20,- DM
Ös 170,-
sfr 20,-

SONDERHEFT
Joyce Nr. 1

JOYCE

eine Publikation der PC Schneider International

Sonderheft
Joyce
1/87



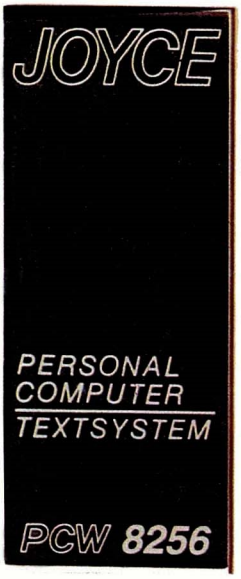
Mathematische Kurse:
Programmbeispiele

Tips:
zu BASIC, LOGO, dBase II

Masken-generator:
Komfortable Erstellung von Bildmasken

XBIOS:
Im Herzen von CP/M Plus





Schneider

HELLIGKEIT

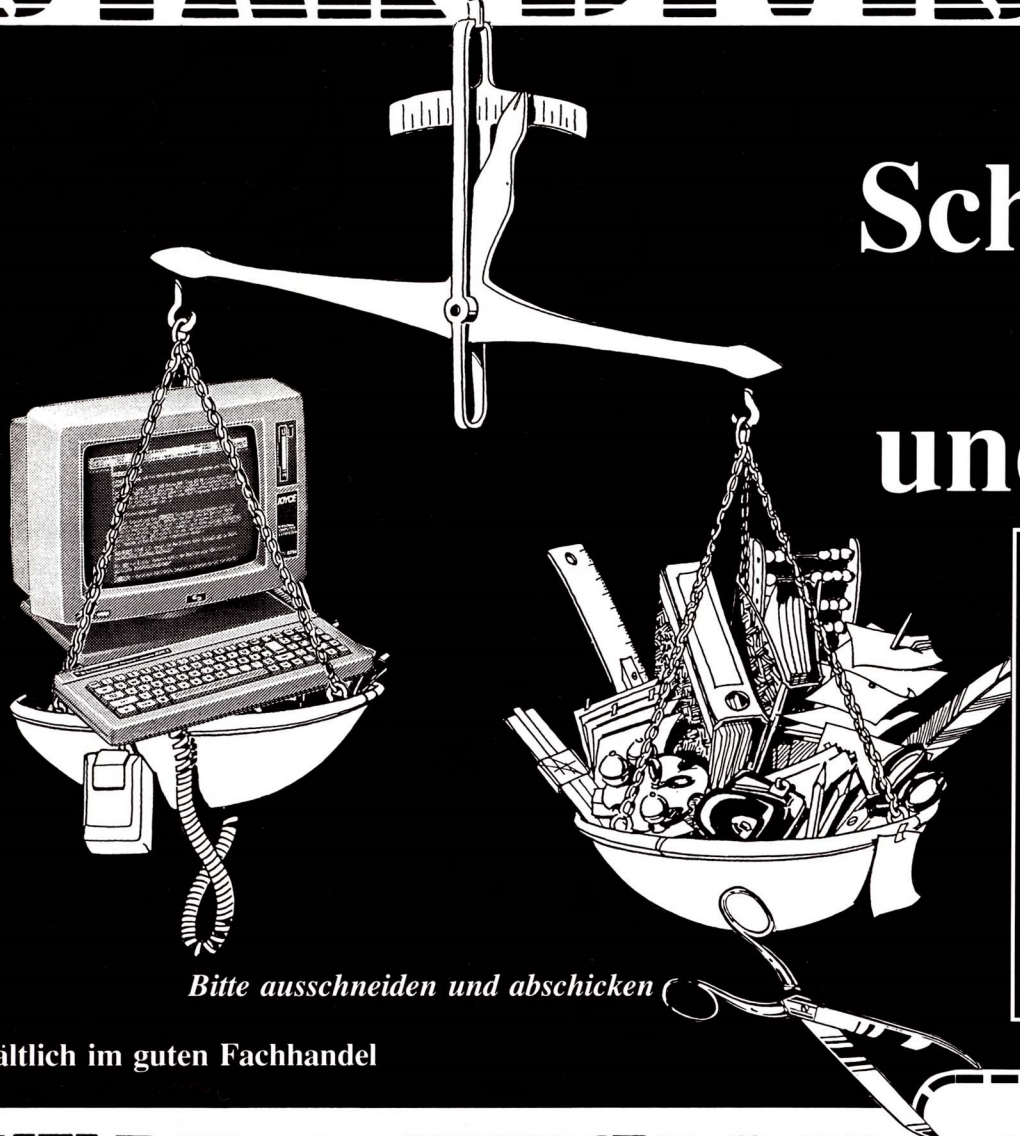
Die Zukunft hat begonnen!

Super-Software von

Günstige Preise

STAR-DIVISION

für Schneider Joyce und CPC



CPC
STAR-WRITER I
DATEI-STAR
Mouse Operating System
STATISTIC-STAR
FIBU-STAR Plus
u.v.m

Joyce
Joyce-Mailing-System
STATISTIC-STAR
STAR-BASE
BUSINESS-STAR
BUISINESS-STAR Plus
FIBU-STAR
u.v.m.

Bitte ausschneiden und abschicken

Erhältlich im guten Fachhandel

STAR-DIVISION

Uelzener Straße 12 • 2120 Lüneburg • Tel.: (0 41 31) 40 25 50

Vertrieb für die Schweiz.
VCS Video-Computershop • Schaffhauserstr. 473 • P.O.Box 103 • CH-8052 Zürich • Tel.:01/3 022600

Vertrieb für Österreich:
Ueberreuther-Media • Alserstr. 24 • A-1091 Wien • Tel.: 4 81 53 80

Copyright by STAR-DIVISION GmbH 1987

Info-Coupon

Ich möchte mehr Informationen über das Software-Lieferprogramm Joyce ☐ CPC ☐

Name, Vorname

Strasse, Nr.

PLZ, Ort

Bitte ausschneiden, auf eine Postkarte kleben oder in einen Briefumschlag stecken und an STAR-DIVISION GmbH schicken.

Impressum

Herausgeber:
Christian Widuch

Chefredakteur:
Stefan Ritter (SR)

Redaktion:
Michael Ebbrecht (ME)

Satz:
Claudia Küllmer, Silvia Führer, Martina Siebert

Gestaltung:
Renate Köberich, Gerd Köberich,
Petra Biehl, Helmut Skoupy

Reprografie:
Helmut Skoupy

Grafik/Illustration:
Heinrich Stiller

Fotografie
Gerd Köberich

Freie Mitarbeiter:
W.Huske, M.Anton, A.Wilke, R.Bourdon, H.-J.Meyer, T.Westermann, S.Stürmer, T.Metzler, G.Ott, F.Jochem, W.Otternberg, A.Freystatzky, T.Schlote, K.-P.Roethke, K.Wiltsch, M.Böhmer, R.Sprunkel, B.Breede, P.Bosch, E.O.Dumke, U.Schmidt, H.Hörster, R.Herrmann.

Anzeigenleitung:
Wolfgang Schnell
Es gilt die Anzeigenpreisliste Nr. 2a vom 15.10.86

Anschrift Verlag/Redaktion:
DMV Daten & Medien Verlagsgesellschaft mbH
Postfach 250
3440 Eschwege
Telefon: (0 56 51) 87 02

Vertrieb:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck:
Druckerei Jungfer, 3420 Herzberg

Bezugspreise:
Schneider PC International
Joyce Sonderheft 1/87
erscheint einmal jährlich.

Einzelpreis DM 20,–/sfr. 20,–/ÖS 170,–

Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernimmt der Verlag keine Haftung.

Honorare nach Vereinbarung (die Zustimmung zum Abdruck wird vorausgesetzt).

Das Urheberrecht für veröffentlichte Manuskripte liegt ausschließlich beim Verlag. Nachdruck sowie Vervielfältigung oder sonstige Verwertung von Texten, nur mit schriftlicher Genehmigung des Verlages.

Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.



Liebe Leser!

endlich ist es nun soweit – das erste JOYCE-Sonderheft liegt vor Ihnen. Bei der Planung dieses Hefts konnten die Erfahrungen, die wir mit den CPC-Sonderheften gesammelt haben, nur eingeschränkt verwendet werden, da der Kreis der JOYCE-User hauptsächlich aus Anwendern besteht. Diese Tatsache schlägt sich in der Auswahl der Beiträge nieder; ich bin sicher, daß Sie den Mangel an Spieleprogrammen nicht beklagen werden. Eine zweite Schwierigkeit bei der Zusammenstellung bestand in der Wahl des Informations- »Niveaus« der Beiträge. Eingefleischte »Betriebssystemler« werden nicht viel Interesse an BASIC- und LocoScript-Tips aufbringen; die Nur-Anwender unter Ihnen sind möglicherweise über den Anteil »trockener« Materie im Heft verärgert. Ich bin jedoch sicher, daß das Heft in der vorliegenden Form für jeden Bereich etwas zu bieten hat. Zahlreiche Tips zur Programmierung in BASIC und zur Bedienung von LocoScript haben einen gleichberechtigten Platz neben einem Exkurs in mathematische Gefilde und einem Aufsatz, der sich mit dem Aufbau des erweiterten BIOS von CP/M PLUS befaßt. Ich möchte es hier nicht versäumen, mich für die aktive Mitarbeit der freien Autoren zu bedanken, die durch Verbesserungsvorschläge und Tips zum Gelingen dieser Ausgabe beigetragen haben. Auch Sie, liebe Leser, haben durch Ihre Nachfrage nach einem Sonderheft einen Teil der Arbeit beigetragen. Bevor ich Sie nun in das Heftinnere entlasse, möchte ich noch einen Aufruf verlauten lassen: Nur durch Ihre Mitarbeit ist es möglich, die Qualität (und Quantität!) der JOYCE-Beiträge zu erhalten. Bitte teilen Sie mir Ihre Anregungen und Ihre Kritik zur Joyce-Rubrik in der PC International und zu diesem Sonderheft mit. Ebenso sind Textbeiträge, Anwendungs- und auch Spieleprogramme jederzeit willkommen. In der Erwartung, Sie bald im zweiten Joyce-Sonderheft begrüßen zu können, wünsche ich Ihnen viel Spaß bei der Lektüre des Ersten.

Herzlichst

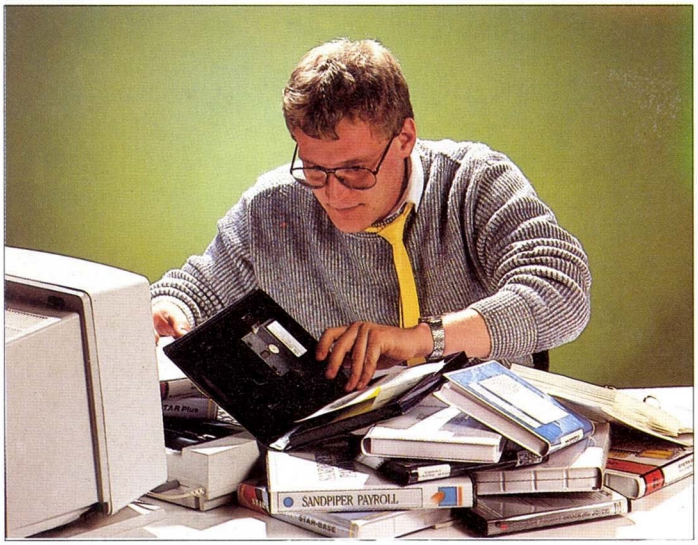


Michael Ebbrecht
Redakteur Joyce



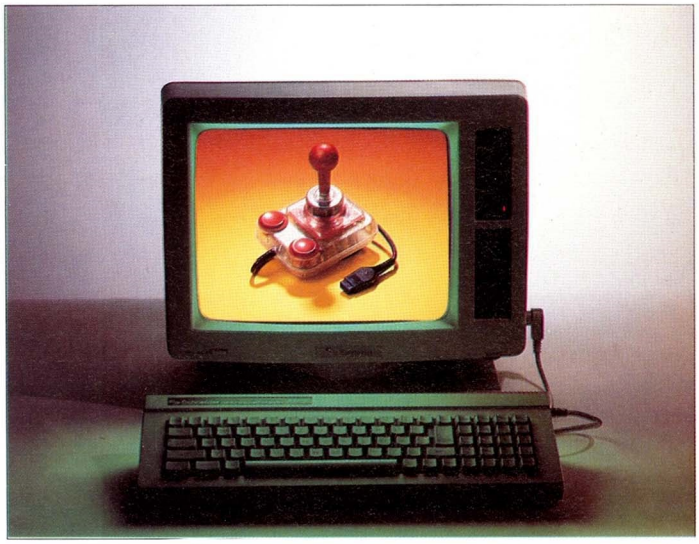
Was Hammer und Säge für den Handwerker, ist der Maskengenerator für den Programmierer.

S. 8



Die Marktübersicht auf Seite 22 soll Ihnen die Auswahl der für Joyce verfügbaren Soft- und Hardware erleichtern.

S. 22



Der Joystick am Joyce – mit wenigen Handgriffen selbst installiert. Ausführliche Bauanleitung auf

S. 74

Berichte:

Überblick über Hard- und Software zum Joyce soll die ***Marktübersicht*** verschaffen..... **S. 22**

Ein ***Interview*** mit Herrn Laaks aus dem Hause Schneider gibt Informationen zur gegenwärtigen und zukünftigen Situation der PCWs..... **S. 72**

Tips zum Softwarekauf – was Sie vor dem Erwerb von Anwendersoftware beachten sollten ..... **S. 120**

Hardware:

Hardcopy ohne Ausstieg: eine kleine Umbauanleitung für das Joyce-Netzteil beseitigt den gelegentlichen Ausstieg des Rechners beim Bildschirmausdruck..... **S. 66**

RAM-Erweiterung – detaillierte Hinweise erleichtern Ihnen den Selbsteinbau der Käfer..... **S. 70**

JOYceSTICK: mit Minimalaufwand passen Sie beliebige Joysticks selbst an Joyce an!..... **S. 74**

Programme:

Der ***CHECKSUMMER*** aus der PC International ermöglicht Ihnen das fehlerfreie Abtippen der abgedruckten Programme..... **S. 6**

»***Mit Hammer und Säge***« stellt Ihnen einen Maskengenerator der Spitzenklasse vor- Bildschirmmasken erstellen wird zum Kinderspiel!..... **S. 8**

Arcus Sinus bietet Ihnen den ersten Teil eines mathematischen Exkurses und die Implementation der SIN/COS-Funktionen im Bogenmaß..... **S. 36**

DIFFINT stellt als Aufsatz und Programm zur Differential/Integralrechnung den zweiten Teil des mathematischen Ausflugs dar..... **S. 26**

CHI-QUADRAT-Test ist ein Programm zur Überprüfung der Richtigkeit wissenschaftlicher Arbeitshypothesen – auch für Laien interessant!..... **S. 40**

Assembler/Disassembler stellen für den Z-80-Maschinencode-Programmierer leistungsfähige Werkzeuge dar..... **S. 48**

Mau-Mau ist die Joyce-Version des bekannten Kartenspiels..... **S. 61**

CP/M:

Im Herzen von CP/M – das erweiterte BIOS des JOYCE wird hier »zerpflückt«; die Ergebnisse der Analyse stehen als Utilities zur Verfügung..... **S. 76**

Tips:

Tips zur ***Basic-Programmierung unter LocoScript*** und zur ***Belegung der Sondertasten*** mit Sätzen erhalten Sie auf..... **S. 98**

Tips zum Softwarekauf
Eine detaillierte Anleitung erleichtert Ihnen das ***Erstellen einer Startdiskette*** und einer ***Tastenumbelegung***.... **S. 102**

Zum Thema ***LOGO*** stellen wir einige nützliche Prozeduren vor..... **S. 104**

Fußnoten unter LocoScript sind möglich – wie's geht, erfahren Sie auf..... **S. 109**

Wie ***dBaseII auf dem PCW 8512*** sinnvoll installiert werden kann, erläutert der Beitrag auf..... **S. 112**

Einige ***Tips zu DR Graph*** machen die Arbeit komfortabler – und schneller!..... **S. 114**

Einfache ***Schriftartanwahl*** des Joyce-Druckers ermöglicht Ihnen ein kleines Basic-Programm..... **S. 116**

Ordnung in Ihre Diskettensammlung bringt ein komfortables ***Diskverwaltungsprogramm***..... **S. 117**

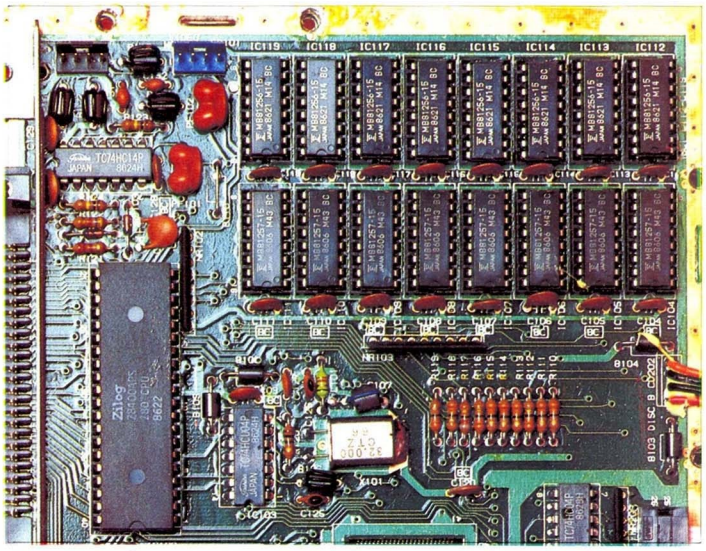
Ein ***Umwandlungsprogramm*** zur Erstellung von ASCIIDateien aus LocoScripttexten ermöglicht die Weiterverwendung Ihrer Texte in anderen Textverarbeitungen..... **S. 101**

Rubriken:

Editorial..... **S. 3**

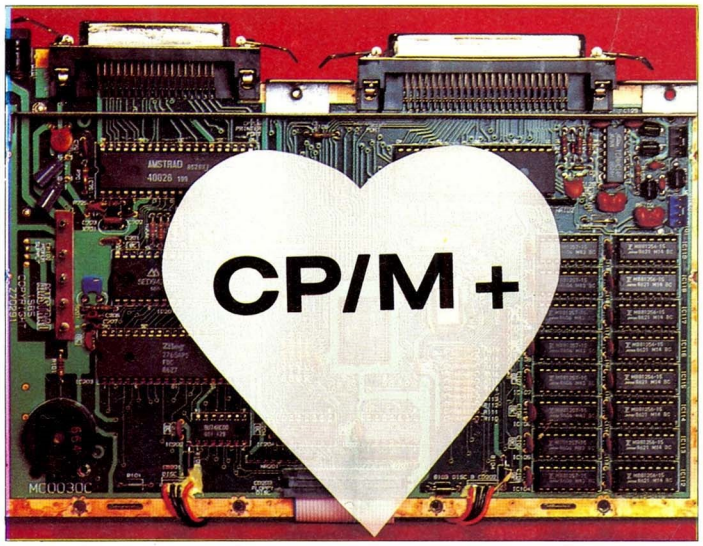
Impressum..... **S. 3**

Inserentenverzeichnis..... **S. 115**



Wie Sie die Ram-Disk des PCW 8256 selbst auf satte 512 kB erweitern, ist im Artikel auf Seite 70 ausführlich beschrieben.

S. 70



Mit dem Herzen von CP/M – dem extended BIOS – befaßt sich die ausführliche Exkursion auf Seite 76

S. 76



Ordnung in Ihrer Diskettensammlung – kein Problem mit unserer Diskverwaltung. Beschreibung und Programm auf Seite 117

S. 117

Prüfsummenprogramm für Joyce

Listings für die CPC-Modelle werden in dieser Zeitschrift seit einiger Zeit mit Prüfsummen abgedruckt. Für Joyce-Programme fehlte dieser Service bislang. Hier ist nun ein universelles Prüfziffernprogramm für den Joyce. Es erzeugt Prüfziffern für viele Programmiersprachen (Assembler, BASIC, Cobol usw.).

Die Prüfziffern (PZ) werden durch ein spezielles Verfahren ermittelt. Daher werden im Gegensatz zum Quersummenverfahren auch sog. »Zahlendreher« erkannt; Die Zeile '100 GOTO 1230' ergibt daher eine andere PZ als '100 GOTO 1320', obwohl die Quersummen gleich sind. Außerdem fallen nur zweistellige PZ an, um die Prüfung zu vereinfachen.

Texte nach dem vereinbarten REM-Zeichen (bzw. -Wort) werden nicht geprüft, können also weggelassen werden. Zeilennummern und Leerstellen im Programmtext fließen jedoch in die PZ ein. Um das Abtippen zu erleichtern, werden aufeinanderfolgende Leerstellen durch »...« (Punkte) ersetzt. Ausgenommen sind einzelne Leerstellen (bzw. die letzte Leerstelle einer Folge), um die Listings gut lesbar zu halten.

Das zu prüfende Programm wird ohne Prüfziffern, »...« (Punkte) als Leerstellen und mit Leerstelle hinter den REM-Zeichen erfaßt. Nach der Sicherung auf Disk A: wird es zusätzlich auf der Memory-Disk M: im ASCII-Format abgelegt. Der Programmname ist beliebig, die Erweiterung muß ».ASC« sein, z.B. SAVE "M:TEST.ASC",A .

```
Variablenliste der Datei 'm:check.bas'
   200        1060
   270         290
   380         410
   440         470
   500         530
   650         610
   690         650   710  1050
   730         740
   760         730
   860         880
   900         840
   930         950
   970         900
  1000        1020
  1040         970
  1060         690  1040
ausgabe$       160   380   390   420
bell$           60    80  1120
clen           770   770   790
datei$         160   250   270   280   310   560   570   590   630   670

datnam$        560   580
druck$         820   860   870   870   880   910   930   940   940   950
               980  1000  1010  1010  1020
erste          520   530   530   710
esc$            60    70    70    80    80   180   180   620   620
fehl$           80   290   410   470   530
fehler$       1090  1100  1110  1120
in$            270   280   280   280   280   290   290   300   310   330
               340   340   350   350   360   380   390   390   390   390
               400   420   440   450   450   460   480   500   510   510
               520   540
invoff$         70   230  1120
invon$          70   230  1120
prnnam$        570   660
rmk$           160   330   340   360   590   630   670   760   770
start$         160   500   510   540
suchdat$       240   250   250
summe          780   800   800   850   920   990
teil$          860   860   930   930  1000  1000
wahl           400   410   610   650   840   900   970
x              730   730   740   760   770   770   790   800   800
zeile$         700   710   730   740   760   770   800   820
zlen           460   470   470   860   870   930   940  1000  1010
zlen$          160   440   450   480
```

Nach RUN"CHECK.BAS" beantworten Sie die Fragen nach Dateinamen, REM-Zeichen, Ausgabe, Zeilenlänge und Startzeile. Falls die in den spitzen Klammern < > vorgegebenen Werte beibehalten werden sollen, brauchen Sie nur mit der RETURN-Taste zu bestätigen.

Die Ausgabe kann auf dem Monitor <M>, dem Printer <P> oder in eine sequentielle Datei <D> erfolgen. Mit <f5> kann die Ausgabe angehalten bzw. fortgesetzt und mit <CAN> abgebrochen werden.

Die bei der Ausgabe auf Disk erzeugte Datei (.PRN) kann ggf. mit RPED oder ED weiterverarbeitet werden. Auch mit LocoScript kann die Datei mit der Option »TEXT EINFÜGEN« gelesen werden, um z.B. Programmbeispiele mit PZ in Texte einzubinden.

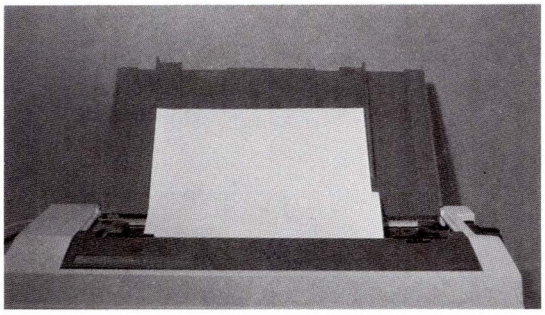
Das Programm wird durch Eingabe von »end« als Dateiname beendet.

