zusätzlich sinh, cosh, tanh, $\Delta$ %, $\sqrt[\#]{\quad}$ , $x!$ , Statistikteil (Mittelwert, Standardabwe- ichung, $\Sigma x$ , $\Sigma x^2$ . Anzahl der eingegebenen Werte), 2 Klammer- ebenen
Umrech- nungen	keine	keine	keine	keine	keine	(Grad/Minuten/ Sekunden $\leftrightarrow$ )

							Dezimalgrad; Polarkoordinaten ↔ Kartesische Koordinaten
π-Taste	ja	nein	nein	nein	ja	ja	
Anzahl der angezeigten Stellen	8 + Vorzeichen, Normal- und Exponential-Darstellung	8 + Vorzeichen	8 + Vorzeichen	8 + Vorzeichen	wie MR 412	wie MR 412	8 für Mantisse 2 für Exponent 2 für Vorzeichen
Stromversorgung	2 Steck Silberoxid-Knopfzellen, Format SR 44	wie MR 412	2 Steck Silberoxid-Knopfzellen, Format SR 54	2 Steck Primärelemente, Format R 6	wie MR 412	wie MR 412	wie MR 412
Betriebsdauer mit einem Batteriesatz	ca. 2000 Stunden	ca. 10000 Stunden	ca. 5000 Stunden	ca. 5000 Stunden	wie MR 412	wie MR 412	wie MR 412
weitere Gebrauchswertmerkmale	Automatische Abschaltung nach ca. 6 min.	Digitaluhr mit Anzeige von Datum, Wochentag, Stunde u. Minute; Wecker	wie MR 411	wie MR 411	Automatische Abschaltung nach 6 min	—	

und der Uhrzeit in 12-Stunden-Darstellung, Stoppuhrbetrieb und Wecken möglich.

Die 3 verschiedenen Typen besitzen bei gleichen elektrischen und rechtechnischen Eigenschaften zur Erreichung gezielter Gebrauchswerteigenschaften unterschiedliche Größen. Der MR 511 als kleinstes der drei Geräte besitzt das Scheckkartenformat, der MR 411 das Normalformat und der MR 4110 die Pultform.

MR 609/MR 610 Beides sind wissenschaftliche Taschenrechner.

Der MR 610 ist das höherwertige Gerät, weil man mit ihm eine größere Zahl von Funktionen berechnen kann (insbesondere Statistikteil) und weil er eine um 2 Stellen größere Ziffernanzeige besitzt.

Die Tabelle 1 enthält die wesentlichsten Kennwerte aller genannten Taschenrechner.

In der Tabelle wurde bewußt auf die Darstellung der Speichermöglichkeit, des Vorzeichenwechsels, der verschiedenen Löschmöglichkeiten und der Besonderheiten der Normal- und der Exponentialdarstellung von eingegebenen bzw. ausgegebenen Zahlen verzichtet. Darauf wird im Abschnitt über den Umgang mit Taschenrechnern eingegangen.

Mit den technischen Daten der Taschenrechner wird zwar eine nüchterne Beschreibung der wesentlichsten Gebrauchswerteigenschaften gegeben, aber damit erhält man kein abgerundetes Bild über das jeweilige Produkt. Für den Umgang mit einem Taschenrechner sind neben seinen funktionellen Eigenschaften noch andere Einflüsse zu beachten, die für den Nutzer wichtig sind. So sind zum Beispiel die Größe und die Ablesbarkeit der Ziffern und Symbole der Anzeige, die Größe der Tasten und ihr Abstand zueinander, die deutliche Anbringung der Ziffern und Symbole

auf die Tastatur sowie die Form- und Farbgestaltung des Gerätes, aber auch die Gestaltung und der Inhalt der Bedienungsanleitung bedeutungsvoll.

Bei der Auswahl eines Taschenrechners spielen also eine ganze Reihe von Faktoren eine Rolle, deren Vor- und Nachteile es abzuwägen gilt, um sich für einen bestimmten Typ zu entscheiden. Ein Beispiel soll diese Aussage verdeutlichen.

Wer zum Beispiel einen Rechner mit Uhr benötigt, hat die Qual der Wahl zwischen den 3 Typen MR 411, MR 511 und MR 4110. Die Geräte unterscheiden sich lediglich in ihrer Geometrie, ihre funktionellen Eigenschaften sind ansonsten gleich.

Den Typ MR 511 sollte jemand nutzen, der nur gelegentlich rechnet, aber Wert auf ständige Verfügbarkeit des Gerätes bei geringem Platzbedarf (es genügt bereits eine Hemdtasche) legt. Die Abmessungen der Tasten und ihr geringer Abstand zueinander erfordern »Fingerspitzengefühl« bei der Betätigung.

Sie sind unzweckmäßig für den Dauerbetrieb, jedoch notwendig zur Erreichung der kleinen Abmessungen des Gerätes.

Der MR 411 stellt die Normalausführung dar. Er besitzt Abmessungen, die eine gute Bedienbarkeit ermöglichen und sich daher für eine dauerhafte Benutzung eignen, wobei ein mobiler Einsatz dennoch leicht möglich ist. Für dieses Gerät ist Platz in jeder Aktentasche.

Der MR 4110 wurde speziell für die Benutzung am Arbeitsplatz konstruiert. Durch die Pultform und seinen rutschsicheren Stand ist er stets auf dem Schreibtisch oder dem Labormeißplatz gut bedienbar, und seine Anzeige befindet sich in einem günstigen Blickwinkel. Sozusagen als Abfallprodukt dieser konstruktiven Gestaltung konnten im Sockelteil des Gerätes die kosten-

günstigen, stabförmigen Primärbatterien des Formates R 6 untergebracht werden.

Dieses Beispiel zeigt, daß man sich vor dem Kauf eines Taschenrechners sehr gut überlegen muß, welches Einsatzgebiet des Gerätes den Vorrang hat, um danach den optimalen Typ auszuwählen.

Man muß aber feststellen, daß mit den vorgestellten 6 verschiedenen Gerätetypen nicht alle Wünsche der Taschenrechneranwender erfüllbar sind.

Der Vergleich des zur Zeit vorhandenen Taschenrechnersortimentes mit dem im internationalen Maßstab angebotenen zeigt, daß programmierbare Taschenrechner aus der eigenen Produktion in der DDR fehlen. Statt dessen wurden auf der Leipziger Frühjahrsmesse 1984 erstmals eigene Kleincomputer vorgestellt.

Es wird eingeschätzt, daß mit dieser Geräteart neue Einsatzgebiete der Kleinstrechenteknik erreicht werden, die zu einer Rationalisierung des numerischen Rechnens an vielen Stellen führen wird.

Das vorgestellte Taschenrechnersortiment ist das Ergebnis der bisherigen angestregten Entwicklungsarbeit in mehreren Betrieben des Kombines Mikroelektronik.

Da auf diesem Gebiet ständig weitergearbeitet wird, wird sich das Gerätesortiment kontinuierlich vervollständigen.

### 3. Der Aufbau der Taschenrechner

Mit der Entwicklung und Produktion der neuen Taschenrechner der MR-Serie ist es gelungen, eine Gerätefamilie zu schaffen, die trotz großer Gebrauchswertunterschiede durch ihre funktionalen Möglichkeiten, prinzipiell gleichartige konstruktive Gestaltung aufweist. Dadurch ist es möglich, ihre Her-

stellung mit gleichen Technologien und gleichen oder modifizierten Fertigungsausrüstungen durchzuführen. Hinzu kommt noch, daß oft gleiche Teile in verschiedenen Gerätetypen Verwendung finden.

Das alles bietet die Gewähr dafür, daß Taschenrechner kostengünstig und in großen Stückzahlen hergestellt werden können.

Kernstück eines jeden Taschenrechners und bestimmend für seine rechentechnischen Möglichkeiten ist ein hochintegriertes mikroelektronisches Bauelement, der Taschenrechnerschaltkreis. In ihm laufen alle elektronischen Operationen in kürzester Zeit ab. Seine Fähigkeiten bestimmen weitestgehend die Eigenschaften eines Taschenrechners.

Taschenrechnerschaltkreise sind speziell für diesen Einsatzzweck zugeschnittene Bauelemente der Mikroelektronik.

Die Schaltkreise der 6 vorgestellten Taschenrechner werden mit einer Technologie hergestellt, die eine extrem niedrige Stromaufnahme (nur wenige  $\mu\text{A}$ ) bei einer Betriebsspannung von 3 V sichert. Sie ist unter der Bezeichnung CMOS-Technologie bekannt. Übrigens werden für die 6 verschiedenen Taschenrechner nicht 6, sondern nur 3 verschiedene Taschenrechnerschaltkreise eingesetzt, da alle Uhrenrechner auf dem gleichen Schaltkreis basieren.

Zum Betrieb des Taschenrechnerschaltkreises sind nur noch wenige andere elektronische Bauelemente notwendig. Zu denen gehört auch die Anzeige. Ihre technischen Kennwerte sind ebenfalls sehr wichtig für die Eigenschaften eines Taschenrechners. Verwendet werden in allen Geräten Flüssigkristallanzeigen, auch als LCD bekannt, wobei 4 verschiedene Typen zur Anwendung gelangen. Flüssigkristallanzeigen zeich-

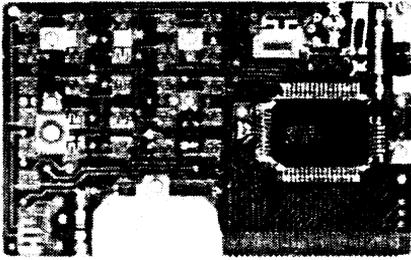


Bild 2a. Bestückte Leiterplatte

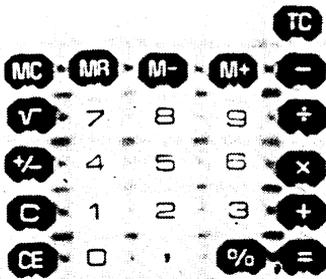


Bild 2b. Tasten

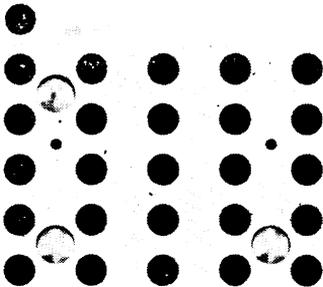


Bild 2c. Schaltmatte mit Lochmuster



Bild 2d. LCD-Anzeige



Bild 2e. Connector

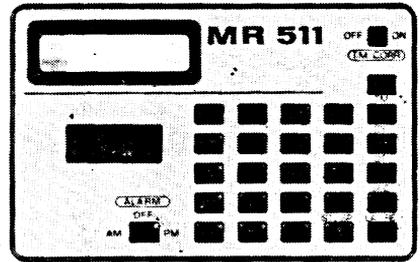


Bild 2f. obere Schalenhälfte

Bild 2g. untere Gehäuseschale

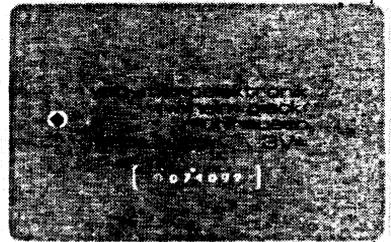


Bild 2. Bauteile des Taschenrechners MR 511

nen sich durch niedrige Stromaufnahme aus, haben eine geringe Einbauhöhe, aber benötigen zum Ablesen der Informationen eine fremde Lichtquelle.

Das Bauelementekonzept der wichtigsten Bauelemente eines Taschenrechners – nämlich Taschenrechnerschaltkreis – Flüssigkristallanzeige – Stromversorgung, sind aufeinander abgestimmt. Dadurch ist es möglich, daß die Rechner MR 412 und MR 609 außer diesen 3 genannten Bauelemen-

ten nur noch 4 und der MR 610 nur noch 3 weitere benötigen.

Bei den Uhren-Rechnern kommen wegen des Uhrenbetriebes noch ein paar hinzu.

Bild 2 enthält alle zum Aufbau des Taschenrechnertyps MR 511 notwendigen Bauteile, die nun kurz erläutert werden sollen. Die anderen Rechner sind prinzipiell gleichartig aufgebaut. Alle Bauelemente, außer Anzeige und Batterie, befinden sich auf einer Leiterplatte (Bild 2a). Diese trägt gleichzeitig Kontaktierungsflächen für die Tastatur und die Schiebeschalter, für die Anzeige und die Batteriekontaktierung. Alle diese Stellen sind zur Gewährleistung einer guten Kontaktsicherheit oberflächenveredelt. Alle übrigen Stellen der Leiterplatte sind durch einen Abdecklack vor Fremdeinflüssen geschützt.

Die Tastatur ist keine separate Baugruppe, sondern sie ist erst nach Montage des gesamten Gerätes funktionsfähig. Sie besteht aus den Tasten (Bild 2b), der Schaltmatte (Bild 2c), der Lochmaske und den Schaltsegmenten auf der Leiterplatte. Kontaktgebend sind dabei die Schaltsegmente der Leiterplatte und die Schaltmatte. Bei Betätigung einer Taste wird der Kontaktpunkt der Schaltmatte auf die Leiterplatte gedrückt und überbrückt dort elektrisch die zugehörigen Schaltsegmente. Die Schaltmatte ist also das funktionsbestimmende Element der Tastatur und ist gleichzeitig kontaktgebend und Rückstellelement für die Taste. Sie ist ein Formteil aus Silikon Gummi, in welches zur Erzielung der elektrischen Leitfähigkeit Rußbestandteile eingelagert wurden. Die Kontaktpunkte sind in einem bestimmten Abstand zueinander, dem Rastermaß, angeordnet.

Nachfolgende Übersicht zeigt die verschiedenen Raster der Taschenrechner.

Raster in mm	angewendet bei Rechnertyp
7 × 10	MR 511
12 × 12	MR 411, MR 4110
10 × 11,5	MR 412, MR 609, MR 610

Das Rastermaß der Schaltmatte und damit der Tasten sowie die Größe der Tasten sind neben den elektrischen Kennwerten der Schaltmatte für die gute und sichere Bedienbarkeit der Tastatur bestimmend.

Wichtig ist weiterhin eine weitere Eigenheit der Schaltmatte, die durch die Formgebung des Kontaktpunktes bestimmend wird. Es handelt sich um den Schnappeffekt. Bei der Benutzung einer Tastatur mit so kurzem Tastenhub wie bei einem Taschenrechner, wird es allgemein als vorteilhaft empfunden, wenn man eine Rückmeldung über eine erfolgreich betätigte Taste erhält. Diese Rückmeldung könnte zum Beispiel bei der Zahleneingabe über eine visuelle Kontrolle der Anzeige erfolgen. Aber das würde zu lange dauern. Besser wird es empfunden, wenn man die Rückmeldung über die Taste selbst erhält. Dazu dient der Schnappeffekt der Schaltmatte. Im Ruhezustand befindet sich der Kontaktpunkt der Schaltmatte in ca. 1,5 mm Abstand vom Schaltsegment der Leiterplatte. Drückt man eine Taste nieder, so spürt man deutlich einen Widerstand. Bei dessen Überwindung springt der Kontaktpunkt schnell auf die Leiterplatte. Der Kontakt wird geschlossen, und der Bediener hat den Eindruck, daß eine sichere Kontaktgabe erfolgt ist. Dieser geschilderte Effekt tritt allerdings bei der Tastatur des MR 511 nicht so deutlich auf, weil bei diesem Gerät der Tastenhub zu klein ist.

Die Lochmaske stellt ein Isolierteil zwischen der Schaltmatte und der Leiterplatte dar, die im Bereich der

Schaltsegmente der Leiterplatte ihre Durchbrüche hat.

Die Anzeige (Bild 2d) ist mit der Leiterplatte über ein Andruckkontaktelement, den Connector (Bild 2e), elektrisch verbunden. Dieser Connector ist elastisch und besteht aus leitfähigen und nicht leitfähigen Segmenten, die im geringen Abstand von ca. 0,4 mm angeordnet sind. Dadurch wird es möglich, die schmalen und eng angeordneten Leiterzüge der Leiterplatte und der Anzeige miteinander zu verbinden.

Das Gehäuse bildet das tragende und schützende Element für das gesamte Innenleben. Es besteht aus 2 Gehäuseschalen (außer beim pultförmigen Gerät MR 4110).

Die dem Benutzer zugewandte obere Schalenhälfte (Bild 2f) besteht aus dem tragenden Element des Rechners, der Trägerplatte, die entweder mit einer schalenförmigen (bei MR 411) oder mit einer plattenförmigen (bei den übrigen 5 Typen) Abdeckung versehen ist. Diese Abdeckung trägt einen Strukturschliff, ist bedruckt und auf die Trägerplatte geklebt.

Die Trägerplatte besitzt die Tasterdurchbrüche, und sie muß verwindungssteif sowie stoß- und kratzfest sein, denn in Verbindung mit der eingeschraubten Leiterplatte sorgt sie für die mechanische Stabilität des Rechners.

Außerdem sind auf ihrer Innenseite Elemente für die Lagefixierung der Anzeige, die Führungen der Schiebeshalter, Befestigungsmöglichkeiten für die Leiterplatte und die Batteriekontakte sowie Positionen für die 2 Knopfzellen vorhanden.

Die Trägerplatte wird zur Erfüllung all dieser Aufgaben aus einem hochwertigen Thermoplast im Spritzgießverfahren hergestellt.

Die untere Gehäuseschale (Bild 2g) besteht aus Aluminium und wird zu-

letzt am Rechner angebracht. Sie muß vom Benutzer des Rechners zum Batteriewechsel entfernt werden.

Beim MR 4110 wird anstelle einer unteren Gehäuseschale ein pultförmiger Plastsockel verwendet. In seinem Inneren befindet sich auch das mit einem Deckel verschlossene Batteriefach.

Die Uhrenrechner verfügen über einen Wecker, dessen Weckzeit über die Tastatur und mit Hilfe des Schiebeshalters AM/PM gesetzt werden kann. Als akustischer Signalgeber ist in den Rechnern ein piezoelektrischer Summer eingesetzt. Dieser wird direkt vom Taschenrechnerschaltkreis elektrisch angesteuert.

Taschenrechner zeichnen sich durch einen sehr gedrungenen Aufbau aus. Nur dadurch ist es möglich, die benötigten handlichen Abmessungen, insbesondere die geringe Dicke, zu erreichen. Gerade die geringe Dicke der Taschenrechner macht diese Geräte so handlich und sichert ihre Flexibilität.

#### **4. Über den Umgang mit Taschenrechnern**

Taschenrechner sind hochwertige elektronische Geräte. Wer ein solches Gerät erwirbt, möchte es für viele Jahre nutzbringend anwenden. Das erfordert aber auch, daß das Gerät sachgerecht und pfleglich behandelt wird. In den Bedienungsanleitungen sind dazu Hinweise gegeben, von denen hier einige kurz dargestellt werden sollen:

- Gerät nicht außerhalb des angegebenen Bereiches der Umgebungstemperatur 0 bis +40°C betreiben, starke Sonnenbestrahlung vermeiden
- Rechner vor starken Erschütterungen und vor Wasser schützen
- keinen starken Druck auf die Anzeige ausüben (Bruchgefahr der Gläser)

- Verunreinigungen mit einem weichen mit Netzmittel (z. B. Fit) angereicherten Tuch entfernen; keine Lösungsmittel verwenden.

Wenn man diese Hinweise beherzigt, wird man lange Freude bei der Benutzung des Rechners haben und froh sein, daß er die oft ermüdende numerische Rechnung schnell und fehlerfrei durchführt. Voraussetzung dafür ist aber, daß man seinen Rechner richtig bedient.

Wie das gemacht wird, steht in der zu jedem Rechner gehörigen Bedienungsanleitung. Es ist deshalb notwendig,

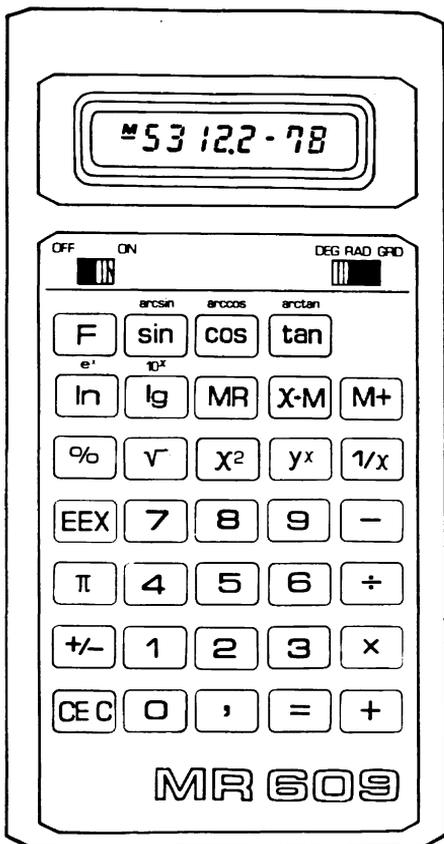
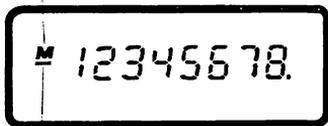


Bild 3. Tastatur des MR 609

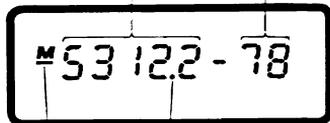
Zeichen „Speicher belegt“



Negatives Vorzeichen

Bild 4. Normaldarstellung von Zahlen

Mantisse Exponent



Negatives Vorzeichen Komma

Bild 5. Exponentialdarstellung von Zahlen

daß man nach dem Erwerb eines Taschenrechners diese Bedienungsanleitung erst einmal aufmerksam liest und am besten den Rechner dabei, wie beschrieben, gleichzeitig handhabt. Die Rechenbeispiele oder bei Uhren-Rechnern die Beispiele für die Bedienung bei Uhrenbetrieb werden helfen, den Rechner nach kurzer Zeit fehlerfrei zu bedienen. Es ist aber keineswegs beschämend, wenn man von Zeit zu Zeit mal wieder einen Blick in die Bedienungsanleitung werfen muß. Bei einfachen Rechnern wird das viel weniger notwendig sein, als zum Beispiel beim MR 610, bei dem man sicherlich nicht täglich alle Funktionen benötigt. Insbesondere bei der Benutzung der Umwandlungen wird sich das als notwendig erweisen.