(H.-J. Meyer)

```
LISTING >CHECK    <, REMARK = >REM<.
<89> 10 REM check.bas 04.11.86 (C) H.J.Meyer Harsum
<78> 20 :
<55> 30 ON ERROR GOTO 1080
<80> 40 :
<85> 50 DEFINT c,s,w,x:DEFSNG e,z:WIDTH 90
<58> 60 bell$=CHR$(7):esc$=CHR$(27)
<69> 70 invon$=esc$+"p":invoff$=esc$+"q":REM invers an/
     aus
<23> 80 fehl$=bell$+esc$+"A"+esc$+"K"+CHR$(13):REM Zeil
     e löschen
<85> 90 :
<97> 100 REM Voreinstellung
<83> 110 DATA END:.... REM Dateiname (bzw. end=Programm
      beenden)
<26> 120 DATA REM:.... REM Remark is REM ( in Assembler
     = ; in Cobol = * usw.)
<22> 130 DATA M:..... REM Ausgabe (M=Monitor/P=Printer
     /D=Disk)
< 6> 140 DATA 40:..... REM Zeilenlänge (10-80)
<33> 150 DATA 0:..... REM Startzeile (0-65534)
<45> 160 READ datei$,rmk$,ausgabe$,zlen$,start$
<28> 170 :
<39> 180 PRINT esc$"H"esc$"E";
<32> 190 :
<39> 200 PRINT:PRINT CHR$(150)STRING$(35,154)CHR$(156)
<10> 210 PRINT CHR$(149)" J O Y C E - C H E C K S U M M
      E R "CHR$(149)
<83> 220 PRINT CHR$(147)STRING$(35,154)CHR$(153)
<22> 230 PRINT invon$:FILES "m:*.ASC":PRINT:PRINT invof
     f$
<58> 240 suchdat$=FIND$ ("M:*.asc")
<94> 250 IF suchdat$<>""THEN datei$=LEFT$(suchdat$,8)
<27> 260 :
<85> 270 PRINT"Dateiname","<"datei$">",:INPUT in$
<37> 280 in$=UPPER$(in$):IF in$="" THEN in$=datei$
<25> 290 IF VAL(in$)<>0 OR LEN(in$)>8 THEN PRINT fehl$;
     :GOTO 270
<35> 300 IF in$="END" THEN PRINT:END
<76> 310 datei$=in$
<20> 320 :
<78> 330 PRINT"REM-Zeichen","<"rmk$">",:INPUT in$
<84> 340 IF in$="" THEN in$=rmk$
<23> 350 IF in$="NO" THEN in$=CHR$(255)
<92> 360 rmk$=in$
<30> 370 :
<36> 380 PRINT"Ausgabe","<"ausgabe$">",:INPUT in$
<56> 390 in$=UPPER$(in$):IF in$="" THEN in$=ausgabe$
< 3> 400 wahl=INSTR("MPD",in$)
<22> 410 IF wahl=0 THEN PRINT fehl$;:GOTO 380
<37> 420 ausgabe$=in$
<23> 430 :
```

```
<89> 440 PRINT"Zeilenlänge","<"zlen$">",:INPUT in$
<35> 450 IF in$="" THEN in$=zlen$
<39> 460 zlen=VAL(in$)
<40> 470 IF zlen<10 OR zlen>80 THEN PRINT fehl$;:GOTO 4
     40
<30> 480 zlen$=in$
<35> 490 :
<19> 500 PRINT"Startzeile","<"start$">",:INPUT in$
<81> 510 IF in$="" THEN in$=start$
<54> 520 erste=VAL(in$)
<50> 530 IF erste<0 OR erste>65534! THEN PRINT fehl$;:G
     OTO 500
<46> 540 start$=in$
<28> 550 :
<73> 560 datnam$="m:"+datei$+".ASC"
<30> 570 prnnam$="a:"+datei$+".PRN"
<35> 580 PRINT:OPEN"I",1,datnam$
<74> 590 PRINT"LISTING >"datei$"<, REMARK = >"rmk$"<.":
     PRINT
<19> 600 :
< 5> 610 IF wahl<>2 THEN 650
<50> 620 LPRINT esc$"l"CHR$(10);esc$"M";
<18> 630 LPRINT "LISTING >"datei$"<, REMARK = >"rmk$"<.
     ":LPRINT
<27> 640 :
<16> 650 IF wahl<>3 THEN 690
<92> 660 OPEN"O",2,prnnam$
<30> 670 PRINT #2,"LISTING >";datei$;"<, REMARK = >";rm
     k$;"<.":PRINT #2," "
<35> 680 :
<45> 690 IF EOF(1) THEN GOTO 1060
<85> 700.. LINE INPUT #1,zeile$
< 0> 710.. IF VAL(zeile$)<erste THEN 690
<24> 720.. :
<16> 730.. x=INSTR(zeile$,SPACE$(2)):IF x=0 THEN 760
<33> 740.. MID$(zeile$,x,1)=".":GOTO 730
<30> 750.. :
<13> 760.. x=INSTR(zeile$,rmk$+" ")
<44> 770.. IF x=0 THEN clen=LEN(zeile$):ELSE clen=x+LEN
     (rmk$)-1
<68> 780.. summe=0
<44> 790.. FOR x=1 TO clen
<85> 800.... summe=(summe+(ASC(MID$(zeile$,x,1))AND 127
     )*x)MOD 99
<70> 810.. NEXT
<67> 820.. druck$=zeile$
<27> 830.. :
<48> 840.. IF wahl<>1 THEN 900
<47> 850.. PRINT USING"<##> ";summe;
<50> 860.. teil$=LEFT$(druck$,zlen):PRINT teil$
<47> 870.. druck$=MID$(druck$,zlen+1)
<61> 880.. IF LEN(druck$)>0 THEN PRINT SPACE$(5);:GOTO
     860
<39> 890.. :
<27> 900.. IF wahl<>2 THEN 970
<69> 910.. PRINT USING"< ##### >";VAL(druck$);
<77> 920.. LPRINT USING"<##> ";summe;
< 6> 930.. teil$=LEFT$(druck$,zlen):LPRINT teil$
<42> 940.. druck$=MID$(druck$,zlen+1)
<73> 950.. IF LEN(druck$)>0 THEN LPRINT SPACE$(5);:GOTO
      930
<34> 960.. :
<63> 970.. IF wahl<>3 THEN 1040
<83> 980.. PRINT USING"< ##### >";VAL(druck$);
<83> 990.. PRINT #2,USING"<##> ";summe;
<67> 1000.. teil$=LEFT$(druck$,zlen):PRINT #2,teil$
<88> 1010.. druck$=MID$(druck$,zlen+1)
<18> 1020.. IF LEN(druck$)>0 THEN PRINT #2,SPACE$(5);:G
     OTO 1000
< 1> 1030.. :
<28> 1040.. IF INKEY$=CHR$(8) THEN GOTO 1060
<92> 1050 GOTO 690
<85> 1060 CLOSE:PRINT:GOTO 200
<20> 1070 :
<78> 1080 REM Fehlerbehandlung
<49> 1090 fehler$=" Fehler Nr."+STR$(ERR)+" in Zeile Nr
     ."+STR$(ERL)+". "
<97> 1100 IF ERR=53 THEN fehler$=" Datei nicht vorhande
     n. "
<20> 1110 IF ERL=230 THEN fehler$=" Keine ASCII-Datei v
     orhanden. "
<71> 1120 CLOSE:PRINT invon$;fehler$;invoff$;bell$:PRIN
     T
<64> 1130 IF ERL=230 THEN RESUME 270 ELSE RESUME 200
```

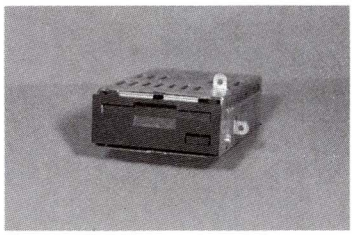

ZUBEHÖR für alle Schneider-Computer



Papierführung für JOYCE-Drucker DM 39,–

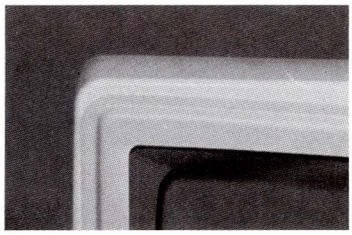
- Mit zwei verstellbaren Seitenführungen
- Gerader Papiereinzug
- Ersetzt die vorhandene Klappe
- Lieferbar in anthrazit und auch in blau, gelb, grün, orange und rot

DAS MUSS FÜR JEDEN PROFESSIONELLEN JOYCE-USER!



FD 2 (2. Laufwerk) 1 MB DM 598,–

RAM-Erweiterung von 256 KB auf 512 KB (Original 257er Bausteine!) DM 129,–



Bildschirmfilter DM 59,–

Alle Händleranfragen sind willkommen!

Weiteres Zubehör:

Für CPC 464, 664, 6128:

Bildschirmfilter Farbmonitor CTM 640/644	DM 44,–
Grünmonitor GT 64/65	DM 39,–

Für den neuen PC:

Hard-Disk-Kit 20 MB/30 MB	DM 1398,–/1698,–
Hard-Disk-Card 20 MB/30 MB	DM 1398,–/1698,–
RAM-Erweiterung 128 KB	DM 99,–
Bildschirmfilter s/w und color	DM 59,–

Das komplette Zubehör ist bei den Schneider-Computer-Händlern erhältlich!

Deutschland:
ABD Computer · Zettachring 12 · 7000 Stuttgart 80
Telefon 07 11-7 15 00 37

Österreich:
Wagner Electronics · Hauptstraße 171 · 3001 Mauerbach
Telefon 0222-97 21 66

Hammer und Säge ...

... sind für den Tischler so wichtig, wie ein Maskengenerator für den Programmierer.
»Das mit dem Hammer ist mir klar«, werden Sie sagen, »aber wozu einen Maskengenerator?«.
Wenn sie schon einmal versucht haben mit Hilfe endloser PRINT Zeilen einen, das Auge erfreuenden Bildaufbau zu erzeugen, wissen Sie sicherlich wie mühselig und langweilig so etwas werden kann. Noch schlimmer wird's wenn auch noch diverse INPUTs vom Programm verlangt werden, und der Benutzer vor seinen (Eingabe) Fehlern geschützt werden soll.
Wie schön wäre es doch, könnte man sich ganz auf die zu bewältigende Programmieraufgabe konzentrieren und dem Programm die Ein- und Ausgabe einfach nach Belieben überstülpen.
Mit dem Maskengenerator und den dazugehörigen Maskenroutinen, befreien wir uns also von lästiger Routinearbeit, zaubern hübsche Startbilder und entledigen uns der häßlichen Fehlermeldung »Improper argument«.
Dadurch, daß mit Hilfe des Maskengenerators Eingabefelder definiert werden können, bekommen wir auch einen Zugang zur Struktur unserer Daten. Ein Maskengenerator ist deshalb Grundlage vieler Datenbankprogramme von PROMPT bis CONCEPT 16. (Letzteres gibs leider, leider nicht für den Joyce).

Zum Maskengenerator ...

Nachdem die Aufgaben eines Maskengenerators abgesteckt sind, können wir nun untersuchen, wie diese Aufgaben programmtechnisch zu lösen sind. Bild 1 zeigt, daß der Maskengenerator aus zwei Hauptteilen besteht.
Das ist zum einen der Maskeneditor, der fast wie ein Texteditor zu bedienen ist, und mit dem sowohl die Form der Maske als auch deren Funktion festgelegt wird. In unserem Beispiel ist dies eine Eingabemaske mit 2 Feldern. Wenn dieser Arbeitsgang abgeschlossen ist, tritt der Generator in Aktion, der eine Liste erstellt, in der z.B. die Position und die Länge eines Eingabefeldes festgehalten werden. Diese Liste heißt Parameter-Datei.
Zu guter Letzt werden die Maske (als Text-Datei) und die Parameter-Datei auf Diskette gespeichert, wo sie den Maskenroutinen und somit dem Anwenderprogramm zur Verfügung stehen.

... gehören die Maskenroutinen

Bild 2 zeigt die einzelnen Teile der Maskenroutinen, die in das Anwenderprogramm eingebunden werden.
Maske und Parameter-Datei müssen von der Diskette gelesen und die Maske muß auf den Bildschirm gebracht werden.



Wenn es Eingabefelder in der Maske gibt, können diese mit dem Zeileneditor bearbeitet werden. Die Eingaben können mit Hilfe eines Typprüfers auf Eingabefehler hin überprüft werden, bevor sie an das Programm weitergegeben werden.

Im Allgemeinen

»Maskengeneratoren gibts wie Sand am Meer«; werden Sie sich jetzt denken und mit Recht fragen, warum Sie gerade mit dem hier Vorgestellen arbeiten sollen. Damit Ihnen die nötige Tipparbeit sinnvoll erscheint, wollen wir Ihnen gleich zu Beginn die Vorteile dieses Maskengenerators zukommen lassen.

Und im Besonderen

- Die Sonderzeichen des Joyce können voll verwendet werden, was dazu beiträgt, daß Ihre Programme ein recht professionelles Aussehen erhalten.
- Sie unterliegen keiner Beschränkung bei der Wahl der Feldtypen, da Sie diese im Anwenderprogramm selbst definieren.

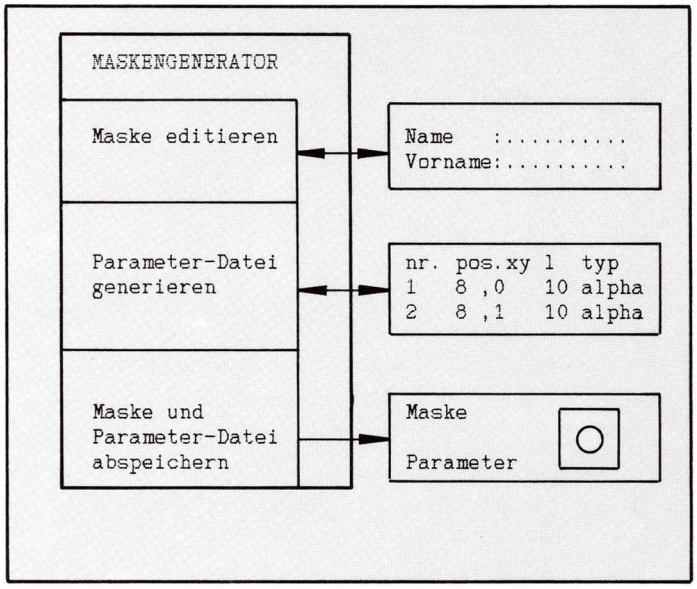


Bild 1: Aufbau eines Maskengenerators

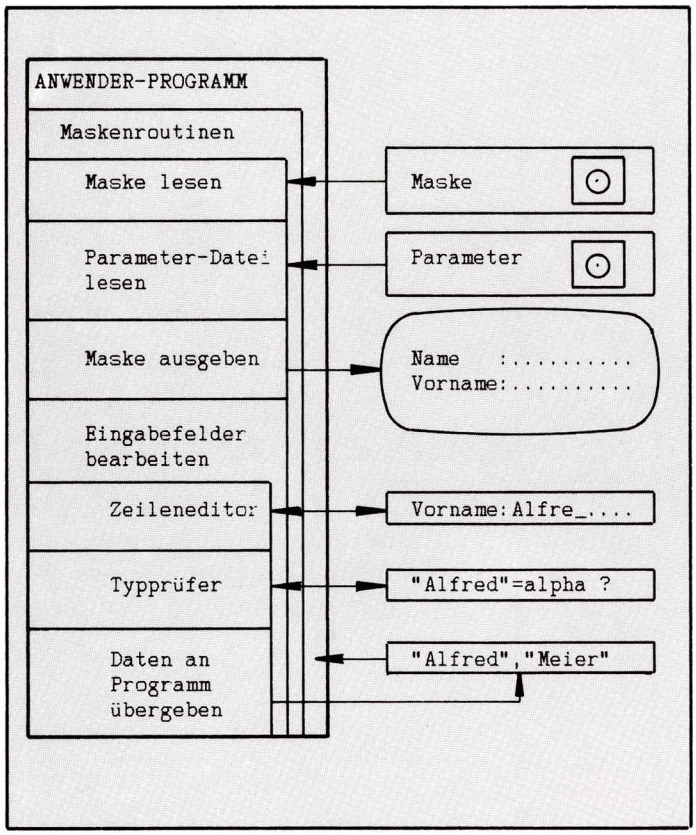


Bild 2: Maskenroutinen

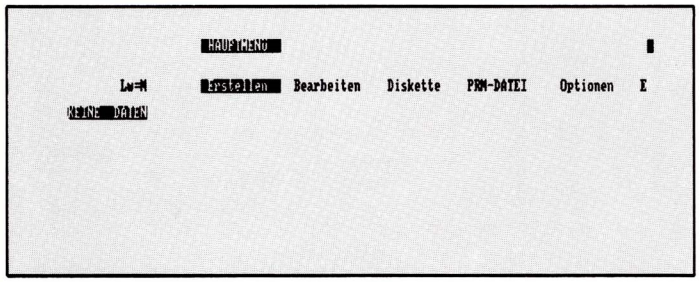


Bild 3: Hauptmenue ohne Menuemaske

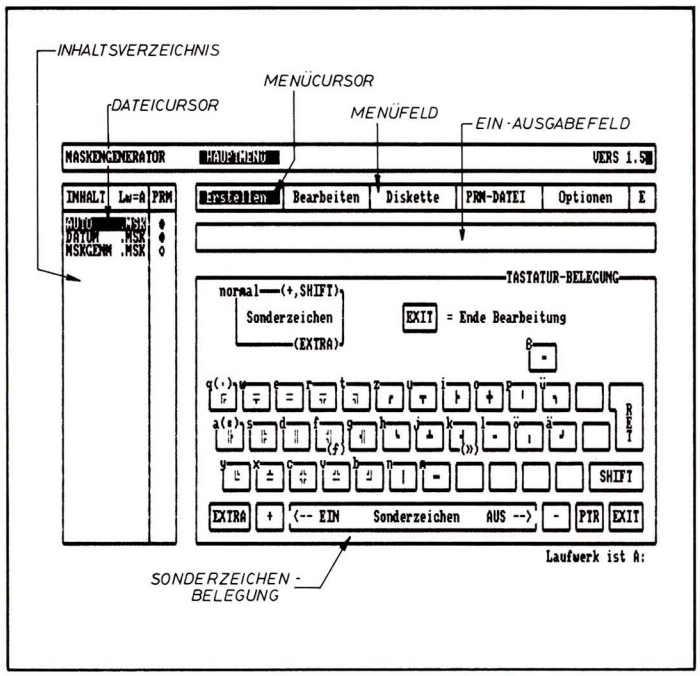


Bild 4: Hauptmenue mit Menuemaske und Sonderzeichentastatur

- Ebenfalls grenzenlos ist die Anzahl der Eingabefelder und deren Verteilung auf dem Bildschirm.
- Die Maske kann nach dem Generieren der Parameter-Datei weiter bearbeitet werden. So läßt sich z.B. aus 8 Pünktchen (.....) für das Datum, der besser geeignete Hinweis JJ.MM.TT einfügen.
- Für Programme, die mit JETSAM Dateien arbeiten sollen, lassen sich schon in der Maske die Werte für Schlüssel-Reihe und Schlüssel-Länge bestimmen.

Haben Sie jetzt bei mindestens einem Punkt gerufen: »Das wollte ich doch schon immer!«, sollten Sie das Programm eingeben, um sich im nächsten Absatz der Bedienung des Programms zu widmen.

Users only

Nach vollbrachter Eingabe und Start mit RUN (vorher das Abspeichern nicht vergessen) meldet sich das Programm wie in Bild 3 gezeigt. Wem das zu mager erscheint, der schaue auf Bild 4, das die Menue-Maske des Programms zeigt. Das Erstellen dieser Menue-Maske ist die beste Übung um sich mit dem Bildschirm-Editor vertraut zu machen. Die Maske muß unter dem Namen <MSKGENM> abgelegt werden, um vom Programm beim Start automatisch erkannt zu werden.

Es war schon immer so, daß die Götter vor den Erfolg den Schweiß gesetzt haben. Oder den Erwerb der Sammeldiskette, da sich die Menue-Maske natürlich mit auf der Diskette befindet.

Auf Bild 4 sieht man auch die Bezeichnungen Menue-Feld, Ein- Ausgabe-Feld und Inhaltsverzeichniss. Der Menue-Cursor (Inverses Feld) läßt sich mit den Cursor-Tasten (links,rechts) bewegen. Alle, auf diese Art gewählten Funktionen, werden mit der ENTER oder RETURN Taste ausgelöst. Die nötigen Eingaben fragt das Programm im Dialog ab, wobei Fehlbedienungen soweit wie möglich durch Sicherheitsabfragen vermieden werden. Die möglichen Funktionen zeigt Bild 5. Maskennamen werden ohne Dateityp angegeben. Das Programm ergänzt die Dateitypen automatisch.

- **MSK für Masken-Dateien**
- **PRM für die dazugehörigen Parameter-Dateien**

Im Inhalsverzeichnis können Sie mit dem Datei-Cursor (Cursor-Tasten hoch,runter) spazieren gehen. Das ist auch nötig, denn alle gewählten Funktionen beziehen sich in der Regel auf die, durch den Datei-Cursor markierte Maske. Die Spalte PRM im Inhaltsverzeichnis zeigt an, ob zu der entsprechenden Maske schon eine Parameter-Datei existiert.

- **○ = keine PRM-Datei**
- **# = PRM-Datei vorhanden**

Befinden sich auf dem angewählten Laufwerk keine Masken-Dateien, so erscheint die Meldung »keine Daten«.

Der Editor gibt sich die Ehre

Durch wählen der Funktionen Erstellen oder Bearbeiten gelangen Sie in den Maskeneditor. Der Editor stellt Ihnen eine volle Bildschirmseite (31 Zeilen) zur Verfügung, auf der Sie nach Herzenslust mit den Cursor-Tasten umherwandern können. Da Sie mit dem Editor nicht einfügen, sondern nur überschreiben können, haben die DELTasten die gleiche Wirkung wie die Cursor-Tasten links, rechts.
Die Sonderzeichen, die der Joyce darstellen kann, erreichen Sie indem Sie die »+« Taste betätigen. Die Belegung der Sonderzeichen-Tastatur zeigt Bild 4. Mit der »−« Taste schalten Sie die Sonderzeichen wieder aus.
Text und Sonderzeichen lassen sich beliebig auf dem Bildschirm verteilen. Nur Stellen, an denen später Eingaben erfolgen sollen (Eingabefelder) verlangen die Einhaltung gewisser Regeln.

Regelwerk

Um dem Generator mitzuteilen, daß Sie an beliebiger Stelle ein Eingabefeld haben möchten, müssen Sie dort das Zeichen "»" (EXTRA + »k«) gefolgt von einigen Parametern setzen.
Die Angabe »20,2 bedeutet z.B., daß ein Eingabefeld mit der Länge 20 erzeugt werden soll. Der zweite Parameter (2) bezeichnet den Feldtypen.(Zu den Feldtypen später mehr. Wir gehen zunächst einmal davon aus, daß der Typ 2 die Eingabe einer Realzahl erlaubt.)
Beim Arbeiten mit JETSAM-Dateien können Sie zusätzlich die Parameter für Schlüssel-Länge und Schlüssel-Reihe angeben. Um dem Generator dies mitzuteilen, muß irgendwo in der Maske das Zeichen (EXTRA + f) erscheinen.
Bei Eingabe von "»" ohne weitere Parameter, wird nur die entsprechende Cursor-Position in die Parameter-Datei übernommen. Das hilft beim Festlegen von Bildschirmpositionen, an denen Ausgaben erfolgen sollen.
Entspricht die Maske ihren Wünschen, verlassen Sie den Editor mit der EXIT-Taste. Ihre Maske wird dann auf dem aktuellen Laufwerk gespeichert.
Eine Zusammenfassung der Editorfunktionen zeigt Bild 6.

Steuerwerk

Um die Arbeit des Generators und der Maskenroutinen zu verdeutlichen, benötigen wir jetzt ein Beispiel. Nehmen wir einmal an, wir wollten ein Programm zur Berechnung des Durchschnittverbrauchs unseres Automobils schreiben. Das ist ein zugegebenermaßen recht simples Beispiel, an dem sich aber alle Vorgänge gut beschreiben lassen.
Unsere Beispielmaske zeigt Bild 7. Die Maske wird eingegeben und der Editor mit der EXIT-Taste verlassen. Jetzt wird der Generator durch die Funktion PRMDatei aufgerufen. Dabei kann man verfolgen, wie der Generator die

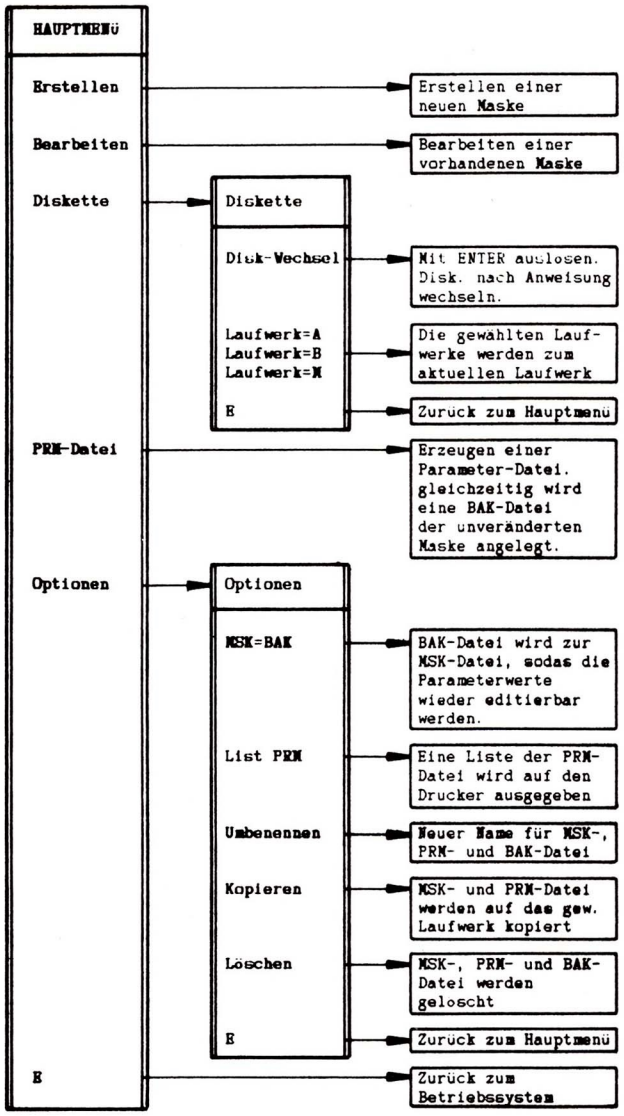


Bild 5: Die Funktionen des Maskengenerators

```
- "+" Taste = Sonderzeichen EIN
- "-" Taste = Sonderzeichen AUS
- "»" EXTRA + "k" = Feldposition

- Cursor-Tasten hoch,runter,links,rechts
- DEL > und < DEL wirken wie die entsprechenden
              Cursor-Tasten.

Feldbezeichner in der Maske:

z.B.          »20,1

    Cursorpos.  Länge  Typ

Für JETSAM-Parameter-Dateien

     1 mal an beliebiger Stelle "ƒ" (EXTRA + "f")

z.B.          »20,1,8,0
                                Schl.-Reihe
  Cursorpos.  Länge  Typ  Schl.-Länge

Aus »6,1 wird beim erstellen der PRM-Datei .....
```

Bild 6: Die Funktionen des Maskeneditors.

Eingabefelder erkennt, die Parameter ausradiert und an Stelle derselben Pünktchen setzt, deren Anzahl genau der gewünschten Feldlänge entspricht. Das Ergebnis zeigt Bild 8.

Die erzeugte Parameter-Datei kann man sich mit der Funktion "List PRM" auf dem Drucker ausgeben lassen. (Bild 9). Das Listing der Parameter-Datei gibt auch genau deren Datenstruktur wieder.

- Format der PRM-Datei:
  2 Byte Integer (MKI−)
  (0,0) = 1 für JETSAM-Parameter-Datei
  0 für normale Parameter-Datei
  (0,1) = Anzahl der Felder
  (0,2) = Recordlänge (Summe der Länge aller Felder)
  alle Weiteren:
  (x,0) = Cursorposition X
  (x,1) = Cursorposition Y
  (x,2) = Feld-Länge
  (x,3) = Feld-Typ
  zusätzlich bei JETSAM-Parameter-Datei:
  (x,4) = Schlüssel-Länge
  (x,5) = Schlüssel-Reihe

Zunächst erscheint es so, als ob Ihre Parameter damit ein für alle Mal verschwunden wären. Dem ist aber nicht so. Vom Generator wird eine BAK-Datei (Sicherungs-Datei) der unveränderten Maske angelegt. Mit der Funktion »MSK=BAK« ist es möglich die ursprüngliche Maske wieder zur Bearbeitung aufzurufen.

Damit sind die wesentlichen Bestandteile des Maskengenerators beschrieben, und wir können uns nun den Maskenroutinen zuwenden.

Routiniers

Immer wenn Programmierer zu faul sind, daß Rad tausend Mal zu erfinden, schreiben sie eine Routine, die bei Bedarf mit MERGE eingebunden wird.

Ebenso ist's mit den Maskenroutinen, die alle Funktionen, die zum Arbeiten mit Masken nötig sind, zur Verfügung stellen.

Listing 3 zeigt das komplette Beispielprogramm für die Verbrauchsberechnug, damit Sie die Anwendung der Routinen gleich in der Praxis mitverfolgen können.

Listing 2 zeigt die Maskenroutinen. Nach der Eingabe sind sie mit >save "maskrout< abzuspeichern und können dann mit >merge "maskrout< in Ihr Programm eingebunden werden.

Im Listing der Maskenroutinen sind ab Zeile 10000 die Bildschirmkommandos zu sehen. Sie sind nicht nur für die Maskenroutinen zu gebrauchen, sondern natürlich auch vom Anwenderprogramm aus zu benutzen. So löscht z.B. >PRINT cls$;< den Bildschirm. >PRINT FNat$(10,5);"A";< schreibt in Zeile 5, Spalte 10 den Buchstaben A.