An Hand der Tastatur (in Bild 3) und der Informationen der Anzeige (in Bild 4 und 5) des Rechners MR 609 sollen noch einige bisher nicht genannte Eigenschaften erläutert werden. Diese treffen weitestgehend auch auf den Rechner MR 412 zu.

Bei der Betrachtung der Tastatur fällt auf, daß nicht für jede Funktion eine separate Taste vorhanden ist, sondern

es liegt bei 5 Tasten eine Doppelbelegung vor (z.B. ln und e<sup>x</sup> für eine Taste).

Diese Doppelbelegung ist notwendig, um eine zu große Abmessung des Rechners zu vermeiden. Besonders auffällig ist das beim MR 610, der bereits 40 Tasten hat, von denen 17 Stück doppelt belegt sind.

Die 2. Tastenfunktion wird dann wirksam, wenn man zunächst die F-Taste und danach die Funktionstaste betätigt.

Ein Beispiel soll das erläutern:

• Soll der natürliche Logarithmus von 3 ermittelt werden, so sind folgende Tasten zu betätigen:

3  $\boxed{\ln}$  Man erhält sofort das Ergebnis 1,0986123.

Soll aber die Funktion e<sup>3</sup> errechnet werden, so ist die Tastenreihenfolge

3  $\boxed{F}$   $\boxed{\ln}$  und man erhält 20,085537.

Die Funktionsberechnung erfolgt also so, daß zunächst der Zahlenwert eingegeben und danach die Funktionstaste betätigt wird. Ausnahmen von dieser Regel bilden y<sup>x</sup> und %.

Bei y<sup>x</sup> wird wie folgt gearbeitet:

falls an einigen Beispielen erläutert werden (vgl. S. 15 oben).

Beim Rechner MR 610 ist statt der  $\boxed{\%}$ -Taste eine über die  $\boxed{F}$ -Taste erreichbare  $\boxed{\Delta \%}$ -Taste vorhanden, mit der man noch andere Möglichkeiten der Prozentrechnung hat, die hier allgemein erläutert werden sollen:

Tastenbetätigung	mathematische Bedeutung
a + b $\boxed{F}$ $\boxed{\Delta \%}$	$\frac{a+b}{b} \cdot 100$
a - b $\boxed{F}$ $\boxed{\Delta \%}$	$\frac{a-b}{b} \cdot 100$
a × b $\boxed{F}$ $\boxed{\Delta \%}$	$\frac{a \cdot b}{100}$
a : b $\boxed{F}$ $\boxed{\Delta \%}$	$\frac{a}{b} \cdot 100$

Man erkennt, daß man prozentuale Zu- und Abschläge mit diesem Rechner nicht so einfach berechnen kann wie beim MR 609. Dazu muß man den Speicher benutzen.

Doch bleiben wir weiter beim MR 609. Die Bilder 4 und 5 zeigen Zahlendar-

Aufgabe	Tastenbetätigung	Anzeige
2 <sup>4</sup>	2 $\boxed{y^x}$ 4 $\boxed{=}$	16.
2,5 <sup>4</sup>	2 $\boxed{,}$ 5 $\boxed{y^x}$ 4 $\boxed{=}$	39,0625
2,5 <sup><math>\frac{1}{4}</math></sup>	2 $\boxed{,}$ 5 $\boxed{y^x}$ 4 $\boxed{1/x}$ $\boxed{=}$	1,25743

Das letzte Beispiel verdeutlicht auch gleich, daß der Exponent nicht ganzzahlig sein muß. In diesem Beispiel wurde dargestellt, wie man die Kehrwerttaste mit benutzen kann. Es wäre ebenso möglich gewesen, als Exponent den Wert 0,25 einzugeben. Die Benutzung der %-Taste soll eben-

stellungen der Anzeige in 2 verschiedenen Darstellungsarten. Da ist zunächst die Normaldarstellung in Bild 5. Diese Art der Darstellung tritt auf, wenn der Betrag der eingegebenen Zahlen oder Ergebnisse im Bereich  $\geq 10^{-7}$  und  $< 10^8$  liegen, oder anders dargestellt  $0,0000001 \leq |x| \leq 99999999$ . Da-

## Prozentrechnung

Aufgabe	Tastenbetätigung	Anzeige
15% von 500	500 $\times$ 15 $\%$ $=$	75.
Wieviel Prozent sind 30 von 150?	30 $\div$ 150 $\%$ $=$	20.

## Prozentuale Zu- und Abschläge

Aufgabe	Tastenbetätigung	Anzeige
15% Steigerung von 350	350 $+$ 15 $\%$ $=$	402,5
10% Abschlag von 250	250 $-$ 10 $\%$ $=$	225

## Aufeinanderfolgende Zu- und Abschläge

Aufgabe	Tastenbetätigung	Anzeige
10% Steigerung von (15% Steigerung von 350)	350 $+$ 15 $\%$ $+$ 10 $\%$ $=$	442,75
10% Abschlag von (15% Abschlag von 350)	350 $-$ 15 $\%$ $-$ 10 $\%$ $=$	267,75

bei kann also die Zahl  $x$  positives oder negatives Vorzeichen haben.

Überschreitet der Betrag der Zahl  $x$  obigen Bereich im Ergebnis einer Rechnung, so stellt der Rechner die Zahl automatisch in Exponentialdarstellung, entsprechend Bild 5 dar. Die dargestellte Zahl bedeutet  $-5312,2 \cdot 10^{-78}$ . Es werden also die Mantisse mit Vorzeichen sowie der Exponent mit Vorzeichen angezeigt. Die Basis 10 kann nicht sichtbar gemacht werden, weil der Taschenrechnerschaltkreis und die Anzeige dafür nicht ausgelegt sind.

Beim höherwertigen Rechner MR 610 wird allerdings die 10 dargestellt.

Die Exponentialdarstellung tritt aber nicht nur im Ergebnis einer Rechnung auf, sondern man kann sie bei der Zahleneingabe ebenfalls wählen. Dazu gibt man zunächst die Mantisse ein, dann betätigt man die  $\boxed{\text{EEX}}$ -Taste und danach kann man 2 Stellen für den Exponenten eingeben. Sowohl die Mantisse als auch der Exponent können dabei auch ein negatives Vorzeichen erhalten über die  $\boxed{+/-}$ -Taste.

Ein paar Beispiele sollen diese Eingabeart verdeutlichen (vgl. S. 16 oben).

Da der Exponent 2 Stellen hat, kann

Eingezugende Zahl	Tastenbetätigung	Anzeige
$5 \cdot 10^7$	5 <b>EEX</b> 7	5. 07
$5 \cdot 10^{-7}$	5 <b>EEX</b> <b>+/-</b> 7	5. -07
$-5 \cdot 10^{-7}$	5 <b>+/-</b> <b>EEX</b> <b>+/-</b> 7	-5. -07

der Rechner Zahlen im Bereich von  $10^{-99}$  bis  $10^{99}$  darstellen.

Der MR 609 hat aber auch die Eigenschaft, daß er die Exponentialdarstellung dann wählt, wenn das Ergebnis einer Rechnung im Betrag zwischen 0 und 1 liegt, aber die 8 Stellen der Anzeige nicht ausreichen. So zeigt er bei der Rechnung  $2:3$  nicht 0,6666666, sondern in Exponentialschreibweise das Ergebnis 6,6666-01 an.

Man muß also bei Benutzung dieses Rechners mit der Exponentialdarstellung umgehen können.

Wir wollen uns nun den Speichertasten des MR 609 zuwenden, die folgende Bedeutung haben:

**M +** Die in der Anzeige dargestellte Zahl wird zum Speicherinhalt addiert. Soll die Zahl vom Speicherinhalt subtrahiert werden, so muß zuvor die **+/-** Taste betätigt werden.

**x → M** Die in der Anzeige dargestellte Zahl wird gespeichert, wobei eine eventuell vorher abgespeicherte Zahl überschrieben wird.

**MR** Der Speicherinhalt wird angezeigt. Wenn der Speicher belegt ist, so wird das in der Anzeige durch das Symbol M dargestellt.

Will man den Inhalt des Speichers löschen, so benutzt man dazu bei vielen Rechnern die Taste **MC**. Beim MR 609 ist eine solche Taste nicht vorhanden. Das Löschen des Speicherinhaltes geschieht dadurch, daß man eine Null eingibt und dann die Taste **x → M** betätigt. Damit ist der Speicher gelöscht, d.h., sein Inhalt ist Null und das Symbol M in der Anzeige verlischt ebenfalls.

Ein Beispiel soll den Umgang mit den Speichertasten erläutern (s. unten).

Der Speicher wird immer dann benötigt, wenn man Summen oder Differenzen von Zwischenergebnissen bilden muß. Man erspart sich dadurch unnötiges Notieren von Zahlenwerten. Es gibt sehr häufig Anwendungsfälle in der Praxis, in denen der Speicher benötigt wird.

Der Inhalt des Speichers wird aber nicht nur dann gelöscht, wenn eine Null eingegeben wird, sondern auch beim

Aufgabe	Tastenbetätigung	Anzeige
$123 \cdot 45,6$	123 <b>×</b> 45 <b>,</b> 6 <b>=</b> <b>M +</b>	M 5608,8
$+ 789 \cdot 12,3$	789 <b>×</b> 12 <b>,</b> 3 <b>=</b> <b>M +</b>	M 9704,7
$- 25,8 \cdot 36,9$	25 <b>,</b> 8 <b>×</b> 36 <b>.</b> 9 <b>=</b> <b>+/-</b> <b>M +</b>	M - 952,02
14361,48	<b>MR</b>	M 14361,48

Ausschalten der Stromversorgung verliert der Speicher seinen Inhalt.

Bevor man den Speicher benutzt, sollte man sich angewöhnen, durch Blick auf die Anzeige zu kontrollieren, ob der Speicher tatsächlich leer ist. In diesem Fall ist das Symbol »M« auf der Anzeige nicht zu sehen.

Nun wollen wir eine weitere Eigenschaft der Taschenrechner kennenlernen, die dem Benutzer sehr hilfreich sein kann und ihm Zeit bei der Lösung seiner mathematischen Probleme spart. Gemeint ist die Eigenschaft der Taschenrechner, die unter dem Begriff Konstantenrechnung in den technischen Daten der Geräte erwähnt wird.

Die Konstante ist dabei nicht etwa eine Größe wie sie aus der Physik oder Chemie bekannt ist (z.B. Gravitationskonstante), sondern eine Größe,

die in einer Folge von Rechnungen immer wiederkehrend vorkommt. Das kann zum Beispiel der Preis eines zu verkaufenden Gegenstandes sein, der mit der verkauften Stückzahl multipliziert werden muß, um den Rechnungsbetrag zu erhalten.

Die Konstantenrechnung und ihre Handhabung ist bei den verschiedenen Gerätetypen auch verschieden.

So verfügen die Rechner MR 411, MR 511 und MR 4110 über die Möglichkeit der Konstantenrechnung bei Multiplikation und Division, während bei den Geräten MR 412, MR 609 und MR 610 die Konstantenrechnung bei allen 4 Grundrechenarten möglich ist.

Wir wollen uns diese günstige Art der Berechnung wieder an Rechenbeispielen des MR 609 ansehen. Die Konstante ist daher unterstrichen.

Aufgabe	Tastenbetätigung	Anzeige
$2 + \underline{3} + \underline{3} = 8$	2 <input type="text" value="−"/> 3 <input type="text" value="="/> <input type="text" value="="/>	8
$4 + \underline{3} = 7$	4 <input type="text" value="="/>	7
$9 - \underline{4} - \underline{4} = 1$	9 <input type="text" value="−"/> 4 <input type="text" value="="/> <input "="" type="text" value="="/>	1
$8 - \underline{4} = 4$	8 <input "="" type="text" value="="/>	4
$5 \cdot \underline{3} \cdot \underline{3} = 45$	5 <input type="text" value="×"/> 3 <input type="text" value="="/> <input "="" type="text" value="="/>	45
$2 \cdot \underline{3} = 6$	2 <input "="" type="text" value="="/>	6
$24 : \underline{2} : \underline{2} = 6$	24 <input type="text" value="÷"/> 2 <input type="text" value="="/> <input "="" type="text" value="="/>	6
$22 : \underline{2} = 11$	22 <input "="" type="text" value="="/>	11
$2^{\underline{3}} = 8$	2 <input type="text" value="y^x"/> 3 <input "="" type="text" value="="/>	8
$5^{\underline{3}} = 125$	5 <input "="" type="text" value="="/>	125

Man erkennt, daß diese Eigenschaft der Taschenrechner-Tastenbetätigungen und damit Fehlermöglichkeiten erspart. Man muß aber beachten, daß die Konstante bei manchen Rechnern als erster oder zweiter Operator eingegeben werden muß, je nachdem, ob es sich um Multiplizieren oder Dividieren handelt. Das ist bei den Rechnern MR 411, MR 511 und MR 4110 der Fall. Das nachfolgende Beispiel soll diese Feststellung verdeutlichen.

In den Bedienungsanleitungen der Taschenrechner sind stets eine Reihe von Rechenbeispielen enthalten, die das gleiche Ziel verfolgen. Sie sind überschaubar dargestellt und sollen dem Anwender eine nützliche Hilfe sein beim Umgang mit seinem Rechner. Gerade bei selten benutzten Funktionen lernt man sein Gerät dann am schnellsten wieder zu bedienen, wenn man sich die Rechenbeispiele dazu ansieht. Nun haben wir uns mit den grundsätz-

Aufgabe	Tastenbetätigung	Anzeige
$15 \times 22 = 330$	15 $\times$ 22 $=$	330.
$15 \times 60 = 900$	60 $=$	900.
$15 \times 35 = 525$	35 $=$	525.
$12 : 3 = 4$	12 $\div$ 3 $=$	4.
$15 : 3 = 5$	15 $=$	5.
$18 : 3 = 6$	18 $=$	6.

Bei obigen Rechnern kann man mit Hilfe der Konstantenrechnung auch potenzieren.

Das Quadrat einer Zahl  $a$  errechnet man mit folgender Tastenbetätigung:

$a \times =$ . Und die Aufgabe  $1,5^5 = ?$

wird wie folgt gelöst:

1,5  $\times = = = =$ .

Das Ergebnis lautet 7,59375.

Die vielen Beispiele sollten dem Leser zeigen, wie man grundsätzlich mit dem jeweiligen Taschenrechnertyp umgehen muß, um richtige Rechenergebnisse zu erhalten.

lichen Eigenschaften von Taschenrechnern der DDR-Produktion befaßt. Jeder Leser, der eines dieser verschiedenen Geräte erwerben möchte, sollte sich genau überlegen, welchem Typ er den Vorrang gibt. Deshalb sollen als kleine Hilfestellung ein paar Hinweise für den Anwendungsbereich der verschiedenen Geräte gegeben werden.

### Taschenrechnertyp und Anwendung

**MR 511, MR 411, MR 4110** dort, wo neben der Rechenmöglichkeit auch Uhr und Wecker benötigt werden, also beim Sport, bei der Arbeitsnormung, zum alltäglichen Gebrauch, z. B. als Kurzzeitwecker.

Der **MR 511** ist der kleinste der 3 Uhrenrechner. Er ist auf Grund der kleinen Tasten nicht für Dauerbetrieb zu empfehlen.

Der **MR 411** ist der Normalrechner dieser Dreiergruppe, und der **MR 4110** sollte auf Grund seiner Pultform für weitestgehend stationären Betrieb Anwendung finden. Aber seine Abmessungen sind immer noch so günstig, daß man ihn in jeder Aktentasche bequem transportieren kann.

**MR 412** im Haushalt und im Berufsleben, wenn man keine höheren mathematischen Funktionen benötigt.

**MR 609, MR 610** Schüler der POS<sup>1)</sup>, der EOS und der Berufsschulen; Studenten der technischen Fachrichtungen; Labors, Entwicklungsstellen, Konstruktionsbüros; an Hoch- und Fachschulen.

Dieses hier beschriebene Taschenrechnersortiment ist nicht als endgültig zu betrachten. Entsprechend der technischen und technologischen Entwicklung wird es vervollständigt. Dem interessierten Beobachter wird empfohlen, die Fachliteratur auf diesem Gebiet aufmerksam zu studieren oder anläßlich von Besuchen der Leipziger

<sup>1)</sup> Ab Schuljahr 1985/86 wird in den 7. Klassen der Schulrechner SR 1 eingesetzt.

Messe den neuesten Stand der Erweiterung des Sortimentes kennenzulernen.

Der Taschenrechner als hochwertiges elektronisches Konsumgut hat in den letzten Jahren eine rasante Entwicklung durchgemacht und ist weltweit sehr stark verbreitet. Er wird mehr und mehr zu einer Selbstverständlichkeit im Beruf und im Heim. Man möchte und kann nicht mehr auf ihn verzichten. Die technische Entwicklung auf dem Gebiet der Kleinrechner bleibt aber nicht stehen.

Neben der weiteren Verbreitung der programmierbaren Taschenrechner zeigt sich bereits jetzt eine neue Geräteart, die sich international großer Beliebtheit erfreut. Es sind die Klein- und Personalcomputer, die Bedeutung erlangen werden für Schule, Beruf, Heim und Hobby.

Autor:

*Dipl.-Ing. Dieter Ehram*

DDR 5700 Mühlhausen  
Pfafferode 116

Haupttechnologe im  
VEB Mikroelektronik  
„Wilhelm Pieck“ Mühlhausen  
im VEB Kombinat Mikroelektronik

# Polycomputer 880 – Anwendung und Erweiterungsmöglichkeiten



Das Mikrorechnerlernsystem Polycomputer 880 wird seit 1982 vom VEB Polytechnik Karl-Marx-Stadt produziert und hat mittlerweile eine recht große Verbreitung gefunden. Die technischen Unterlagen des Grundgeräts sind in [1] zu finden, daher sollen hier nur die wichtigsten Merkmale dieses Geräts angeführt werden (Bild 1):

– CPU U880 mit 921,6 kHz Taktfrequenz (7,3728 MHz-Quarz)

- 1 KByte RAM ( $8 \times U202$ )
- 2 KByte ROM (U505) mit Monitorprogramm
- 2 freie Fassungen für ROM/EPROM U505/U555 (u. U. auch 2716, 2732)
- 8stellige Siebensegmentanzeige
- 27 nichtrastende Tasten, davon vier zur Hardwaresteuerung (Reset, Monitorunterbrechung, Einzelzyklusbetrieb, Zyklusschritt)

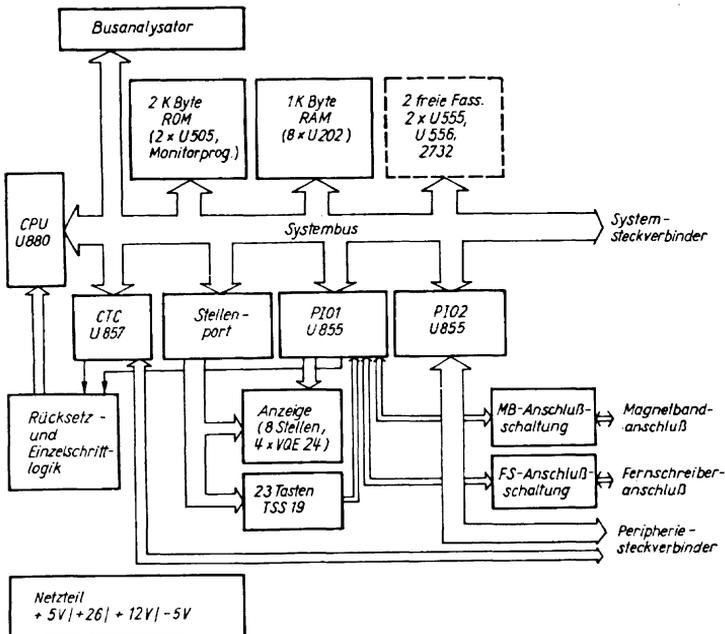


Bild 1  
Blockschaltbild  
Polycomputer 880

- 2 PIOs U855 (davon eine vollständig für den Anwender frei)
- 1 CTC U857 (Kanäle 1-3 für den Anwender)
- Busanalysator zur binären Anzeige aller Adreß-, Daten- und Steuerbusleitungen; Einzelzyklusbetrieb möglich
- Systembus für Erweiterungen herausgeführt
- Magnetbandanschluß 1200 Bit/s (450 KByte je K60-Kassette!)
- Fernschreiberanschluß (Zweidraht-Stromschleife 40 mA)
- interne Netzteile +5 V/+26 V/+12 V/-5 V mit größeren Reserven
- kompakter, gut transportfähiger Aufbau (Koffer)
- Anleitungs- und Unterlagenmaterial (u.a. Datensammlung aller wesentlichen Mikrorechner-Schaltkreise; Stromlaufpläne; Quellcode des Monitorprogramms und verschiedener Zusatzprogramme für Selbsttest usw.).