Die verwendeten Variablennamen sind wohl selbsterklärend. Vor der Benutzung müssen die Kommandos durch GOSUB 10000 initialisiert werden.

Die weiteren Funktionen sind

Zeile 12100, Maske lesen.
Die MSK-Datei einer Maske wird gelesen. Vor dem Aufruf mit GOSUB 12100, muß der Variablen mskname$ der Maskenname übergeben werden. Der Name wird ohne Typangabe zugewiesen. (Im Beipiel Zeile 1020 und 1050)

Zeile 12300, Parameter-Datei lesen.
Die Parameter-Datei einer Maske wird gelesen. (Im Beispiel Zeile 1060)

Zeile 12500, Maske ausgeben.
Die Maske wird auf den Bildschirm geschrieben. Dieses Unterprogramm kann innerhalb des Anwenderprogramms auch mehrmals aufgerufen werden. (Im Beispiel Zeile 1070)

Zeile 12600, Maske löschen,
Sollen in einem Anwenderprogramm mehrere Masken verwendet werden, so ist vor dem Laden einer neuen Maske, die alte Maske durch Aufruf dieses Unterprogramms zu löschen. Es werden alle in den Maskenroutinen verwendeten Felder gelöscht, damit diese neu dimensioniert werden können.

Zeile 12700, Ein Eingabefeld bearbeiten
Mit diesem Unterprogramm läßt sich gezielt ein einzelnes

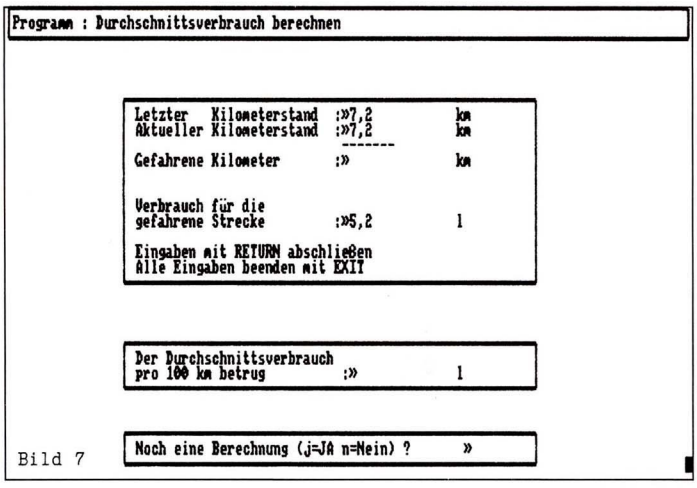


Bild 7

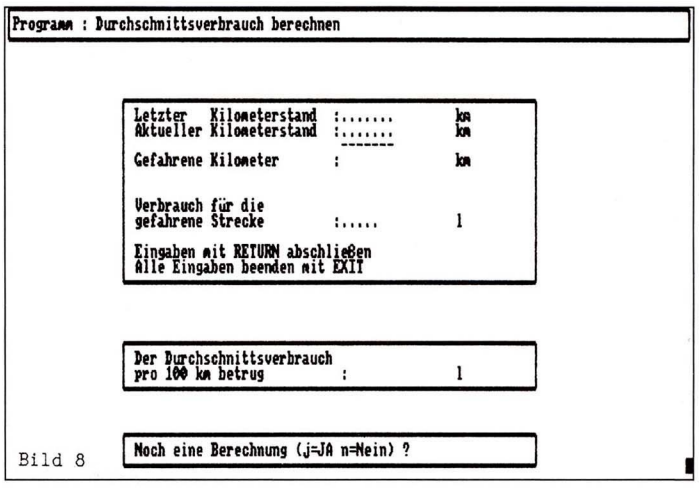


Bild 8

SchneiderData

SD15 der Typenraddrucker für alle Schneider Computer

- ★ hat das richtige Schriftbild für Anspruchsvolle
- ★ schreibt schnell (bis zu 15 Zeichen/sec.) und leise (kleiner 65 dB)
- ★ bedruckt Etiketten, Endlospapier, Briefpapier, auch DIN A4 quer
- ★ ist durchschlagend: Original plus 4 Kopien
- ★ läßt vielfältige Gestaltungsmöglichkeiten zu: drei verschiedene Zeichendichten, Schatten- und Fettdruck, autom. Unterstreichen voll und wortweise, hoch- und tiefstellen, halber Zeilenschritt vorwärts und rückwärts
- ★ druckt bidirektional mit Druckwegoptimierung
- ★ gibt LocoScript-Texte (JOYCE) über das Programm LOCO15 (DM 59,–) aus
- ★ hat serienmäßig parallele und serielle Schnittstelle, Traktor und Selbsttest
- ★ ist der Typenraddrucker von SchneiderData für Schneider Computer

698,–
unverbindliche Preisempfehlung



SchneiderData · Rindermarkt 8 · D-8050 Freising · Telefon 08161/2877

Eingabefeld bearbeiten. Das entsprechende Eingabefeld ist der Variablen feld zu übergeben. Sollte in unserem Beispiel das erste Eingabefeld (Letzter Kilometerstand) bearbeitet werden, so würde der Aufruf lauten >feld=1: GOSUB 12700<. Bei der Zählung der Felder müssen auch die definierten Cursorpositionen mitgezählt werden. So ist das Feld (Verbrauch f. d. gef. Strecke) als Feld 4 anzusprechen. Feld 3 ist das Ausgabefeld (Gef. Kilometer). Bei der Bearbeitung eines Eingabefeldes wird ein Zeileneditor benutzt, dessen Funktionen Bild 10 zeigt. Nach Rückkehr aus dem Unterprogramm befindet sich der Eingegebene String in der Variablen dat$(feld). (Im Beispiel die Zeilen 1100,1110 und 1120).

Soll ein bereits vorhandener Wert geändert werden, so ist dieser vor dem Aufruf des Unterprogramms ebenfalls der Variablen dat−(feld) zuzuweisen. Die Variable dat$( ) ist also unsere Schnittstelle zu den Maskenroutinen.

Zeile 13100, Eingabefelder bearbeiten

Dieses Unterprogramm entspricht weitgehend dem Vorangegangenen, mit dem Unterschied, daß alle Eingabefelder der Maske bearbeitet werden. Von dieser Möglichkeit macht auch unser Beispielprogramm Gebrauch. Mit dem Cursor-Tasten kann beliebig von Feld zu Feld gesprungen werden.

Wird eine Eingabe mit Return abgeschlossen, springt der Cursor zum nächsten Eingabefeld. Erst die Betätigung der EXIT-Taste bricht die Eingabe ab. (Im Beispiel Zeile 1080)

Funktion FNgo$(f)

Die Funktion FNgo$(f) positioniert den Cursor auf einem beliebigen Feld. Das wird im Beispielprogramm zur Ausgabe der errechneten Werte und zum Löschen der Ausgabefelder benutzt. (im Beispiel die Zeilen 1160, 1210, 1220, 1310 und 1320)

Ein Kapitel für sich der Typprüfer

Sie werden sich jetzt fragen, wo denn die versprochene freie Wahl der Feldtypen bleibt. Die Aussicht jeden Eingabefehler abfangen zu können ist schließlich sehr verlockend. Bei der Definition der Maske hatten wir für alle Eingabefelder den Feldtypen 2 angegeben und blindlings dem Hinweis vertraut, daß dieser Typ die Eingabe einer Realzahl erlaubt. Das können Sie jetzt überprüfen, indem Sie versuchen etwas anderes einzugeben als eine Realzahl ohne Exponent.

Der Zeileneditor wird jede Fehleingabe brüsk mit einem warnenden Piepen zurückweisen, und nicht eher Ruhe geben, bis die Eingabe entsprechend korrigiert ist.

In Theorie ...

Dieses gar erstaunliche Verhalten hat seinen Ursprung in der Automatentheorie.

»Aha«, werden Sie denken, »jetzt schnappt der Autor über«. Aber keine Sorge, so schlimm wird's nicht werden.

Die Automatentheorie beschreibt Maschinen, deren Verhalten eindeutig vorhersehbar ist. Solche Maschinen können z.B. Blumen- oder Fahrkartenautomaten sein. Aber auch Microprozessoren lassen sich damit beschreiben.

Jeder Automat hat eine Sprache, was beim Microprozessor leicht zu erkennen ist. Aber auch der Blumenautomat hat eine Sprache, den die Menge der Eingabesignale, die ein Automat kennt, nennt man das Alphabet des Automaten. So sind z.B. »Geld einwerfen« und »Warenschacht öffnen« Eingabesignale, die ein solcher Automat versteht. Jedes gültige Eingabesignal versetzt den Automaten in einen neuen Zustand.

Auch der hier beschriebene Typprüfer ist ein solcher Automat. Er hat die Aufgabe, solange nichts zu tun, wie die Zeichen, die er empfängt, seiner Sprache entsprechen. Erst wenn ein eingebenes Zeichen nicht in seiner Sprache vorkommt, meldet er dies mit dem erwähnten Piepen. Der Vorteil den der Typprüfer gegenüber dem simplen Blumenautomaten hat, ist seine Programmierbarkeit.

Um das Verhalten eines jeglichen Automaten zu beschreiben, benutzt man gerne eine sogenannte Übergangsmatrix. Die Übergangsmatrix zeigt, welches Zeichen (Eingabesignal) als Nachfolger des vorangehenden Zeichens in einer Zeichenkette erlaubt ist.

Bild 11 zeigt eine Übergangsmatrix für Realzahlen ohne Exponent. An dieser Matrix kann man erkennen, daß als erstes Zeichen ein Vorzeichen, eine Ziffer oder ein Dezimalpunkt erlaubt sind.

```
Parameter-Datei für AUTO

Cursorpositionen : 6
Record-Länge     : 19

Nr    cursor X  cursor Y  Feld-Länge   Typ
 1        44        7         7         2
 2        44        8         7         2
 3        44       10         0         0
 4        44       14         5         2
 5        45       24         0         0
 6        60       29         0         0
```

Bild 9

```
- Zeichen werden immer eingefügt, nicht
  überschrieben
- Cursor-Tasten links,rechts
- DEL > und < DEL
- ALT + DEL > = von Cursor bis Ende Zeile löschen
- ALT + < DEL = Cursor an Zeilenanfang setzen
- Wird das letzte Zeichen gemäß Zeilenlänge
  erreicht, warnt ein Piepton.
- Bei Eingabefehlern warnt ein Piepton und der
  Cursor steht an der Stelle, an der der Fehler
  erkannt wurde.
```

Bild 10: Die Funktionen des Zeileneditors.

Wurde als erstes Zeichen der Dezimalpunkt eingegeben, so kann man in der Zeile »alt«, Spalte »Dezimalpunkt« ablesen, daß hiernach nur eine Ziffer folgen kann, während die Eingabe eines Vorzeichens an dieser Stelle auf alle Fälle ein Fehler wäre.
Es läßt sich nun leicht einsehen, daß mit Hilfe einer Übergansmatrix nicht nur Zahlen auf ihre richtige Eingabe hin überprüft werden können, sondern jede beliebige Zeichenfolge, die gewissen Regeln folgt.

... und Praxis

Da bleibt nur noch ein Problem: Wie sag ich's meinem Computer ?
Dazu sehen wir uns das Listing 2 (Maskenroutinen) ab Zeile 15200 an. Dort finden wir die Daten, die der Typprüfer zum Erkennen einer Realzahl benötigt.
In Zeile 15205 wird dem Typprüfer mitgeteilt, daß er 5 verschiedene Zustände erkennen muß. (Als da wären »start«, »+−«, »0−9«, ».« und wieder »0−9«).
Der Zustand »start« wird nur als leerer String dargestellt, muß aber immer vorhanden sein. Und dann folgt eine Liste der erlaubten Zustände, die direkt aus der Übergangsmatrix gebildet wird. Nach dem Zustand "start" dürfen nur die Zustände 2 (Vorzeichen), 3 (0-9) oder 4 (.) folgen. Die Nullen besagen, daß es keine weiteren gültigen Zustände gibt. Sie müssen eingegeben werden, damit der Typprüfer eine vollständige 5*5 Matrix einlesen kann.
Ebenso wird mit den anderen erlaubten Zuständen verfahren. Zunächst werden die gültigen Zeichen in einem String definiert. (Zeilen 15220, 15230, 15240 und 15250) Dann wird eine Liste der DANACH erlaubten Zustände, eingegeben.
Eine 1 in der Zustandsliste bedeutet, daß die Eingabe hier auch beendet werden darf. So erhalten die Zustände 3 und 5 in der Übergangsmatrix, wie auch im Listing, den Nachfolgezustand »Ende erlaubt«.

Typisch

Das ist eigentlich schon alles. Wir müssen nur noch klären, wie der Typprüfer an die richtigen Daten gelangt. In der Zeile 12780 und 13230 findet sich ein >ON typ GOSUB<. Die zweite Position in dieser Zeile (also Typ 2) verzweigt nach 14200.
Dort findet sich zunächst ein >RESTORE 15300<. Damit wird dem Typprüfer mitgeteilt, daß er an der besagten Stelle die Daten für den geforderten Typ findet.
Nun wird der Zeileneditor aufgefordert die Eingabe bitteschön entgegen zu nehmen, (GOSUB 13300) und der Typprüfer gebeten das Eingegebene auf seine Richtigkeit hin zu testen. (GOSUB 13000)
Der Typprüfer übergibt in der Variablen e sein Urteil. Ist e=0, war die Eingabe korrekt und das Programm kehrt zur Eingaberoutine zurück. Falls irgendwo ein Fehler entdeckt wurde muß die ganze Prozedur wiederholt werden.(IF e THEN typanfang)
Diese 4 Programmzeilen sind für jeden definierten Feldtypen verbindlich.
Die einzige Ausnahme bildet der Typ 1. Er nimmt alles entgegen was Eingegeben wird, da der Typprüfer gar nicht erst bemüht wird (Zeile 14100). Typ 0 soll nicht verwendet werden, da dieser den Eingaberoutinen zeigt, daß sich an der betreffenden Stelle nur eine Cursorposition befindet.
Für den Fall, daß Sie einen neuen Typen definieren, dürfen Sie nicht vergessen die Zeilen 12780 und 13230 anzupassen.
Daß auch ziemlich komplizierte Eingaben, wie ein Datum überprüft werden können, zeigt der Typ 4, dessen Übergangsmatrix in Bild 12 gezeigt ist. Damit wird der 30. Februar oder der 31. November unmöglich.

Zustand / alt / neu	1 start	2 +-	3 0-9	4 .	5 0-9
1 start	/	nein	nein	nein	nein
2 +-	ja	nein	nein	nein	nein
3 0-9	ja	ja	ja	nein	nein
4 .	ja	ja	ja	nein	nein
5 0-9	nein	nein	nein	ja	ja
6 ende	nein	nein	ja	nein	ja

Bild 11: Übergangsmatrix für eine Realzahl

Zustand / alt / neu	1 start	2 0-9	3 0-9	4 .	5 0	6 1	7 1,3 5,7 8	8 4,6 9	9 0,2	10 1	11 2	12 .	13 .	14 .	15 0	16 1,2	17 3	18 1-9	19 0-9	20 0,1	21 3	22 0
1 start	/	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
2 0-9	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
3 0-9	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
4 .	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
5 0	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
6 1	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
7 1,3,5 7,8	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
8 4,6,9	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
9 0,2	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
10 1	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
11 2	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
12 .	n	n	n	n	n	ja	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n	n
13 .	n	n	n	n	n	n	ja	n	ja	n	n	n	n	n	n	n	n	n	n	n	n	n
14 .	n	n	n	n	n	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n	n
15 0	n	n	n	n	n	n	n	n	n	n	ja	ja	ja	n	n	n	n	n	n	n	n	n
16 1,2	n	n	n	n	n	n	n	n	n	n	ja	ja	ja	n	n	n	n	n	n	n	n	n
17 3	n	n	n	n	n	n	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n	n
18 1-9	n	n	n	n	n	n	n	n	n	n	n	n	n	ja	n	n	n	n	n	n	n	n
19 0-9	n	n	n	n	n	n	n	n	n	n	n	n	n	n	ja	n	n	n	n	n	n	n
20 0,1	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	ja	n	n	n	n	n	n
21 3	n	n	n	n	n	n	n	n	n	n	n	ja	n	n	n	n	n	n	n	n	n	n
22 0	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	ja	n	n
23 ende	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	ja	ja	ja	n	ja

Bild 12:Übergangsmatrix für die Datumseingabe

Zum Schluß

Wir hoffen, daß Ihnen der Maskengenerator und die Maskenroutinen das (Programmierer-) Leben etwas erleichtern. Sollte noch die eine oder andere Unklarheit geblieben sein, hilft das Studium der Listings sicherlich weiter. Da es nichts gibt, was man nicht noch besser machen könnte, wird PC International Verbesserungen und Kritik gerne aufnehmen und an die Leser weitergeben, soweit diese von allgemeinem Interesse sind.

Weitere Literatur und Quellen:

c't 12.84 und 1.85 — Eine Beschreibung wie man von einem Maskengenerator zum Programmgenerator gelangt

c't 8.85 — »Bernie und die Automaten« ist eine leicht verständliche Einführung in die Automatentheorie.

CHIP M-BASIC — Listing und Beschreibung eines Maskengenerators

(T.Schlote)

```
LISTING >MASKGEN<, REMARK = >REM<.

<83> 1000 '
<74> 1010 'MASKENGENERATOR.. Vers.1.3. Mai 1986
<89> 1020 '
<52> 1030 'T.Schlote, 6415 Petersberg
<95> 1040 '
<98> 1050 '
< 2> 1060 '
<19> 1070 'INITIALISIERUNG
< 8> 1080 '
< 5> 1090 DEFINT a-z:WIDTH 255:RESTORE:OPTION FILES "A"
<49> 1100 DIM v$(30)..... :'EDITOR-
     FELD
<64> 1110 DIM m$(30)..... :'MENüMAS
     KEN-FELD
<71> 1120 DIM sz(62)..... :'SONDERZ
     EICHEN
<70> 1130... FOR i=1 TO 62
<64> 1140..... READ sz(i)
< 0> 1150... NEXT
<17> 1160 DIM inhalt$(24):DIM inhalt(24):imax=0
<81> 1170 im$=".MSK":ip$=".PRM":ib$=".BAK"... :'DATEIAT
     TRIBUTE
<77> 1180 DIM wahl$(2,5)..... :'TEXTE F
     üR MENü
<12> 1190... FOR i=0 TO 2:FOR j=0 TO 5
< 0> 1200..... READ wahl$(i,j)
<64> 1210... NEXT:NEXT
<91> 1220 wahl=0:w=0
<13> 1230 x=0:y=0:n=0..... :'CURSORP
     OSITION.....
<23> 1240 po=0..... :'SONDERZ
     EICHEN AUS
<79> 1250 l$=STRING$(68,32)
<27> 1260 lw$="A":mlw$="ABM":z$=".":num$="0123456789"
<27> 1270 alpnum$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"+num$
<96> 1280 GOSUB 10000:'BILDSCHIRMKOM.
<42> 1290 GOSUB 5100 :'EDITOR-FELD LöSCHEN
<12> 1300 GOSUB 7100 :'MENüMASKE LADEN
<13> 1310 GOSUB 7200 :'INHALTSVERZEICHNIS LADEN
<93> 1500 '
<96> 1510 'HAUPTMENü
< 0> 1520 '
<37> 1530 PRINT statoff$;home$;cls$;
<62> 1540... FOR i=0 TO 30:PRINT m$(i);:NEXT
<51> 1550 PRINT staton$;FNat$(9,4);"Lw=";lw$;
<69> 1560... GOSUB 3000
<47> 1570 PRINT FNat$(21,1);invon$;" HAUPTMENü. ";invof
     f$;
<64> 1580... n=0
<94> 1590... GOSUB 3100
<96> 1600..... ON wahl+1 GOSUB 4000,4200,4400,2000,490
     0,5000
<85> 1610 IF n=1 THEN 1530 ELSE 1570
```

```
<84> 2000 '
<60> 2010 'PARAMMETER-DATEI ERSTELLEN
<90> 2020 '
<61> 2030 PRINT FNat$(21,1);invon$;" PRM-DATEI. ";invof
     f$;
<58> 2040... IF kd=1 THEN 7700
<48> 2050 IF inhalt(aktdat)=0 THEN 2130
<32> 2060... PRINT FNat$(21,7);CHR$(7);"PRM-Datei exist
     iert";
<32> 2070... PRINT " >Löschen und Weiter=RETURN Abbruch
     =EXIT";
<63> 2080... x$=INPUT$(1):PRINT FNat$(21,7);l$;
<62> 2090..... IF ASC(x$)<>13 THEN RETURN
<69> 2100... KILL LEFT$(inhalt$(aktdat),8)+ip$
<25> 2110... PRINT FNat$(15,aktdat+6);CHR$(187);
<96> 2120... inhalt(aktdat)=0
<71> 2130... mskname$=inhalt$(aktdat)
< 0> 2140... prmname$=LEFT$(mskname$,8)+ip$
<26> 2150 PRINT FNat$(21,7);"PRM-Dateiname :";prmname$;
     " >Weiter=RETURN Abbruch=EXIT";
<59> 2160... x$=INPUT$(1):PRINT FNat$(21,7);l$;
<58> 2170..... IF ASC(x$)<>13 THEN RETURN
<17> 2180... PRINT FNat$(21,7);"Bitte Warten... Aktiv i
     n Zeile :";
< 0> 2190... GOSUB 6200:GOSUB 6400
<49> 2200 'ANZAHL DER FELDER ERMITTELN
<39> 2210... anz=0:jetsam=0:l=0
<14> 2220..... FOR i=0 TO 30:FOR j=0 TO 89
< 9> 2230..... IF MID$(v$(i),j+1,1)=CHR$(172) THEN
     anz=anz+1
<27> 2240..... IF MID$(v$(i),j+1,1)=CHR$(176) TH
     EN jetsam=1
<97> 2250..... NEXT:PRINT FNat$(55,7);i;:NEXT
<44> 2300 'DIMENSIONIEREN
<13> 2310... IF jetsam=0 THEN DIM f(anz,3) ELSE DIM f(a
     nz,5)
<49> 2320..... GOSUB 5200:PRINT statoff$;
<76> 2330..... f(0,0)=jetsam:f(0,1)=anz
<45> 2400 'DATEN DER FELDER ERMITTELN
<74> 2410... inx=1
<18> 2420..... FOR i=0 TO 30:FOR j=0 TO 89
< 0> 2430..... IF MID$(v$(i),j+1,1)=CHR$(172) THEN
     GOSUB 2700:inx=inx+1
<39> 2440..... NEXT:NEXT
< 9> 2500 'FELDER ZEICHNEN
<34> 2510... l=0:c=ASC(z$)
<46> 2520..... FOR k=1 TO anz
<78> 2530..... j=f(k,0):i=f(k,1)
<18> 2540..... IF f(k,2)=0 THEN c=32:GOSUB 2900:c=A
     SC(z$):GOTO 2580
< 8> 2550 FOR n=1 TO f(k,2)
<36> 2560... IF j=89 THEN f(k,2)=j-f(k,0):GOTO 2570 ELS
     E GOSUB 2900:j=j+1
<15> 2570 NEXT
<36> 2580..... NEXT
<18> 2590... f(0,2)=0:FOR i=1 TO anz:f(0,2)=f(0,2)+f(i,
     2):NEXT
<32> 2600 GOSUB 6000:GOSUB 6500
<50> 2610... inhalt(aktdat)=1:ERASE f:n=1
<97> 2620 RETURN
<98> 2700 '
<30> 2710... f(inx,0)=j:f(inx,1)=i
<31> 2720... p=2:IF jetsam=1 THEN pmax=5 ELSE pmax=3
<71> 2730... n$="":l=0
<98> 2740 IF j+l=89 OR p>pmax THEN IF n$="" OR p>pmax T
     HEN RETURN ELSE 2780
<70> 2750... x$=MID$(v$(i),j+2+l,1)
<31> 2760..... IF INSTR(num$,x$) THEN n$=n$+x$:l=l+1:c
     =32:GOSUB 2900:GOTO 2740
<46> 2770..... l=l+1:c=32:GOSUB 2900
<31> 2780... f(inx,p)=INT(VAL(n$))
<56> 2790..... p=p+1:n$=""
<57> 2800 GOTO 2740
< 3> 2900 '
< 8> 2910... v$(i)=LEFT$(v$(i),j+l)+CHR$(c)+RIGHT$(v$(i
     ),90-(j+l+1))
< 1> 2920..... PRINT FNat$(j+l,i);CHR$(c);
< 7> 2930 RETURN
<85> 3000 '
<83> 3010 'INHALTSVERZEICHNIS ANZEIGEN
<91> 3020 '
<80> 3030... FOR i=0 TO imax
<80> 3040..... PRINT FNat$(1,i+6);inhalt$(i);FNat$(15,
     i+6);
<97> 3050..... IF inhalt(i)=1 THEN PRINT CHR$(188);
     ELSE PRINT CHR$(187);
```

```
< 3> 3060... NEXT
<81> 3070... aktdat=0
< 5> 3080 RETURN
<87> 3100 '
<75> 3110 'WÄHLEN
<93> 3120 '
<43> 3130... c=0
<64> 3140 WHILE c<>13
<79> 3150... PRINT FNat$(21,4);
<31> 3160..... FOR i=0 TO 5
<26> 3170..... IF i=wahl THEN PRINT invon$;
<20> 3180..... PRINT FNat$(21+(13*i),4);wahl$(w,
     i);invoff$
<32> 3190..... NEXT
<28> 3200... GOSUB 3500
<79> 3210... PRINT FNat$(88,1);
<91> 3220... c=ASC(INPUT$(1))
<70> 3230..... IF c=6 THEN wahl=wahl+1:IF wahl=6 THEN
     wahl=0:GOTO 3270
<95> 3240..... IF c=1 THEN wahl=wahl-1:IF wahl=-1 THEN
      wahl=5:GOTO 3270
<54> 3250..... IF c=31 THEN GOSUB 3300
<71> 3260..... IF c=30 THEN GOSUB 3400
<38> 3270 WEND
<55> 3280... PRINT FNat$(21,7);l$;
<12> 3290 RETURN
<91> 3300 '
<45> 3310 'CURSOR UP
<97> 3320 '
<55> 3330... GOSUB 3600
<94> 3340... aktdat=aktdat-1
<75> 3350..... IF aktdat<0 THEN aktdat=0
< 5> 3360 RETURN
<93> 3400 '
<37> 3410 'CURSOR DOWN
< 0> 3420 '
<57> 3430... GOSUB 3600
<52> 3440... aktdat=aktdat+1
<19> 3450..... IF aktdat>imax THEN aktdat=imax
< 7> 3460 RETURN
<95> 3500 '
<77> 3510 'AKTUELLE DATEI INVERS SETZEN
< 2> 3520 '
<73> 3530... PRINT FNat$(1,aktdat+6);invon$;inhalt$(akt
     dat);invoff$;
< 3> 3540 RETURN
<97> 3600 '
<81> 3610 'AKTUELLE DATEI NORMAL SETZEN
< 4> 3620 '
<50> 3630... PRINT FNat$(1,aktdat+6);inhalt$(aktdat);
< 5> 3640 RETURN
<86> 4000 '
<80> 4010 'MASKE ERSTELLEN
<92> 4020 '
<78> 4030 PRINT FNat$(21,1);invon$;" ERSTELLEN. ";invof
     f$;
< 1> 4040... PRINT FNat$(21,7);"NAME :";
<47> 4050... GOSUB 7500:PRINT FNat$(21,7);l$;
<42> 4060... mskname$=UPPER$(LEFT$(mskname$,8))+im$
<16> 4070... PRINT FNat$(21,7);"NAME :";mskname$;
<78> 4080..... x$=STRIP$(FIND$(mskname$))
<91> 4090..... IF x$<>"" THEN PRINT CHR$(7);" Maske
      existiert >Abbruch=EXIT";:c=1
<94> 4100..... IF x$="". THEN PRINT " >Weiter= R
     ETURN Abbruch=EXIT";:c=0
<46> 4110... x$=INPUT$(1):PRINT FNat$(21,7);l$;
<60> 4120..... IF c=1 OR ASC(x$)<>13 THEN RETURN
<28> 4130... GOSUB 5100:GOSUB 5200:GOSUB 5300:GOSUB 600
     0:GOSUB 7200:n=1
<95> 4140 RETURN
<90> 4200 '
<65> 4210 'MASKE BEARBEITEN
<96> 4220 '
<19> 4230 PRINT FNat$(21,1);invon$;" BEARBEITEN ";invof
     f$;
<64> 4240... IF kd=1 THEN 7700
<81> 4250... mskname$=inhalt$(aktdat)
<23> 4260 PRINT FNat$(21,7);"NAME :";mskname$;" >Weiter
     =RETURN Abbruch=EXIT";
<41> 4270... x$=INPUT$(1)
<40> 4280..... IF ASC(x$)<>13 THEN PRINT FNat$(21,7);l
     $;:RETURN
<68> 4290... GOSUB 6200:GOSUB 5200:GOSUB 5300:GOSUB 600
     0:n=1
<87> 4300 RETURN
<94> 4400 '
<44> 4410 'DISKETTE
< 1> 4420 '
<39> 4430 PRINT FNat$(21,1);invon$;". DISKETTE. ";invof
     f$;
<60> 4440... RESET
<18> 4450... w=1:wahl=0:GOSUB 3100
< 3> 4460..... ON wahl+1 GOSUB 4500,4610,4710,4810
<87> 4470... w=0:wahl=2:GOSUB 7400:GOSUB 7200:GOSUB 300
     0
<14> 4480 RETURN
<96> 4500 '
<89> 4510 PRINT FNat$(21,7);"Disketten-Wechsel möglich
     ! >Danach RETURN Abbruch=EXIT";
<57> 4520... x$=INPUT$(1):PRINT FNat$(21,7);l$;
< 1> 4530 RETURN
<98> 4600 '
<44> 4610. OPTION FILES "A":PRINT FNat$(9,4);"Lw=A";:lw
     $="A":RETURN
< 1> 4700 '
<67> 4710. OPTION FILES "B":PRINT FNat$(9,4);"Lw=B";:lw
     $="B":RETURN
< 3> 4800 '
< 3> 4810. OPTION FILES "M":PRINT FNat$(9,4);"Lw=M";:lw
     $="M":RETURN
< 5> 4900 '
<46> 4910 'OPTIONEN
<11> 4920 '
<50> 4930 PRINT FNat$(21,1);invon$;". OPTIONEN. ";invof
     f$;
<36> 4940... w=2:wahl=0:GOSUB 3100
<71> 4950..... ON wahl+1 GOSUB 8030,8200,8500,8700,900
     0
< 3> 4960... w=0:wahl=4
<21> 4970 RETURN
<87> 5000 '
< 2> 5010 'EXIT
<93> 5020 '
<86> 5030... PRINT FNat$(21,7);CHR$(7);"Zurück nach CP/
     M >Ja=EXIT";
<29> 5040... x$=INPUT$(1)
<11> 5050..... IF ASC(x$)<>27 THEN PRINT FNat$(21,7);l
     $:RETURN
< 4> 5060 RESET:SYSTEM
<89> 5100 '
<82> 5110 'VIDEOSPEICHER LöSCHEN
<95> 5120 '
<56> 5130... FOR i=0 TO 30
<55> 5140..... v$(i)=STRING$(90,32)
< 4> 5150... NEXT
< 3> 5160 RETURN
<91> 5200 '
<68> 5210 'VIDEOSPEICHER ZEIGEN
<97> 5220 '
<70> 5230... PRINT statoff$;home$;
<61> 5240... FOR i=0 TO 30
<76> 5250..... PRINT v$(i);
< 9> 5260... NEXT
<57> 5270... PRINT staton$;FNat$(0,0);
<11> 5280 RETURN
<93> 5300 '
<18> 5310 'EDITOR
< 0> 5320 '
< 2> 5330... x=0:y=0:PRINT statoff$;home$;
<47> 5340... x$=INPUT$(1):c=ASC(x$):IF x$=esc$ THEN PRI
     NT staton$;:x=0:y=0:RETURN
<82> 5350.... IF c=13 THEN PRINT x$;:c=30:x=0
<28> 5360..... IF po=1 THEN IF c<64 OR c>126 THEN ELSE
     x$=CHR$(sz(c-64))
<76> 5370 IF c=1 OR c=127 THEN IF x>0. THEN x=x-1:PRINT
      FNat$(x,y);.. :GOTO 5340 ELSE 5340
<43> 5380 IF c=6 OR c=7.. THEN IF x<89 THEN..... PRINT
      cright$;:x=x+1:GOTO 5340 ELSE 5340
< 8> 5390 IF c=31..... THEN IF y>0. THEN..... PRINT
      cup$.. ;:y=y-1:GOTO 5340 ELSE 5340
<33> 5400 IF c=30..... THEN IF y<30 THEN..... PRINT
      cdown$ ;:y=y+1:GOTO 5340 ELSE 5340
< 9> 5410..... IF c=22 THEN po=1:GOTO 5340 ELSE IF c=28
      THEN po=0:GOTO 5340
<92> 5420 IF c<32 THEN 5340
<19> 5430 IF (y<=30 AND x<=90) THEN PRINT x$;:v$(y)=LEF
     T$(v$(y),x)+x$+RIGHT$(v$(y),90-(x+1))
< 3> 5440..... IF x<89 THEN x=x+1 ELSE PRINT FNat$(x,y)
     ;
<52> 5450 GOTO 5340
<88> 6000 '
<51> 6010 'ABSPEICHERN DER MASKE OHNE PARAMETER
```

```
<94> 6020 '
<71> 6030... OPEN "R",#1,mskname$,90
<81> 6040... FIELD #1 , 90 AS zeile$
<46> 6050..... FOR i=0 TO 30
<27> 6060..... LSET zeile$=v$(i)
< 0> 6070..... PUT #1
<30> 6080..... NEXT
<94> 6090... CLOSE #1..
<85> 6100 RETURN
<92> 6200 '
<64> 6210 'EINLESEN DER MASKE OHNE PARAMETER
<98> 6220 '
< 6> 6230... GOSUB 5100
<78> 6240... OPEN "R",#1,mskname$,90
<88> 6250... FIELD #1 , 90 AS zeile$
<53> 6260..... FOR i=0 TO 30
< 9> 6270..... GET #1
<44> 6280..... LSET v$(i)=zeile$
<37> 6290..... NEXT
<33> 6300... CLOSE #1
<92> 6310 RETURN
<96> 6400 '
< 0> 6410 'BAK-DATEI ERSTELLEN
< 3> 6420 '
<44> 6430... mskname$=LEFT$(mskname$,8)+ib$
<12> 6440... GOSUB 6000
<61> 6450... mskname$=LEFT$(mskname$,8)+im$
<10> 6460 RETURN
<98> 6500 '
<52> 6510 'PARAMMETER-DATEI ABSPEICHERN
< 5> 6520 '
<69> 6530... mskname$=LEFT$(mskname$,8)+ip$
<55> 6540..... IF jetsam<>0 THEN 6630
<54> 6550... OPEN "R",#1,mskname$,8
<53> 6560... FIELD #1,8 AS w$
<15> 6570..... FOR i=0 TO anz
<18> 6580..... LSET w$=MKI$(f(i,0))+MKI$(f(i,1))+MK
     I$(f(i,2))+MKI$(f(i,3))
<16> 6590..... PUT #1
<18> 6600..... NEXT
<42> 6610... CLOSE #1
< 2> 6620 RETURN
< 4> 6630... OPEN "R",#1,mskname$,12
<52> 6640... FIELD #1,12 AS w$
<11> 6650..... FOR i=0 TO anz
< 4> 6660 LSET w$=MKI$(f(i,0))+MKI$(f(i,1))+MKI$(f(i,2)
     )+MKI$(f(i,3))+MKI$(f(i,4))+MKI$(f(i,5))
<36> 6670..... PUT #1
<24> 6680... NEXT
<66> 6690... CLOSE #1
<97> 6700 RETURN
< 5> 6800 '
<54> 6810 'PARAMETER-DATEI LADEN
<11> 6820 '
<75> 6830... mskname$=LEFT$(mskname$,8)+ip$
<57> 6840... OPEN "R",#1,mskname$,8
<70> 6850... FIELD #1,2 AS xw$,2 AS yw$,2 AS anz$,2 AS
     typ$
<36> 6860... GET #1
<65> 6870... jetsam=CVI(xw$):anz=CVI(yw$):IF jetsam=0 T
     HEN 6940
<84> 6880 CLOSE #1:OPEN "R",#1,mskname$,12
<18> 6890 FIELD #1,2 AS xw$,2 AS yw$,2 AS anz$,2 AS typ
     $,2 AS ans$,2 AS rw$
<20> 6900... GET #1
<62> 6910 DIM f(anz,5)
<36> 6920 f(0,0)=1:f(0,1)=anz:f(0,2)=CVI(anz$)
<91> 6930... GOTO 6960
<88> 6940... DIM f(anz,4)
<95> 6950... f(0,0)=0:f(0,1)=anz:f(0,2)=CVI(anz$)
<38> 6960..... FOR i=1 TO anz
<23> 6970..... GET #1
<17> 6980..... f(i,0)=CVI(xw$):f(i,1)=CVI(yw$).
<87> 6990..... f(i,2)=CVI(anz$):f(i,3)=CVI(typ$)
<12> 7000..... IF jetsam=1 THEN f(i,4)=CVI(ans$):f(
     i,5)=CVI(rw$)
<10> 7010..... NEXT
<16> 7020 CLOSE #1
<93> 7030 RETURN
<91> 7100 '
<94> 7110 'MENü-MASKE EINLESEN
<97> 7120 '
<89> 7130 mskname$=STRIP$(FIND$("MSKGENM.MSK"))
<88> 7140... IF mskname$="" THEN 7190
<42> 7150... GOSUB 6200
<52> 7160..... FOR i=0 TO 30
<92> 7170..... m$(i)=v$(i)
<33> 7180..... NEXT
<14> 7190 RETURN
<93> 7200 '
<87> 7210 'INHALTSVERZEICHNIS ERSTELLEN
< 0> 7220 '
<38> 7230... imax=-1:i=0
<17> 7240... inhalt$(i)=STRIP$(FIND$("*"+im$,i+1))
<67> 7250 WHILE inhalt$(i)<>"" AND i<24
<71> 7260... x$=STRIP$(FIND$(LEFT$(inhalt$(i),8)+ip$))
<68> 7270..... IF x$<>"" THEN inhalt(i)=1 ELSE inhalt(
     i)=0
<17> 7280... i=i+1
<32> 7290... inhalt$(i)=STRIP$(FIND$("*"+im$,i+1))
<23> 7300 WEND
<54> 7310... imax=i-1:kd=0
<89> 7320..... IF imax=-1 THEN kd=1:inhalt$(0)="KEINE.
      DATEN"
< 0> 7330 RETURN
<97> 7400 '
< 8> 7410 'INHALTSVERZEICHNIS LöSCHEN
< 4> 7420 '
< 8> 7430... FOR i=0 TO 23
<76> 7440..... PRINT FNat$(1,i+6);STRING$(12,32);FNat$
     (15,i+6);" ";
<12> 7450... NEXT
<11> 7460 RETURN
< 0> 7500 '
<91> 7510 'MASKENNAMEN EINLESEN
< 6> 7520 '
<86> 7530 mskname$="":INPUT x$:x$=UPPER$(x$)
<49> 7540... IF LEN(x$)>8 THEN x$=LEFT$(x$,8):PRINT CHR
     $(7);
< 3> 7550..... FOR i= 1 TO LEN(x$)
<75> 7560..... n$=MID$(x$,i,1)
<61> 7570..... IF INSTR(alpnum$,n$) THEN mskname
     $=mskname$+n$ ELSE PRINT CHR$(7);
<59> 7580..... IF i=1 AND INSTR(num$,n$) THEN
      mskname$="M":PRINT CHR$(7);
<44> 7590..... NEXT
<76> 7600..... IF mskname$="" THEN PRINT CHR$(7);:m
     skname$="MASKE"
<12> 7610... mskname$=mskname$+STRING$(8-LEN(mskname$),
     32)
< 3> 7620 RETURN
< 4> 7700 '
<19> 7710 'MELDUNG "KEINE DATEN
<10> 7720 '
<30> 7730 PRINT FNat$(21,7);CHR$(7);"Keine Daten >Abbru
     ch=EXIT";
<90> 7740... y$=INPUT$(1):PRINT FNat$(21,7);l$;:n=0
<14> 7750 RETURN
<90> 8000 '
<77> 8010 'UNTERPROGRAMME FüR OPTIONEN
<96> 8020 '
< 0> 8030 '
<39> 8040 'BAK=MSK
< 6> 8050 '
<30> 8060 PRINT FNat$(21,1);invon$;". BAK=MSK.. ";invof
     f$;
<73> 8070... IF kd=1 THEN 7700
< 5> 8080... x$=STRIP$(FIND$(LEFT$(inhalt$(aktdat),8)+i
     b$))
<82> 8090..... IF x$<>"" THEN 8130
< 6> 8100... PRINT FNat$(21,7);CHR$(7);"BAK-Datei nicht
      vorhanden >Abbruch=EXIT";
<50> 8110... x$=INPUT$(1):PRINT FNat$(21,7);l$;
<93> 8120 RETURN
<17> 8130... KILL inhalt$(aktdat)
<61> 8140... NAME LEFT$(inhalt$(aktdat),8)+ib$ AS inhal
     t$(aktdat)
< 3> 8150 RETURN
<94> 8200 '
< 2> 8210 'PARAMETER-DATEI DRUCKEN
< 1> 8220 '
<22> 8230 PRINT FNat$(21,1);invon$;". LIST-PRM. ";invof
     f$;
<68> 8240... IF kd=1 THEN 7700
<18> 8250... IF inhalt(aktdat)=1 THEN 8280
<47> 8260 PRINT FNat$(21,7);CHR$(7);"Keine PRM-Datei vo
     rhanden >Abbruch=EXIT";
<98> 8270... x$=INPUT$(1):PRINT FNat$(21,7);l$:RETURN
<76> 8280 mskname$=LEFT$(inhalt$(aktdat),8)+ip$
<36> 8290 GOSUB 6800
<53> 8300... IF f(0,0)=1 THEN LPRINT "JETSAM ";
<69> 8310... LPRINT "Parameter-Datei für ";LEFT$(msknam
```

```
    e$,8)
< 7> 8320... LPRINT
< 3> 8330... LPRINT "Cursorpositionen :";f(0,1)
<24> 8340... LPRINT "Record-Länge.... :";f(0,2)
<16> 8350... LPRINT
<64> 8360... LPRINT "Nr... cursor X. cursor Y. Feld-Län
    ge.. Typ. ";
<33> 8370... IF f(0,0)=1 THEN LPRINT "Schlüssel-L.. Rei
    he" ELSE LPRINT
<36> 8380..... FOR i=1 TO f(0,1)
<71> 8390 LPRINT i TAB(10) f(i,0) TAB(20) f(i,1) TAB(30
    ) f(i,2) TAB(40) f(i,3) TAB(50);
<98> 8400... IF f(0,0)=1 THEN LPRINT f(i,4) TAB(60) f(i
    ,5) ELSE LPRINT
< 0> 8410..... NEXT
< 6> 8420... LPRINT CHR$(12)
<92> 8430... ERASE f
< 6> 8440 RETURN
< 1> 8500 '
<60> 8510 'UMBENENNEN
< 7> 8520 '
< 3> 8530 PRINT FNat$(21,1);invon$;" UMBENENNEN ";invof
    f$;
<74> 8540... IF kd=1 THEN 7700
<93> 8550... PRINT FNat$(21,7);"Neuer Name :";
<80> 8560... GOSUB 7500:mskname$=mskname$+im$:x$=STRIP$
    (FIND$(mskname$))
<82> 8570..... IF x$="" THEN 8600
<87> 8580... PRINT FNat$(21,7);CHR$(7);mskname$;" exist
    iert >Abbruch=EXIT";
<57> 8590... x$=INPUT$(1):PRINT FNat$(21,7);1$;:RETURN
<75> 8600 PRINT FNat$(21,7);invon$;inhalt$(aktdat);invo
    ff$;" wird ";invon$;mskname$;invoff$;
<97> 8610 PRINT " >Weiter=RETURN Abbruch=EXIT";
<70> 8620... x$=INPUT$(1):IF ASC(x$)<>13 THEN PRINT FNa
    t$(21,7);1$;:RETURN
< 2> 8630... x$=STRIP$(FIND$(LEFT$(inhalt$(aktdat),8)+i
    b$))
<61> 8640 IF x$<>"" THEN NAME LEFT$(inhalt$(aktdat),8)+
    ib$ AS LEFT$(mskname$,8)+ib$
<74> 8650 IF inhalt(aktdat)=1 THEN NAME LEFT$(inhalt$(a
    ktdat),8)+ip$ AS LEFT$(mskname$,8)+ip$
<57> 8660... NAME inhalt$(aktdat) AS mskname$
<38> 8670... GOSUB 7400:GOSUB 7200:GOSUB 3000:PRINT FNa
    t$(21,7);1$;
<22> 8680 RETURN
< 5> 8700 '
<11> 8710 'KOPIEREN
<11> 8720 '
< 6> 8730 PRINT FNat$(21,1);invon$;". KOPIEREN. ";invof
    f$;
<78> 8740... IF kd=1 THEN 7700
<15> 8750 RESET
<24> 8760... PRINT FNat$(21,7);"Laufwerk=";lw$;". Ziel-
    Laufwerk (";
<60> 8770 IF lw$="A" THEN PRINT "B,M"; ELSE IF lw$="B"
    THEN PRINT "A,M"; ELSE PRINT "A,B";
<74> 8780... PRINT ") =";
<85> 8790... x$=INPUT$(1):x$=UPPER$(LEFT$(x$,1))
<66> 8800..... IF x$="" OR x$=lw$ OR INSTR(mlw$,x$)=0
    THEN PRINT CHR$(7);:GOTO 8790
<87> 8810... PRINT x$;
<82> 8820 mskname$=inhalt$(aktdat)
<14> 8830... GOSUB 6200:IF inhalt(aktdat)=1 THEN GOSUB
    6800 ELSE DIM f(1,1)
<54> 8840..... PRINT FNat$(21,7);1$;
<81> 8850 PRINT FNat$(21,7);inhalt$(aktdat);" Nach ";x$
    ;" >Weiter=RETURN Abbruch=EXIT";
<88> 8860... n$=INPUT$(1):PRINT FNat$(21,7);1$;
< 1> 8870..... IF ASC(n$)<>13 THEN ERASE f:RETURN
< 7> 8880... mskname$=inhalt$(aktdat)
<50> 8890... OPTION FILES x$
<55> 8900..... n$=STRIP$(FIND$(mskname$))
<69> 8910..... IF n$="" THEN 8960
<63> 8920 PRINT FNat$(21,7);CHR$(7);"Datei existiert au
    f ";x$;" >Löschen und Weiter=RETURN";
<20> 8930 PRINT " Abbruch=EXIT";
<84> 8940... n$=INPUT$(1):PRINT FNat$(21,7);1$;
<96> 8950..... IF ASC(n$)<>13 THEN ERASE f:RETURN
<18> 8960... GOSUB 6000:IF inhalt(aktdat)=1 THEN GOSUB
    6500
<70> 8965... OPTION FILES lw$:ERASE f
<25> 8970 RETURN
<91> 9000 '
<67> 9010 'LöSCHEN
<97> 9020 '
<54> 9030 PRINT FNat$(21,1);invon$;". LöSCHEN.. ";invof
    f$
<65> 9040... IF kd=1 THEN 7700
<88> 9050... PRINT FNat$(21,7);inhalt$(aktdat);" Lösche
    n >Ja=SHIFT J Abbruch=EXIT";
<40> 9060 x$=INPUT$(1):PRINT FNat$(21,7);1$;
<92> 9070... IF x$<>"J" THEN RETURN
<87> 9080 IF inhalt(aktdat)=1 THEN KILL LEFT$(inhalt$(a
    ktdat),8)+ip$
< 9> 9090... x$=STRIP$(FIND$(LEFT$(inhalt$(aktdat),8)+i
    b$))
<17> 9100..... IF x$<>"" THEN KILL x$
<12> 9110... KILL inhalt$(aktdat)
< 1> 9120... GOSUB 7400:GOSUB 7200:GOSUB 3000
<97> 9130 RETURN
< 2> 9500 '
<97> 9510 'VORBESETZUNG
< 8> 9520 '
<15> 9530 '.... A.. B.. C.. D.. E.. F.. G.. H.. I.. J
<98> 9540 DATA 128, 32, 32, 32, 32, 32, 32, 32, 32, 32
<67> 9550 '.... K.. L.. M.. N.. O.. P.. Q.. R.. S.. T
<86> 9560 DATA. 32, 32, 32, 32, 32, 32,144, 32, 32, 32
< 4> 9570 '.... U.. V.. W.. X.. Y.. Z.. ä.. ö.. ü.. ^
<80> 9580 DATA. 32, 32, 32, 32, 32, 32, 32, 32, 32, 32
<40> 9590 '.... _.. `.. a.. b.. c.. d.. e.. f.. g.. h
<67> 9600 DATA. 32, 32,130,137,143,133,138,141,136,147
<48> 9610 '.... i.. j.. k.. l.. m.. n.. o.. p.. q.. r
<12> 9620 DATA 151,155,157,152,154,149,159,145,134,142
<84> 9630 '.... s.. t.. u.. v.. w.. x.. y.. z.. ä.. ö
<16> 9640 DATA 135,140,158,139,132,129,131,150,153,148
<45> 9650 '.... ü.. ß
<78> 9660 DATA 156,146
< 6> 9700 '
<32> 9710 DATA " Erstellen. "," Bearbeiten ",". Diskett
    e. "," PRM-DATEI. "
<76> 9720 DATA ". Optionen. "," E "
<65> 9730 DATA "Disk-Wechsel"," Laufwerk=A "," Laufwerk
    =B "," Laufwerk=M "
<76> 9740 DATA "............ "," E "
<30> 9750 DATA " MSK = BAK. ",". List PRM. "," Umbenenn
    en ",". Kopieren. "
<33> 9760 DATA ". Löschen.. "," E "
<97> 10000 '
<28> 10010 'BILDSCHIRMKOMMANDOS
< 6> 10020 '
< 6> 10030. esc$.... = CHR$(27):'ESCAPE
<50> 10040. home$... = esc$+"H":'CURSOR IN ZEILE 0, SPA
    LTE 0
<63> 10050. eras$... = esc$+"J":'VON CURSOR BIS ENDE SE
    ITE LöSCHEN
< 6> 10060. cls$.... = esc$+"E":'BILDSCHIRM LöSCHEN
<55> 10070. cup$.... = esc$+"A":'CURSOR EINE ZEILE HOCH
<64> 10080. cdown$.. = esc$+"B":'CURSOR EINE ZEILE RUNT
    ER
< 6> 10090. cright$. = esc$+"C":'CURSOR EIN ZEICHEN WEI
    TER
< 4> 10100. cleft$.. = esc$+"D":'CURSOR EIN ZEICHEN ZUR
    üCK
<36> 10110. staton$. = esc$+"1":'STATUSZEILE EIN
<52> 10120. statoff$ = esc$+"0":'STATUSZEILE AUS
<84> 10130. invon$.. = esc$+"p":'INVERS EIN
<37> 10140. invoff$. = esc$+"q":'INVERS AUS
<27> 10150. 'CURSOR AUF ZEILE x, SPALTE y SETZEN
<97> 10160... DEF FNat$(x,y)=esc$+"Y"+CHR$(32+y)+CHR$(3
    2+x)
<69> 10170 RETURN
```

```
LISTING >MASKROUT<, REMARK = >REM<.

<97> 10000 '
<28> 10010 'BILDSCHIRMKOMMANDOS
< 6> 10020 '
< 6> 10030. esc$.... = CHR$(27):'ESCAPE
<50> 10040. home$... = esc$+"H":'CURSOR IN ZEILE 0, SPA
    LTE 0
<63> 10050. eras$... = esc$+"J":'VON CURSOR BIS ENDE SE
    ITE LöSCHEN
< 6> 10060. cls$.... = esc$+"E":'BILDSCHIRM LöSCHEN
<55> 10070. cup$.... = esc$+"A":'CURSOR EINE ZEILE HOCH
<64> 10080. cdown$.. = esc$+"B":'CURSOR EINE ZEILE RUNT
    ER
< 6> 10090. cright$. = esc$+"C":'CURSOR EIN ZEICHEN WEI
    TER
< 4> 10100. cleft$.. = esc$+"D":'CURSOR EIN ZEICHEN ZUR
```

```
     üCK
<36> 10110. staton$. = esc$+"1":'STATUSZEILE EIN
<52> 10120. statoff$ = esc$+"0":'STATUSZEILE AUS
<84> 10130. invon$.. = esc$+"p":'INVERS EIN
<37> 10140. invoff$. = esc$+"q":'INVERS AUS
<27> 10150. 'CURSOR AUF ZEILE x, SPALTE y SETZEN
<97> 10160... DEF FNat$(x,y)=esc$+"Y"+CHR$(32+y)+CHR$(3
     2+x)
<43> 10170. 'CURSOR AUF EIN MASKENFELD SETZEN
<78> 10180... DEF FNgo$(f)=FNat$(f(f,0),f(f,1))
<44> 10200 RETURN
< 2> 12000 '
<84> 12010 'ROUTINEN ZUR HANDHABUNG VON MASKEN
<10> 12020 '
<71> 12030 'Von den Maskenroutinen benutzte Variable
<86> 12031 '(Variable mit * dürfen im Anwenderprogramm
<68> 12032 ' nicht anderweitig benutzt werden)
<29> 12033 '
<98> 12040 ' m$(30)* , z$(20)* , mskname$ , i$.
<82> 12050 ' f(x,4)* , z(20,20)*
<20> 12060 ' anz* , feld , typ , xa , xm , xp , z , zl
<14> 12070 ' az , sp , sm , e , c , i , j
<34> 12080 '
<76> 12081 'das FELD dat$(anz) dient der Datenein- und
<73> 12082 'ausgabe
< 5> 12100 '
<15> 12110 'MASKE LESEN
<13> 12120 '
<40> 12130 'EINGABEN : Name der Maske in mskname$
<38> 12140 '.......... (max 8 Zeichen)
<28> 12150... DEFINT z,f
<45> 12160... DIM z$(25),z(25,25)
<50> 12170... DIM m$(30)
<37> 12190... OPEN "R",#1,mskname$+".MSK",90
<73> 12200.... FIELD #1,90 AS zeile$
<50> 12210..... FOR i=0 TO 30
<93> 12220...... GET #1
<57> 12230....... m$(i)=zeile$
<37> 12240..... NEXT
< 7> 12250.... CLOSE #1
<72> 12260 RETURN
<11> 12300 '
<18> 12310 'PARAMETER-MASKE LESEN
<19> 12320 '
<46> 12330 'EINGABEN : Name der Maske in mskname$
<44> 12340 '.......... (max 8 Zeichen)
<34> 12350... OPEN "R",#1,mskname$+".PRM",8
<51> 12360.... FIELD #1,2 AS w$(0),2 AS w$(1),2 AS w$(2
     ),2 AS w$(3)
<57> 12365 'Bei JETSAM-Datei >2 AS w$(4),2 AS w$(5)< an
     hängen
<30> 12370..... GET #1
<71> 12380...... anz=CVI(w$(1))
<74> 12390...... DIM f(anz,4)
<25> 12400....... FOR i=0 TO anz
<97> 12405 'BEI JETSAM-Datei >FOR j=0 TO 5<
<62> 12410........ GET #1,i+1
<73> 12420........ FOR j=0 TO 3
<70> 12430......... f(i,j)=CVI(w$(j))
<79> 12440........ NEXT
<25> 12450....... NEXT
<17> 12460.... CLOSE #1
<47> 12470.... DIM dat$(anz)
<86> 12480 RETURN
<17> 12500 '
< 0> 12510 'MASKE AUSGEBEN
<25> 12520 '
<28> 12530... WIDTH 255
<69> 12540... PRINT statoff$;cls$;home$;
<67> 12550.... FOR i=0 TO 30
<43> 12560..... PRINT m$(i);
<39> 12570.... NEXT
<78> 12580... PRINT staton$;FNat$(0,0);
<93> 12590 RETURN
<20> 12600 '
<20> 12610 'MASKE LöSCHEN
<28> 12620 '
<59> 12630.. ERASE z$,m$,dat$,z,f
<76> 12640 RETURN
<23> 12700 '
<54> 12710 'EIN EINGABEFELD BEARBEITEN
<31> 12720 '
<20> 12730 'EINGABEN : Nummer des zu bearbeitenden Feld
     es
<33> 12740 '.......... in der Variablen feld
<11> 12750... typ=f(feld,3)
<62> 12760... i$=dat$(feld)
<22> 12765... c=32
< 8> 12770... xa=f(feld,0):xm=f(feld,0)+f(feld,2):xp=xa
     :zl=f(feld,1):sp=1:sm=f(feld,2)
<51> 12780..... ON typ GOSUB 14100,14200,14300,14400 :
     'Hier weitere Typen anhängen
<85> 12790... dat$(feld)=i$
<66> 12800 RETURN
< 4> 13000 '
<49> 13010 'TYP-PRüFER
<12> 13020 '
<51> 13030... READ az:FOR i=1 TO az:READ z$(i):FOR j=1
     TO az:READ z(i,j):NEXT:NEXT
<43> 13040... sp=1:z=1:i=1
<14> 13050.... IF z(z,i)=0 THEN e=1:PRINT CHR$(7);:RETU
     RN
<21> 13060..... IF MID$(i$,sp,1)="" THEN 13080
<78> 13070..... IF INSTR(z$(z(z,i)),MID$(i$,sp,1))THEN
      z=z(z,i):sp=sp+1:i=1:GOTO 13050
<27> 13080..... IF z(z,i)<>1 OR sp<=LEN(i$) THEN i=i+
     1:GOTO 13050
<80> 13090 RETURN
< 7> 13100 '
< 6> 13110 'EINGABEFELDER BEARBEITEN
<15> 13120 '
<35> 13140... feld=1:c=32
<91> 13150.... WHILE NOT(c=27)
<94> 13160.... c=32
<11> 13170..... IF feld >f(0,1) THEN feld=1
<75> 13180..... IF feld <1 THEN feld=f(0,1)
<32> 13190..... typ=f(feld,3)
<92> 13200..... IF typ=0 THEN feld=feld+1:GOTO 13170
<50> 13210..... i$=dat$(feld)
<24> 13220..... xa=f(feld,0):xm=f(feld,0)+f(feld,2
     ):xp=xa:zl=f(feld,1):sp=1:sm=f(feld,2)
<18> 13230..... ON typ GOSUB 14100,14200,14300,14
     400 :'Hier weitere Typen anhängen
< 7> 13240..... dat$(feld)=i$
<80> 13250..... IF c=13 OR c=30 THEN feld=feld+1
<19> 13260..... IF c=31 THEN feld=feld-1
<74> 13270.... WEND
<82> 13280 RETURN
<13> 13300 '
<48> 13310 'ZEILEN-EDITOR
<21> 13320 '
<70> 13330... e=0:c=32
<63> 13340... WHILE NOT(c=13 OR c=31 OR c=30 OR c=27)
<91> 13350.... PRINT FNat$(xa,zl);i$;MID$(m$(zl),xa+LEN
     (i$)+1,xm-LEN(i$));FNat$(xa+sp-1,zl);
< 1> 13360..... x$=INPUT$(1):c=ASC(x$):IF c=127 THEN c=
     8
<54> 13370..... IF c=6 AND sp<=LEN(i$) THEN sp=sp+1:xp
     =xp+1:GOTO 13460
< 3> 13380..... IF c=1 AND sp>1 THEN sp=sp-1:xp=xp-1:
     GOTO 13460
<18> 13390..... IF c=11 THEN i$=LEFT$(i$,sp-1):GOTO
     13460
<66> 13400..... IF c=24 THEN sp=1:xp=xa:GOTO 13460
< 1> 13410..... IF c=7 THEN GOSUB 13480:GOTO 13460
<73> 13420..... IF c=8 THEN GOSUB 13510:GOTO 1346
     0
<18> 13430..... IF c<32 THEN 13460
<11> 13440.... i$=LEFT$(i$,sp-1)+x$+MID$(i$,sp,sm):i$=L
     EFT$(i$,sm)
<66> 13450..... IF sp<sm THEN sp=sp+1:xp=xp+1 ELSE PRIN
     T CHR$(7);
<21> 13460... WEND
<84> 13470 RETURN
<48> 13480 '
<22> 13490... IF sp<=LEN(i$) THEN i$=LEFT$(i$,sp-1)+RIG
     HT$(i$,LEN(i$)-sp)
<59> 13500 RETURN
<23> 13510 '
<80> 13520... IF sp>1 AND LEN(i$)>=1 THEN i$=LEFT$(i$,s
     p-2)+RIGHT$(i$,LEN(i$)-(sp-1)):xp=xp-1:sp=sp-1
<71> 13530 RETURN
< 6> 14000 '
<18> 14010 'TYPEN
<14> 14020 '
<44> 14100 'TYP 1... UNIVERSALTYP
<50> 14110... GOSUB 13300:RETURN
<12> 14200 '
<22> 14210 'TYP 2... DEZIMALZAHL OHNE EXPONENT
<20> 14220 '
<48> 14230... RESTORE 15200..... :'An dieser Stelle
     stehen die Daten des Typs
<82> 14240.... GOSUB 13300..... :'Zeilen-Editor auf
     rufen
```

```
<54> 14250..... GOSUB 13000........ :'Typ-Prüfer aufruf
     en
<46> 14260...... IF e THEN 14230... :'Wenn Fehler dann
     nochmal
<80> 14270 RETURN
<15> 14300 '
<69> 14310 'TYP 3... NAME mit abk. durch Punkt
<23> 14320 '
<75> 14330... RESTORE 15300
<69> 14340.... GOSUB 13300
<79> 14350..... GOSUB 13000
<46> 14360...... IF e THEN 14330
<83> 14370 RETURN
<18> 14400 '
<84> 14410 'TYP 4... DATUM
<26> 14420 '
<98> 14430... RESTORE 15400
<72> 14440.... GOSUB 13300
<82> 14450..... GOSUB 13000
<74> 14460...... IF e THEN 14430
<86> 14470 RETURN
< 8> 15000 '
<86> 15010 'DATEN FüR DIE UNTERSCHIEDLICHEN TYPEN
<16> 15020 '
<20> 15200 'TYP 2... DEZIMALZAHL OHNE EXPONENT
<87> 15205 DATA 5
<82> 15210 DATA ""
<22> 15215..................... DATA 2,3,4,1,0
<56> 15220 DATA "+-"
< 9> 15225..................... DATA 3,4,0,0,0
<97> 15230 DATA "0123456789"
<49> 15235..................... DATA 3,4,1,0,0
<33> 15240 DATA "."
<78> 15245..................... DATA 5,1,0,0,0
< 6> 15250 DATA "0123456789"
<82> 15255..................... DATA 5,1,0,0,0
<81> 15300 'Typ 3... NAME mit abk. durch Punkt
<78> 15305 DATA 4
<85> 15310 DATA ""
< 0> 15315..................... DATA 2,1,0,0
<17> 15320 DATA "ABCDEFGHIJKLMNOPQRSTUVWXYZÄÜÖ"
<75> 15325..................... DATA 3,4,1,0
<11> 15330 DATA "abcdefghijklmnopqrstuvwxyzäöüß"
<79> 15335..................... DATA 3,4,1,0
<36> 15340 DATA "."
<45> 15345..................... DATA 1,0,0,0
<82> 15400 'TYP 4... DATUM
<92> 15410 DATA 22
<92> 15420 DATA ""
<86> 15430 DATA. 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0
< 8> 15440 DATA "0123456789"
< 8> 15450 DATA. 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<16> 15460 DATA "0123456789"
<29> 15470 DATA. 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<55> 15480 DATA "."
<47> 15490 DATA. 5, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<52> 15500 DATA "0"
< 5> 15510 DATA. 7, 8,11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<73> 15520 DATA "1"
<39> 15530 DATA. 9,10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<93> 15540 DATA "13578"
<48> 15550 DATA 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<25> 15560 DATA "469"
<69> 15570 DATA 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<26> 15580 DATA "02"
<64> 15590 DATA 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<68> 15600 DATA "1"
<48> 15610 DATA 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<89> 15620 DATA "2"
<69> 15630 DATA 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<45> 15640 DATA "."
<88> 15650 DATA 15,16,17, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<53> 15660 DATA "."
< 0> 15670 DATA 15,16,21, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<61> 15680 DATA "."
<61> 15690 DATA 15,16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<58> 15700 DATA "0"
<17> 15710 DATA 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<21> 15720 DATA "12"
<38> 15730 DATA 19, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<14> 15740 DATA "3"
<40> 15750 DATA 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<78> 15760 DATA "123456789"
<98> 15770 DATA. 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<33> 15780 DATA "0123456789"
< 7> 15790 DATA. 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<88> 15800 DATA "01"
<77> 15810 DATA. 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
< 9> 15820 DATA "3"
<61> 15830 DATA 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
<77> 15840 DATA "0"
<93> 15850 DATA. 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
LISTING >AUTOMSK <, REMARK = >REM<.

<87> 1000 'DURCHSCHNITTSVERBRAUCH BERECHNEN
<86> 1010 '
<76> 1020. mskname$="auto"............. :'NAME DER MASK
     E
<92> 1030 '
<27> 1040. GOSUB 10000................. :'BILDSCHIRMKOM
     ANDOS INITIALISIEREN
<51> 1050. GOSUB 12100................. :'MASKE LESEN
<54> 1060. GOSUB 12300................. :'PARAMETER-DAT
     EI LESEN
<84> 1070. GOSUB 12500................. :'MASKE AUSGEBE
     N
<96> 1080. GOSUB 13100................. :'EINGABEFELDER
      BEARBEITEN
<11> 1090 '
<55> 1100. aa=VAL(dat$(1))............. :'STRINGS IN RE
     ALS UMWANDELN
<48> 1110. ab=VAL(dat$(2))
<89> 1120. b=VAL(dat$(4))..............  :'FELD 4, WEIL
     FELD 3 = AUSGABEFELD
<94> 1130 '
< 2> 1140. a=ab-aa..................... :'GEFAHRENE KIL
     OMETER BERECHNEN
< 1> 1150 '
<40> 1160. PRINT FNgo$(3);a;........... :'ERGEBNIS AUF
     FELD 3 AUSGEBEN
< 7> 1170 '
<59> 1180. c=a/100:d=b/c............... :'VERBRAUCH BER
     ECHNEN
<50> 1190. d=INT((d+0.005)*100)/100.... :'ERGEBNIS RUND
     EN
<87> 1200 '
<45> 1210. PRINT FNgo$(5);d;........... :'ERGEBNIS AUF
     FELD 5 AUSGEBEN
<70> 1220. PRINT FNgo$(6);............. :'CURSOR AUF FE
     LD 6 SETZEN
<96> 1230 '
<41> 1240. x$=INPUT$(1)................ :'ABFRAGE OB NO
     CH EINE BERECHNUNG
< 3> 1250 '
<67> 1260. FOR I=1 TO 4................ :'DIE DATENFELD
     ER WERDEN GELöSCHT, DAMIT
<92> 1270... dat$(I)=""................ :'SIE NICHT NOC
     H EINMAL ANGEZEIGT WERDEN
<38> 1280. NEXT I
<15> 1290 '
<92> 1300.............................. :'AUSGABEFELDER
      LöSCHEN
<56> 1310. PRINT FNgo$(3);"....... ";
<97> 1320. PRINT FNgo$(5);"....... ";
<98> 1330 '
< 7> 1340. IF x$="j" THEN GOTO 1080 ELSE END
<69> 1350. END
```