Im Unterschied zu neueren Hobbycomputern ist am Grundgerät kein Bildschirmanschluß vorgesehen (man kann ihn aber nachrüsten); ebenso ist die RAM-Kapazität des Grundgeräts relativ klein. Die Ausstattung des Geräts ist an der Hauptanwendung als Mikrorechnerlernsystem zum Kennenlernen der Funktionsweise der Mikrorechnerhardware und ihres Zusammenwirkens mit (maschinennaher) Software orientiert. Bei dieser Anwendung ist die bereits im Grundgerät enthaltene Busanalysatorfunktion sehr wertvoll; diese ist in anderen Systemen nur durch mehr oder weniger aufwendige Zusätze oder überhaupt nicht möglich.

Das System gestattet die Eingabe und den Test von Programmen auf Maschinencode-niveau, d.h. in hexadezimaler Darstellung. Der Programmtest wird durch die Funktionen des in Festwertspeichern mitgelieferten Monitorprogramms unterstützt:

- Speicheranzeige und -änderung mit Weiterschalten vor/zurück
- Programmstart mit Prüfpunkt(en)
- schrittweise Programmausführung (befehls- oder zyklusweise)
- Registeranzeige und -änderung
- E/A-Kanäle lesen und schreiben
- Datenblöcke im Speicher verschieben, Speicher mit konstanten Daten füllen
- Abspeicherung und Laden von Speicherinhalten zum/vom Magnetband mit Fehlersicherung (10 Sekunden für 1 KByte)

Neben der Anwendung in der maschinennahen Ausbildung ist das Gerät auch gut für kleine Prozeßsteuerungen, Meß-, Prüf- und Rationalisierungsmittel geeignet. Die Kompaktheit des Systems ist hier vorteilhaft; für viele solcher Anwendungen sind die Bedien- und Anzeigemöglichkeiten des Grundgeräts ausreichend.

Die Einsatzmöglichkeiten lassen sich durch verschiedene Zusätze beträchtlich erweitern. Diese Zusatzeinrichtungen können am 58poligen Systemsteckverbinder (Rechnerbus) oder am 26poligen Peripheriesteckverbinder (peripherieseitige Anschlüsse von PIO und CTC) angeschlossen werden. Nachfolgend sollen einige dieser Zusätze vorgestellt werden.

### E/A-Experimentiermodul

Diese Baugruppe ermöglicht ein intensives Kennenlernen und Experimentieren mit den Schaltkreisen U855-PIO und U857-CTC sowie die Arbeit mit analogen und digitalen Prozeßschnittstellen. Die Baugruppe wurde in [2] beschrieben; sie nutzt PIO und CTC des Grundgeräts (Anschluß am Peripheriesteckverbinder). Der Stromlaufplan (Bild 2) läßt folgende wesentliche Bestandteile erkennen:

- Lautsprecher oder magnetische Hörkapsel zur Tonausgabe

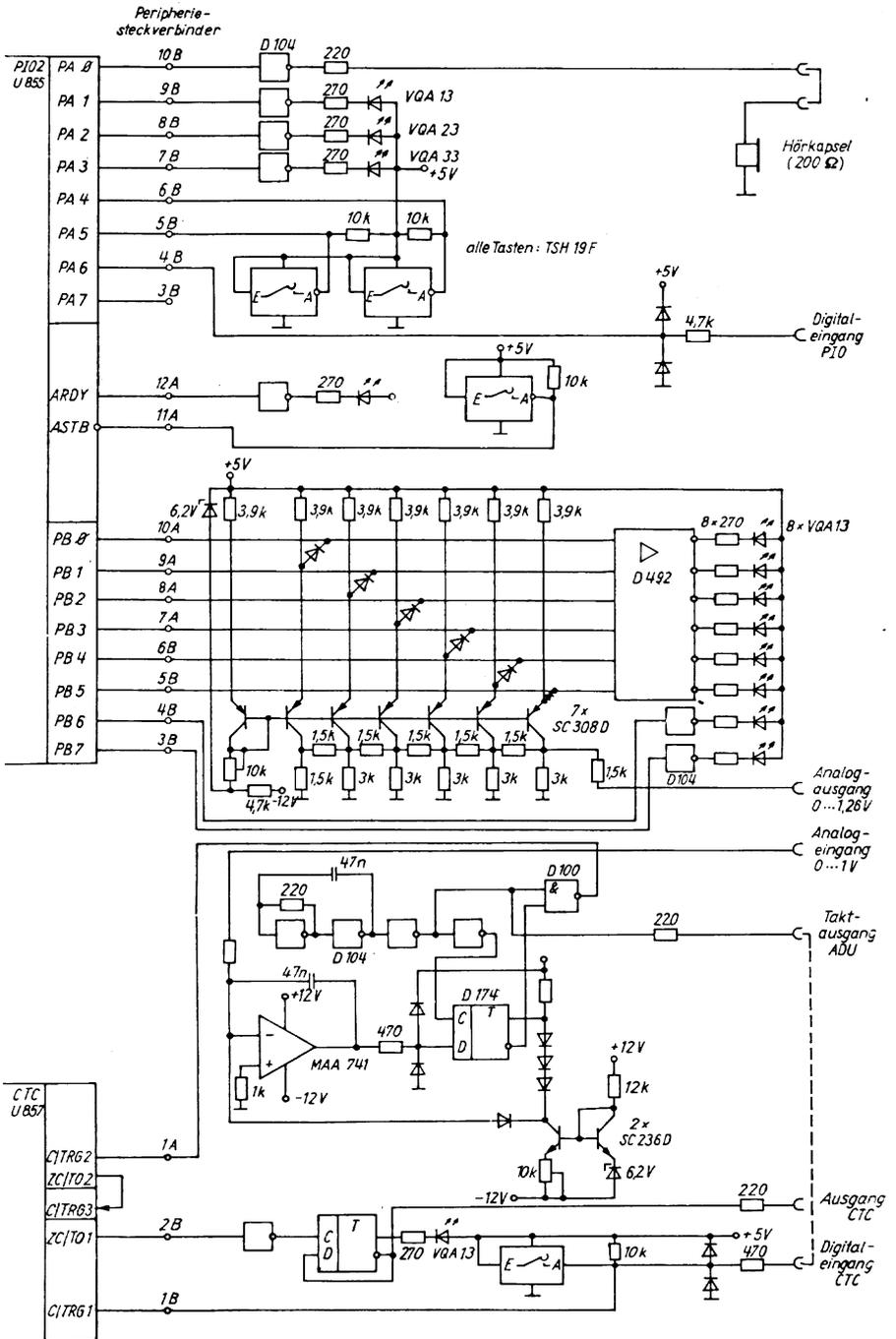


Bild 2. Stromlaufplan E/A-Experimentiermodul

- 4 Tasten (zwei PIO-Datenleitungen, PIO-STB, CTC-Zählengang)
- 11 Leuchtdioden
- D/A-Wandler (6 Bit-Momentanwertumsetzer)
- A/D-Wandler (Spannungs/Frequenz-Wandler mit Frequenzmessung durch CTC; Genauigkeit  $8 \dots 12$  Bit je nach den eingesetzten Bauelementen).

### Bildschirmanschluß

Diese ebenfalls in [2] bereits vorgestellte Baugruppe wird an den Systembus angeschlossen und hat folgende Merkmale:

- 16 Zeilen mit je 64 Zeichen werden dargestellt

- Groß- und Kleinbuchstaben, alle Sonderzeichen, Pseudografik
- Video- oder VHF-Anschluß

### Tastaturanschluß

Eine alphanumerische Tastatur kann über den Peripheriesteckverbinder an beide Ports der Anwender-PIO im Polycomputer angeschlossen werden. Der Stromlaufplan (Bild 3) zeigt, daß außer den Tasten fast keine weiteren Bauelemente erforderlich sind. Die Tasten selbst brauchen nur einen Schließler zu haben, neben Mikrotastern sind auch Leitgummitastaturen o.ä. verwendbar; ein eventuelles Kontaktprellen wird vom Programm unterdrückt.

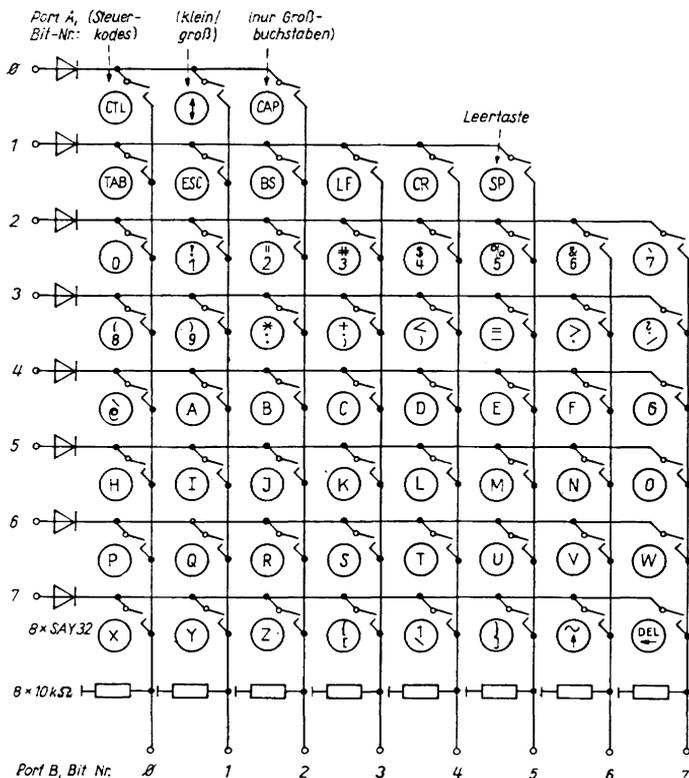


Bild 3. Tastaturverdrahtung

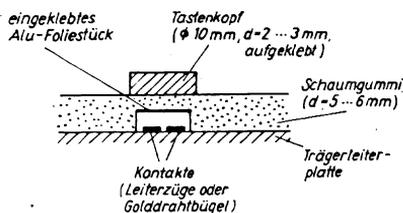


Bild 4. Vorschlag für eine einfache Tastaturmechanik

Bild 4 zeigt eine Aufbauvariante für die Tastaturmechanik, die sehr billig ist und ohne spezielle Materialien auskommt. Über der Leiterplatte mit den Kontakten ist ein Schaumgummistück angeordnet, das an jeder Tastenposition eine Vertiefung erhalten hat (LötKolben), in die ein Stück Alufolie eingeklebt ist. Auf der dem Bediener zugewandten Seite können »Tasten« in Form kleiner Plasteplättchen aufgeklebt werden. Falls nicht die Möglichkeit besteht, die Kontaktflächen auf der Leiterplatte zu veredeln, sollten im Interesse einer langen Funktionssicherheit besser kleine Bügel aus Golddraht als Kontakte genutzt werden.

Die Tastatur arbeitet so, daß vom Rechner aus (Port A der Anwender-PIO) zunächst eine Zeilenleitung 1-Pegel erhält, alle übrigen 0-Pegel. Anschließend liest der Rechner die Belegung der Spaltenleitungen über Port B der PIO. Falls eine Taste in der gerade aktiven Zeile gedrückt ist, wird 1-Pegel auf einer Spaltenleitung erkannt.

Die Anordnung der Tasten in der Matrix erfolgte so, daß ein einfaches Programm (Bild 5) ohne Kodetabelle direkt den 7-Bit-Kode (ASCII) einer gedrückten Taste ermitteln kann. Das dargestellte Programm realisiert eine Tastaturabfrage und -auswertung mit folgenden Sonderfunktionen:

– »Shift«-Taste zur Umschaltung Groß/Kleinbuchstaben bzw. Ziffern/Sonderzeichen

– »Control«-Taste zur Ausgabe der Steuerkodes (00-1Fh) anstelle der zusätzlich mitbetätigten Buchstaben-taste

– »Repeat«-Taste zur ständigen Aus-sendung eines Zeichens

– »Caps-Only«-Taste: Modumschal-tung bei jeder Betätigung, im »Caps-Only«-Modus werden nur die Groß-buchstaben erzeugt (bequemere Ar-beit mit Basic, Assembler . . .)

Die gleichzeitige Betätigung mehrerer Tasten wird ignoriert; bei Erkennen des Schließens oder Öffnens einer Taste wird eine Entprellzeit von etwa 10 ms eingefügt.

### Zusatzspeicher

Der Speicherbereich des Geräts läßt sich durch Erweiterungsbaugruppen, die an den Systembus angeschlossen werden wesentlich vergrößern. Der geringst-Aufwand im Verhältnis zur Speicherkapazität ergibt sich dabei mit dynamischen RAM-Schaltkreisen (U256 – 16 Kbit  $\times$  1). Mit 16 Speicherschaltkreisen U256 kann eine 32-KByte-Speichererweiterung realisiert werden; die Ansteuerlogik benötigt 6 . . . 8 weitere (TTL-) Schaltkreise und kann z. B. nach [3] aufgebaut werden.

Ein statischer RAM ist bezüglich der elektrischen Bedingungen unkritischer (und daher einfacher aufzubauen und in Betrieb zu setzen), außerdem kann damit der Maschinenzklusbetrieb weiter uneingeschränkt genutzt werden (im dynamischen RAM kommt es in längeren WAIT-Phasen zum Datenverlust wegen des ausbleibenden Auf-frischens).

Bild 6 zeigt eine 4-KByte-Zusatzspeicherbaugruppe mit 8 statischen RAM-Schaltkreisen U214/U224 (1 KBit  $\times$  4). Die Speicherbaugruppe

ADR	OBJ-KODE	ANW	ALPHATASTATUR QUELLANWEISUNG	SEITE 1 POLY880-ASM 1.0
		1	;TASTATURABFRAGE FUR PC880	
		2	;LIEFERT NZ,WENN TASTE ERKANNT; KODE IN A	
		3	TASFRA:	
0000	C5	4	PUSH	BC
0001	D5	5	PUSH	DE
0002	E5	6	PUSH	HL
0003	219900	7	LD	HL,TASZUS
0006	012802	8	LD	BC,226H ;2-ANF.SPALTE
		9	NEXSPA:	
0009	78	10	LD	A,B
000A	D384	11	OUT	(SPIOD),A
000C	DB86	12	IN	A,(ZPIOD)
000E	A7	13	AND	A
000F	200A	14	JR	NZ,TASACT ;GEDRUECKTE TASTE
0011	CB00	15	RLC	B
0013	3866	16	JR	C,TASFER ;KEINE TASTE BETAETIGT
0015	79	17	LD	A,C
0016	C608	18	ADD	A,8
0018	4F	19	LD	C,A
0019	18EE	20	JR	NEXSPA
		21	TASACT:	
001B	0F	22	RRCA	
001C	3803	23	JR	C,CFER
001E	0C	24	INC	C
001F	18FA	25	JR	TASACT
		26	CFER:	
0021	3E01	27	LD	A,1
0023	D384	28	OUT	(SPIOD),A
0025	DB86	29	IN	A,(ZPIOD)
0027	47	30	LD	B,A ;UNSCHALTSTASTEN
0028	CB58	31	BIT	REPEAT,B
002A	2802	32	JR	Z,NOREP
002C	CBB6	33	RES	GDR,(HL)
		34	NOREP:	
002E	79	35	LD	A,C
002F	FE30	36	CP	30H
0031	382C	37	JR	C,SONTAS
0033	FE3C	38	CP	3CH
0035	3818	39	JR	C,ZIFTAS
0037	FE40	40	CP	40H
0039	381C	41	JR	C,ZEITAS
		42	;BUCHSTABENTASTE	
003B	CB40	43	BIT	CTRL,B
003D	2804	44	JR	Z,NORMAL
003F	CBB7	45	RES	6,A
0041	1822	46	JR	KODFER
		47	NORMAL:	
0043	CB7E	48	BIT	CAPS,(HL)
0045	281E	49	JR	Z,KODFER
0047	CB48	50	BIT	SHIFT,B
0049	201A	51	JR	NZ,KODFER
004B	CBEF	52	SET	5,A
004D	1816	53	JR	KODFER
		54	ZIFTAS:	
004F	CB48	55	BIT	SHIFT,B
0051	2812	56	JR	Z,KODFER
0053	CBA7	57	RES	4,A ;ZWEITBELEGUNG
0055	180E	58	JR	KODFER

ADR	OBJ-KODE	ANW	ALPHATASTATUR QUELLANWEISUNG	SEITE 2 POLY880-ASM 1.0
		59	ZEITAS:	
0057	CB48	60	BIT	SHIFT,B
0059	200A	61	JR	NZ,KODFER
005B	CBA7	62	RES	4,A
005D	1806	63	JR	KODFER
		64	SONTAB:	
005F	116900	R 65	LD	DE,SONTAB-28H
0062	83	66	ADD	A,E
0063	5F	67	LD	E,A
0064	1A	68	LD	A,(DE)
		69	KODFER:	
0065	CB76	70	BIT	GEDR,(HL)
0067	200J	71	JR	NZ,KEINEU
		72	;NEUE TASTE,	20MS WARTEN
0069	CBF6	73	SET	GEDR,(HL)
006B	CD8600	R 74	CALL	WAIT10
006E	FE80	75	CP	80H ;CAPS-ONLY-TASTE?
0070	2005	76	JR	NZ,RETU
0072	7E	77	LD	A,(HL)
0073	EE80	78	XOR	CAPHAS
0075	77	79	LD	(HL),A
		80	KEINEU:	
0076	AF	81	XOR	A
		82	RETU:	
0077	E1	83	POP	HL
0078	D1	84	POP	DE
0079	C1	85	POP	BC
007A	C9	86	RET	
		87	TASFER:	
007B	CB76	88	BIT	GEDR,(HL)
007D	28F7	89	JR	Z,KEINEU
		90	;GERADE LOSGELASSEN	
007F	CEB6	91	RES	GEDR,(HL)
0081	CD8600	R 92	CALL	WAIT10
0084	18F0	93	JR	KEINEU
		94	;	
		95	WAIT10:	
0086	F5	96	PUSH	AF
0087	11E402	97	LD	DE,20000/27 ;20MS
		98	WAITS:	
008A	1B	99	DEC	DE
008B	7A	100	LD	A,D
008C	B3	101	OR	E
008D	20FB	102	JR	NZ,WAITS
008F	F1	103	POP	AF
0090	C9	104	RET	
		105	;SONDERZEICHENTABELLE	
		106	SONTAB:	
0091	09	107	DEFB	09H ;TABULATOR
0092	1B	108	DEFB	1BH ;ESCAPE
0093	08	109	DEFB	08H ;BS: RUECKWAERTSSCHRITT
0094	0A	110	DEFB	0AH ;LF: ZEILENSCHALTUNG
0095	0D	111	DEFB	0DH ;CR: WAGENRUECKLAUF
0096	20	112	DEFB	20H ;SP: LEERZEICHEN
0097	7F	113	DEFB	7FH ;US1: RESERVE
0098	80	114	DEFB	80H ;CAPS ONLY
		115	;	
		116	; RAM - B E R E I C H	

ADR	OBJ-KODE	ANW	ALPHATASTATUR QUELLANWEISUNG	SEITE POLY880-ASM 1.0
0099		117	TASZUS DEFS 1	;TASTATURZUSTAND
009A		118	KURSOR DEFS 2	;KURSORPOS.
		119	;	
		120	;BITS IN TASZUS	
		121	GEDR EQU 6	
		122	CAPS EQU 7	
		123	;BITS DER UMSCHALTSTASTEN	
		124	CTRL EQU 0	
		125	SHIFT EQU 1	
		126	CONLY EQU 2	
		127	REPEAT EQU 3	
		128	;	
		129	CAPMAS EQU 80H	
		130	;	
		131	;PIO-ADRESSEN	
		132	SPIOD EQU 84H	
		133	SPIOC EQU 85H	
		134	ZPIOD EQU 86H	
		135	ZPIOC EQU 87H	

Bild 5. Tastaturabfrageprogramm

belegt den Adreßbereich 8000h – 8FFFh; im Adreßbereich 9000h – 9FFFh sind wegen der einfachen Schaltungstechnik die Datenbustreiber ebenfalls aktiv, dieser Bereich ist also nur auf dieser Baugruppe unter Benutzung der Bustreiber noch belegbar (z.B. durch acht weitere RAM-Schaltkreise).