DMV präsentiert:

Joyce-Programmsammlung Vol. 1

***Hochwertige Software** zu **Niedrigpreisen** finden Joyce-Anwender jetzt im Rahmen einer Programmsammlung in der Angebotspalette des DMV-Verlages.*

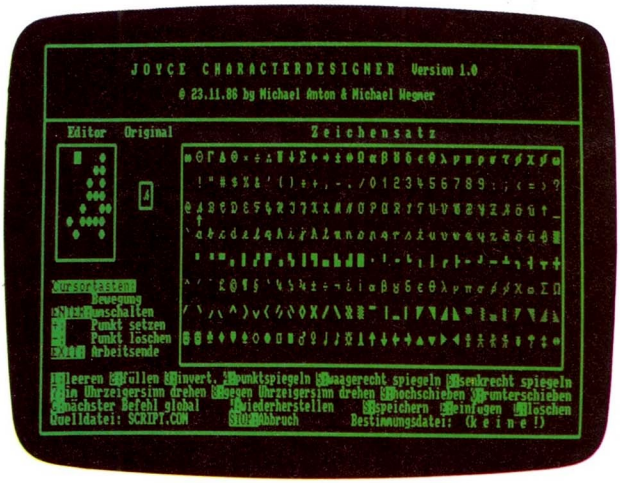
Jede Ausgabe aus dieser Reihe enthält eines oder zwei Programme, die aus verschiedenen Anwendungsgebieten kommen. Diese Serie erscheint in unregelmäßiger Reihenfolge und wird als komplettes Programmpaket mit 3"-Diskette und Bedienungsanleitung ausgeliefert.

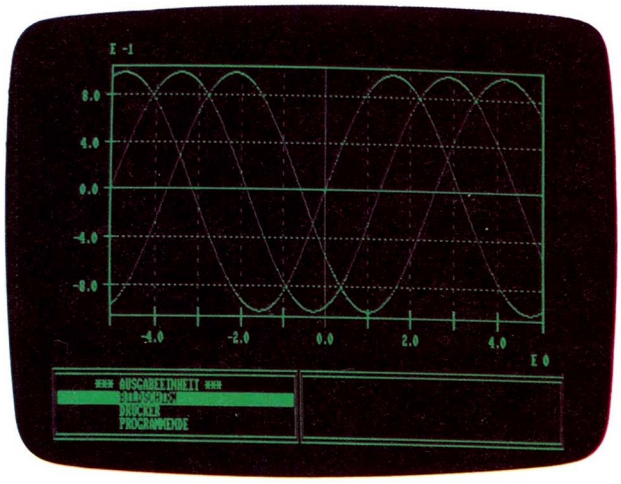
CHARACTER DESIGNER – DAS Werkzeug zur Erstellung eigener Zeichensätze und MGX-Mathematical Graphics Extension- ein Funktionenplotter mit außergewöhnlichen Leistungsmerkmalen.

1. Der Character Designer

Funktion: Komfortable Erstellung eigener Zeichensätze auf PCW 8256/8512 und deren Darstellung am Bildschirm! Ausdruck von ASCII-Files in diesem Zeichensatz unter CP/M Plus

Leistungsumfang: CD.COM ist der Character-Designer, der Editierung oder Neuerstellung von Zeichensätzen und deren Speicherung erlaubt. Zeichensätze können als COM-File abgespeichert werden, d.h. auch unter CP/M aufgerufen und somit aktiviert werden. **CD-PRINT** druckt vorformatierte ASCII-Texte auf dem Joyce-Drucker oder anderen Druckern in dem gewünschten Zeichensatz aus.
CRAZY, ORIGINAL, LOCOCHAR und **SCRIPT** sind mitgelieferte Zeichensätze. **SCRIPT** ist ein Schreibschrift-Zeichensatz und wurde freundlicherweise von der Firma Gerdes Imperial Software zur Verfügung gestellt. **SETUP.COM** erlaubt als Zugabe die Vorwahl einiger Systemparameter, z.B. die der seriellen Schnittstelle, der Tastaturgeschwindigkeit und der Floppy-Steprate.





2. MGX

Funktion: Graphische Darstellung von mathematischen Funktionen und beliebigen Meßreihen auf Bildschirm oder im Großformat auf dem Drucker

Leistungsumfang: Neben den arithmetischen Grundfunktionen sind auch weitere Funktionen darstellbar, die z.B. unter Mallard-Basic nicht zur Verfügung gestellt werden. Es können mehrere Funktionen und Meßreihen (diese wiederum mit mehreren Meßwerten gleichzeitig dargestellt werden). Die X-undYBereiche können manuell oder automatisch bestimmt werden, ebenso ist die Darstellung der Graphen mit **X/Y-Gitter, X oder Y-Gitter** oder ohne Gitter möglich. Es stehen **drei Auflösungsstufen** zur Verfügung, wobei mit Erhöhung der Auflösung auch die Arbeitsgeschwindigkeit abnimmt. Letztendlich kann noch eine **beliebige Überschrift** in die Funktion/Meßreihe eingefügt werden.
Bei der Druckausgabe steht als Standard das **A-4-Format** zur Verfügung; dieses ist jedoch in weiten Grenzen frei variierbar.

Die JOYCE Programmsammlung VOL.I ist ab 1. Mai beim Verlag zum Preis von DM 59,– erhältlich.

Nebenstehenden Coupon ausfüllen und senden an:

DMV-Verlag,
Redaktion PC International
Fuldaerstr. 6, 3440 Eschwege, Postfach 250

Name: _____

Anschrift: _____

Hiermit bestelle ich die Joyce-Programmsammlung Vol.1

☐ 3" Diskette 59,– DM
zuzüglich 3,– DM Porto/Verpackung
(im Ausland zuzüglich 5,– DM Porto/Verpackung)

☐ bitte liefern Sie per NN (nur innerhalb der BRD)

☐ V-Scheck liegt bei

Lieferung erfolgt ab 1. Mai 1987

Marktübersicht

Teil I Hardware und Zubehör
Teil II Anwenderprogramme
Teil III Sprachen
Teil IV Spiele

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Teil I			
Bildschirmfilter	59,–		13
Bildschirmfilter	79,–		16
Bildschirmfilter	89,–		3
Bildschirmfilter	98,–		4
CPS 8256	139,–	Parall./seriell. Port	29
Druckerverlängerungskabel	59,–		29
Echtzeituhrenmodul	138,–		18
Farbband	19,90		29
Grafpad 3	549,–	Grafik-Tablett	23
Harddisk 20 MB	Anfrage		16
Harddisk 10 MB	1.898,–		18
Harddisk 20 MB	2.498,–		
Joyce Mousepack	249,–	Maus u. Treibersoftware	24
Joystickadapter	98,–		18
Joystickadapter	158,–	mit 3 Kanal Sound Synthesizer	18
Light pen	278,–	Freihandzeichnen auf Monitor	18
Multicenter Turbo	Anfrage	Netzwerkzentrale	44
Papierführung	37,–		29
RAM-Aufrüstsatz auf 512 kB	79,–		33
RAM-Aufrüstsatz auf 512 kB	80,–		35
RAM-Aufrüstsatz auf 512 kB	99,–		16
RAM-Aufrüstsatz auf 512 kB	109,–		29
RAM-Aufrüstsatz auf 512 kB	119,–		22
RAM-Aufrüstsatz auf 512 kB	129,–		34
RAM-Aufrüstsatz auf 512 kB	148,–		3



Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Schneider Data SD 15	698,-	Typenraddrucker für Joyce (für DM 59,- Aufpreis komfortable Anpassung für Joyce)	17
Schneider Data SD 24	1298,–	24-Nadel-Drucker f. Joyce	17
Zweitfloppy FD2	548,–		19
Zweitfloppy FD2	564,–		16
Zweitfloppy FD2	579,–		13
Zweitfloppy FD2	628,–		35
Zweitfloppy FD2	648,–		22
Zweitfloppy FD2	698,–		3
ID-55	589,–	5,25"-Zweitlaufwerk	25
ID-35	489,–	3,5"-Zweitlaufwerk	25
Teil II			
Adressverwaltung	98,–		12
Adressverwaltung	129,–		7
Agenturverwaltung	620,–	Versicherungspaket	5

Bezeichnung	Preis in DM	Bemerkung	Bezugs-quelle
ALGO-Handwerk	998,–	Paket für Handwerker	46
ALGO-Haus	998,–	Hausverwaltung 46	
ALGO-Kart I	79,–	Dateiverwaltung	46
ALGO-Rhytmus I	149,–	Adressen-Briefverwaltung	46
Appartment/Hotel-Abrechng.	1.298,–		30
Arche	79,–	Dateiverwaltung	26
Astrologie	85,–		4
Auftrag	398,–	Auftragsbearbeitung	12
Auftragsbearbeitung	448,–		3
Bäcker-Software	349,–	Bäcker-Abrechnung	38
Bibliomat	185,–	Buchhandelspaket	5
Biorhythmus	45,–		4
BUSINESS-PACK	199,–	Adressverw., Faktura, Lagerverw.	55
Buchhaltung	577,–	Finanzbuchhaltung	7
Business-Star Plus	298,–	Auftragsbearbeitung	9
Carat-Auftrag	498,–	Auftragsbearbeitung	5
Carat-Kasse Plus	198,–	Kassenverwaltung	5
Carat-Lager Plus	288,–	Lagerverwaltung	26
Carat-Lager Plus	298,–	Lagerverwaltung	5
ComPack	798,–	Lager, Auftragsbearb., Faktura u. Fibu	37
Cracker	198,–	Tabellenkalkulation	17
Datamat	298,–	Dateiverwaltung	10
Datei Star 2.0	98,–	Dateiverwaltung	9
Datenbank	198,–		7
Datenrem	68,–	Dateiverwaltung	15
dB Faktur	798,–	Fakturierung (nur mit dBase)	39
dB Fibu	798,–	Fibu (nur mit dBase)	39
dB Kunden	398,–	Kundenverw. (nur mit dBase)	39
dB Lager	398,–	Lagerverw. (nur mit dBase)	39
dB Liquidat	798,–	Arzt-Abrechnung (nur mit dBase)	39
dB Triat	498,–	Triathlonauswert. (nur mit dBase)	39
dBase II	179,–	Dateiverwaltung	26
dBase II	189,–	Dateiverwaltung	11
dBase II	199,–	Dateiverwaltung	3
Dental	998,–	Zahnarzt-Abrechnung	2
DR Draw	179,–	Zeichenprogramm	26

Bezeichnung	Preis in DM	Bemerkung	Bezugs-quelle
DR Draw	199,–		3
DR Graph	179,–	Diagrammerstellung	26
DR Graph	199,–		3
Einn.-Überschuß-Rechnung	398,–		30
ELOCA	198,–	Kassenverwaltung	41
EASYFAKT	500,–	Fakturierung	47
EASYFIBU	500,–	Finanzbuchhaltung	47
Etatgraph	58,–	Haushaltsbuch mit Grafik	15
Faktura/Lager/Datei	178,–		7
Fakturem	98,–	Fakturierung	18
Fakturierung	94,–		33
Fibuking	136,–	Finanzbuchhaltung	15
Fibu-Soft	198,–	Finanzbuchhaltung	2
Fibu Star plus	298,–	Finanzbuchhaltung	9
Finanzbuchhaltung	194,–		33
Finanzbuchhaltung	278,–		12
Finanzbuchhaltung	499,–		42
Finanzmathematik	95,–		7
Gehirn-Trust	198,–	elektr. Kurzzeitsgedächtnis	51
Hausverwaltung	570,–	Paket für Hauswirte	7
IDA komplett	649,–	Faktura, Fibu, Statistik	25
Immobilienvermittlung	498,–		48
JOYCE-Administration	857,–	Fibu, Faktura, Dateiverw.	51
Konkursverwaltung	1200,–	Anwaltspaket	5
Kunden-Faktura-Lager	298,–		54
Leistungsstatistik	Anfrage		43
Loco 15	59,–	ASCII-Umwandlung für SD 15	17
Loco 24	Anfrage	dto. für SD 24	17
LocoScript-Einführung	29,50	Diskette u. Buch	37
Mercator	399,–	Finanzbuchaltung	40
MICA	198,–	CAD	1
Multi-Database & Toolkit	49,–	Dateiverwaltung	31
Multiplan	179,–	Tabellenkalkulation	26
Multiplan	189,–	Tabellenkalkulation	11
Multiplan	199,–	Tabellenkalkulation	3

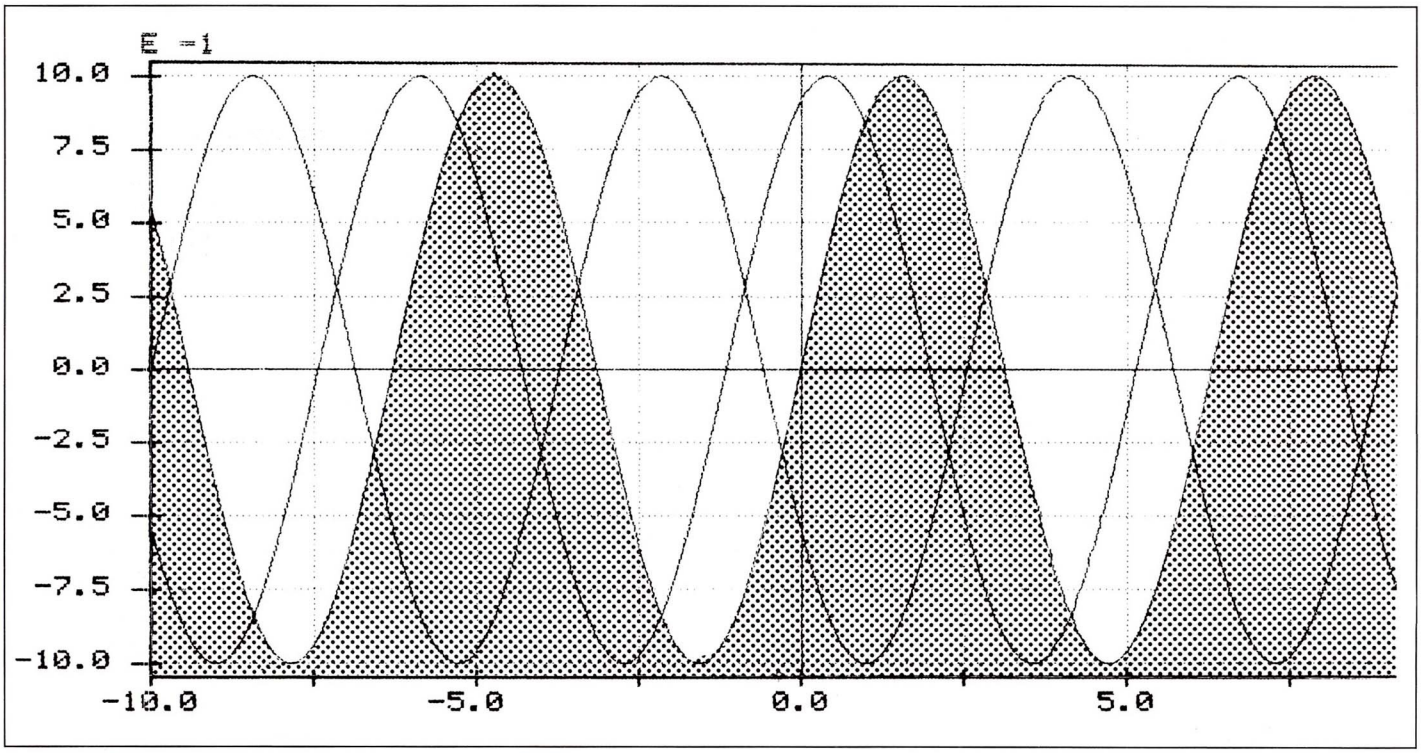
Bezeichnung	Preis in DM	Bemerkung	Bezugs-quelle
Orgafibu	274,–	Fibu 50 Konten	50
Orgafibu	297,–	Fibu 100 Konten	50
Orgafakt	548,–	Fakt./Adressverw./Mahnw.	50
Polyfont	75,–	Schriftmuster für Polyprint	45
Polymail	175,–	Serienbriefe/Adressverw. für Polyprint	45
Polyplot	175,–	Geschäftsgraphiken	45
Polyprint	175,–	Textverarbeitung	45
Praktische Textverarbeitung	89,–	Ergänzung Locoscript	21
Privatliquidation	Anfrage	Arzt-Abrechnung	43
Profirem	178,–	Faktura, Lager u. Adressen	15
Prompt	69,–	Dateiverwaltung	18
Prompt-Druck	39,–	Ergänzung Locoscript	18
Quick-Bill 2.0	168,–	Angebotsrechnung	16
Quick-Calc 2.1	148,–	Finanzbuchhaltung	16
Quick-Cine 2.0	98,–	Videoarchiv	16
Quick-Data 3.0	228,–	GSX-Geschäftssoftware	16
Quick-Mahn 3.0	68,–	Mahn-Modul	16
Quick-Name 2.0	128,–	Adressverwaltung	16
Quick-Stat 3.0	78,–	Statistik-Modul	16
Quick-Zahl 3.0	119,–	Zahlkarten-Modul	16
RH-Dat	89,–	Adressverwaltung	1
Salon	398,–	Friseurpaket	5
Schneider-Soft	1990,–	Schreiner-Abrechnung	2
Sofort-Druck	Anfrage	Schreibmaschinenmodus für Joyce	17
Standardleistungs-buch für Bauwesen	Anfrage		13
Star Base	198,–	Dateiverwaltung	9
Star Mail 2.0	98,–	Ergänzung Locoscript	9
SuperCalc2	198,–	Kalkulationsprogramm englische Anleitung	37
Super Copy	79,–	Kopierprogramm	18
Tasword 8000	139,90	Textverarbeitung	56
Turbo-Adress	149,–	Aressverwaltung	1
Turbo-Adress	169,–	Adressverwaltung	18
Turbo Faktum	199,–	Faktura	18
Universelle Büro	199,–	Büroverwaltung	52
varDAT II	198,–	Benutzeroberfläche für dBase II	53
Vereinsverwaltung	198,–	nur Joyce+	1
Vereinsverwaltung	248,–		3
Vereinsverwaltung	249,–		19
Verbentrainer	59,–		18
Videoarchiv verwaltung	98,–		7
Vokabeltrainer	59,–		26
Vokabeltrainer	69,–		18
Vokabi	58,–	Vokabeltrainer	15
Wärmebedarfsrechng.	248,–		7
Wärmebedarfsrechng.	298,–		8
Window Adresse	98,–	Adressverwaltung	6
Window Kasse	98,–	Kassenverwaltung	6
Window Lager	198,–	Lagerverwaltung	6
Wordstar	179,–	Textverarbeitung	26
Wordstar	189,–	Textverarbeitung	11
Wordstar	199,–	Textverarbeitung	3
Teil III			
CBasic Compiler	169,–		11
CBasic Compiler	174,–		33
COM	139,–	Kommunikationsprogramm	49
COMAL-80	148,–		36
FORTH	139,–	Forth-Compiler	49
Mac Lin	350,–	Programmkerngenerator	22
MACRO	139,–	Makroassembler	49
MI-C	445,–	mehr. Versionen	14
Microsoft-Basic	98,–		30
Nevada Cobol	168,–		19
Nevada-Cobol-Compiler	135,–		19
Nevada Fortran	168,–		19
Nevada-Fortran-Compiler	135,–		19
Oxford Pascal	89,–		20
PROFI	139,–	Assemlerprogrammsammlung	49
Small C	99,–	Entwicklungssystem	33
Turbo-Lader Business	148,–		33

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Turbo-Lader Grundpaket	138,–		33
Turbo-Lader Science	189,–		33
Turbo Pascal	219,–	ohne Grafik	11
Turbo Pascal	225,–	ohne Grafik	3
Turbo Pascal	279,–	mit Grafik	11
Turbo Pascal	285,–	mit Grafik	18
Turbo-Toolbox	125,–		18
Turbo-Toolbox	225,–		33
Turbo-Tutor	104,–	englisch/deutsch	33
Teil IV			
Aftershock	69,90	Adventure	18
Annals of Rome	69,90	Strategiespiel	32
Batman	von 39,–bis 59,–	Action-Adventure	18,32
Bounder	39,90	Action-Spiel	32
Cyrus Chess	von 49,90 bis 69,–	Schachprogramm	18,32
Cyrus II-3D	128,–	Schachproramm	37
3D Clock Chess	von 49,– bis 79,90	Schachprogramm	3,18,20
Fairlight	59,–	Action-Adventure	18
Football Fortunes	59,90	Strategiespiel	56
Frank Bruno's Boxing	56,90	Action-Spiel	56
Hitchhikers	99,–	Textadventure	18
Infocom-Adventures	je 79,90		56
Lord of the Rings	99,–	Adventure	18
Lotto 6 aus 49	59,–		4
Pawn	69,90	Grafikadventure	32
Ping Pong	149,90		55
S.A.D.Raid	59,–	Action	18
Space Invaders	49,90	Action	32
Strike Force Harrier	von 54,90 bis 69,90		20,32
Super-Lotto RUN 49	99,–		31
Trivial Pursuit	59,90	Frage/Antwortspiel	56

Wir haben uns bemüht, diese Marktübersicht aktuell und vollständig zu gestalten. Dennoch ist es möglich, daß inzwischen Preisänderungen bzw. Aktualisierungen aufgetreten sind. Bei Produkten mit Mehrfachnennungen sind Preisvergleiche möglich. *Die Redaktion*

Adressen für Joyce Soft- und Hardware

1. GfA Systemtechnik	Heerdter Sandberg 30	4000 Düsseldorf 11
2. M. Kempenich	Zetastr. 13	6220 Rüdesheim
3. Schauties	Bachstr. 52	7980 Ravensburg
4. mükra	Schöneberger Str. 5	1000 Berlin 42
5. Carat-Soft	Nelkenstr. 27	4600 Dortmund 41
6. Micro Market Worms	Joh.-Westermann-Platz 1	4780 Lippstadt
7. Computer Studio	am Rätschenbach 9	8058 Erding
8. WHS-Soft	Hohenzollernstr. 9	7447 Aichtal
9. Star Division	Uelzener Str. 12	2120 Lüneburg
10. Data Becker	Merowinger Str. 30	4000 Düsseldorf 1
11. Waldeck	Tulpenstr. 30	2870 Delmenhorst
12. Byte me	Wilhelmstr. 7	5240 Betzdorf
13. G+W	Hartmeyerstr. 50	7400 Tübingen
14. H. Rose EDV	Bogenstr. 32	4390 Gladbeck
15. Van der Zalm	Schieferstätte	2949 Wangerland
16. Werder	Bramfelder Ch. 215	2000 Hamburg 71
17. Schneider Data	Rindermarkt 8	8050 Freising
18. Weeske	Potsdamer Ring	7150 Backnang
19. Computer Max	Postfach 1461	7920 Heidenheim
20. M. Naujoks	Rottmannstr. 40	6900 Heidelberg
21. DMV-Verlag	Fuldaer Str. 6	3440 Eschwege
22. SFK	Delsterner Str. 23	5800 Hagen
23. PiZie Data	Mittelstr. 61	4322 Sprockhövel
24. Gerdes	Heidegartenstr. 36	5300 Bonn 1
25. Innowave Data	Kurt-Schuhmacher Str.26	3000 Hannover
26. TG-Soft	Offersdorf 5	8491 Rimbach
27. Boll Computer- und Software-Vertrieb	Wilhelmsthaler Str. 26 B	3501 Ahnatal-Weimar
28. Syndrom Computer GmbH	Edwaldstr. 181	4352 Herten
29. Vortex-Versand	Falterstr.	7101 Flein
30. ESCHCOMP	Postfach	8391 Pelesrent
31. INTERLOGIK-Peter Herzog	Grabenweg 30	8240 Schönau am Königssee
32. Waldeck-Software	Tulpenstr. 30	2870 Delmenhorst
33. R. Schuster Electronic	Obere Münsterstr. 33-35	4620 Castrop-Rauxel
34. ABD Computer	Zettachring 12	7000 Stuttgart 80
35. G Computerstore	Hochstr. 11	8500 Nürnberg
36. D-Belz		2270 Utersum
37. Schneider AG	Postfach 120	8939 Türkheim
38. ABS COMPU-TERLAND	Huglfinger Str.4	8121 Polling
39. Computerladen Braun	Kreuzstr. 9	8940 Memmingen
40. COMPUTER-STUDIO-OFFENBACH	Kl. Biergrund 17	6050 Offenbach
41. COMPUTER TEAM CUXHAVEN	Am Hauptbahnhof	2190 Cuxhaven
42. DAST-Mobil	Saarbrückerstr. 56	4670 Lünen
43. Effekta Chr. Walls	Am Wiggert 9c	4500 Osnabrück
44. Answer Management Consulting GmbH	Postfach	7000 Stuttgart
45. Franzis-Software	Karlstr. 37-41	8000 München
46. A.H.W. Goscg OHG	Grindelallee 138	2000 Hamburg 13
47. Gruse Elektronik	Stieglitzweg 1	3300 Braunschweig
48. Hailer & Gensberger	Albstr. 18	7317 Wendlingen
49. Holtkötter	Albert-Schweitzer-Ring 15	2000 Hamburg 70
50. ORGASOFT	Graneggstraße 43	7732 Niedereschach
51. Richter's	Thürmchenswall 3	5000 Köln
52. Schmidt-Soft	Blumenstr. 15	5600 Wuppertal 1
53. SOFTDESIGN	Bleichstr. 25	4040 Neuss
54. Vertretung der Industrie	Brachvogelweg 9	8037 Olching
55. ZS-Soft	Postfach 2361	8240 Berchtesgaden
56. Sunshine Software	Auf dem Schilde 14	5270 Gummersbach



Text zum Programmpaket »Funktionsanalyse«

Programm: »Diffint«

1) ALLGEMEINES

Jeder, der das Gymnasium hinter sich gebracht oder das Abitur vor sich hat, kann sich unter den Begriffen Differential- bzw. Integralrechnung etwas vorstellen.
Einige rümpfen die Nase, wenn sie diese Wörter nur hören; andere sind ganz wild darauf, möglichst komplizierte Funktionsverläufe zu differenzieren oder zu integrieren.
Dieser Aufsatz und das hier vorgestellte Programm soll weder für diejenigen sein, die mit Mathematik »überhaupt nichts am Hut haben«, (die Computerei möglicherweise als Selbstzweck betrachten und lieber »Rambo im Computerurwald wild umherschießen lassen«), noch soll es für Mathematik-Freaks sein, die diese Wissenschaft lediglich in einer abgehoben theoretischen Weise betreiben.
Vielmehr sind alle die angesprochen, die über ein gesundes Fundament an mathematischem Basiswissen verfügen, um in der schulischen- oder beruflichen Alltagspraxis vorkommende Problemstellungen zu lösen. Am häufigsten kommen diese selbstverständlich in Naturwissenschaft und Technik vor; allerdings dient praktische Mathematik natürlich auch in anderen Bereichen, wie z.B. im kaufmännischen Sektor dazu, Probleme stärker zu systematisieren und damit vernünftige, logische Lösungen zu finden.
Das Gebiet der sog. praktischen Mathematik befaßt sich in erster Linie mit funktionalen Zusammenhängen. Als Beispiel hierfür denke man nur an irgendeine Reihe dutzender Meßwerte, die als Tabelle nur wenig aussagen, die jedoch als Funktion dargestellt, (evtl. sogar graphisch), eine enorme Aussagekraft besitzen und daher Bewertungen oder Prognosen erleichtern.
In der Praxis liegen oft lediglich solche Meßwerte vor; die zugehörige Funktion ist unbekannt. Sie zu bestimmen ist übrigens ebenfalls Aufgabe praktischer Mathematik, soll aber hier nicht Gegenstand der Ausführungen sein.
Da sich die wenigsten Vorgänge in der Natur durch einfache, triviale Funktionen in der Form $y=f(x)$ beschreiben lassen, weil häufig viele Parameter anstatt konstant, variabel sind, stößt man in allen Gebieten der Naturwissenschaften oder der Technik auf Differentialgleichungen.
Diese werden, sofern möglich, mit Hilfe sog. Randbedingungen gelöst; es haben sich verschiedenste Lösungsalgorithmen je nach Art des Problems bewährt. Wird nicht gerade die Laplace-Transformation oder ein für viele ähnlich »exotisches« Verfahren benutzt, kommt man nicht umhin, Funktionen zu differenzieren bzw. zu integrieren.
Durch Differentiation bestimmt man Gradienten, d.h. Änderungen einer Größe in bestimmten Punkten im Funktionsverlauf. So gibt z.B. die erste Ableitung einer Funktion die Steigung der Tangente in einem gewählten Punkt des Graphen an; die zweite Ableitung die Änderung der Steigung, d.h. die Krümmung.
Mit Hilfe der Differentialrechnung werden beispielsweise Kurvenverläufe auf Maxima- oder Minimawerte untersucht.
Die Integralrechnung in ihrer einfachsten Form befaßt sich dagegen mit der Berechnung von Flächen zwischen Funktionsverläufen und Abszisse in gewählten oder durch die Art des Problems vorgeschriebenen Grenzen. So kann die Integration einer Kurve eine »völlig neue« physikalische Größe ergeben. Ein anschauliches Beispiel hierfür ist das

aus der Physik bekannte Geschwindigkeits-Zeit-Diagramm. Ein Beispiel aus der Technik liefert eine einfache Feder. Drückt man die Feder um einen Weg s zusammen, so wirkt auf die Hand die Federkraft F, und zwar je mehr, umso stärker man drückt. Zusammenfassend dienen also Differential- und Integralrechnung in der Praxis meist dazu:

1) Differentialgleichungen zu lösen.
2) Markante Kurvenpunkte zu ermitteln (Minima, Maxima, Wendepunkte, Sattelpunkte, etc.), mit anderen Worten: Stichwort Kurvendiskussion.
3) Flächen unter Kurven zu bestimmen.

Die Differentiation einer Funktion wird in der Regel stets in einem Punkt ausgeführt; die Funktionsintegration wird entweder unbestimmt durchgeführt (zur Lösung benötigt man Randbedingungen), oder bestimmt mit Integrationsgrenzen. Diese Bildung bestimmter Integrale ist auch Grundlage der nachfolgenden Programmbeschreibung. Man wählt hierbei auf der Abszisse einen Minimum- und einen Maximumwert (Integrationsgrenzen), denkt sich in diesen beiden Punkten je eine vertikale Gerade bis zum Kurvenverlauf und berechnet durch Integration die hierdurch begrenzte Fläche A.
Das vorliegende Programm »Diffint« ist dazu geeignet, sowohl Differentiale, als auch Integrale zu bilden. Besonderheit ist dabei, daß die abzuleitende oder zu integrierende Funktion zwar bekannt sein kann, aber nicht muß. Liegen z.B. nur einige Meßpunkte vor, so reichen diese für die Berechnungen aus.

2) GRUNDLAGEN

Um zu verstehen, was sich im Programm »Diffint« abspielt, benötigt man einige Grundlagen der numerischen Mathematik.
Es ist dem Rechner nicht möglich, Ableitungen oder Integrale analytisch zu berechnen, üblicherweise werden daher numerische Näherungsverfahren eingesetzt, die mit den analytischen Ergebnissen mehr oder weniger genau übereinstimmen.
Das Programm besteht aus zwei Hauptteilen, nämlich Integral- und Differentialteil.
Beim Integralteil findet die »Simpson-Regel« Anwendung, die in jeder besseren Formelsammlung nachzuschlagen ist. Sie liefert genauere Ergebnisse als beispielsweise Trapez- oder Rechteckregel und wird daher auch am häufigsten eingesetzt.
Voraussetzungen für das Programm sind, wie bereits erwähnt, entweder die Funktion selbst, die zu integrieren ist, (mit Angabe der Intervallgrenzen), oder einzelne Meßpunkte, auch Stützstellen genannt, die allerdings äqidistant sein müssen; d.h. der Abstand h zweier benachbarter Meßpunkte muß über den gesamten Integrationsbereich gleich sein. Ist dieses erfüllt, gilt die »Simpson-Regel« [1]:

$$A = h*[y(0)+y(n)+2*(y(2)+y(4)+\ldots+y(n-2))+4*(y(1)+y(3)+\ldots+y(n-1))]\ /3$$

Hierbei bedeuten:

A	: Fläche unter der Kurve (Integration).
h	: Äquidistanter Abstand der x-Koordinaten zweier benachbarter Stützpunkte.
0 bis n	: Indizes der y-Koordinaten der Stützpunkte (y(0) ist der kleinste, y(n) der größte Wert).
y(0) −y(n)	: y-Koordinaten der Stützpunkte.

Wie man sieht, benötigt man nicht die x-Werte der Stützpunkte; die y-Werte, die Integrationsgrenzen und die Angabe des Abstandes h reichen zur Bestimmung des Integrals A aus. Voraussetzung ist eine ungerade Anzahl von Stützpunkten; (diese bewirkt eine Unterteilung des Integrationsbereiches (=Bereich zwischen y(0) und y(n)) in eine gerade Anzahl von Subintervallen.

Der Differentialteil basiert auf den Formeln von [2]. Sind mindestens drei (äquidistante!) Stützstellen bekannt, so können in diesen Punkten Ableitungen gebildet werden. Die Berechnung von Ableitungen im letzten eingegebenen Punkt y(n) ist allerdings nur möglich, wenn mindestens noch die darauffolgende Stelle y(n+1) bekannt ist.
Je mehr Stützstellen eingegeben werden, desto genauer erfolgt die Berechnung der Ableitungen und desto größer ist der berechenbare Ableitungsgrad.

Soll beispielsweise der erste von drei eingegebenen Werten differenziert werden, so kann lediglich die erste Ableitung gebildet werden; sind dagegen die nachfolgenden vier Stellen bekannt, so rechnet das Programm bis hin zur dritten Ableitung.

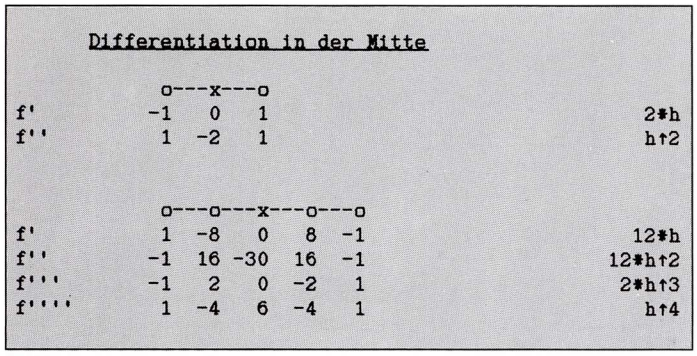
Man unterscheidet prinzipiell zwei Fälle. Zum einen den, daß bei einem Kollektiv von, (immmer äquidistanten!), Stützstellen die erste auf ihre Ableitungen hin untersucht werden soll; zum anderen den, daß dieses für den Wert in der Mitte des Kollektivs durchgeführt werden soll. Bei Letzterem muß die Anzahl der Stützstellen ungerade sein. Aus Tab. 1 kann man hierzu die Einzelheiten entnehmen.

Danach ist der y-Wert einer jeden Stützstelle mit einem Faktor zu multiplizieren, (der jeweils unter der Stelle steht); für jede Stelle erhält man somit ein Produkt. Die Summe aller dieser Produkte wird durch ein Vielfaches bzw. Potenzen des Abstandes h dividiert. Die Werte stehen in Tab.1 in jeder Zeile ganz rechts. Das Ergebnis dieser Division ergibt die jeweilige Ableitung f', f'', etc. im betrachteten Punkt.

Differentiation im ersten Wert

```
                x---o---o
f'             -3   4  -1                                  2*h

                x---o---o---o
f'            -11  18  -9   2                              6*h
f''             2  -5   4  -1                              h↑2

                x---o---o---o---o
f'            -25  48 -36  16  -3                          12*h
f''            35 -104 114 -56 11                          12*h↑2
f'''           -5  18 -24  14  -3                           2*h↑3
```

Differentation im ersten Wert



Differentiation in der Mitte

```
          o---x---o
f'       -1   0   1                              2*h
f''       1  -2   1                              h↑2


          o---o---x---o---o
f'        1  -8   0   8  -1                      12*h
f''      -1  16 -30  16  -1                     12*h↑2
f'''     -1   2   0  -2   1                      2*h↑3
f''''     1  -4   6  -4   1                       h↑4
```

Differentiation in der Mitte

Anmerkungen

h=Abstand der x-Koordinaten zweier benachbarter Stützstellen (muß zwischen allen Stellen gleich sein!). Mit x ist derjenige Stützwert dargestellt, in dem abgeleitet wird.

Beispiel 1:
Sind drei Stellen bekannt, so sieht die Berechnungsformel für die erste Ableitung in der ersten Stelle wie folgt aus:

f'=[-3*y(i) + 4*y(i+h) − y(i+2*h)]=2*h

Beispiel 2:
Sind fünf Stellen bekannt, und soll die dritte Ableitung in der mittleren Stelle berechnet werden, so ergibt sich folgende Formel:

f'''=[−y(i−2*h) + 2*y(i−h) + 0*y(i) − 2*y(i+h) + y(i+2h)]=2*h↑3

3) BEDIENUNG DES PROGRAMMES

Nach Start des Programmes wird gefragt, ob man differenzieren oder integrieren möchte. Egal, wofür man sich entscheidet, es erscheint nun die Frage, ob eine Funktion vorliegt oder die Berechnung über bekannte Stützstellen erfolgen soll.

Beim Vorhandensein einer Funktion wird sowohl im Differentialteil als auch im Integralteil von "Diffint" darauf hingewiesen, daß zunächst diese Funktion im Programm einzugeben ist (die entsprechende Zeilennummer ist dem Bildschirmtext zu entnehmen).

Hierbei ist der Ablauf des Programmes zu unterbrechen und mittels edit (Zeilennummer)- Kommando die entsprechende Zeile um die Funktion zu ergänzen (y(i)=, steht bereits dort, so daß nur die rechte Gleichungsseite hinzugefügt werden muß). Nach Betätigen der RETURN-Taste ist nun erneut zu starten. Es ist sinnvoll, die Eingabe der Funktion bereits vor dem Starten vorzunehmen; man spart sich dann das Unterbrechen.

Beim Differentialteil des Programmes wird nun nach dem x-Wert der Stelle gefragt, an der differenziert werden soll. Nach deren Eingabe erfolgt die Ausgabe der ersten bis vierten Ableitung in der Form:

y'(x)=Ergebnis
↑
Stelle

Ist keine Funktion bekannt, so erfolgt die Berechnung stattdessen über bekannte Stützstellen. Deren Anzahl ist einzugeben, ebenso der Abstand der x-Werte zweier benachbarter Stellen. Da diese äquidistant sein müssen, ist diese Schrittweite zwischen allen Werten stets gleich.

Als nächstes sind die y-Werte der Stützpunkte einzugeben. Diese werden indiziert in der Form:

y(i)=Wert

Es muß bei dem Wert mit der Eingabe begonnen werden, zu dem der betragsmäßig kleinste x-Wert gehört; dann wird der y-Wert eingegeben, zu dem der nächstgrößere x-Wert gehört usw. Entscheidend für die Reihenfolge der Eingabe sind also die jeweiligen Zahlenwerte der x-Koordinaten der Stützstellen, nicht deren Vorzeichen!

Die letzte notwendige Eingabe bezieht sich auf die Stelle an der abgeleitet werden soll. Hier ist der entsprechende Wert für i anzugeben; im Ergebnis erscheint dann die Stelle i in der Klammer (z.B. y''(i)=Wert). Dieser Wert in der Klammer ist also nicht die x-Koordinate der gewählten Stelle, sondern kennzeichnet die Stelle durch den entsprechenden i-Wert.

Im letzten eingegebenen Punkt kann nicht differenziert werden, es ist hierfür der nächstfolgende nötig. Dieses wird durch eine entsprechende Fehlermeldung berücksichtigt; ebenso, wenn an einer Stelle abgeleitet werden soll, die nicht eingegeben wurde.

Beim Integralteil von »Diffint« wird ganz analog die Frage nach der Funktion gestellt. Auch hier ist das Programm zu deren Eingabe zu unterbrechen, es sei denn, die Funktion wurde bereits vor dem Start eingegeben. Danach sind die x-Werte für die obere und untere Integrationsgrenze einzugeben; daraufhin erscheint das Ergebnis.

Es ist übrigens ohne Probleme möglich, als untere Grenze eine negative, und als obere Grenze eine positive Zahl einzugeben. Das Programm berechnet in diesem Fall zuerst die Integration im positiven Bereich (mit Null als unterer Grenze), dann im negativen und addiert beide Teilergebnisse.