Bei der Benutzung von RAM-Schaltkreisen mit getrennten Datenein- und -ausgängen oder von EPROMs können die Datenbustreiber (2 × DS8216) entfallen.

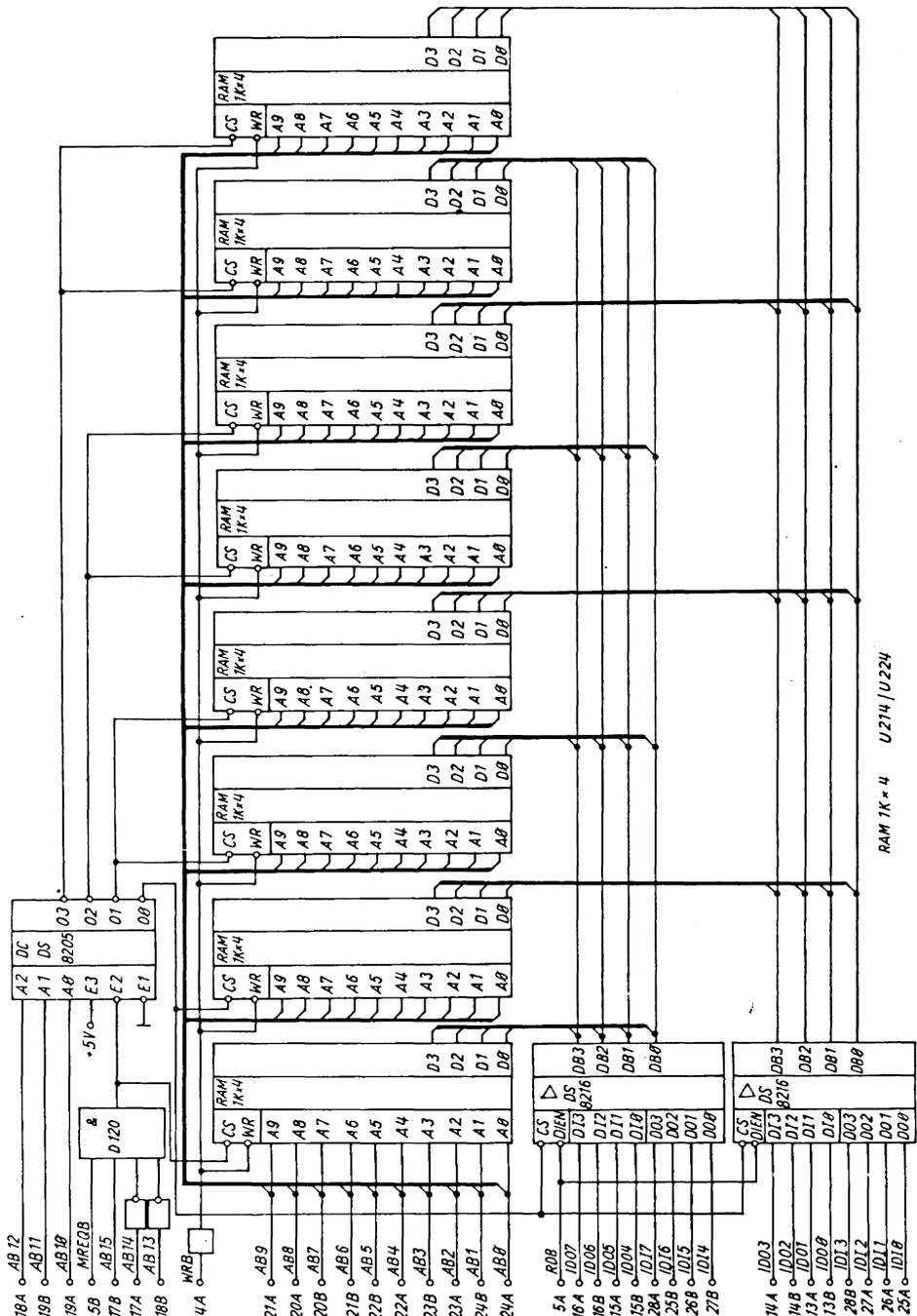
### EPROM-Programmiereinrichtung

Gerade bei kleineren Steuerungsanwendungen muß das Programm meist ständig verfügbar in EPROMs (löschrare Festwertspeicher) abgelegt werden. Damit benötigt man eine Möglichkeit zum Programmieren dieser Schaltkreise, die durch einen sehr einfachen Zusatz zum Polycomputer geschaffen werden kann (Bild 7).

Der Anschluß der Programmierbaugruppe erfolgt am Peripheriesteckverbinder. Um mit den 16 Anschlüssen der dort verfügbaren PIO auszukommen, wird die EPROM-Adresse auf dem Programmierzusatz durch einen Binärzähler erzeugt. Der Einsatz von CMOS-Schaltkreisen an dieser Stelle erspart die sonst (für EPROMs U555) notwendigen Pegelanhebungswiderstände. Die dargestellte Variante kann die Typen U555 (1 KBit × 8) und U556 (2 KBit × 8) programmieren; eine Erweiterung auf andere Typen (bis 64 KBit × 8) ist leicht möglich. Statt einer Umschaltung werden getrennte Fassungen für die verschiedenen Typen verwendet.

Falls die eingesetzten 2 × 4-Bit-Zähler U4520 nicht greifbar sein sollten, können an dieser Stelle auch 2 × 4-Bit-Schieberegister U4015 (mit Programmänderung) eingesetzt werden.

Bild 6. Zusatzspeicher 4 Kbyte stat. RAM



RAM 7K x 4 U 2714 / U 224

Systemsteckverbinder

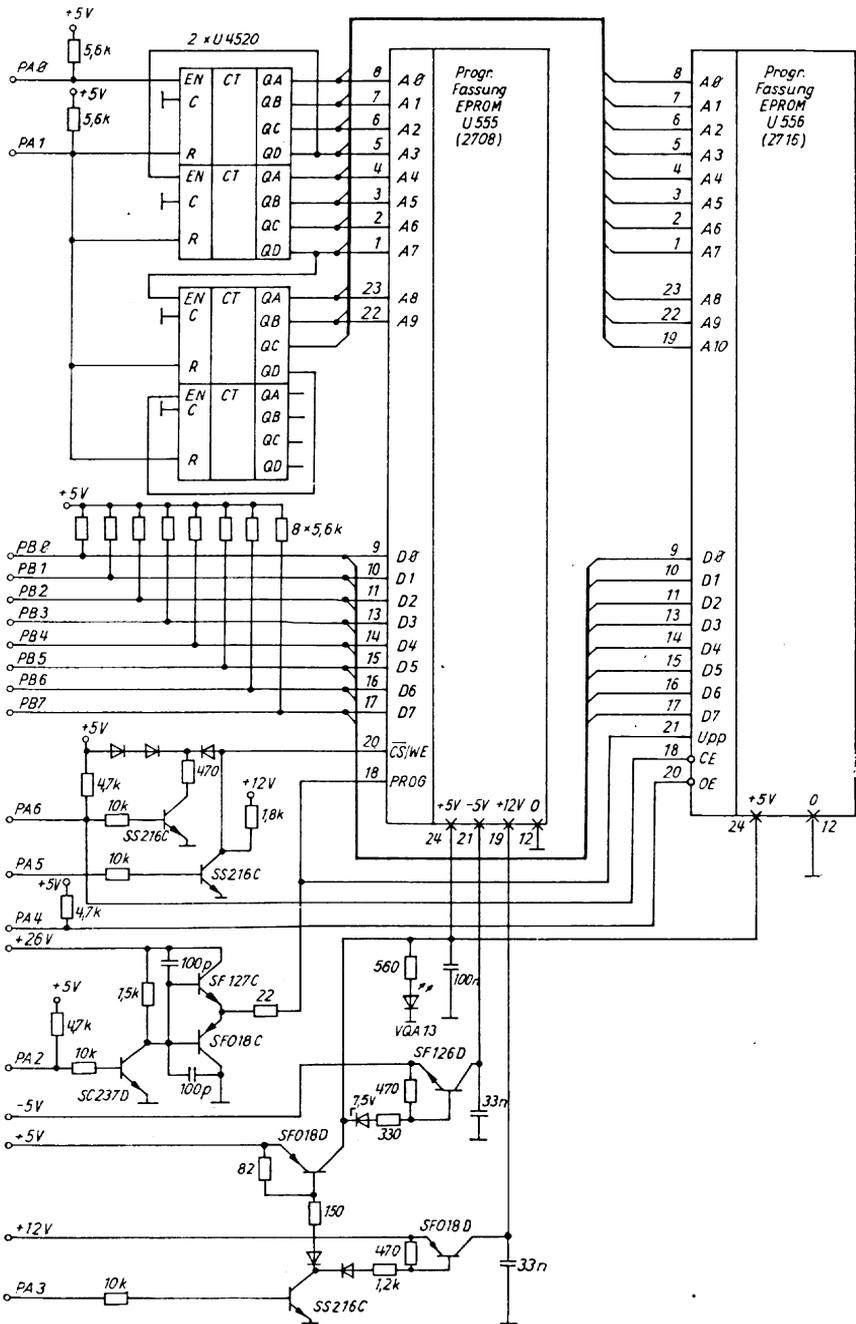


Bild 7. EPROM-Programmierzusatz

```

1 ;PROGRAMMIERUNG EPROM U555C
2 ;UEBERGABE DER DATENANFANGSADRESSE IN REG. IY
3 ;ANW.-PIO, PORT A PROG.; ANFANGSDATEN LADEN
0000 3EFF 4 PROG08: LD A,OPFH ;BITHODE
0002 D385 5 OUT (85H),A
0004 3EAF 6 LD A,10101111B ;UB EIN, /CS=0
0006 D384 7 OUT (84H),A ;DATENADRESSE
0008 3E00 8 LD A,0
000A D385 9 OUT (85H),A ;ALLE BITS - AUSGABE
10 ;ANLIEGEN DER BETRIEBSSPANNUNG ABWARTEN
11 LD B,0
12 DJNZ #
13 IN A,(84H)
14 RES 5,A ;CS/WE := 12V
15 OUT (84H),A
16 LD C,100 ;100 ZYKLEN A 0,5 MS
17 ;PROG.-ZYKLUS: ADRESSZAEBLER RESET,
0018 DB84 18 LOOP08: IN A,(84H)
001A CBCF 19 SET 1,A
001C D384 20 OUT (84H),A
001E CB8F 21 RES 1,A
0020 D384 22 OUT (84H),A
0022 FDE5 23 PUSH IY
0024 E1 24 POP HL ;HL := ANFANGSADRESSE
0025 110004 25 LD DE,1024 ;EPROM-LAENGE
26 ;PROGRAMMIEREN DER EINZELNEN SPEICHERZELLEN
27 ;DATENAUSGABE AUF PORT B
0028 3EFF 28 LOOP18: LD A,OPFH
002A D387 29 OUT (87H),A
002C 3E00 30 LD A,0
002E D387 31 OUT (87H),A ;BITMODE, AUSGABE
0030 7E 32 LD A,(HL)
0031 2F 33 CPL ;DATENNEG. FUER PC880-EPROM
0032 D386 34 OUT (86H),A ;SONST NOP STAT CPL
35 ;PROGRAMMIERIMPULS AUSGEBEN
0034 DB84 36 IN A,(84H)
0036 CB97 37 RES 2,A
0038 D384 38 OUT (84H),A ;PROG := 26V
003A 0623 39 LD B,35 ;0,5 MS
003C 10FE 40 DJNZ #
003E CBD7 41 SET 2,A
0040 D384 42 OUT (84H),A ;PROG := 0
43 ;ADRESSE WEITERSCHALTEN
0042 CB87 44 RES 0,A
0044 D384 45 OUT (84H),A.
0046 CBC7 46 SET 0,A
0048 D384 47 OUT (84H),A
004A 23 48 INC HL
004B 1B 49 DEC DE
004C 7A 50 LD A,D
004D E3 51 OR E
004E 20D8 52 JR NZ,LOOP18
53 ;ZYKLUS BEENDET
0050 0D 54 DEC C
0051 20C5 55 JR NZ,LOOP08
56 ;PROGRAMMIERUNG BEENDET, SPANNUNGEN ABSCHALTEN
0053 3EFF 57 LD A,OPFH
0055 D385 58 OUT (85H),A
0057 D385 59 OUT (85H),A ;BITHODE, EINGABE
0059 76 60 HALT

```

Bild 8. Programmieroutine für EPROM U 555

Eine einfache Routine zum Programmieren des Typs U555 ist in Bild 8 dargestellt. Sie realisiert das Programmieren des vollständigen EPROMs, falls auch Teilstücken programmiert werden sollen, ist sie zu modifizieren. Die Programme zum Bedienerdialog und Kontroll-Lesen sind hier aus Umfangsgründen nicht dargestellt.

## Literatur

[1] BURKHARDT, S.; HÜBNER, U.: Technik und Anwendung des Mikrorechnerlernsystems Polycomputer 880. – In: radio fernsehen elektronik 33 (1984) 5

[2] HÜBNER, U.: Zusatzgeräte für den Polycomputer 880. – In: radio fernsehen elektronik 33 (1984) 7

[3] KIESER, H.; MEDER, M.: Mikroprozessortechnik. – Berlin, 1982.

Autor:

*Dr. sc. techn. Uwe Hübner*

DDR 1142 Berlin

Brodowiner Ring 7

Wissenschaftlicher Mitarbeiter  
im Institut für Nachrichtentechnik  
Forschungszentrum des VEB Kombinat  
Nachrichtenelektronik

---

# Spielereien mit dem Taschenrechner

Sollten Sie ein höflicher Mensch sein, der während ein Konferenz, während des Unterrichts oder dgl. durch das laute Gerede seines Nachbarn gestört wird, so nehmen Sie doch einfach Ihren Taschenrechner zur Hand und tippen die Ziffernfolge

3 5 1 3 7 . 1 3 5

ein. Wenn Sie dann Ihren Taschenrechner um 180° drehen und ihn Ihrem Nachbarn unter die Nase halten, dann wird er sicher verstehen, was Sie von ihm wollen.

Wenn Ihr Nachbar jedoch schlagfertig und dabei nicht so höflich sein sollte wie Sie, dann wird er seinerseits zum Taschenrechner greifen und die Ziffernfolge

7 3 5 3 . 3 1 5

eintippen, ihn ebenfalls um 180° drehen und Ihnen sein Machwerk vor Augen führen.

Suchen Sie nach weiteren Beispielen solcher Zahlenspielereien!

---

# Wie kann ein Programm systematisch entworfen werden?



## 1. Einleitung

Gegenwärtig werden Mikrorechner in nahezu allen Bereichen der Volkswirtschaft, Wissenschaft und Technik eingesetzt. Charakteristisch ist auch, daß der immer preiswerter und leistungsfähiger werdende Heimcomputer unsere private Sphäre zu durchdringen beginnt. Während noch vor 10 Jahren der Programmwurf und die Programmierung einer begrenzten und speziell dafür ausgebildeten Berufsgruppe vorbehalten war, beschäftigen sich heute schon weite Kreise der Werktätigen und unserer Jugendlichen mit dem Entwurf und der Entwicklung von Programmen. Infolge der weiteren Durchdringung aller Bereiche unserer Gesellschaft mit der Mikrorechentechnik ist auch der Tag nicht mehr fern, wo die noch junge Wissenschaft »Informatik« neben der Mathematik und Physik zu den Grundlagenfächern in unseren Schulen gehören wird.

In diesem Beitrag soll aus den genannten Gründen eine Einführung in die Grundlagen des systematischen Programmwurfes gegeben werden, die an einem Beispiel erläutert wird.

## 2. Warum systematischer Programmwurf?

In der konventionellen Programmentwicklung wurde die Erarbeitung

eines Programmsystems in drei Etappen vorgenommen, die die Grundlage für eine Arbeitsteilung mit personeller Trennung in drei selbständig arbeitende Kollektive war:

### – *Problemanalyse*

Von der Problemanalysegruppe wurde die Problemdokumentation erarbeitet, die grob gegliederte problemorientierte Programmablauf- und Datenflußpläne umfaßte.

### – *Organisation*

Die Organisationsgruppe erarbeitete detaillierte Programmablauf- und Datenflußpläne sowie Testdatensätze für den autonomen und komplexen Test der später zu erarbeitenden Programmmoduln. Die Organisation legte die Arbeitsteilung während der Programmierung fest.

### – *Programmierung*

Die Programmierergruppe erarbeitete auf der Grundlage der detaillierten Programmablaufpläne die Programmmoduln in einer vorgegebenen Programmiersprache für ein vorgegebenes Betriebssystem und führte den autonomen und komplexen Test durch.

Die horizontale Aufteilung der Programmentwicklungsetappen auf verschiedene Kollektive hat sich auf Grund der notwendigen Abstimmungen zwi-

schen den Kollektiven und des auftretenden Informationsverlustes als ineffektiv erwiesen. Die moderne Softwaretechnologie geht deshalb von einer vertikalen Aufteilung des Gesamtproblems auf einzelne Programmierer aus, wobei jeder Programmierer für alle Arbeitsetappen zu einem vorgegebenen Teilproblemkreis verantwortlich ist. Der Programmentwicklungsprozeß wird in folgende Arbeitsetappen untergliedert:

- Spezifizierung,
- Entwurf,
- Implementierung,
- Verifizierung,
- Dokumentierung.

Dieser technologische Prozeß kommt damit insbesondere auch der Programmentwicklung für die Mikrorechner und Heimcomputer entgegen, wobei sich der Einzelne jeweils nacheinander an die Lösung der Teilprobleme macht und für diese die Arbeitsetappen selbst ausführt. Die Softwaretechnologie muß gewährleisten, daß mit möglichst geringem personellem und materiellem Aufwand qualitativ hochwertige Programme entwickelt werden.

Kriterien eines qualitativ hochwertigen Programms sind:

#### - *Richtigkeit*

Das Programm realisiert vollständig und korrekt den spezifizierten Funktionsumfang.

#### - *Zuverlässigkeit*

Fähigkeit des Programms, auf fehlerhafte Eingabedaten und gewisse Hardwarefehler so zu reagieren, daß der Fehler erkannt und entweder umgangen oder durch Fehlerausdrucken identifiziert wird. Das Programm muß funktionsfähig bleiben und trotz der Fehlersituation ein maximal wertbares Resultat liefern.

#### - *Änderbarkeit*

Ein Programm muß leicht änderbar sein. Die Änderung ist zur Beseitigung von Programmfehlern, Anpassung an veränderte Nutzungsbedingungen oder Weiterentwicklung erforderlich. Die Änderbarkeit setzt eine transparente Programmstruktur voraus.

#### - *Transparenz*

Ein Programm ist transparent, wenn die Funktionen des Gesamtsystems grob überschaubar und die einzelnen Funktionskomponenten auf die Programmstruktur abbildbar sind.

Weitere Kriterien sind Effektivität, Portabilität, Sauberkeit und Virtualität, die für Heimcomputeranwendungen nicht vordergründig sind. Der Brennpunkt, auf den sich alle Kriterien eines »guten« Programms konzentrieren, ist eine gut strukturierte, transparente Programmstruktur, die nur das Ergebnis eines systematischen Programmentwurfs sein kann.

Betrachten wir im weiteren die wesentliche Etappe des Programmentwurfs zuerst am Beispiel eines Problems aus der Textverarbeitung. Zunächst aber die Spezifikation des zu lösenden Problems.

### **3. Spezifikation einer Textverarbeitungsaufgabe**

Die Spezifikation ist eine schriftliche Aufgabenstellung für das zu entwerfende Programm. Die Beschreibung des Problems kann verbal oder formal oder in »gemischter« Form erfolgen. Wichtig ist die Eindeutigkeit, Exaktheit und Vollständigkeit der Anforderungen an das Programmsystem. Sie sollte aber andererseits auch keine Vorgriffe auf die nachfolgenden Arbeitsetappen enthalten.

Schwerpunkte der Spezifikation sind:

- Darstellung der Struktur und des Inhaltes der Ein- und Ausgabeinformationen,
- Beschreibung der Operationen mit den Eingabeinformationen,
- Forderungen zum Fehlerverhalten,
- Randbedingungen, mögliche Einschränkungen, mögliche zukünftige Erweiterungen usw.

Nun zum Beispiel der Textverarbeitung. Es soll ein Textverarbeitungssystem entwickelt werden, das folgende Funktionen realisiert:

- Eingabe der Texte von verschiedenen Eingabemedien,
- Korrektur der gespeicherten Texte,
- Druckaufbereitung der fehlerfreien Texte,
- Ausgabe der druckreifen Texte (auf Drucker),
- Ausgabe der gespeicherten Texte auf verschiedene andere Medien.

Das Hauptproblem ist die Druckaufbereitung der Texte. Das Programm hat die Aufgabe, einen vorliegenden Text so zeilenweise aufzubereiten, daß eine vorgeschriebene Zeilenlänge nicht überschritten wird. Dabei ist ein sogenannter Flatterrandausgleich durchzuführen. (Der Flatterrand, der bei der Schreibmaschinenschrift am rechten Rand des Blattes unvermeidlich ist, aber häßlich aussieht, soll bei unserem Textverarbeitungssystem durch Aufteilen der restlichen Leerzeichen der Zeile auf die Wortlücken beseitigt werden.) Auf Grund der Kompliziertheit der deutschen Schriftsprache kann vorerst auf eine Silbentrennung verzichtet werden. Eine spätere Ergänzung soll aber prinzipiell möglich sein. Verbunden werden soll die Druckaufbereitung mit einer Vorschubsteuerung. Vor einem Absatz soll eine Leerzeile und vor einem neuen Abschnitt

sollen zwei Leerzeilen eingeschoben werden.