Ist die Funktion unbekannt, müssen auch hier die y(i)-Werte für die Stützstellen eingegeben werden. Ihre Anzahl muß ungerade sein! Zuvor sind noch die Schrittweite zwischen zwei Stützstellen und deren Anzahl einzugeben. Bei Eingabe der y(i)-Werte muß analog zu oben Erwähntem mit demjenigen begonnen werden, dem der betragsmäßig kleinste x-Wert zugeordnet ist. Ausgegeben wird schließlich die Integration der Stützpunkte zwischen der ersten (y(i)) und der letzten (y(n)) eingegebenen Stützstelle (Integrationsgrenzen) und der Abszisse. In dem Fall, daß die x-Werte einiger Stützpunkte im negativen, andere im positiven Bereich liegen, müssen zwei Rechengänge ausgeführt werden. Man integriert zunächst die Stellen mit positiven x-Werten bis Null und merkt sich das Ergebnis; danach die mit negativen x-Werten und addiert schließlich beide Teilresultate.

JÜRGEN SIEBERT

PRAKTISCHE TEXTVERARBEITUNG MIT JOYCE



DMV

Bunt gemischtes
Anwender-Paket für
den Joyce-Texter:
Buch + Diskette
mit Wort-Experimenten,
Tips und Tricks, einem Layout-
Archiv, dem LocoScript-Software-
Training, literarischen Text-Beispie-
len, »historischen Simulationen, Insider-
Plaudereien und feuilletonistischen Ab-
wegen«, theoretischen Notizen und vie-
les mehr.

Ein Wegweiser durch die Welt der Text-
verarbeitung.

Abwechslungsreicher Lesestoff für Neu-
linge und Fortgeschrittene, der über die
Grenzen des Computerschreibtischs hin-
ausführt.

Auf Diskette:

1. Der **»LocoSript-Zettelkasten«** bietet direkten Zugriff auf zahlreiche LocoSript-Standard-Schablonen in über 50 Dateien.
   Dazu: Mustertexte, Editierübungen, Schriftbeispiele, Serienbriefe, Spaltendruck, Tabellen, Telefon-register, typografische Experimente, Text-Kostproben, und vieles mehr.
2. **WordStar-Tastendefinitionsdatei** mit Referenzkarte, WS-Stapeldatei für Startdiskette (beides individuell anzu-passen).
3. **LOGO- und BASIC-Programme** zur Erstellung computergenerierter Sätze und Gedichte.



Buch + Diskette
für LocoScript und WordStar-Autoren

210 Seiten, Einband: Leinen-Hardcover
Buch und 3''-Diskette zum Preis von 89,– DM
(unverbindliche Preisempfehlung)

Zu beziehen über den Computerhandel und den guten Fachbuchhandel
oder direkt beim Verlag. Händleranfragen erwünscht.

DMV Verlag · Fuldaer Straße 6 · Postfach 250 · 3440 Eschwege

— Bitte Bestellkarte benutzen —

Wichtig

Sowohl bei der Differentiation als auch bei der Integration ist zu empfehlen, sich ein Bild von der zu bearbeitenden Funktion zu machen, bzw. sich die Lage der Stützpunkte im Koordinatensystem zu veranschaulichen. In letzterem Fall sind eventuelle Ausreißerpunkte zu eliminieren, da sie die Genauigkeit der Berechnungen z.T. stark beeinflussen können. Bei starken Oszillationen der Meßpunkte ist intervallweises Vorwärtsarbeiten im Sinne genauerer Ergebnisse empfehlenswert. Beim Bearbeiten von Funktionen führen mathematisch falsche Eingaben zu falschen Ergebnissen; so dürfen beispielsweise bei der Funktion y=ln(x) keine negativen x-Werte oder Null eingegeben werden. Ebenso sollten große (abhängig von der Funktion bzw. vom Verlauf der Stützwerte) Eingabewerte vermieden werden. Die im Programm verwendeten numerischen Beziehungen (Tab. 1) stellen Näherungen dar, die vom exakten analytischen Ergebnis mehr oder weniger abweichen; dieses ist stets zu bedenken, insbesondere bei komplizierten Kurvenverläufen. Prinzipiell gilt, je kleiner die eingegebenen Zahlen sind, desto genauer ist das Ergebnis. Auch die im Computer entstehenden Ungenauigkeiten durch die Dezimal-Binärumwandlung fallen bei großen Werten stärker ins Gewicht.

Welche der in Tab. 1 aufgeführten Differentiationsformeln zur Anwendung kommt, hängt von der Anzahl der eingegebenen Stützwerte ab, und der Lage des Punktes, in dem abgeleitet werden soll. Das Programm wählt in jedem Einzelfall selbständig die Formel aus, die die meisten Ableitungen bilden kann bzw. die höchste Genauigkeit besitzt. Bei Eingabe einer Funktion wird die letzte in der Tabelle aufgeführte Formel angewendet.

Differentiationsbeispiel

Gegeben ist die Funktion y=3*x↑2. Eingabe dieser Funktion sowie der Intervallgrenzen -2 und 3 führt bezüglich der Stelle x=1 zu folgenden Ergebnissen:

f'(x=1)=f'(i=3)=6; f''(1)=6; f'''(1) = f''''(1)=0

Eingabe der Stützstellen: y=3; y=0; y=3; y=12; y=27 ergibt bei einer Schrittweite von 1 für die Ableitungen im dritten Punkt (x=1; y=3; i=3) dasselbe Resultat.

Zur Überprüfung: Analytische Rechnung liefert:

f'(1)=6*x=6 ; f''(1)=6 ; f'''(1) = f''''(1)= 0

Die numerische Lösung entspricht also der analytischen.

Integrationsbeispiel

Die einfache Funktion y=x soll in den Grenzen von x=-2 bis x=4 integriert werden (Schrittweite ist wieder 1). Nach Eingabe der Funktion ergibt sich als Lösung für die Fläche unter der Kurve: 10

Ist die Funktion unbekannt, erfolgt die Eingabe der Stützstellen:

y=-2; y=-1; y=0; y=1; y=2; y=3; y=4,
(Schrittweite=1).

Hier tritt nun der Fall auf, daß die x-Werte einiger Stützstellen positiv, die anderer negativ sind. Daher gibt man zunächst von Null ausgehend, die, rechts von der Ordinate ein. Ergebnis der Integration ist 8. Danach führt man dieses für die Stellen, links der Ordinate, wieder mit Null beginnend durch. Ergebnis: 2. Addition beider Teilergebnisse liefert schließlich das Endresultat von 10, welches sich durch die analytische Überprüfung bestätigt.

Auch hier stimmen also numerische und analytische Lösung überein.

4) PROGRAMMBESCHREIBUNG

»Diffint« teilt sich in zwei Hauptteile auf: Differentialteil (bis Zeile 1150) und Integralteil (Z. 1160 bis Ende). Es soll hier nicht das Programm bis ins letzte Detail erörtert werden; vielmehr möchte ich neben einem groben Überblick auf Besonderheiten oder wichtige Schritte eingehen.

Differentialteil:

Zeile 200 enthält die linke Seite der Funktion, deren rechte Seite dort eingegeben wird. Ausgehend von dieser Funktion berechnet das Programm fünf Stützstellen (Z. 140-230) mit dem Ziel, in jedem Fall die letzte der in Tab. 1 aufgeführten Formeln anzuwenden. Dabei wird eine Schrittweite von nur 0.2 gewählt; dieses führt zu höherer Genauigkeit bei komplizierten (z.B. oszillierenden) Funktionen (Z. 140).

Ist die Funktion unbekannt, werden in Z. 260-420 die Stützwerte eingegeben. Die Zeilen 560-1050 beinhalten die Koeffizienten aus Tab. 1; die Entscheidungsprozedur in den Zeilen 390,400,440520 sorgt dafür, daß die eingegebenen Stützstellen der für sie »optimalen« Formel aus der Tabelle zugewiesen werden. »Optimal« heißt größtmögliche Genauigkeit bzw. Auswahl der Formel, mit der der höchste Ableitungsgrad errechnet werden kann. Sind z.B. acht Stützstellen bekannt, für deren dritte die Ableitungen berechnet werden sollen, so könnte man dies nach der vierten Formel aus Tab. 1 tun; man erhält dann Ableitungen bis zum zweiten Grad. Benutzt man stattdessen die dritte Formel, so erreicht man Ableitungen bis zum dritten Grad. Dieses und die für diesen Fall nötige Umindizierung (Z.520-550) wird, wie bereits erwähnt, automatisch von »Diffint« durchgeführt.

Die eigentliche Berechnungsformel, in die dann die gefundenen, optimalen Koeffizienten eingesetzt werden, befindet sich für die Differentiation am Anfang (s. Tab.1) in Z.790-850; für die Differentiation in der Mitte in Zeile 1060-1150.

Die auf drei Dezimalstellen gerundeten y(i)-Werte der Stützstellen werden mit Tausend multipliziert und dadurch in ganze Zahlen umgewandelt (Z.220,340,350). Nach der Berechnung wird dann wieder durch Tausend dividiert. Sinn dieser Umrechnung ist die Minimierung der Rechenvor-

gänge mit Dezimalzahlen und den damit verbundenen Ungenauigkeiten bei der Dezimal-Binärumwandlung im Rechner (je nach Wahl der Funktion oder der Stützwerte können sich hierdurch Fehler entwickeln, die nicht mehr vernachlässigt werden können).

Integralteil:

Ist die zu integrierende Funktion bekannt, wird in den Zeilen 1680-1790) die Berechnung der Koeffizienten durchgeführt, die in die »Simpson-Regel« (Z.1830) eingesetzt werden. Die Ausgabe des Ergebnisses erfolgt mit drei Nachkommastellen.

Die Anzahl der Intervalle, in die der Integrationsbereich unterteilt wird, ist abhängig von der Wahl der Integrationsgrenzen (Z. 14601470). In Z. 1330-1450 werden die Vorzeichen der eingegebenen Integrationsgrenzen erkannt. Sind diese ungleich, wird der gesamte Rechengang einmal für den positiven und einmal für den negativen Bereich ausgeführt und die Teilergebnisse addiert (Z.1390-1441, 1601, 1611, 1831-1851).

Sollen Stützstellen statt einer Funktion eingegeben werden, springt das Programm in Zeile 1900. Hier werden ebenfalls die Koeffizienten ermittelt, die dann in Z.1830 eingesetzt, die Fläche unter der Kurve ergeben.

B) Programm »Kurdis«

1) Allgemeines

Wie bereits am Anfang erwähnt, befasst sich »Kurdis« mit Kurvendiskussion. Voraussetzung ist eine bekannte Funktion, die in Programmzeile 170 einzusetzen ist. Im nachfolgenden Programmausdruck ist als Beispiel die Funktion y=sin(x) angegeben. In einem Intervall, dessen Grenzen während des Programmablaufes einzugeben sind, werden Nullstellen, Extrema und Wendepunkte der Funktion berechnet.

Die Kenntnis solcher markanter Kurvenpunkte ist beispielsweise in der technischen Praxis bei komplizierten Funktionen, deren Verläufe nicht ohne weiteres vorstellbar sind, von großer Bedeutung. Grobe Abschätzungen sind hierdurch bereits möglich, ohne den exakten Verlauf der Funktion in allen Details zu kennen.

Die Bedienung des Programmes ist sehr einfach und erklärt sich beim Laufenlassen von selbst. Auch das Programm selbst ist einfach gehalten; auf die detallierte Beschreibung der verschiedenen Iterationsprozesse soll hier verzichtet werden. Es sei vielmehr an die Initiative des Lesers appeliert, an einem verregneten Abend einmal das Programm »zu Fuß« durchzuspielen; für ein tieferes Verständnis die beste Methode.

Je nach eingegebener unterer Intervallgrenze fällt, von dieser Grenze ausgehend, die Funktion mit zunehmenden x-Werten ab oder steigt an. Daher wird ausgehend von Zeile 220 entweder nach Z. 380 oder nach Z. 610 gesprungen. Bei ungünstiger Wahl der Integrationsgrenzen, d.h. wenn in unmittelbarer Nähe der Grenzen, aber außerhalb des Integrationsbereichs makante Punkte liegen, werden diese noch mitangezeigt (Z. 500, 660).

Die Berechnung der Wendepunkte dauert in der Regel etwas länger, als die Bestimmung der Nullstellen bzw. Extrema, da die Anzahl der Iterationsschritte demgegenüber größer ist. Generell kann man das Verfahren zur Bestimmung der Nullstellen und Extrema als ein stückweises »Abtasten« der Funktion auffassen. Vorteil dieser Methode ist, daß auch relative Extrema mit auf dem Monitor erscheinen. Insbesondere bei komplizierten Funktionen (z.B. Schwingungsverläufen mit zahlreichen Oszillationen) ist die Kenntnis der relativen Extrema häufig wichtiger als die der absoluten Maxima bzw. Minima. Letztere können übrigens leicht aus sämtlichen auf dem Bildschirm befindlichen Extrema ersehen werden; sie besitzen ganz einfach den größten Betrag.

Bei der Bestimmung der Wendepunkte dagegen wird die »Krümmung« der Kurve verfolgt. Diese ist ja bekanntlich mittels zweiter Ableitung ausdrückbar und so wird festgestellt, an welchem Punkt eine positive Biegung in eine negative übergeht bzw. umgekehrt; diese Stelle ist dann ein Wendepunkt. Für die Bestimmung der zweiten Ableitung wird aus dem inzwischen mehrmals erwähnten, in Tab. 1 am Schluß stehenden Formelpaket die zweite benutzt.

Es sei nochmals darauf hingewiesen, daß die Berechnungen je nach der Wahl der Integrationsgrenzen unterschiedlich lange dauern können. Sie sind erst abgeschlossen, wenn in der linken unteren Ecke des »Joyce«-Monitors ein »Ok« erscheint.

Die Genauigkeit der Ergebnisse sind in der Praxis völlig ausreichend. Lediglich bei großen Zahlenwerten können sie in einigen Fällen um 0,5% bis höchstens ca. 1% neben den exakten Werten liegen. Es ist daher empfehlenswert, zu große Integrationsintervalle zu vermeiden. (Bei einer Sinusfunktion als Grenzen -800 und 1000 anzugeben, ist sicherlich nicht im Sinne schneller und genauer Rechnung und im übrigen auch in den seltensten Fällen praxisrelevant!)

C) Verbindung beider Programme

Die beiden oben erläuterten Programme können sowohl einzeln, voneinander völlig unabhängig, als auch verbunden eingesetzt werden.

Oben beschriebenes bezieht sich stets auf den Einzelbetrieb; beim Aneinanderhängen ist es sinnvoll, vor beide Programme einen einfachen Funktionsplotter zu stellen, der (bei bekannter Funktion) oft entscheidend die Vorstellung der Funktion erleichtert. Ob nun »Diffint« oder »Kurdis« zuerst bearbeitet werden soll, ist jedem selbst überlassen. Die Verknüpfung der Programme dürfte für jeden, der Basic auch nur zu 70% beherrscht, kein Thema sein.

Beide Programme zusammen decken einen weiten Bereich an Funktionsanalyse ab, und sind daher ein sinnvolles, wichtiges Arbeitsmittel für den naturwissenschaftlich interessierten Anwender.

D) Quellennachweis

[1] Giek: Techn. Formelsammlung, 27. Aufl., Giek-Verlag/Heilbronn

[2] Schefel: Einf. in numerische Mathematik

(R.Bourdon)

LISTING

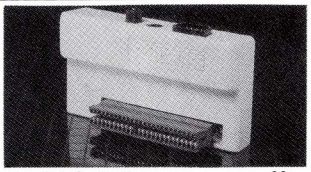
```
>DIFFINT <, REMARK = >REM<.

<71> 10 REM                          *** PROGRAMMNAME: "DIFFINT" *
<29> 20 PRINT:INPUT "MöCHTEN SIE DIFFERENZIEREN, SO GEB
     EN SIE EIN d EIN UND DRüCKEN RETURN ;.. MöCHTEN SI
     E INTEGRIEREN, SO GEBEN SIE EIN I EIN UND DRüCKEN
     RETURN :",Antwort$
<89> 30 IF Antwort$="i" THEN GOTO 1160
<98> 40 IF Antwort$="d" THEN GOTO 50 ELSE GOTO 20
<73> 50 DIM y(100)
<46> 60 PRINT:INPUT "IST DIE ZU DIFFERENZIERENDE FUNKTI
     ON BEKANNT, SO GEBEN SIE EIN f EIN; SIND STüTZSTEL
     LEN DER FUNKTION BEKANNT, SO GEBEN SIE BITTE EIN s
      EIN !",Antwort$
<52> 70 IF Antwort$="s" THEN GOTO 290
<98> 80 IF Antwort$<>"f" THEN GOTO 60
<42> 90 PRINT:PRINT "FALLS NOCH NICHT GESCHEHEN, UNTERB
     RECHEN SIE DAS PROGRAMM MITTELS... STOP-TASTE, GEB
     EN DIE FUNKTION. MITTELS.. EDIT 200 - KOMMANDO IN
     ZEILE 200 EIN UND STARTEN.... ERNEUT."
<39> 100 PRINT:INPUT "AN WELCHER STELLE DER FUNKTION SO
     LL DIFFERENZIERT WERDEN (x-KOORDINATE.. EINGEBEN)?
     ",k
<19> 110 REM..... ********************************
     ********************
<27> 120 REM..... * DIFFERENTIALBERECHNUNG üBER
     BEKANNTE FUNKTION *....
<23> 130 REM..... ********************************
     ********************
<76> 140 Schrittweite=0.2
<43> 150 FOR x=(k-0.4) TO (k+0.5) STEP Schrittweite
<68> 160 i=i+1
<31> 170 REM..... ********************************
     ********************
<93> 180 REM....          ... * FUNKTIONSDEFINITION UND STüTZST
     ELLENERZEUGUNG *
<35> 190 REM..... ********************************
     ********************
< 4> 200 y(i)=
<54> 210 y(i)=ROUND(y(i),3)
<94> 220 y(i)=y(i)*1000
<44> 230 NEXT x
<12> 240 Stuetzstellenanzahl=5
<17> 250 GOTO 930
<82> 260 REM..... ********************************
     **********************
<52> 270 REM..... * DIFFERENTIALBERECHNUNG üBER BE
     KANNTE STüTZSTELLEN *
<86> 280 REM..... ********************************
     **********************
<57> 290 PRINT:INPUT "WIEVIEL STüTZSTELLEN SIND BEKANNT
      ? :",Stuetzstellenanzahl
<57> 300 PRINT:INPUT "GEBEN SIE DEN ABSTAND DER x-KOORD
     INATEN ZWEIER BENACHBARTER STüTZSTELLEN EIN (DIESE
     R ABSTAND MUß BEI ALLEN STELLEN DER GLEICHE SEIN!)
     :",Schrittweite
<55> 310 PRINT:PRINT "GEBEN SIE NUN DIE y-WERTE DER STü
     TZSTELLEN EIN, BEGINNEND BEI DEM BETRAGSMäßIG KLEI
     NSTEN."
< 2> 320 FOR i=1 TO Stuetzstellenanzahl
<83> 330 PRINT:PRINT "y(";i; ")=";:INPUT ". ",y(i)
<61> 340 y(i)=ROUND(y(i),3)
< 2> 350 y(i)=y(i)*1000
< 0> 360 NEXT i
<94> 370 PRINT:PRINT "DIE EINGEGEBENEN y-WERTE SIND IN
     DER FORM y(i) DARGESTELLT."
<13> 380 PRINT:INPUT "UM DEN y(i)-WERT DER STELLE i ANZ
     UGEBEN, FüR DIE DIE ABLEITUNG GEBILDET. WERDEN SOL
     L, GEBEN SIE DEN ENTSPRECHENDEN ZAHLENWERT FüR i E
     IN:",i
<96> 390 IF i<Stuetzstellenanzahl THEN GOTO 450
<71> 400 IF i=Stuetzstellenanzahl GOTO 430
<45> 410 PRINT:PRINT "SIE MöCHTEN IN EINEM WERT DIFFERE
     NZIEREN, DER NICHT EINGEGEBEN WURDE.... GEBEN SIE
     NUN ERNEUT DIE DATEN, INKLUSIVE DIESES WERTES EIN
     !"
< 9> 420 GOTO 290
<26> 430 PRINT:PRINT "SIE MöCHTEN IM LETZTEN EINGEGEBEN
     EN WERT (y(i)=y(n)) DIFFERENZIEREN ;... HIERZU BEN
     öTIGT DAS PROGRAMM DEN n+1 -TEN WERT. DAS DIFFEREN
     TIALPROGRAMM STARTET NUN..... ERNEUT. GEBEN SIE DI
     ESEN WERT BITTE MIT AN !"...
<13> 440 GOTO 290
<63> 450 IF NOT i=1 THEN GOTO 490
<64> 460 IF Stuetzstellenanzahl=3 THEN GOTO 590
<86> 470 IF Stuetzstellenanzahl=4 THEN GOTO 630
<32> 480 IF Stuetzstellenanzahl>4 THEN GOTO 700
<45> 490 IF y(i+2)=0 AND y(i+3)=0. THEN GOTO 860
<40> 500 IF. y(i-2)=0 AND (Stuetzstellenanzahl=3 OR Stu
     etzstellenanzahl=4) THEN GOTO 860
<57> 510 IF NOT (y(i-2)=0 AND Stuetzstellenanzahl>4) TH
     EN GOTO 930
<72> 520 IF Stuetzstellenanzahl=5 AND i=3 THEN GOTO 860
<83> 530 FOR j=1 TO Stuetzstellenanzahl
<60> 540 y(j)=y(i+k)
<94> 550 k=k+1
<12> 560 NEXT j
<10> 570 IF Stuetzstellenanzahl=5 OR (Stuetzstellenanza
     hl=6 AND i=3) THEN GOTO 630
<72> 580 GOTO 700
<86> 590. a=-3 : b=4 : c=-1 : d=0 : e=0 : g=(2*Schrittw
     eite)
<94> 600 GOSUB 800
<85> 610 PRINT "f'(";i;") = ",Ableitung
<32> 620 END
<40> 630. a=-11 : b=18 : c=-9 : d=2 : e=0 : g=(6*Schrit
     tweite)
< 3> 640 GOSUB 800
<93> 650 PRINT "f'(";i;") = ",Ableitung
<78> 660. a=2 : b=-5 : c=4 : d=-1 : e=0 : g=(Schrittwei
     te^2)
< 9> 670 GOSUB 800
<64> 680 PRINT "f''(";i;") = ",Ableitung
<46> 690 END
<20> 700 a=-25 : b=48 : c=-36 : d=16 : e=-3 :g=(12*Schr
     ittweite)
<97> 710 GOSUB 800
<88> 720 PRINT "f'(";i;") = ",Ableitung
<95> 730 a=35 : b=-104: c=114 : d=-56 : e=11 : g=(12*Sc
     hrittweite^2)
< 4> 740 GOSUB 800
<59> 750 PRINT "f''(";i;") = ",Ableitung
<27> 760 a=-5 : b=18 : c=-24 : d=14 : e=-3 : g=(2*Schri
     ttweite^3)
<10> 770 GOSUB 800
<69> 780 PRINT "f'''(";i;") = ",Ableitung
<47> 790 END
<36> 800 IF. Stuetzstellenanzahl<5 THEN GOTO 820
<65> 810 i=i+1
<60> 820 Ableitung=(a*y(1)+b*y(2)+c*y(3)+d*y(4)+e*y(5))
     /(g*1000)
<47> 830 IF Stuetzstellenanzahl<5 THEN GOTO 850
<87> 840 i=i-1
<84> 850 RETURN
<48> 860 a=0 : b=-1 : c=0 : d=1 : e=0 : g=2*Schrittweit
     e
<90> 870 GOSUB 1060
< 2> 880 PRINT "f'(";i;") = ",Ableitung
<66> 890 a=0 : b=1 : c=-2 : d=1 : e=0 : g=(Schrittweite
     ^2)
<77> 900 GOSUB 1060
<53> 910 PRINT "f''(";i;") = ",Ableitung
<35> 920 END
<58> 930 a=1 : b=-8 : c=0 : d=8 : e=-1 : g=12*Schrittwe
     ite
<85> 940 GOSUB 1060
<96> 950 PRINT "f'(";i;") = ",Ableitung
<71> 960 a=-1 : b=16 : c=-30 : d=16 : e=-1 : g=(12*Schr
     ittweite^2)
<91> 970 GOSUB 1060
<67> 980 PRINT "f''(";i;") = ",Ableitung
<78> 990 a=-1 : b=2 : c=0 : d=-2 : e=1 : g=(2*Schrittwe
     ite^3)
<17> 1000 GOSUB 1060
<78> 1010 PRINT "f'''(";i;") = ",Ableitung
<23> 1020 a=1 : b=-4 : c=6 : d=-4 : e=1 : g=(Schrittwei
     te^4)
<26> 1030 GOSUB 1060
<70> 1040 PRINT "f''''(";i;") = ",Ableitung
<81> 1050 END
<95> 1060 IF Antwort$="s" THEN GOTO 1080
<48> 1070 i=3
<93> 1080. Ableitung=a*y(i-2)+b*y(i-1)+c*y(i)+d*y(i+1)+
     e*y(i+2)
<42> 1090 g=1000*g
<49> 1100 Ableitung#=CDBL(Ableitung)/g
<76> 1110 Ableitung=Ableitung#
<64> 1120 Ableitung=ROUND(Ableitung,1)
```

```
<19> 1130 IF Antwort$="s" THEN GOTO 1150
<93> 1140 i=k
<95> 1150 RETURN
<28> 1160 REM..... ****************************
     **********************
<77> 1170 REM  ..... * INTEGRALBERECHNUNG DURCH SIMPSO
     N-INTEGRATION *
<34> 1180 REM..... ****************************
     **********************
<74> 1190 DIM x(1000),y(1000)
<80> 1200 PRINT:INPUT " IST DIE ZU INTEGRIERENDE FUNKTI
     ON BEKANNT (ja/nein) ";Antwort$
<98> 1210 IF Antwort$="nein" THEN GOTO 1900
<17> 1220 IF Antwort$<>"ja" THEN GOTO 1200
< 9> 1230 REM..... ****************************
     **********************
<52> 1240 REM..... * INTEGRATIONSRECHNUNG ÜBER
      BEKANNTE FUNKTION *
<15> 1250 REM..... ****************************
     **********************
< 6> 1260 PRINT:PRINT "GEBEN SIE, FALLS NOCH NICHT GESC
     HEHEN, DIE ZU INTEGRIERENDE FUNKTION EIN IN DER FO
     RM y(i)=f(x) IN ZEILE 1670 MITTELS STOP-TASTE UND
     EDIT 1670-KOMMANDO... STARTEN SIE DANN NOCHMALS VO
     N BEGINN AN MITTELS RUN-KOMMANDO !"
<66> 1270 PRINT:INPUT "GEBEN SIE DEN KLEINSTEN WERT DES
      INTEGRATIONSBEREICHS EIN (UNTERE GRENZE):",a
<38> 1280 PRINT:INPUT "GEBEN SIE DEN GRÖßTEN WERT DES I
     NTEGRATIONSBEREICHS EIN (OBERE GRENZE):",b
<90> 1290 REM..... ****************************
     **********************
<60> 1300 REM  * ERZEUGUNG EINER GERADEN, GLEICHGROßEN
     ANZAHL VON INTERVALLEN *.....
<36> 1310 REM..... *... ÜBER DEN GESAMMTEN INTEGRATIO
     NSBEREICH (VON a BIS b)..... *
<71> 1320 REM..... ****************************
     **********************
<78> 1330 a2=a
<90> 1340 d=0
< 0> 1350 e=0
<27> 1360 i=0
<96> 1370 t=0
< 5> 1380 Flaeche1=Flaeche
<71> 1390 IF SGN(a2)=SGN(b) THEN GOTO 1440
<70> 1400 q=q+1
<65> 1410 a=0
<73> 1420 IF q=1 THEN GOTO 1440
<28> 1430 b=a2
<13> 1440 a1=ABS(a)
<35> 1450 b1=ABS(b)
<58> 1460 IF (b1-a1)<0.5 THEN GOTO 1540 ELSE GOTO 1470
<65> 1470 Intervallanzahl=(b1-a1)*10
<57> 1480 Intervallanzahl=ROUND (Intervallanzahl,0)
<63> 1490 FOR k=0 TO Intervallanzahl STEP 2
<55> 1500 IF k=Intervallanzahl THEN GOTO 1590
<11> 1510 IF k>Intervallanzahl THEN GOTO 1550
<11> 1520 NEXT k
<37> 1530 GOTO 1550
<85> 1540 k=6
<87> 1550 Intervallanzahl=k
<64> 1560 REM..... ****************************
     ****
```

```
<85> 1570 REM  ..... * INTEGRATIONSRECHNUNG *
<70> 1580 REM..... ****************************
     ****
<40> 1590 Schrittweite=(b1-a1)/Intervallanzahl
<83> 1600 x(i)=a1+t
<43> 1610 IF a<0 AND b<0 THEN GOTO 1620 ELSE GOTO 1630
<98> 1620 x(i)=-x(i)
<76> 1630 x=x(i)
<69> 1640 REM..... ****************************
     ***
<95> 1650 REM....       ..... * FUNKTIONSDEFINITION
     *
<75> 1660 REM..... ****************************
     ***
<34> 1670 y(i)=
<64> 1680 IF i=Intervallanzahl THEN GOTO 1720
<43> 1690 t=t+Schrittweite
<63> 1700 i=i+1
<81> 1710 GOTO 1600
<42> 1720 FOR i=2 TO (Intervallanzahl-2) STEP 2
<84> 1730 d=d+y(i)
<98> 1740 NEXT i
< 9> 1750 d=d*2
< 8> 1760 FOR i=1 TO (Intervallanzahl-1) STEP 2
<11> 1770 e=e+y(i)
<11> 1780 NEXT i
<55> 1790 e=e*4
<88> 1800 REM..... ******************
<73> 1810 REM  ..... * SIMPSON-REGEL *
<94> 1820 REM..... ******************
<67> 1830 Flaeche=ABS((d+e+y(0)+y(Intervallanzahl))*Sch
     rittweite/3)
<49> 1840 IF q=0 THEN GOTO 1870
<66> 1850 IF q=1 THEN GOTO 1340
<24> 1860 Flaeche=Flaeche+Flaeche1
<37> 1870 Flaeche=ROUND (Flaeche,3)
<80> 1880 PRINT:PRINT "DIE INTEGRATION ERGIBT EINE FLÄC
     HE VON:",Flaeche
<10> 1890 END
< 5> 1900 REM..... ********************************
     ******************
<85> 1910 REM..... * INTEGRATIONSRECHNUNG ÜBER BEKAN
     NTE STÜTZSTELLEN *
<11> 1920 REM..... ********************************
     ******************
<67> 1930 PRINT:INPUT "GEBEN SIE DIE (UNGERADE) ANZAHL
     DER ZU BENUTZENDEN STÜTZSTELLEN EIN:",.. Stuetzste
     llenanzahl
<11> 1940 PRINT:INPUT "GEBEN SIE DEN ABSTAND DER x-KOOR
     DINATEN ZWEIER BENACHBARTER STÜTZSTELLEN EIN . ( D
     IESE ABSTÄNDE MÜSSEN ÜBER DEN GESAMTEN INTEGRATION
     SBEREICH KONSTANT SEIN! ):",..... Schrittweite
<71> 1950 PRINT:PRINT "GEBEN SIE DIE y-WERTE DER STÜTZS
     TELLEN EIN, BEGINNEND BEI DEM..... BETRAGSMÄß
     IG KLEINSTEN."
<75> 1960 Stuetzstellenanzahl=Stuetzstellenanzahl-1
< 3> 1970 FOR i=0 TO Stuetzstellenanzahl
<36> 1980 PRINT:PRINT "GEBEN SIE DEN y-WERT DER STÜTZST
     ELLE EIN:"
<70> 1990 PRINT:PRINT "y(";i; ")=";:INPUT ". ",y(i)
<73> 2000 NEXT i
<34> 2010 FOR i=1 TO (Stuetzstellenanzahl-1). STEP 2
```