Folgende Steuerzeichen sollen für die Steuerung der Textaufbereitung verwendet werden:

- # % - neuer Abschnitt (steht vor einer Überschrift),
- # # - neuer Absatz (steht vor einem Absatz),
- # / - Ende des Textes.

Das Steuerzeichen #/ kann ausgelassen werden. Im weiteren müßte nun die Spezifikation für die anderen Funktionen folgen, auf die wir aber aus Platzgründen hier verzichten wollen.

#### 4. Entwurf des Programmsystems

Die Arbeitsetappe des Entwurfs wird durch die Anwendung des Prinzips der wiederholten Dekomposition (Zerlegung) der Aufgabe in funktionell abgeschlossene Teilaufgaben charakterisiert.

Da ein Programmsystem in der Regel relativ groß ist, hat es sich in der Praxis bewährt, das Programmsystem in mehrere Moduln zu zerlegen. Es werden drei Modultypen - Quellmodul, Objektmodul und Lademodul - unterschieden. Ein Programmsystem wird somit bei der Modularisierung in mehrere Quellmoduln zerlegt. Jeder Quellmodul stellt eine autonome Übersetzungseinheit dar und wird mit dem Übersetzer (Assembler, Compiler) in einen Objektmodul übersetzt. Anschließend erfolgt mit dem Programmverbinder die Umwandlung eines oder

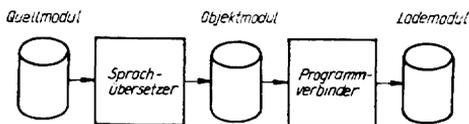


Bild 1. Methodik der Programmentwicklung

mehrerer Objektmoduln in einen ausführbaren Lademodul (siehe Bild 1). Die Arbeitsetappe des Entwurfs untergliedert sich durch die Modularisierung in zwei Arbeitsschritte:

- Systementwurf auf der Ebene von Moduln (Grobentwurf),
- Modulentwurf (Feinentwurf) der einzelnen Moduln.

Betrachten wir zunächst den Systementwurf. Bei der Modularisierung wird davon ausgegangen, daß relativ unabhängige, selbständige Funktionen als getrennte Quellmoduln programmiert werden. Damit verringert sich der Programmentwicklungsaufwand im ganzen, da sich der Korrektur- und Übersetzungsaufwand wesentlich reduziert. Sehr wichtig ist auch, daß sich die Testung der Moduln vereinfacht, je kleiner die Quellmoduln werden.

Moduln, die völlig selbständig abgearbeitet werden können, werden mit dem Programmverbinder aus der Objektform in einen autonomen Lademodul umgewandelt.

Moduln, die nur in Verbindung mit anderen Moduln voll arbeitsfähig sind, müssen nach der autonomen Testung zu einem Lademodul integriert werden. Diese Integration wird mit dem Programmverbinder durchgeführt, der mehrere Objektmoduln zu einem Lademodul verbinden kann. Nach der Programmverbindung muß ein abschließender Integrationstest vorgenommen werden, um weitere mögliche Fehler zu lokalisieren.

Bei einfachen Sprachverarbeitungssystemen wird in der Regel in den absoluten Maschinencode übersetzt, so daß sich ein Programmverbinder erübrigt. In solchen Systemen muß die Modulintegration von Hand durchgeführt werden. Wichtig ist auf jeden Fall, daß für die Modulintegration ein entsprechendes Modulinterface er-

arbeitet wurde, das die Schnittstelle bestimmt, über die Moduln miteinander kommunizieren.

Die Zerlegung eines Programmsystems in mehrere Lademoduln hat den Vorteil, daß nicht der gesamte Programmcode zur Programmaufzeit im Hauptspeicher erforderlich ist, d.h., durch die Dekomposition in mehrere Lademoduln kann ein größeres Programmsystem auch auf einem Heimcomputer mit relativ kleinem Hauptspeicher implementiert werden. Da die verschiedenen Lademoduln zeitlich nacheinander in den Hauptspeicher geladen werden, können sie nicht mehr auf direkte Art wie bei der Modulintegration miteinander kommunizieren, sondern indirekt über spezielle Hauptspeicherbereiche (COMMON-Bereiche) oder externe Speichermedien (Magnetbandkassette, Diskette o.ä.). Das Zusammenspiel der einzelnen Lademoduln wird während des Systementwurfs über sogenannte Datenflußpläne festgelegt.

Betrachten wir nun den Systementwurf am Beispiel der spezifizierten Textverarbeitungsaufgabe.

Ausgehend von den fünf genannten Funktionen kann das System in fünf Lademoduln zerlegt werden. Es bietet sich aber auf Grund der Gleichartigkeit der ersten und fünften Funktion an, die Eingabe und die Ausgabe der Texte von/auf verschiedene Medien zusammenzufassen. In verschiedenen Betriebssystemen werden dafür bereits Standarddienstprogramme unter der Bezeichnung PIP (Peripheral Interchange Program) angeboten. Auch für die Korrektur von Texten stehen in der Regel schon mehrere Lösungen für mehr oder weniger komfortable Bildschirm- bzw. Dialogeditoren zur Verfügung.

Die Druckaufbereitung der fehlerfreien Texte und die Ausgabe der druckreifen

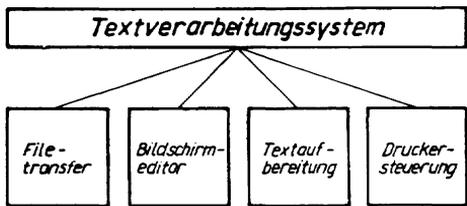


Bild 2. Struktur des Textverarbeitungssystems

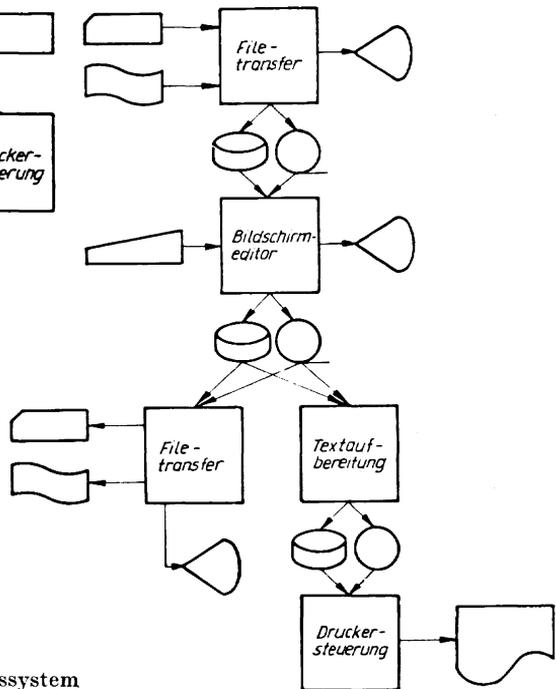


Bild 3. Datenfluß im Textverarbeitungssystem

Texte auf Drucker könnte eventuell zusammengefaßt werden. Da aber die Ausgabe auf Drucker sehr stark von der gerätetechnischen Lösung des Druckers und des Druckeranschlusses an den Mikrorechner abhängig ist, empfiehlt es sich, beide Funktionen zu trennen. Eine solche Trennung hätte außerdem den Vorteil, daß die Texte nach der Druckaufbereitung noch einmal mit dem Bildschirmeditor nachgearbeitet werden könnten (Einfügen von Steuerzeichen, wie Zeilenvorschub, Blattwechsel usw.). Nach diesen Überlegungen besteht das Textverarbeitungssystem gemäß Bild 2 aus vier Modulen.

Der Datenaustausch zwischen den Lademodulen erfolgt zweckmäßigerweise über externe Speichermedien, wie Magnetbandkassette oder Diskette, je nach Verfügbarkeit. Der

grundsätzliche Datenfluß im Textverarbeitungssystem ist in Bild 3 dargestellt.

Im weiteren betrachten wir den Modulentwurf am Beispiel des Textaufbereitungsprogramms.

## 5. Der Modulentwurf

Das Ziel des Modulentwurfs besteht darin, durch eine schrittweise Verfeinerung den Algorithmus des Moduls so zu entwerfen, daß er sich mühelos programmieren läßt und zu einem transparenten, gut strukturierten Programm führt. Wie in jeder Ingenieurwissenschaft haben auch in der Programmierung grafische Mittel einen festen Platz. Seit den Anfängen der Programmierungstechnik wird zur Darstellung von Algorithmen die Methode des Programmablaufplans ver-

wendet. Sie ist eine Form eines gerichteten Graphen, die wegen ihrer Universalität weit verbreitet ist (siehe Heft 2, Beitrag zur Programmierung einer einfachen Aufzugssteuerung). Der Nachteil der Programmablaufpläne besteht darin, daß sie den systematischen Programmentwurf nicht unterstützen und nicht zum strukturierten Entwerfen zwingen.

Wesentlich günstiger für den Entwurf vieler Softwaresysteme ist die Struktogrammtechnik. Sie ist ein universelles grafisches Darstellungsmittel für den Softwareentwurf, gestattet eine schrittweise Verfeinerung und zwingt zum strukturierten Entwurf. Die von ihr verwendete Symbolik ist sehr einfach, setzt keine besonderen EDV-Kenntnisse voraus und läßt sich leicht in höheren Programmiersprachen formulieren.

Bei der Struktogrammtechnik wird ein Algorithmus oder ein Programm aus Bausteinen (Strukturblöcken) zusammengesetzt. Bei schrittweiser Verfeinerung kann jeder Baustein in weitere Bausteine bis hinunter auf die Ebene der Programmiersprache (Machinesprache) zerlegt werden. Betrachten wir zunächst die Grundbausteine der Struktogrammtechnik.

Der wichtigste Grundbaustein ist die Sequenz (Bild 4). Sie besteht aus einer unverzweigten Folge von Strukturblöcken. Die Strukturblöcke können bei der schrittweisen Verfeinerung durch Anweisungen der Programmiersprache oder andere Grundbausteine, wie Alternative, Fallauswahl oder Schleife, weiter untersetzt werden.

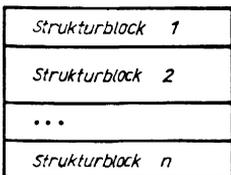


Bild 4. Sequenz

Bild 5  
Unvollständige  
Alternative

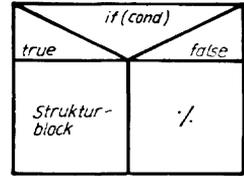
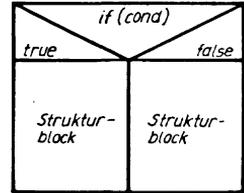


Bild 6  
Vollständige  
Alternative



Die Alternative kommt als unvollständige oder vollständige Alternative (Bild 5 und 6) vor. In Abhängigkeit von einer Bedingung *cond*, die den Wahrheitswert *true* (wahr) oder *false* (nicht wahr) annehmen kann, wird entweder der linke oder rechte Strukturblock ausgeführt. Bei der unvollständigen Alternative ist der falsche Zweig leer. Zur besseren Verdeutlichung des Übergangs von den Struktogrammen zum Programm soll im weiteren zu den einzelnen Strukturblöcken ihre Realisierung in höheren Programmiersprachen angegeben werden. Auf Grund ihrer guten Eignung zur strukturierten Programmierung und weiten Verbreitung wird auf die Programmiersprachen PASCAL und C Bezug genommen. In PASCAL werden die vollständige und unvollständige Alternative wie folgt programmiert:

```

if <cond> then <strukturblock i>
  else <strukturblock j>;
if <cond> then <strukturblock i>;

```

In C haben die Alternativen folgende Notationen:

```

if (<cond>) <strukturblock i>;
else <strukturblock j>;
if (<cond>) <strukturblock i>;

```

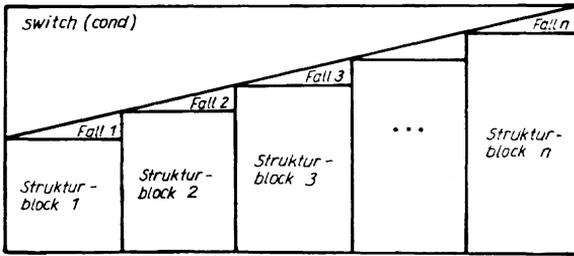


Bild 7. Fallauswahl

Die Fallauswahl (Bild 7) ist eine Erweiterung der Alternative auf mehr als zwei Fälle. Sie beruht auf der case- oder switch-Anweisung von solchen höheren Programmiersprachen, wie PASCAL oder C. Die Bedingung cond ist die sogenannte Schaltervariable, die die Auswahl der einzelnen Strukturblöcke bewirkt. In PASCAL kann die Fallauswahl wie folgt programmiert werden:

```

case <cond> of
  <fall 1> : <strukturblock 1>;
  <fall 2> : <strukturblock 2>;
  <fall 3> : <strukturblock 3>;
  :
  <fall n> : <strukturblock n>; end

```

In C hat die Fallauswahl folgende Notation:

```

switch (<cond>){
  case <fall 1>: <strukturblock 1>; break;
  case <fall 2>: <strukturblock 2>; break;
  case <fall 3>: <strukturblock 3>; break;
  :
  default : <strukturblock n>; break;}

```

Schleifen kommen in der Regel als abzählbare Schleife, als Abweisschleife oder als Nichtabweisschleife vor (Bild 8, 9 und 10).

Die abzählbare Schleife gestattet die Ausführung eines Strukturblockes für alle Werte einer Laufvariablen von e1 bis e2. In manchen Programmiersprachen, wie FORTRAN, C oder ALGOL ist zusätzlich eine Schrittweitensteuerung (e3) möglich. In PASCAL ist nur eine Schrittweite von 1 (to) oder -1 (downto) möglich:

```

for <name> := <e1> to <e2> do
  <strukturblock>;
for <name> := <e1> downto <e2> do
  <strukturblock>;

```

In C gibt es eine verallgemeinerte for-Anweisung:

```

for (<e1>; <e2>; <e3>)
  <strukturblock>;

```

Die Abweis- bzw. Nichtabweisschleifen werden insbesondere für Iterationen benötigt, wo die Anzahl der Schleifen von vornherein nicht abzählbar ist. Bei der Abweisschleife wird erst die Bedin-

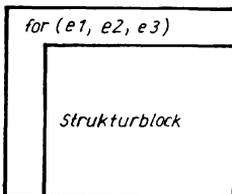


Bild 8. Abzählbare Schleife

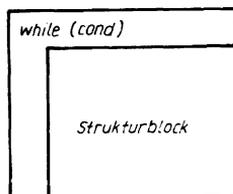


Bild 9. Abweisschleife

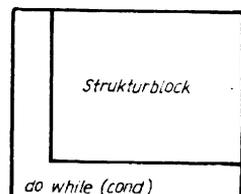


Bild 10. Nichtabweisschleife

gung cond geprüft, und wenn sie den Wahrheitswert true ergibt, wird der Strukturblock ausgeführt. Im Unterschied dazu wird bei der Nichtabweisschleife nach der Ausführung des Strukturblocks geprüft, ob die Bedingung cond den Wahrheitswert true ergibt und der Strukturblock wiederholt werden muß. Folgende Realisierungen sind in PASCAL und C möglich:

Abweisschleife:

In PASCAL:  
**while** <cond> **do** <strukturblock>  
**end**;

In C:  
 while (<cond>) <strukturblock>;

Nichtabweisschleife:

In PASCAL:  
**repeat** <strukturblock> **until** <cond>;

In C:  
 do <strukturblock> while (<cond>;

## 6. Systematischer Entwurf des Moduls „Textaufbereitung“ (1. und 2. Abstraktionsstufe)

Nach den Grundlagen der Struktogrammtechnik soll nun der systematische Entwurf am Beispiel des Moduls »Textaufbereitung« erläutert werden. Die grundsätzliche Arbeitsweise des Moduls läßt sich an Hand des Struktogramms Bild 11 erläutern.

Im ersten Strukturblock werden das Ein- und das Ausgabefile eröffnet. Anschließend erfolgt in einer Nichtabweisschleife die Textaufbereitung. Wenn das Textende erreicht ist, werden im letzten Strukturblock die Files wieder abgeschlossen.

Die eigentliche Textaufbereitung kann wiederum in drei Strukturblocke unterteilt werden:

– *Einlesen einer Textzeile*

Aus dem Eingabefile werden die Text-

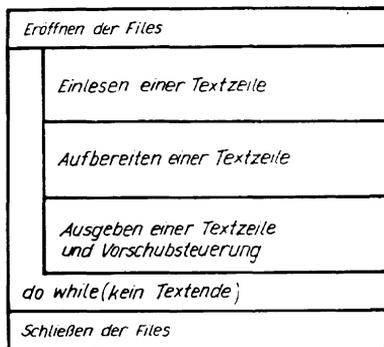


Bild 11. Grundsätzliche Funktionsweise des Moduls »Textaufbereitung«

zeichen eingelesen und in einem Wortpuffer zwischengespeichert. Das Wortende wird durch ein Leerzeichen oder durch das Zeilenende erkannt. Danach wird das Wort in einen Zeilenpuffer übernommen. Es werden so lange Worte in den Zeilenpuffer übernommen, bis der Zeilenpuffer bis zur vorgegebenen Druckzeilenlänge gefüllt ist bzw. bis die Kapazität des Zeilenpuffers für die Übernahme eines Wortes nicht mehr ausreichend ist.

– *Aufbereiten der Textzeile*

Im Zeilenpuffer werden die Wortlücken gesucht und zusätzliche Leerzeichen so lange eingefügt, bis das letzte Wort die vorgegebene Druckzeilenlänge erreicht hat. Durch dieses Verfahren wird der Flatterrandausgleich vorgenommen.

– *Ausgeben einer Textzeile und Vorschubsteuerung*

Es wird die aufbereitete Textzeile in das Ausgabefile ausgegeben. Dabei werden die Steuerzeichen ausgewertet und entsprechend Leerzeilen in den Text eingefügt.

Auf dieser Abstraktionsebene sollte man noch einmal die Frage der Zerlegung des Moduls in mehrere Quellmoduln diskutieren. Prinzipiell wäre

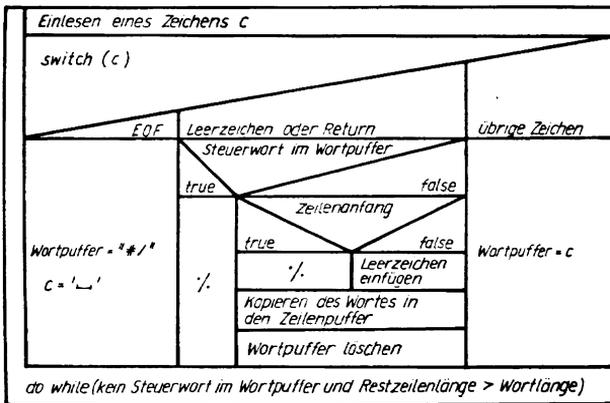


Bild 12  
Verarbeitungsmodul 1

es möglich, mit dem Struktogramm in Bild 11 eine Dekomposition in vier Quellmoduln vorzunehmen:

- Steuermodul für Fileeröffnung, Schleifenorganisation, Aufruf der Verarbeitungsmoduln und Fileabschluss;
- Verarbeitungsmodul 1: Einlesen einer Textzeile;
- Verarbeitungsmodul 2: Aufbereiten einer Textzeile;
- Verarbeitungsmodul 3: Ausgeben einer Textzeile und Vor-schubsteuerung.

Bei wenig Erfahrung ist meist nicht abzuschätzen, welche Größe die einzelnen Probleme annehmen. Deshalb ist es auf jeden Fall günstiger, wenn man den Algorithmus eines jeden Verarbeitungsmoduls getrennt weiterbearbeitet. Stellt sich heraus, daß die einzelnen Moduln relativ klein sind, lassen sie sich im Quelltext schnell zu einem Modul vereinigen. Anderenfalls müßte noch ein Modulinterface zur Parameterübergabe zwischen Steuer- und Verarbeitungsmodul ergänzt werden.

### 6.1. Verarbeitungsmodul 1

Dieser Modul besteht aus einer Nicht-abweisschleife (Bild 12), die wiederholt

wird, solange im Wortpuffer kein Steuerwort (#%, ## oder #/) steht und die Restzeilenlänge größer als die aktuelle Wortlänge ist. Der Strukturblock in der Schleife besteht aus einer Sequenz eines einfachen Strukturblocks und einer Fallauswahl. Im einfachen Strukturblock wird ein Zeichen eingelesen. Bei der Fallauswahl wird das eingelesene Zeichen analysiert, wobei folgende Fälle unterschieden werden:

- EOF  
Es wurde das EOF-Zeichen (End of File) eingelesen. Das File wurde nicht mit dem Steuerwort "#/%" abgeschlossen. Das Steuerwort wird in den Wortpuffer eingetragen.
- Leerzeichen oder Return  
Ein Leerzeichen (SP – Space) oder Zeilenende (CR – Carriage Return bzw. NL – New Line) wurde eingelesen. Damit steht im Wortpuffer ein vollständiges Wort, das in den Zeilenpuffer übernommen werden muß, falls es kein Steuerwort ist. Bei einem Steuerwort ist keine weitere Operation erforderlich. Wenn das zu übernehmende Wort nicht an den Zeilenanfang kommt, so ist zur Trennung vom vorhergehenden Wort ein Leerzeichen in den Zeilenpuffer einzutragen. Nach dem Kopieren

des Wortes wird der Wortpuffer gelöscht.

c) Übrige Zeichen

Bei allen übrigen Zeichen erfolgt eine Übernahme des Zeichens c in den Wortpuffer.

**6.2. Verarbeitungsmodul 2**

Der Verarbeitungsmodul 2 (Bild 13) macht den Flatterrandausgleich, der nicht ausgeführt wird, wenn es sich um

- die letzte Zeile eines Absatzes vor einem neuen Absatz oder vor einer Überschrift handelt,
- die Zeile leer ist oder
- die Zeile nur aus einem Wort besteht.

Zur Analyse, ob die Zeile nur aus einem Wort besteht, wird zuvor die erste Wortlücke gesucht.

Der Flatterrandausgleich selbst könnte in einer Abweisschleife durchgeführt werden. Zweckmäßigerweise wird dabei die Zeile von rechts nach links durchmustert, und jede Wortlücke wird durch Einfügen eines Leerzeichens vergrößert, wobei der rechte Teil des Zeilenpuffers immer um ein Zeichen nach rechts verschoben werden muß. Das Ende ist erreicht, wenn die aktuelle Zeilenlänge der maximalen Zeilenlänge entspricht. Bei wenig Wortlücken und

vielen auszugleichenden Leerzeichen kann es passieren, daß der linke Rand erreicht wird, ohne daß die aktuelle Zeilenlänge der maximalen Zeilenlänge entspricht. In diesem Fall muß der Prozeß der Leerzeicheneinfügung von rechts nach links noch einmal wiederholt werden. Deshalb sind zwei ineinandergeschaltete Schleifen erforderlich. Die äußere Schleife wiederholt die innere Schleife, falls der linke Rand erreicht wird, aber der Flatterrandausgleich noch nicht abgeschlossen ist.

**6.3. Verarbeitungsmodul 3**

Der Verarbeitungsmodul 3 (Bild 14) organisiert im ersten Teil die Ausgabe des aufbereiteten Textes in ein Ausgabefile und im zweiten Teil die Einfügung von zusätzlichen Leerzeilen, wobei jeder Teil aus einer unvollständigen Alternative besteht.

Die Ausgabe des Zeilenpuffers ist nur erforderlich, wenn die aktuelle Zeilenlänge größer als Null ist.

Nach der Ausgabe des Zeilenpuffers muß ein Zeichen »Zeilenvorschub« (NL) ausgegeben und der Zeilenpuffer gelöscht werden.

Im zweiten Teil wird, falls im Wortpuffer ein Steuerwort steht, ein Zeichen

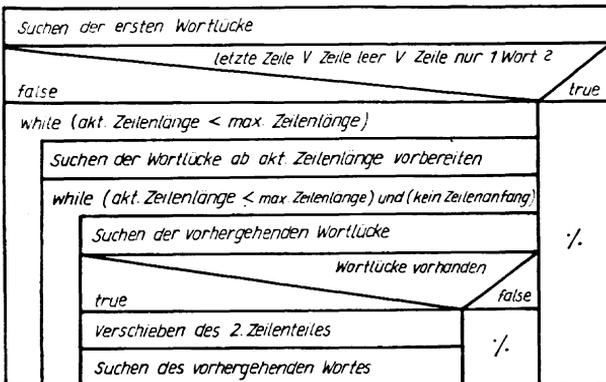


Bild 13  
Verarbeitungsmodul 2

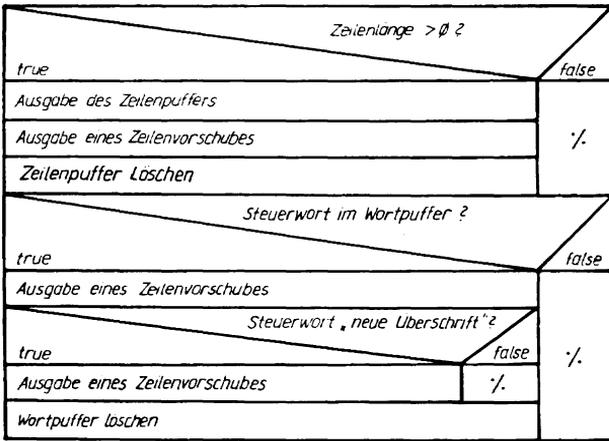


Bild 14  
Verarbeitungsmodul 3

»Zeilenvorschub« ausgegeben. Bei dem Steuerwort " # % " (neuer Abschnitt) werden sogar zwei Zeichen »Zeilenvorschub« ausgegeben. Zum Abschluß wird der Wortpuffer gelöscht.

### 7. Systematischer Entwurf des Moduls „Textaufbereitung“ (3. Abstraktionsstufe)

Die weitere Vervollkommnung und Verfeinerung des Modulentwurfs ist nur möglich, wenn die bisher verwendeten verbalen Aussagen formalisiert werden. Da sich bis zum heutigen Tag spezielle Entwurfssprachen zur Formalisierung des Modulentwurfs nicht durchgesetzt haben, ist es zweckmäßig, dafür Elemente moderner ausdrucksstarker Programmiersprachen wie PASCAL oder C zu benutzen. Zweifellos werden dadurch implementierungstechnische Details in den Modulentwurf hineingetragen, was aber letztlich nicht unerwünscht ist, da sie die

programmtechnische Umsetzung der Struktogramme fördern und bei Kenntnis der höheren Programmiersprachen mühelos ohne weitere Erläuterungen lesbar sind. Auf Grund ihrer Ausdrucksstärke und leichten Erlernbarkeit soll die Formalisierung der Struktogramme in diesem Beitrag unter Anlehnung an die Sprachelemente der bei uns erst in Verbreitung befindlichen Systemprogrammiersprache C erfolgen. Trotzdem soll dabei eine Allgemeingültigkeit gewahrt bleiben. Betrachten wir zuerst vor der Verfeinerung des Modulentwurfs die wichtigsten formalen Ausdrucksmittel.

### Ein- und Ausgabeoperationen

Zur Identifikation der benötigten Files für die Ein- bzw. Ausgabe werden Namen eingeführt, wie z.B. fin – für das Eingabefile und fout – für das Ausgabefile. Die Funktion open (fin) dient zur Eröffnung des Eingabefiles, wäh-

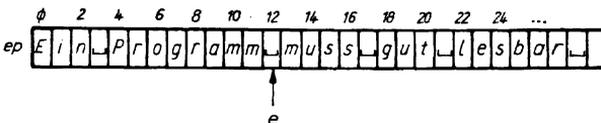


Bild 15. Der Eingabepuffer

rend die Funktion `close (fin)` es wieder abschließt.

Die Funktion `c = getc (fin)` liest vom Eingabefile `fin` ein Zeichen ein. Die zeichenweise Ein- bzw. Ausgabe ist in der Programmiersprache C eine typische Funktion. In anderen Programmiersprachen erfolgt in der Regel eine satzorientierte Ein- bzw. Ausgabe. Die Funktion `getc` muß dann über die Arbeit mit einem *Eingabepuffer* (hier mit `ep` bezeichnet) selbst programmiert werden (Bild 15).

Mit einer Eingabeoperation (`get, read`) wird ein Satz aus dem Eingabefile in den Eingabepuffer `ep` gelesen. Der Zeiger `e` wird auf Byte 0 gesetzt. Bei der Funktion `getc` wird nun das Zeichen `ep [e]` bereitgestellt. Anschließend wird `e` um eins erhöht. Dadurch können nacheinander die einzelnen Zeichen aus dem Eingabepuffer gelesen werden. Wenn der eingelesene Satz verarbeitet ist, so muß der nächste Satz eingelesen werden.

Die Funktion `putc (c, fout)` gibt das Zeichen `c` auf das Ausgabefile `fout` aus. Auch hinter dieser Funktion verbirgt sich die Arbeit mit einem Ausgabepuffer.

### Pufferarbeit

Die Textverarbeitung beruht im allgemeinen auf einer intensiven Arbeit mit Puffern, in denen der Text ge-

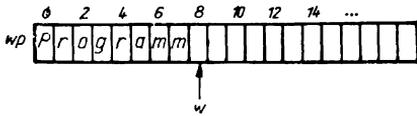


Bild 16. Der Wortpuffer

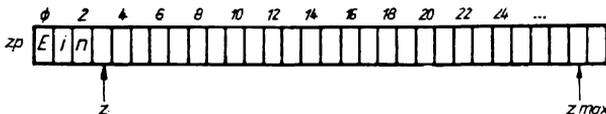


Bild 17. Der Zeilenpuffer

speichert und verarbeitet wird. Die Speicherung von Texten erfolgt heute bei Mikrorechnern allgemein im ASCII-Kode (Amerikanischer Standardkode für den Informationsaustausch). Ein einzelnes ASCII-Zeichen wird in Hochkommas eingeschlossen, z.B. '#', was soviel wie der Hexadezimalcode des ASCII-Zeichen # im Byteformat (23H) bedeutet. ASCII-Zeichenketten werden in Anführungszeichen eingeschlossen, z.B. »Programm«. Wie bereits in den Struktogrammen Bild 12 bis 14 angedeutet, werden bei dieser Aufgabe ein *Wortpuffer* (Bild 16) und ein *Zeilenpuffer* (Bild 17) benötigt.

Der Zugriff auf die Speicherplätze der Puffer erfolgt über ihre Zeiger als Pufferindex, wobei der Index in Anlehnung an C über eckige Klammern spezifiziert wird. Zur Erklärung der Pufferarbeit seien hier einige Beispiele angeführt:

`w = 0`

*/× Löschen des Wortpuffers ×/*

`wp [w] = c`

*/× Eintragen des Zeichens c in den Puffer ×/*

`wp [0] = '#'`

*/× Eintragen des ASCII-Zeichen # in Puffer auf die Position 0 ×/*

`w = w + 1`

*/× Vorsetzen des Pufferindex ×/*

Mit `zmax` wird die maximale Länge der Druckzeile spezifiziert.

### Logische Operationen

Zur Spezifikation von logischen Ausdrücken werden hier die allgemein

üblichen Operatoren benutzt, die teilweise von den Notationen in C abweichend sind, wie:

- not - Negation
- ∧ - Konjunktion
- ∨ - Disjunktion
- = - gleich
- ≠ - ungleich
- < - kleiner als
- > - größer als

Unter Verwendung der dargelegten Prinzipien der Ein- und Ausgabe und der Pufferarbeit sowie Anwendung der beschriebenen Symbolik ist die schrittweise Verfeinerung der Struktogramme unproblematisch, wobei man die einzelnen Strukturblöcke systematisch durchgehen kann. Das Ergebnis dieser dritten Abstraktionsstufe ist in Bild 18 dargestellt, wobei auf Grund der relativ kleinen Größe der Struktogramme (und damit auch der zu entwickelnden Quellmoduln) eine Zusammenfassung zu einem Quellmodul bevorzugt wurde.

### 8. Ergebnisse der Realisierung des Moduls „Textaufbereitung“

Auf der Grundlage des Struktogramms in Bild 18 erfolgte die Implementierung des Moduls in einer höheren Programmiersprache, wobei die Programmierung, Eingabe des Quelltextes, Übersetzung und Testung des Moduls weniger als fünf Stunden beanspruchte.

Probleme der Implementierung von Algorithmen im Zusammenhang mit der Auswahl einer geeigneten Implementierungssprache sind einer weiteren Veröffentlichung vorbehalten, so daß sie hier nicht diskutiert werden sollen. Zur Förderung des Verständnisses für die Arbeitsweise des Moduls sei hier ein Beispiel betrachtet. Vorgegeben ist ein Text gemäß Bild 19, der willkürlich unter Verwendung der beschriebenen

Steuerzeichen zeilenweise eingegeben wurde.

Dieser Text soll einmal für eine Länge der Druckzeile von 35 Zeichen und 60 Zeichen aufbereitet werden (Bild 20 und 21).

Die Ausgabe der Texte erfolgte auf einem Typenraddrucker robotron 1152, dessen Druckwerk der elektronischen Schreibmaschine S 6001 bzw. S 6011 entspricht. An den Beispielen ist sehr gut zu sehen, wie der implementierte Modul die einzelnen Worte in den Zeilen anordnet und den Flatterrandausgleich vornimmt.

### 9. Ein anderes Beispiel — Der Turm von Hanoi

Zum Abschluß sei die Anwendung der Struktogramntechnik noch an einem anderen Beispiel, dem Programm für das Computerspiel »Turm von Hanoi« gezeigt (s. SCHÖNFELDER: »Computerspiele«, Heft 4).

Im Mittelpunkt des Algorithmus steht die Arbeit mit der Bildtabelle BTA, die folgenden Grundaufbau hat:

Spalte		1	2	3	4
Zeilen	Adresse				
1	BTA	A 0	A 0	A 0	0 D
2	BTA + 4	A 0	A 0	A 0	0 D
3	BTA + 8	A 1	A 0	A 0	0 D
4	BTA + 12	A 2	A 0	A 0	0 D
5	BTA + 16	A 3	A 0	A 0	0 D
6	BTA + 20	A 4	A 0	A 0	0 D
7	BTA + 24	A 5	A 0	A 0	0 D
8	BTA + 28	A 6	A 6	A 6	0 D
9	BTA + 32	A 6	A 6	A 6	0 D

Die Spalte 1 kodiert den Aufbau der Pyramide auf Platz 1.

Die Spalten 2 und 3 zeigen den Aufbau der Pyramide für die Plätze 2 und 3. Im Anfangszustand sind die Plätze leer, da die Pyramide sich auf Platz 1 befindet. Die Spalte 4 enthält das

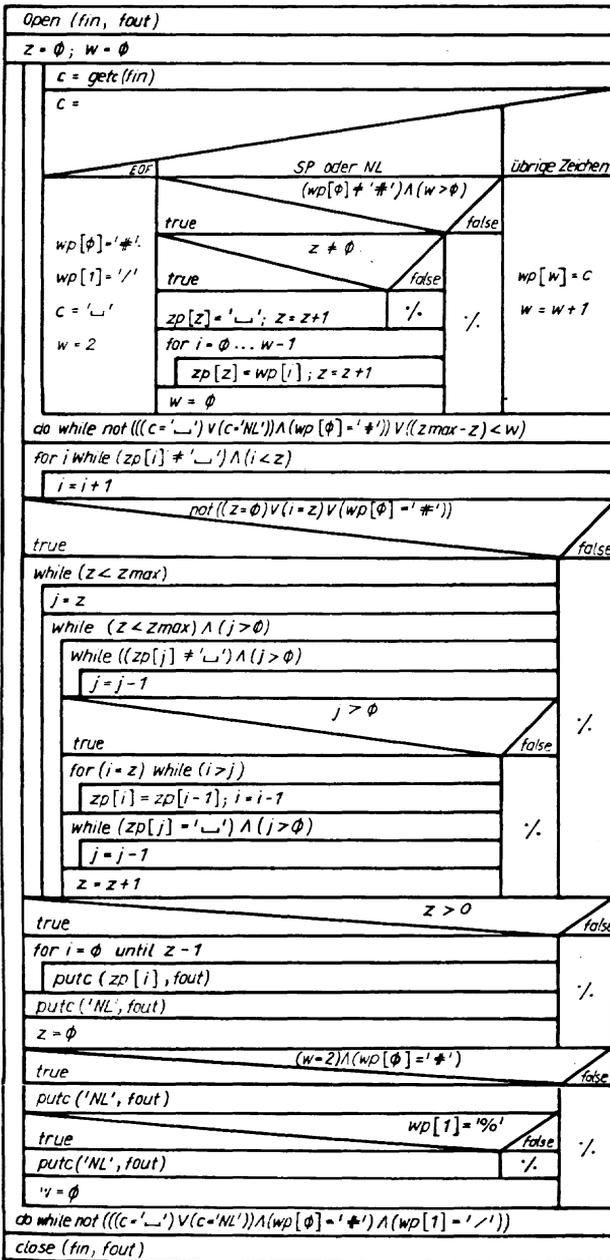


Bild 18. Verfeinertes formalisiertes Struktogramm des Moduls »Textaufbereitung«

## 1. Einfuehrung in den Modul "Textaufbereitung"  
 ## Der Modul "Textaufbereitung" hat die Aufgabe, beliebig geschriebene Texte fuer den Druck mit einer vorgegebenen Zeilenlaenge aufzubereiten. Die Laenge der Druckzeile ist frei wahlbar. Die Aufbereitung wird durch die Steuerworte "Ueberschrift" (##), "Absatz" (##) und "Fileende" (#/) gesteuert.  
 ## 2. Einschraenkungen ## In der gegenwaertigen Ausbaustufe wird keine Silbentrennung realisiert. Diese kann aber in das entwickelte Programm problemlos eingebaut werden, da die Anschlussstelle elementar einfach ist. Das eigentliche Problem ist die Silbentrennung in der deutschen Schriftsprache. #/

Bild 19. Textbeispiel mit Steuerzeichen

1. Einfuehrung in den Modul "Textaufbereitung"

Der Modul "Textaufbereitung" hat die Aufgabe, beliebig geschriebene Texte fuer den Druck mit einer vorgegebenen Zeilenlaenge aufzubereiten. Die Laenge der Druckzeile ist frei wahlbar. Die Aufbereitung wird durch die Steuerworte "Ueberschrift" (##), "Absatz" (##) und "Fileende" (#/) gesteuert.

2. Einschraenkungen

In der gegenwaertigen Ausbaustufe wird keine Silbentrennung realisiert. Diese kann aber in das entwickelte Programm problemlos eingebaut werden, da die Anschlussstelle elementar einfach ist. Das eigentliche Problem ist die Silbentrennung in der deutschen Schriftsprache.

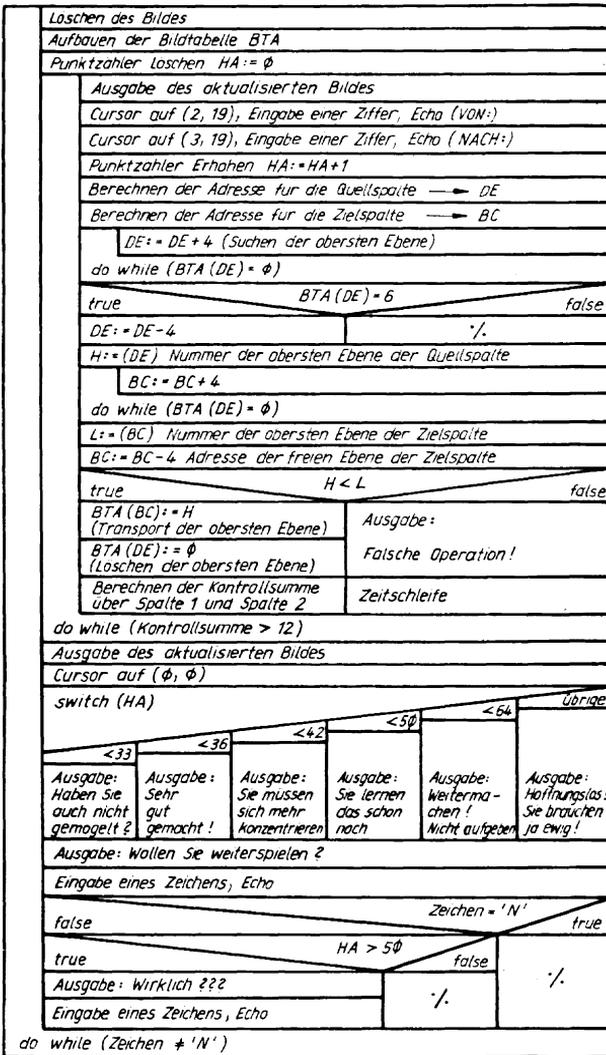
Bild 20. Aufbereiteter Text mit einer Zeilenlänge von 35 Zeichen

1. Einfuehrung in den Modul "Textaufbereitung"

Der Modul "Textaufbereitung" hat die Aufgabe, beliebig geschriebene Texte fuer den Druck mit einer vorgegebenen Zeilenlaenge aufzubereiten. Die Laenge der Druckzeile ist frei wahlbar. Die Aufbereitung wird durch die Steuerworte "Ueberschrift" (##), "Absatz" (##) und "Fileende" (#/) gesteuert.

Bild 21. Aufbereiteter Text mit einer Zeilenlänge von 60 Zeichen

Bild 22. Struktogramm  
des Programms  
»Turm von Hanoi«



Steuerzeichen (Carriage return – Wagenrücklauf). Die Zeilen 8 und 9 enthalten den sogenannten »Baugrund«. Gespeichert werden in der Bildtabelle die entsprechenden Hexadezimalcodes zur Bildsteuerung. Im Struktogramm wird von dieser Bildsteuerungsfunktion abstrahiert, und es werden für die Kodierung der Ebenen der Pyramide nur die niederwertigen 4 Bits betrach-

tet, also 0, 1, 2, 3, 4, 5 und 6 anstelle von A 0, A 1, A 2, A 3, A 4, A 5 und A 6.

Aus der eingegebenen Ziffer für die Quell- und Zielspalte wird die Adresse der Quell- und Zielspalte in der Bildtabelle berechnet, wobei die Doppelregister DE als Zeiger auf die Quellspalte und BC als Zeiger auf die Zielspalte benutzt werden.

Die Schreibweise BTA (DE) soll den entsprechenden Speicherplatz der Quellspalte der Bildtabelle bezeichnen, der durch DE adressiert wird. Durch Addition oder Subtraktion einer Vier wird der Zeiger DE auf die darunter- oder darüberliegende Ebene der Quellspalte bewegt. Das gleiche trifft auch auf den Zeiger BC zu.

Unter Beachtung der getroffenen Festlegungen läßt sich das Struktogramm in Bild 22 leicht lesen. Bestimmt wird an diesem Beispiel jeder feststellen, daß Struktogramme wesentlich zum Verständnis eines Programms beitragen können.

## 10. Schlußbemerkungen

Im Beitrag wurde neben einer Einführung in die Technik der Struktogramme der systematische Programm-entwurf an einem Beispiel aus der Textverarbeitung dargestellt. Es wurde gezeigt, daß mit der schrittweisen Verfeinerung des Entwurfes gut strukturierte und leicht verständliche Programme entwickelt werden können, was insbesondere für die Lebensfähigkeit eines Programms wichtig ist. So kann z. B. später ein Silbentrennungsprogramm zwischen Verarbeitungsmodul 1 und 2 eingefügt werden. Da die Silbentrennung in der deutschen Schriftsprache relativ kompliziert ist, wäre für den Anfang auch eine manuell gestützte Silbentrennung bei besonders langen Worten und einer größeren Restzeichenanzahl auf der Zeile möglich. Praktische Anwendungen zeigten, daß durch eine manuell gestützte Silbentrennung bei vertretbarem Aufwand ein relativ großer Nutzeffekt erreicht werden kann. Es stellte sich dabei auch heraus, daß bei einer Zeilenlänge von 60 Zeichen bei weniger als 8 Restzeichen pro Zeile eine Silbentrennung keinen Nutzeffekt

hat. Je kürzer eine Zeile wird, um so wichtiger ist allerdings die Silbentrennung.

Bei allen positiven Seiten der Struktogrammtechnik hat sie auch negative. Der schwerwiegendste Nachteil ist der relativ große Zeichenaufwand. Trotzdem ist meiner Meinung nach die Anwendung der Struktogrammtechnik in Verbindung mit dem systematischen Programm-entwurf nicht nur für den Anfänger ein Hilfsmittel. Zur Linderung des genannten Nachteils wurden aber auch schon spezielle Programme entwickelt, die nach der Testung der entwickelten Programme die Struktogramme automatisch ausdrucken ([4]). Damit entfällt der Zeichenaufwand bei der Erstellung der Programmdokumentation.

## Literatur

- [1] GEWALD, K.; HAAKE, G.; PFADLER, W.: Software Engineering: Grundlagen und Technik rationeller Programm-entwicklung. – München, Wien: R. Oldenbourg Verlag, 1977
- [2] HERRLICH, O.; LINDNER, U.: Strukturierte Programmierung. – Leipzig: BSB B. G. Teubner, 1981
- [3] NASSI, I.; SHNEIDERMAN, B.: Flow Chart Techniques for Structured Programming. – In: Sigplan Notices 8 (1975) 12
- [4] ROTHARDT, G.: Erfahrungen mit der strukturierten Programmierung. – In: edv-aspekte 3 (1982), S, 12–17

Autor:

*Doz. Dr. sc. techn. Thomas Horn*

DDR 8036 Dresden

Prohliser Straße 2b

Hochschuldozent

an der Sektion Informationsverarbeitung  
der Ingenieurhochschule Dresden

---

# Eine Mondlandung mit dem K 1003



Wer das Raumflugplanetarium in Halle besucht, kann sich die Erde aus einem Raumflugkörper betrachten, etwa so, wie sie SIEGMUND JÄHN bei seiner Dienstreise in den Kosmos zu Gesicht bekam. In Pionierhäusern und Stationen Junger Techniker gibt es Kosmonautenzentren und Verlage und Gesellschaften unterbreiten vielfältige Angebote, um uns allen den Kosmos ein Stück näher zu rücken. Es bietet sich deshalb an, selbst einmal, mit Hilfe eines Rechners, physikalischen Kenntnissen und etwas Geschick, einen Raumflugkörper weich auf dem Mond landen zu lassen. Für unser Anliegen ist es dabei gleichgültig, ob wir, wie die Amerikaner ARMSTRONG und ALDRIN, selbst im Mondlander sitzen oder eine der zahlreichen Mondsonden der UdSSR (Lunik- und Luna-Serie) vom Kontroll- und Leitzentrum auf der Erde steuern. Gefahr besteht für uns weder in dem einen noch im anderen Fall, lediglich die zeitliche Verzögerung der Steuersignale zur Mondsonde und zurück (zusammen etwa 2,5 s) wäre bei einer Fernsteuerung von der Erde aus noch zu berücksichtigen.

Wir wollen die Mondlandung mit dem programmierbaren Tischrechner K 1003 mit Drucker simulieren, wobei einige vereinfachende Annahmen nicht zu umgehen sind. Grundsätzlich läßt der Rechner uns, im Gegensatz zur Praxis,

zwischen den einzelnen Bremsmanövern beliebig lange Zeit zum Überlegen. Der Drucker wird das Protokoll erstellen und uns alle wichtigen Zwischenergebnisse ausdrucken. Er wird uns eine weiche Landung bescheinigen, wenn bei der Höhe  $h = 0\text{m}$ , also dem Erreichen der Mondoberfläche, die Geschwindigkeit  $v$  kleiner als  $2\text{ km/h}$  ist. Der Landevorgang wird durch zwei Umstände erschwert. Zum ersten ist der Treibstoffvorrat für die Bremstriebwerke begrenzt, denn ein Nachtanken ist in diesem Stadium nicht möglich. Zum zweiten können zu starke Bremschübe dazu führen, daß sich der Raumflugkörper wieder vom Mond entfernt und in diesem Fall der Treibstoff für einen zweiten Versuch mit großer Wahrscheinlichkeit nicht ausreicht. Wir werden also Schritt für Schritt und wohl dosiert eine optimale Treibstoffmenge in den Rechner eingeben und damit ein kurzzeitiges Zünden der Bremstriebwerke simulieren. Dabei müssen wir die noch verfügbare Treibstoffmenge, die verbleibende Höhe und die Momentangeschwindigkeit im Auge behalten.

Zunächst aber einiges zu den physikalischen Grundlagen. Wir gehen davon aus, daß eine Abbremsung, also eine kurzzeitige Zündung der Bremstriebwerke, eine gleichmäßig beschleunigte Bewegung erzeugt. Diese Verein-

fachung gestattet uns die Verwendung des

Weg - Zeit - Gesetzes

$$h_t = h_0 - v_0 \cdot t + \frac{a}{2} \cdot t^2 \quad (1)$$

und des Geschwindigkeit - Zeit - Gesetzes

$$v_t = v_0 + a \cdot t. \quad (2)$$

Es gilt:

$h_0$  Entfernung des Raumflugkörpers zum Mond vor dem Bremsschub

$h_t$  Entfernung des Raumflugkörpers zum Mond nach dem Bremsschub

$v_0$  Geschwindigkeit des Raumflugkörpers vor dem Bremsschub

$v_t$  Geschwindigkeit des Raumflugkörpers nach dem Bremsschub

$t$  Zeitdauer des Bremsschubes

$a$  Beschleunigung.

Die Beschleunigung  $a$  setzt sich aus der auf der Oberfläche des Mondes wirkenden Fallbeschleunigung  $a_M = 1,62 \frac{m}{s^2}$  und der bremsenden, also negativen Beschleunigung der Bremstriebwerke  $a_B$  zusammen. Damit gilt

$$a = a_M - a_B. \quad (3)$$

Da in unserem Fall nur Entfernungen von maximal 2000 m von der Mondoberfläche sinnvoll sind, arbeiten wir, vereinfachend für unsere Rechnung, durchgängig mit  $a_M = 1,62 \frac{m}{s^2}$ . Für die Bestimmung der Bremsbeschleunigung  $a_B$  werden wir Festlegungen treffen, die eine direkte Zuordnung zur eingespritzten Treibstoffmenge ermöglichen.

Es ist

$$T_t = T_0 - T_{\text{ein}}, \quad (4)$$

wobei gilt:

$T_0$  Treibstoffvorrat vor dem Bremsschub

$T_t$  Treibstoffvorrat nach dem Bremsschub

$T_{\text{ein}}$  eingespritzte Treibstoffmenge.

Wir treffen folgende Festlegungen:

1. Ein Bremsschub (das entspricht einer Eingabe in den Rechner) dauert  $t = 1$  s.

2. 1 kg Treibstoff erzeugt eine negative Beschleunigung von  $a_B = 1 \frac{m}{s^2}$ .

Damit vereinfachen sich die Gleichungen (1) und (2) wie folgt:

$$h_t = h_0 - v_0 + \left( \frac{1,62 - T_{\text{ein}}}{2} \right) \quad (5)$$

$$v_t = v_0 + (1,62 - T_{\text{ein}}), \quad (6)$$

wobei gilt:

$h$  in m;  $v$  in  $\frac{m}{s}$

$T_{\text{ein}}$  in kg  $\left( 1 \text{ kg} \triangleq 1 \frac{m}{s^2} \right)$ .

Als Fahrrad-, Moped-, Motorrad- und Autofahrer ist uns die Angabe der Geschwindigkeit in km/h geläufiger. Wir rechnen deshalb die Geschwindigkeiten in km/h um und erhalten die zugechnittenen Größengleichungen:

$$h_t = h_0 - \frac{v_0}{3,6} + \left( 0,81 - \frac{T_{\text{ein}}}{2} \right) \quad (7)$$

$$v_t = v_0 + (5,832 - 3,6 \cdot T_{\text{ein}}), \quad (8)$$

wobei gilt:

$h$  in m;  $v$  in  $\frac{km}{h}$

$T_{\text{ein}}$  in kg  $\left( 1 \text{ kg} \triangleq 1 \frac{m}{s^2} \right)$ .

Mit Hilfe der Gleichungen (7) und (8) erhalten wir nach Eingabe der Treibstoffmenge für die Bremstriebwerke

die neue Entfernung  $h_t$  und die neue Geschwindigkeit  $v_t$ . Im Programmablauf müssen aber noch folgende Situationen berücksichtigt werden:

1. Es muß bei jedem Bremsvorgang (also Rechnerdurchlauf) der Tankinhalt überprüft werden, um »Mogeleien« durch den Bediener zu unterbinden. Ist der Tank leer, dann ist nur die Eingabe  $T_{\text{ein}} = 0$  möglich. Der Raumflugkörper nähert sich dann ungebremst mit  $\alpha_M$  der Mondoberfläche.

2. Eine negative Höhe  $h_t$  kann es, sofern man keinen neuen Mondkrater schaffen will, nicht geben. Es bietet sich deshalb an,  $h_t < 0$  als Abbruchbedingung für die bisher gemachten Ausführungen zu nutzen und die Endphase des Landevorganges näher zu betrachten.

In der Endphase interessiert die Aufprallgeschwindigkeit  $v_A$  auf die Mondoberfläche. Wir legen folgende Bereiche fest:

Weiche Landung, wenn  $v_A < 2 \frac{\text{km}}{\text{h}}$ ,

harte Landung, wenn  $2 \leq v_A < 12 \frac{\text{km}}{\text{h}}$ ,

Bruchlandung, wenn  $12 \leq v_A < 35 \frac{\text{km}}{\text{h}}$ ,

Absturz, wenn  $v_A \geq 35 \text{ km/h}$ .

Die Berechnung der Aufprallgeschwindigkeit  $v_A$  setzt die Ermittlung der Zeit  $t$  in der Endphase bis zum Erreichen der Mondoberfläche voraus. Dabei soll natürlich auch die zuletzt eingegebene Treibstoffmenge  $T_{\text{ein}}$  (sofern noch Treibstoff vorhanden ist) mit berücksichtigt werden. Dazu ist das Weg-Zeit-Gesetz (1) nach  $t$  umzustellen. Für  $h_t = 0$  ergibt sich:

$$\frac{a}{2} \cdot t^2 - v_0 \cdot t + h_0 = 0,$$

woraus folgt:

$$t^2 - \frac{v_0}{\frac{a}{2}} \cdot t + \frac{h_0}{\frac{a}{2}} = 0.$$

Daraus folgt die Gleichung:

$$t^2 - \frac{v_0}{0,81 - \frac{T_{\text{ein}}}{2}} \cdot t + \frac{h_0}{0,81 - \frac{T_{\text{ein}}}{2}} = 0.$$

Die Lösungen der Gleichung lauten:

$$t_{A,2} = \frac{v_0}{7,2 \cdot \left(0,81 - \frac{T_{\text{ein}}}{2}\right)} \pm \sqrt{\left(\frac{v_0}{7,2 \cdot \left(0,81 - \frac{T_{\text{ein}}}{2}\right)}\right)^2 - \frac{h_0}{0,81 - \frac{T_{\text{ein}}}{2}}} \quad (9)$$

Wir setzen  $F = \frac{v_0}{7,2 \cdot \left(0,81 - \frac{T_{\text{ein}}}{2}\right)}$ . Der

Faktor  $F$  kann positiv oder negativ sein. Da  $t_A$  nicht negativ werden kann, gilt für  $F \leq 0$  das Pluszeichen vor der Wurzel in Gleichung (9).

Wenn  $F > 0$  ist, dann wird für die Lösung das negative Vorzeichen vor der Wurzel benutzt, damit das kleinste, aber positive  $t_A$  ermittelt wird. Aus der Zeit  $t_A$ , die bis zum Aufprall verbleibt, läßt sich die Aufprallgeschwindigkeit  $v_A$  nach folgender Gleichung ermitteln:

$$v_A = v_0 + (5,832 - 3,6 \cdot T_{\text{ein}}) \cdot t_A. \quad (10)$$

Damit liegen alle Gleichungen zur Simulation der Mondlandung vor. Bild 1 zeigt den daraus abgeleiteten Programmablaufplan (PAP). Aus ihm geht hervor, daß die Anfangswerte  $h_0$ ,  $v_0$ ,  $T_0$  frei wählbar sind. Allerdings ist hier auf sinnvolle Kombinationen zu achten. So sollten die Anfangshöhen zwischen 200 m und 2000 m gewählt werden. Für

die Anfangsgeschwindigkeiten sind Werte zwischen 200 km/h und 900 km/h sinnvoll. Die Geschwindigkeit sollte 3 Stellen nicht überschreiten, da sonst das Druckformat im vorliegenden Programm gestört wird. Der Treibstoffvorrat sollte zwischen 50 kg und 300 kg betragen. Mit dieser freien Wählbarkeit der Anfangswerte wird von vornherein ein bestimmter Schwierigkeitsgrad (z.B. große Geschwindigkeit bei geringem Treibstoffvorrat) festgelegt

und der Variantenreichtum dieses Spiels erhöht.

Im PAP nach Bild 1 wird dann, sofern der Selektor beim K 1003 noch nicht gesetzt ist, die Treibstoffmenge  $T_{ein}$  von Hand eingegeben. Wird versucht, mehr Treibstoff als vorhanden einzugeben, so erfolgt nach einem Hinweis auf dem Drucker der Rücksprung zur gleichen Handeingabe. Dann wird der noch verfügbare Treibstoffrest  $T_t$  errechnet. Bei leerem Tank ist  $T_t = 0$ . In

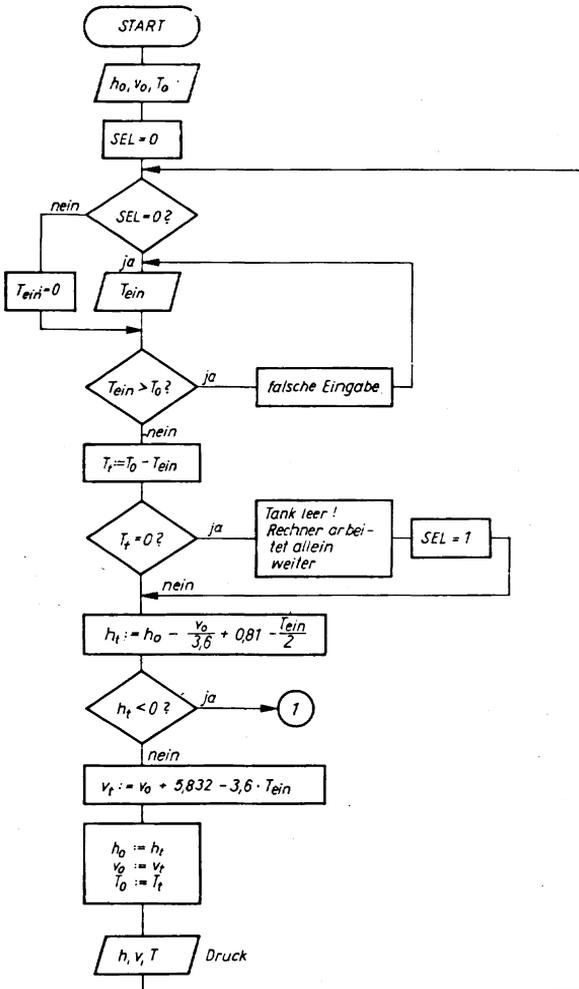
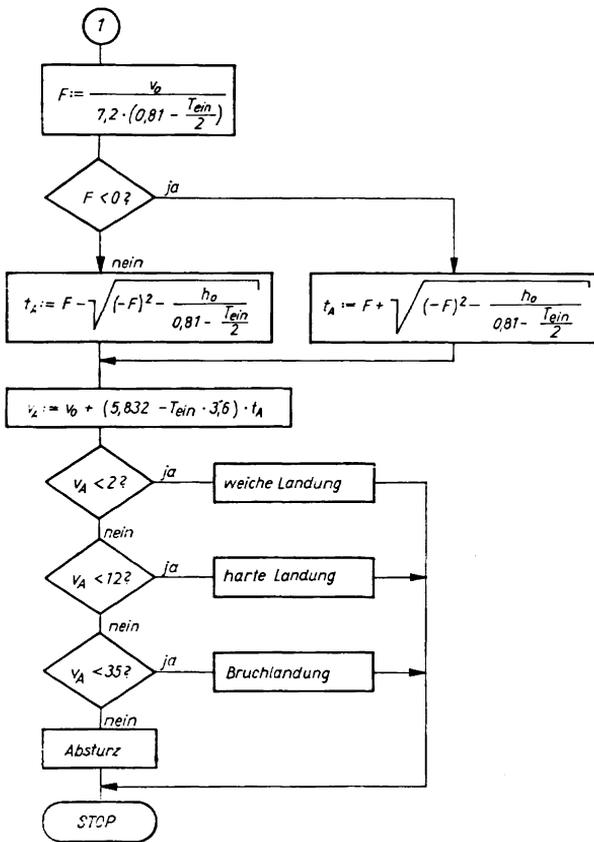


Bild 1. Programmablaufplan zum Mondlandespiel (Bedeutung der Kurzzeichen im Text)



(Bild 1 Fortsetzung)

diesem Fall wird der Selektor gesetzt und damit bewirkt, daß der Rechner vom nächsten Durchlauf an automatisch für  $T_{\text{ein}} = 0$  eingibt. Im Anschluß daran wird die noch verbleibende Entfernung  $h_t$  zwischen Raumflugkörper und Mondoberfläche ermittelt. Falls  $h_t < 0$  ist, wird der Rechengang unterbrochen und mit den Werten  $h_0$ ,  $v_0$  und  $T_{\text{ein}}$  die Aufprallgeschwindigkeit  $v_A$  in der Endphase errechnet.

Für  $h_t \geq 0$  wird die aktuelle Sinkgeschwindigkeit  $v_t$  ermittelt. Nach Umweisung der neuen zu alten Werten wird ein weiterer Rechnerdurchlauf durch Handeingabe von  $T_{\text{ein}}$  oder der automatischen Eingabe von  $T_{\text{ein}} = 0$  durch den Rechner eingeleitet. Das Programm

endet stets mit der Ermittlung der Aufprallgeschwindigkeit  $v_A$ . Mit den vier möglichen Landungsarten lassen sich Texte drucken, die anerkennende, kritische oder humoristische Bemerkungen enthalten. Dem Passagier im Mondlander würden diese Texte bei einer Bruchlandung oder einem Absturz (siehe Bild 1 b) freilich kaum mehr nützen.

Das Rechenprogramm für den K 1003 (Bild 2) umfaßt 962 Befehle. Damit wird etwa ein Viertel des Speicherbereiches der 4-KByte-Version des K 1003 genutzt. Den meisten Speicherplatz

Bild 2. Vollständiges Programm für den K 1003 (Seite 54 bis 59)

0000	NRK	0040	U	0080	W	0120	
0001	1	0041	N	0081	.	0121	K
0002	TEX	0042	G	0082		0122	G
0003	DRU	0043	ZS	0083	V	0123	=
0004	U	0044	ZS	0084	ZS	0124	TEX
0005	1	0045	A	0085	I	0125	STP
0006	DRU	0046	N	0086	N	0126	TXR
0007	4	0047	F	0087		0127	3
0008	0	0048	.	0088	K	0128	TEX
0009	DRU	0049	H	0089	M	0129	DRU
0010	U	0050	O	0090	/	0130	3
0011	2	0051	E	0091	H	0131	0
0012	DRU	0052	H	0092	=	0132	ZS
0013	3	0053	E	0093	TEX	0133	ZS
0014	0	0054		0094	STP	0134	T
0015	DRU	0055	H	0095	TXR	0135	R
0016	U	0056	ZS	0096	2	0136	E
0017	3	0057	I	0097	TEX	0137	I
0018	DRU	0058	N	0098	DRU	0138	B
0019	3	0059		0099	3	0139	S
0020	0	0060	M	0100	0	0140	T
0021	TEX	0061	=	0101	ZS	0141	O
0022	UP	0062	TEX	0102	T	0142	F
0023	NRK	0063	STP	0103	R	0143	F
0024	STM	0064	TXR	0104	E	0144	
0025	S=0	0065	1	0105	I	0145	B
0026	NUM	0066	TEX	0106	B	0146	E
0027	KOM	0067	DRU	0107	S	0147	I
0028	GL	0068	4	0108	T	0148	ZS
0029	TEX	0069	0	0109	.	0149	N
0030		0070	ZS	0110	V	0150	U
0031		0071	A	0111	O	0151	M
0032	M	0072	N	0112	R	0152	=
0033	O	0073	F	0113	R	0153	0
0034	N	0074	.	0114	A	0154	4
0035	D	0075	G	0115	T	0155	
0036	L	0076	E	0116		0156	E
0037	A	0077	S	0117	T	0157	I
0038	N	0078	C	0118	I	0158	N
0039	D	0079	H	0119	N	0159	G

0160	E	0199		0239	4	0279	I
0161	B	0200	M	0240	ST	0280	T
0162	E	0201		0241	ST	0281	E
0163	N	0202		0242	TEX	0282	N
0164	ZS	0203	K	0243	S	0283	
0165	ZS	0204	M	0244	I	0284	S
0166		0205	/	0245	E	0285	I
0167		0206	H	0246		0286	E
0168	H	0207		0247	K	0287	ZS
0169		0208	K	0248	O	0288	K
0170		0209	G	0249	E	0289	O
0171		0210		0250	N	0290	R
0172	V	0211	K	0251	N	0291	R
0173		0212	G	0252	E	0292	E
0174		0213	TEX	0253	N	0293	K
0175		0214	ZS	0254		0294	T
0176	T	0215	STM	0255	D	0295	.
0177		0216	UP	0256	O	0296	M
0178		0217	1	0257	C	0297	D
0179	E	0218	MRK	0258	H	0298	L
0180	I	0219	2	0259	N	0299	G
0181	N	0220	NUM	0260	I	0300	.
0182		0221	3	0261	C	0301	!
0183		0222	GL	0262	H	0302	TEX
0184	I	0223	SEL	0263	T	0303	STM
0185	N	0224	STM	0264		0304	2
0186		0225	3	0265	N	0305	MRK
0187		0226	ST	0266	R	0306	4
0188	I	0227	ST	0267	C	0307	TRX
0189	N	0228	STP	0268	H	0308	4
0190		0229	MRK	0269	T	0309	TEX
0191		0230	3	0270	R	0310	
0192	I	0231	TRX	0271	N	0311	
0193	N	0232	4	0272	K	0312	
0194		0233	TRX	0273	E	0313	
0195	I	0234	3	0274	N	0314	
0196	N	0235	VXY	0275	R	0315	
0197	ZS	0236	SUB	0276	R	0316	
0198		0237	>=0	0277	B	0317	
0199		0238	STM	0278	E	0318	

0319		0359	E	0399	3	0439	STM
0320		0360	E	0400	DP	0440	DP
0321		0361	R	0401	6	0441	1
0322		0362	I	0402	DIV	0442	STM
0323	DRU	0363	ZS	0403	SUB	0443	2
0324	2	0364	I	0404	DP	0444	MRK
0325	0	0365	C	0405	8	0445	7
0326	TEX	0366	H	0406	1	0446	DP
0327	TRX	0367		0407	ADD	0447	8
0328	3	0368	A	0408	TRX	0448	1
0329	TRX	0369	R	0409	4	0449	TRX
0330	4	0370	B	0410	MOD	0450	4
0331	SUB	0371	E	0411	2	0451	MOD
0332	TXR	0372	I	0412	DIV	0452	2
0333	3	0373	T	0413	SUB	0453	DIV
0334	=0	0374	E	0414	<0	0454	SUB
0335	STM	0375	ZS	0415	STM	0455	7
0336	5	0376	A	0416	7	0456	DP
0337	ST	0377	L	0417	ST	0457	2
0338	ST	0378	L	0418	ST	0458	MUL
0339	STM	0379	E	0419	TXR	0459	1/X
0340	6	0380	I	0420	1	0460	TRX
0341	MRK	0381	N	0421	TRX	0461	2
0342	5	0382		0422	2	0462	MUL
0343	SEL	0383	W	0423	MOD	0463	TXR
0344	STM	0384	E	0424	5	0464	5
0345	6	0385	I	0425	DP	0465	MOD
0346	ST	0386	T	0426	8	0466	DP
0347	ST	0387	E	0427	3	0467	8
0348	TEX	0388	R	0428	2	0468	1
0349	T	0389	I	0429	ADD	0469	TRX
0350	A	0390	TEX	0430	3	0470	4
0351	N	0391	S=1	0431	DP	0471	MOD
0352	K	0392	MRK	0432	6	0472	2
0353		0393	6	0433	TRX	0473	DIV
0354	I	0394	TRX	0434	4	0474	SUB
0355	S	0395	1	0435	MUL	0475	1/X
0356	T	0396	TRX	0436	SUB	0476	TRX
0357		0397	2	0437	TXR	0477	1
0358	L	0398	MOD	0438	2	0478	MUL

0472	TRX	0519	ADD	0559		0599	SUB
0480	5	0520	KNO	0560	Y	0600	<0
0481	X+2	0521	KNO	0561	0	0601	STM
0482	VXY	0522	TEX	0562	N	0602	SUB
0483	SUB	0523	S	0563	DRU	0603	ST
0484	QWX	0524	I	0564	3	0604	ST
0485	TRX	0525	E	0565	0	0605	TEX
0486	6	0526		0566	K	0606	M
0487	TRX	0527	S	0567	M	0607	I
0488	5	0528	I	0568	/	0608	T
0489	=0	0529	N	0569	H	0609	
0490	ADD	0530	D	0570	G	0610	I
0491	STM	0531		0571	E	0611	H
0492	8	0532	M	0572	L	0612	R
0493	ST	0533	I	0573	R	0613	E
0494	<0	0534	T	0574	N	0614	M
0495	ADD	0535		0575	D	0615	
0496	STM	0536	E	0576	E	0616	A
0497	8	0537	I	0577	T	0617	B
0498	ST	0538	-	0578	!	0618	-
0499	VXY	0539	N	0579	TEX	0619	Z5
0500	SUB	0540	E	0580	2	0620	S
0501	MRK	0541	R	0581	SUB	0621	T
0502	8	0542		0582	<0	0622	U
0503	3	0543	G	0583	STM	0623	R
0504	DP	0544	E	0584	DIY	0624	Z
0505	6	0545	S	0585	ST	0625	
0506	TRX	0546	C	0586	ST	0626	H
0507	4	0547	H	0587	KNU	0627	A
0508	MUL	0548	W	0588	1	0628	B
0509	5	0549	I	0589	2	0629	E
0510	DP	0550	N	0590	SUB	0630	N
0511	8	0551	D	0591	<0	0631	
0512	3	0552	I	0592	STM	0632	S
0513	2	0553	G	0593	MUL	0633	I
0514	VXY	0554	-	0594	ST	0634	E
0515	SUB	0555	K	0595	ST	0635	Z5
0516	MUL	0556	E	0596	KNU	0636	E
0517	TRX	0557	I	0597	3	0637	I
0518	2	0558	T	0598	5	0638	N

0639	E	0679	S	0719		0759	K
0640	N	0680	T	0720	E	0760	E
0641		0681	U	0721	I	0761	H
0642	N	0682	M	0722	N	0762	R
0643	E	0683		0723	E	0763	
0644	U	0684	I	0724	ZS	0764	I
0645	E	0685	H	0725	B	0765	S
0646	N	0686	R	0726	R	0766	T
0647	ZS	0687	E	0727	U	0767	ZS
0648	M	0688	N	0728	C	0768	L
0649	G	0689		0729	H	0769	E
0650	N	0690	N	0730	L	0770	I
0651	D	0691	A	0731	A	0771	D
0652	K	0692	-	0732	N	0772	E
0653	R	0693	M	0733	D	0773	R
0654	A	0694	E	0734	U	0774	
0655	T	0695	N	0735	N	0775	U
0656	E	0696		0736	G	0776	N
0657	R	0697	E	0737	ZS	0777	M
0658		0698	R	0738	V	0778	O
0659	G	0699	H	0739	E	0779	E
0660	E	0700	A	0740	R	0780	G
0661	-	0701	E	0741	U	0781	-
0662	ZS	0702	L	0742	R	0782	ZS
0663	S	0703	T	0743	S	0783	L
0664	C	0704	.	0744	A	0784	I
0665	H	0705	TEX	0745	C	0785	C
0666	A	0706	STP	0746	H	0786	H
0667	F	0707	NRK	0747	I	0787	.
0668	F	0708	SUB	0748	.	0788	
0669	E	0709	TEX	0749		0789	A
0670	N	0710	S	0750	E	0790	B
0671	,	0711	I	0751	I	0791	S
0672		0712	E	0752	N	0792	C
0673	D	0713		0753	E	0793	H
0674	E	0714	H	0754	R	0794	I
0675	R	0715	A	0755	U	0795	E
0676	ZS	0716	B	0756	E	0796	D
0677	P	0717	E	0757	C	0797	S
0678	O	0718	N	0758	K	0798	-

0799	G	0840		0881		0922	G
0800	R	0841	W	0882	A	0923	
0801	U	0842	A	0883	U	0924	L
0802	E	0843	R	0884	F	0925	A
0803	S	0844	S	0885		0926	E
0804	S	0845	P	0886	D	0927	S
0805	E	0846	O	0887	E	0928	S
0806		0847	R	0888	N	0929	T
0807	N	0848	T	0889	ZS	0930	
0808	I	0849	L	0890	R	0931	T
0809	C	0850	I	0891	U	0932	R
0810	H	0851	C	0892	E	0933	A
0811	T	0852	H	0893	C	0934	-
0812	ZS	0853		0894	K	0935	B
0813	V	0854	H	0895	E	0936	A
0814	E	0855	A	0896	N	0937	N
0815	R	0856	R	0897	.	0938	T
0816	G	0857	T	0898	TEX	0939	-
0817	E	0858	.	0899	STP	0940	F
0818	S	0859	ZS	0900	MRK	0941	A
0819	S	0860	K	0901	DIV	0942	H
0820	E	0861	L	0902	TEX	0943	R
0821	N	0862	E	0903	I	0944	E
0822	.	0863	B	0904	H	0945	R
0823	TEX	0864	E	0905	R	0946	
0824	STP	0865	N	0906	E	0947	E
0825	MRK	0866		0907		0948	R
0826	MUL	0867	S	0908	W	0949	-
0827	TEX	0868	I	0909	E	0950	ZS
0828	I	0869	E	0910	I	0951	B
0829	H	0870		0911	C	0952	L
0830	R	0871	S	0912	H	0953	A
0831	E	0872	I	0913	E	0954	S
0832		0873	C	0914		0955	S
0833	L	0874	H	0915	L	0956	E
0834	A	0875	ZS	0916	A	0957	N
0835	N	0876	E	0917	N	0958	.
0836	D	0877	I	0918	-	0959	TEX
0837	U	0878	N	0919	D	0960	STP
0838	N	0879		0920	U	0961	END
0839	G	0880	A	0921	N		

benötigen auch hier die Textausgaben. so endet die Berechnung der Aufprallgeschwindigkeit schon beim Befehl 0519. Aber auch vom Programmanfang bis zum Befehl 0390 wird der meiste Speicherplatz für Texte benötigt. Man sollte auf diese Texte aber nicht verzichten. Sie machen eine Bedienungsanleitung überflüssig, erhöhen scheinbar die Intelligenz des Rechners und steigern die Freude am Spiel. Das gesamte Programm findet auf 5 Magnetkartenspuren Platz, da ja eine Spur etwa 200 Byte (1 Byte = 1 Befehl) speichert.

Das Programm wird bei der Marke STM gestartet und beginnt beim Befehl 0023. Die ersten 23 Befehle des Programms gehören zum Zeilendruckprogramm für  $h$ ,  $v$  und  $T$ , das als Unterprogramm arbeitet. Aus Zeitgründen wurde es an den Anfang des Programmspeichers gelegt. Das gesamte Programm ist symbolisch, also durch Vergabe von Marken, adressiert. Der geübte Nutzer merkt das, besonders beim Ausdruck des Schlußtextes, da beim K 1003 die Markensuche nur sehr langsam vonstatten geht. Wer das Programm schneller machen will, muß es auf absolute Adressierung umschreiben.

Für das Programm gilt folgender Datenbelegungsplan (Abkürzungen siehe Bild 1 und Text):

<001> =  $h_0, h_t$

<002> =  $v_0, v_t$

<003> =  $T_0, T_t$

<004> =  $T_{\text{ein}}$

<005> =  $F$

<006> = Wurzelausdruck.

Damit kann die Speicherbereichsverteilung (Daten – Programm – Speicher) wie sie beim Einschalten des Rechners vorliegt, beibehalten werden.

Die Bilder 3 bis 5 (S. 61 ff.) zeigen Berechnungsbeispiele für die möglichen Landungsvarianten. Die Beispiele ma-

chen deutlich, daß der Spielverlauf vom Geschick des Nutzers, aber vor allem auch von sinnvollen Ausgangswerten bestimmt wird. Eine Spielvariante für mehrere Personen besteht darin, daß bei gleicher Anfangshöhe und gleicher Anfangsgeschwindigkeit eine minimale Aufprallgeschwindigkeit mit minimalem Treibstoffverbrauch angestrebt wird.

Im Druckprotokoll wird auf ganze Zahlen gerundet, obwohl der Rechner intern mit 12stelliger Genauigkeit rechnet. Diese, in kritischen Situationen störende Ungenauigkeit im Ausdruck wurde dem übersichtlichen Druckbild untergeordnet. Die manchmal auftauchenden negativen Vorzeichen gehören zur Geschwindigkeitsangabe. In diesem Fall haben wir zu stark abgebremst, so daß wir uns wieder von der Mondoberfläche entfernen. Verschiebungen im Druckbild können nur dann auftreten, wenn die Geschwindigkeit auf vier Stellen anwächst oder wenn die Höhe zufällig so gering wird, daß der Rechner automatisch auf Gleitkommaformat umschaltet (siehe auch Bedienhandbuch zum K 1003). Die Rechnung wird aber auch in diesen Fällen exakt fortgesetzt.

Natürlich werden viele weiche Landungen noch nicht zu einem Führerschein in den Kosmos reichen, aber ein Beweis für einen gefühlvollen Umgang mit Physik und Technik sind sie gewiß.

Autor:

*Dr. Hannes Gutzer*

DDR 4090 Halle-Neustadt  
Am Südpark 581/6

Leiter der Informationsstelle  
des Zentralinstituts für Schweißtechnik  
Halle

MONDLANDUNG	10	3	19						
									2
ANF.HOEHE H	9	1	17						
IN M=									0
200	10	7	17						
ANF.GESCHW. V									2
IN KM/H=	7	6	15						
200									2
TREIBST.VORRAT T	6	4	13						
IN KG=									2
90	4	3	11						1
TREIBSTOFF BEI	4	5	10						
NUM=04 EINGEBEN									2
	2	4	8						
H V T EIN									2
IN IN IN IN									
M KM/H KG KG	1	3	6						2
200 200 90									
									10
140 170 80									
									10
89 140 70									
									20
41 73 50									
									25
9- 11 25									
									0
13- 5 25									
									0
15 1 25									
									0
15 7 25									
									2
13 5 23									
									2
11 4 21									
									2

SIE SIND MIT EINER GESCHWINDIGKEIT VON 1KM/H GELANDET!  
 IHRE WEICHE LAN- DUNG LAESST TRA- BANT-FAHRER ER- BLASSEN.

Bild 3. Berech- nungsbeispiel für eine weiche Landung

HONDLANDUNG				235	236	100	
							20
ANF.HOEHE H				160	172	80	
IN M=							20
2000				103	106	60	
ANF.GESCHW. V							10
IN KM/H=				69	76	50	
900							20
Treibst.Vorrat T				39	10	30	
IN KG=							5
300				35	3	25	
							0
Treibstoff bei				36	3	25	
NUH=04 eingeben							0
				36	9	25	
H	V	T	Ein:				0
IN	IN	IN	IN	34	15	25	
M	KM/H	KG	KG				0
				31	21	25	
2000	900	300					0
			20	26	27	25	
1741	834	280					0
			20	19	32	25	
1500	768	260					2
			20	10	31	23	
1278	701	240					8
			20				
1074	625	220					
			20				
880	569	200					
			20				
721	503	180					
			20				
572	437	160					
			20				
441	371	140					
			20				
329	304	120					
			20				

SIE SIND MIT EINER GESCHWINDIGKEIT VON 11KM/H GELANDET!  
Ihre Landung war sofortlich hart.  
KLEBEN SIE SICH EIN AUF DEN RUECKEN.

**Bild 4. Berechnungsbeispiel für eine harte Landung**

MONDLANDUNG

KORREKT, WDLG.!

15

ANF. HOEHE H

IN M=

1000

ANF. GESCHW. V

IN KM/H=

500

TREIBST. VORRAT T

IN KG=

150

TREIBSTOFF BEI

NUM=04 EINGEBEN

H	V	T	EIN
IN	IN	IN	IN
M	KM/H	KG	KG

1000 500 150

0

862 506 150

0

722 512 150

0

581 517 150

30

423 415 120

30

293 313 90

20

197 247 70

30

114 145 40

25

63 61 15

SIE KOENNEN DOCH  
NICHT NACHTANKEN  
ARBEITEN SIE

TANK IST LEER!

ICH ARBEITE

ALLEIN WEITER!

39 12 0

36 18 0

32 24 0

26 30 0

19 36 0

9 42 0

SIE SIND MIT EI-  
NER GESCHWINDIG-  
KEIT VON 47KM/H  
GELANDET!

MIT IHREM AB-  
STURZ HABEN SIE  
EINEN NEUEN  
MONDKRATER GE-  
SCHAFFEN, DER  
POSTUM IHREN NA-  
MEN ERHAELT.

Bild 5. Berech-  
nungsbeispiel  
für einen Ab-  
sturz

---

# Der Taschenrechner hilft, mathematische Gesetzmäßigkeiten zu entdecken

Lösen Sie mit Ihrem Taschenrechner  
die Aufgaben

$$9 \cdot 1 + 2 =$$

$$9 \cdot 12 + 3 =$$

$$9 \cdot 123 + 4 =$$

usw.

Was entdecken Sie? Warum ist das so?

Berechnen Sie die fünften Potenzen  
der natürlichen Zahlen 1, 2, 3, . . .

Was stellen Sie fest, wenn Sie die End-  
ziffern der Ergebnisse betrachten?

Muß das immer so sein?

Berechnen Sie

$$12345679 \cdot 0,9 =$$

$$12345679 \cdot 1,8 =$$

$$12345679 \cdot 2,7 =$$

usw.

---

© VEB Fachbuchverlag Leipzig 1985

1. Auflage

Lizenznummer 114-210/1/85

LSV 1083

Verlagslektor: Helga Fago

Gestaltung: Lothar Gabler

Printed in GDR

Satz und Druck:

Messedruck Leipzig, Bereich Borsdorf

III-18-328

Redaktionsschluß: 15. 3. 1985

Bestellnummer: 5470102

00780

---

---

# Vorschau

## Heft 4

*Horn*: Programmiersprachen – ein Vergleich an Hand von Beispielen

*Gutzer*: Archimedes und die Zahl  $\pi$

*Schönfelder*: Computerspiele – mehr als eine Spielerei



Die Broschürenreihe

## KLEINSTRECHNER-TIPS

behandelt

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

und will dem Laien auf dem Gebiet der Informationsverarbeitung Anregungen geben für seine Arbeit mit

- einfachen oder programmierbaren Tisch- und Taschenrechnern,
- Klein- und Videocomputern,
- Mikrorechnern

und anderen modernen Rechenhilfsmitteln.