Michael Naujoks

Annals of Rome	68,90
Bounder	49,90
Brian Clough's Football Fortunes	49,90
Classic Collection	44,90
Frank Bruno's Boxing	69,90
Jewels of Darkness	79,90
Southern Bell + Heathrow ATC	59,90
Star Glider	79,90
Steve Davis Snooker	49,90
Tau Ceti	69,90
The Pawn	85,90
Trivial Pursuit (Deutsch)	59,90

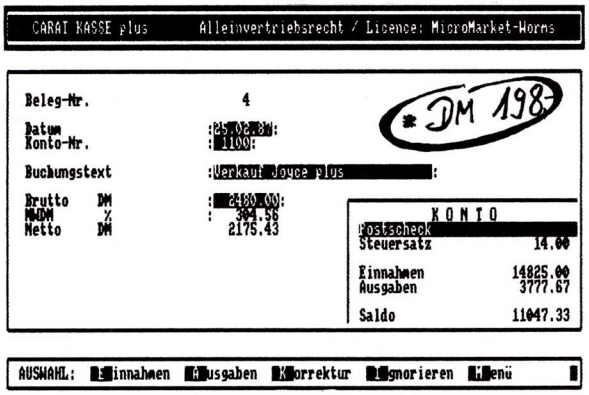


Joystick-Controller	69,—
Sound-Synthesizer + Joystick-Interface	129,—
Echtzeituhren-Modul	129,—
Adapter f. obige Geräte	29,—
256 k Speichererweiterung	109,—

Entwicklung & Vertrieb von Computer Soft- und Hardware
Rottmannstr. 40, 6900 Heidelberg

Hotline: (06221) 46885

Das JOYCE-KASSENBUCH (indiv. Konten) – einfach gut!!!



Version auch für Österreich lieferbar (Vorkasse/Scheck)!
MicroMarket-Worms
Joh.-Westermann-Platz 1, 4780 Lippstadt, Tel. (0 29 41) 5 92 90

```
<68> 2020 d=d+y(i)
<82> 2030 NEXT i
<13> 2040 d=d*4
<96> 2050 FOR i=2 TO (Stuetzstellenanzahl-2). STEP 2
<94> 2060 e=e+y(i)
<94> 2070 NEXT i
<19> 2080 e=e*2
<14> 2090 Intervallanzahl=Stuetzstellenanzahl
<31> 2100 GOTO 1830
```

LISTING

```
>KURDIS<, REMARK = >REM<.

<49> 10 REM..... **********
     **********
< 8> 20 REM..... *..... KURVENDISKUSSI
     ON..... *
<51> 30 REM..... **********
     **********
<92> 40 REM
<78> 41 REM.
<56> 42 REM..... **********
     **********
<26> 44 REM..... *... Programmname: "Ku
     rdis".. *
<66> 47 REM..... **********
     **********
< 9> 48 REM
<97> 50 PRINT :INPUT "SOLLEN AUCH DIE WENDEPUNKTE ERMIT
     TELT WERDEN ? (ja/nein) :",Antwort$
<63> 60 IF Antwort$="nein" THEN GOTO 80
<20> 70 IF Antwort$<>"ja". THEN GOTO 10
<13> 80 PRINT : PRINT "ZUR ERMITTLUNG DER KURVENEXTREMA
      GEBEN SIE DIE FUNKTION EIN (ZEILE 170)"
<32> 90 PRINT: PRINT "GEBEN SIE DIE INTERVALLGRENZEN EI
     N, ZWISCHEN DENEN DIE EXTREMA BESTIMMT. WERDEN SOL
     LEN :"
<50> 100 INPUT "UNTERE GRENZE: a=",a : INPUT "OBERE. GR
     ENZE: b=",b
<25> 110 a1=a
<36> 120 REM..... **********
     **********
<80> 130 REM..... *..... EXTREMA...
     ..... *
<40> 140 REM..... **********
     **********
<92> 150 x=a
<93> 160 DIM y(2000)
<87> 170 DEF FNy=SIN(x*3.141592654#/180)
<73> 180 y1=FNy
< 3> 190 x=x+ABS(b-a)/1000 : i=i+2 : y2=FNy
<83> 200 x=a
<75> 210 i=i-1
<73> 220 IF y1<y2 THEN GOSUB 380 ELSE GOTO 300
<72> 230 GOSUB 590
<34> 240 WHILE x<b
<14> 250 y1=y(i) : x=x+ABS(b-x)/1000
<32> 260 y2=FNy
<39> 270 GOTO 220
<96> 280 WEND
<64> 290 GOTO 1110
<81> 300 GOSUB 610
< 7> 310 y1=y(i) :a=x
<37> 320 GOSUB 380
<33> 330 WHILE x<b
<90> 340 y1=y(i) : x=x+ABS(b-a)/100 : y2=FNy
<36> 350 GOTO 220
<93> 360 WEND
<61> 370 GOTO 1110
< 4> 380 REM..... *** KURVE STEIGT MIT
      x AN ***
< 7> 390 k=40
<42> 400 Schrittweite=ABS(b-a)/k
< 0> 410 x=x+Schrittweite
<63> 420 i=i+1
<52> 430 y(i)=FNy
<22> 440 IF SGN(y1)<>SGN(y(i)) THEN GOSUB 860
<32> 444 PRINT x,y1,y(i)
<85> 450 IF y(i)=y1 THEN GOTO 530
<18> 460 IF y(i)>y1 THEN GOTO 500
<33> 470 x=x-(2*Schrittweite)
< 1> 480 i=i-2
<78> 490 k=k*10
<26> 500 IF b*1.06<x AND b<>0 THEN GOSUB 1110
<24> 510 y1=y(i)
<30> 520 GOTO 400
<97> 530 x=ROUND (x,0)
<14> 540 y(i)=ROUND (y(i),2)
<16> 542 REM IF x>0 AND (x<xmax OR x<xmin) THEN x=1.2*x
     : RETURN
<14> 550 PRINT:PRINT "Maximum :. x=";x , "y=";y(i)
<30> 552 xmax=x
<16> 557 ymax=y(i)
<83> 560 RETURN
<16> 570 REM
<43> 580 REM..... *** KURVE FÄLLT MIT
      x AB ***
<40> 590 y1=y(i)
<34> 600 a=x
<93> 610 k=40
<48> 620 Schrittweite=ABS(b-a)/k
< 6> 630 x=x+Schrittweite
<33> 640 GOTO 670
<10> 650 x=x+Schrittweite
<29> 660 IF b*1.06<x THEN GOSUB 1110
<75> 670 i=i+1
<64> 680 y(i)=FNy
<34> 690 IF SGN(y1)<>SGN(y(i)) THEN GOSUB 860
<27> 698 IF y(i)=ymax THEN x=x+Schrittweite : k=10 : y1
     =y(i) : a=x : GOTO 620
<67> 700 IF y(i)=y1 THEN GOTO 780
<18> 710. IF y(i)<y1 THEN GOTO 760
<26> 720 x=x-(2*Schrittweite)
<93> 730 i=i-2
<71> 740 k=k*10
<55> 750 Schrittweite=ABS(b-a)/k
<36> 760 y1=y(i)
<18> 770 GOTO 650
<10> 780 x=ROUND (x,0)
<26> 790 y(i)=ROUND (y(i),2)
<76> 792 IF x>0 AND (x<xmin OR x<xmax) THEN x=1.2*x : R
     ETURN
<58> 800 PRINT:PRINT "Minimum :. x=";x , "y=";y(i)
<98> 802 xmin=x
<76> 810 RETURN
<41> 820 REM..... **********
     *********
<15> 830 REM..... *..... NULLSTELLEN..
     ..... *
<45> 840 REM..... **********
     *********
<15> 850 REM
<87> 860 x1=x : k1=k : Schrittweite1=Schrittweite : y11
     =y1
<86> 870 IF y11>y(i) THEN GOTO 960
<95> 880 j=1
<60> 890 WHILE y11<0
<56> 900 x=x+Schrittweite1/(0.4*k1)
<47> 910 y11=FNy
<27> 920 y11=ROUND(y11,2)
<60> 930 IF y11=0 THEN GOTO 1070
<95> 940 WEND
<76> 950 GOTO 1030
<85> 960 j=0
<85> 970 WHILE y11>0
<72> 980 x=x+Schrittweite1/(0.4*k1)
<63> 990 y11=FNy
<18> 1000 y11=ROUND(y11,2)
< 5> 1010 IF y11=0 THEN GOTO 1070
<17> 1020 WEND
<22> 1030 IF x>b*1.06 THEN GOTO 1110
<64> 1032 x=x-Schrittweite1 : k1=k1*2
<33> 1050 y11=FNy
<71> 1060 IF j=1 THEN GOTO 880 ELSE GOTO 960
<26> 1070 x=ROUND(x,0)
<59> 1080 PRINT: PRINT TAB(50) "Nullstelle : x=";x,"y="
     ;y11
<48> 1090 x=x1
<80> 1100 RETURN
<28> 1110 REM..... **********
<64> 1120 REM..... * WENDEPUNKTE *
<34> 1130 REM..... **********
<68> 1140 REM
<26> 1150 IF Antwort$="nein" THEN END
<39> 1160 a=a1
<60> 1170 k=20 : Schrittweite=ABS(b-a)/k
<74> 1180 Schrittweite1=Schrittweite
<19> 1190 x=a-2*Schrittweite/k
```

```
<52> 1200 FOR i=1 TO 5
<20> 1210 y(i)=FNy
< 1> 1220 x=x+Schrittweite/k
<85> 1230 NEXT i
<56> 1240 Abl2yi=(-y(1)+16*y(2)-30*y(3)+16*y(4)-y(5))/(
     (Schrittweite/k)^2*12)
<35> 1250 x=x-6*Schrittweite/k
<70> 1260 FOR i=1 TO 5
<74> 1270 y(i)=FNy : x=x+Schrittweite/k
< 1> 1280 NEXT i
<49> 1290 Abl2y1=(-y(1)+16*y(2)-30*y(3)+16*y(4)-y(5))/(
     (Schrittweite/k)^2*12)
< 7> 1300 IF SGN(Abl2y1) <> SGN(Abl2yi) THEN GOTO 1330
<48> 1310 IF x>b THEN END
<53> 1320 x=x+20*Schrittweite/k : Abl2y1=Abl2yi : .GOTO
     1250
<14> 1330 x=x-20*Schrittweite/k
<17> 1340 FOR i=1 TO 5.
< 2> 1350 y1(i)=FNy
<15> 1360 x=x+Schrittweite/k
< 0> 1370 NEXT i
<57> 1380 GOSUB 1430
< 9> 1390 x=x-2*Schrittweite/k
<62> 1400 REM PRINT x,Abl2yi,Abl2y1
<82> 1410 IF SGN(Abl2y1)=SGN(Abl2yi) THEN GOTO 1340
<68> 1420 x=x-10*Schrittweite/k. :. k=k+10 :. GOTO 1340
<60> 1430 Abl2y1#= (-y1(1)+16*y1(2)-30*y1(3)+16*y1(4)-y
     1(5))/(Schrittweite/k)^2*12
<62> 1440 Abl2y1#=ROUND(Abl2y1#,4)
<15> 1450 Abl2y1=Abl2y1#
<33> 1460 IF Abl2y1=0 THEN GOTO 1480
< 8> 1470 RETURN
<12> 1480 x=x-2*Schrittweite/k : x=ROUND(x,0)
<29> 1490 y1(3)=ROUND(y1(3),1)
<45> 1500 IF x<a1 THEN GOTO 1520
<75> 1510 PRINT : PRINT TAB(20) "Wendepunkt : x=";x , "
     y=";y1(3)
<15> 1520 Schrittweite=Schrittweite1 : k=20 : x=x+8*Sch
     rittweite/k
<35> 1530 GOTO 1200
```

Zeitanzeige unter Basic

Das folgende kleine Programm läßt sich als Subroutine in jedes vorhandene Basicprogramm einbinden.
Funktion: Anzeige der aktuellen Uhrzeit in der rechten oberen Bildschirmecke. Die Uhrzeit muß jedoch unter CP/M mit DATE SET gestellt werden. 8080-Programmierern mögen die angegebenen Speicheradressen als Basis für die Erstellung eines kleinen Assemblerprogrammes dienen.

(R.Herrmann)

```
10 'UHRZEIT V1.1
20 'R.Herrmann DEZ.86
30 cls$=CHR$(27)+"E"+CHR$(27)+"H"
40 cursor$= CHR$(27) + "Y"
   + CHR$(32) +CHR$(110)
50 PRINT cls$
60 GOSUB 50000
70 END

50000 'UP Zeit lesen und zeigen
50010 WHILE INKEY$=""
50020 std$=HEX$(PEEK(&HFBF6))
50030 min$=HEX$(PEEK(&HFBF7))
50040 sec$=HEX$(PEEK(&HFBF8))
50050 IF sec$=secold$ THEN GOTO
      50090
50060 IF sec$="0" THEN PRINT
      cursor$; std$; ":"; min$; ":";
      CHR$(27)+ "K"; sec$: GOTO
      50090
50070 PRINT cursor$; std$; ":";
      min$; ":"; sec$
50080 '
50090 secold$=sec$
50100 WEND

42 Zeilen
File: timeshow.asc
Disk: texte sh2
```


Welche Software würden Sie mit auf eine einsame Insel nehmen?

COMAC—KASSE !

Was machen Sie mit dem 10 Tage alten Tankbeleg, den Sie im Kofferraum wiederfinden? COMAC-KASSE sortiert ihn problemlos unter dem richtigen Datum ein.

Was machen Sie mit Ihrer geliebten Umsatzsteuererklärung, die Sie seit einem halben Jahr vor sich herschieben? COMAC-KASSE druckt sie sofort übersichtlich aus.

Was machen Sie, wenn wieder Ebbe in Ihrem Portemonnaie/Ihrem Tresor herrscht? COMAC-KASSE verfaßt sofort eine einsichtige Übersicht Ihrer Einnahmen und Ausgaben.

COMAC-KASSE ist
* eine Einnahmen-Überschußrechnung für alle Gewerbetreibenden
* eine Haushaltskasse für alle Hausmänner und Hausfrauen
* ein Geschäftsjournal für alle Buchhalter

Leistungsmerkmale
* Festlegung beliebig vieler Konten
* nachträgliche Eingabe/Änderung von Buchungen
* 2000 Buchungen (1 Laufwerk), 8000 Buchungen (2 Laufwerke)
* monatliche/vierteljährliche Umsatzsteuererklärung
* Anzeige der Diskettenkapazität etc, etc.

Gesamtpreis incl. Handbuch: DM 128.—
(Demo-Diskette + Handbuch DM 35.—)

COMAC—LV

Leistungsverzeichnis für Ingenieure und Architekten

* minimaler Zeitaufwand bei der Erstellung Ihrer Leistungsverzeichnisse
* umfangreiche Korrekturmöglichkeiten
* freie Eingabe Ihrer Leistungstexte etc. etc. DM 398.—

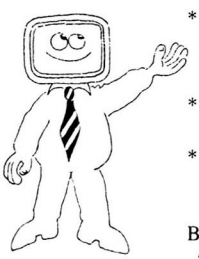
COMAC—LITKASTEN

professionelle Literaturdatenverwaltung für alle Leser

* Ordnung in Ihrem Zettelberg
* umfangreiche Suchmöglichkeiten
* Eingabe von Schlagworten etc. etc. DM 98.—

CMZ-Verlag Winrich C.-W. Clasen
Borgswiese 9-11, 4650 Gelsenkirchen 2
Tel. 0209 - 777896 oder 02261 - 65434

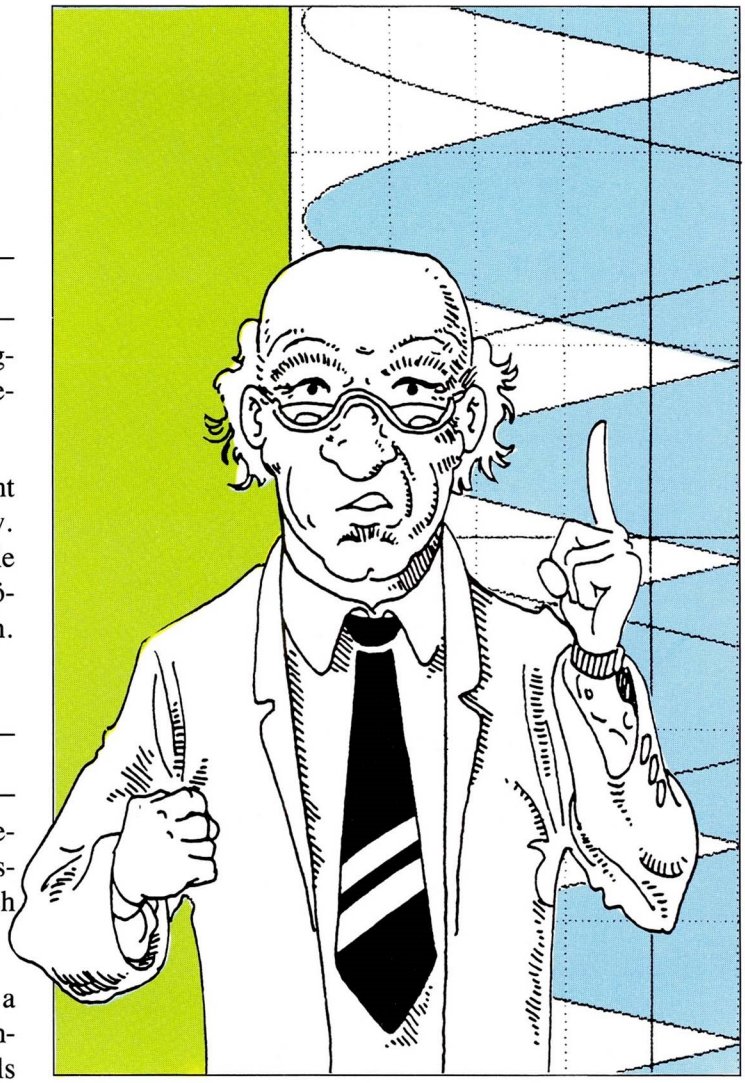
Bestellen und aufkleben!



* Endlos-Haftetiketten für Ihre 3"-Disketten in versch. Packungsgrößen, zweiseitig: z.B. 100 Stck. DM 9.95
* JOYCE Marken-Druckerfarbband DM 18.95
* JOYCE Software und Individuallösungen Verlangen Sie unser Gratis-Info

Büro für Software-Entwicklung, Sonnenstr. 43
5270 Gummersbach, Telefon: 02261/65434

ARCUS SIN und ARCUS COS



1) Allgemeines

Ein Problem beim »Joyce« ist die recht eingeschränkte Möglichkeit, ihn als wissenschaftlichen Taschenrechner zu benutzen.

Dieses ist u.A. bedingt durch das Fehlen der doch recht wichtigen Umrechnungsfunktionen von Arcus-Sinus- bzw. Arcus-Cosinus-Werten in die entsprechenden Winkel. Die Umrechnung von Arcus-Tangens- Werten in die zugehörigen Winkel ist demgegenüber ohne Probleme möglich.

2) Mathematische Grundlagen

Die Trigonometrie als ein kleines Teilgebiet der Mathematik beschäftigt sich mit Winkeln und Seitenverhältnissen in Dreiecken; wir wollen hier im folgenden lediglich rechtwinklige Dreiecke betrachten.

In Bild 1 ist ein solches Dreieck dargestellt; die Seiten a und b bezeichnet man als sog. »Katheten« und die dem rechten Winkel gegenüberliegende (längste) Dreieckseite c als »Hypothenuse«. Für den Winkel α (Alpha) ist a die »Gegenkathete«, da sie gegenüber von α liegt; b heißt »Ankathete«.

Mit Hilfe dieser Begriffe werden nun in der Trigonometrie z.B. Seitenverhältnisse benutzt, um unbekannte Winkel im Dreieck zu berechnen. Man definiert das Verhältnis der Gegenkathete von α (also der Strecke a) zur Hypothenuse (also c) als »Sinus« von α; das Verhältnis von Ankathete b zur Hypothenuse c als »Cosinus« von α und das Verhältnis von Gegenkathete a zu Ankathete b als »Tangens« von α.

In mathematischer Kurzschreibweise, also in Formeln ausgedrückt, stellt sich das wie folgt dar:

$$\sin \alpha = a/c;\ \cos \alpha = b/c;\ \tan \alpha = a/b$$

Analog läßt sich dieses natürlich auch für den Winkel ß (Beta) aufstellen (s. Bild 1):

$$\sin ß = b/c;\ \cos ß = a/c;\ \tan ß = b/a$$

Hierbei ist nun a die Ankathete von ß und b die Gegenkathete.

Vom Taschenrechner her kennt man die Möglichkeit, den Sinuswert zu ermitteln, indem man den bekannten Winkel eintippt und die »sin«-Taste drückt. Möchte man umgekehrt aus einem gegebenen Sinuswert den zugehörigen Winkel berechnen, so gibt man diesen Wert ein und tippt dann »Inv-sin« (Invers-Sinus).

Dabei ist die »Invers«-Operation eine Umkehrfunktion, die in der Trigonometrie üblicherweise als sog. »Arcus«-Funktion bezeichnet wird. Man sagt statt »Invers-Sinus« daher auch »Arcus-Sinus«, kurz »arc sin«. Das Gleiche gilt natürlich entsprechend auch für Cosinus und Tangens.

Der »Joyce« kann wie viele Rechner zwar aus einem vorgegebenen Winkel die zugehörigen Sinus, Cosinus- und Tangenswerte berechnen; allerdings nicht einen bekannten Sinus- bzw. Cosinuswert in den entsprechenden Winkel umformen; d.h. arc sin- und arc cos-Funktion sind auf »Joyce« nicht durchführbar, da er über diese Befehle nicht verfügt. Lediglich die Arcus-Tangensfunktion ist vorhanden.

Hier ist der »Joyce« schwächer als die meisten Taschenrechner, die alle diese Umrechnungen serienmäßig können.

Aus diesen Gründen habe ich die zwei im folgenden beschriebenen Programme entwickelt, die »Joyce« und auch (evtl. umgeschrieben) anderen Computern die Eigenschaften verleihen, nun doch oben beschriebene Operationen durchzuführen.

3) Programmbeschreibung

a) Allgemeines:

Beide Programme, sowohl »sincos1« als auch »sincos2« bieten die Entscheidungsmöglichkeit, einen Winkel durch seinen (bekannten) Sinuswert oder durch seinen Cosinuswert zu ermitteln und können unabhängig voneinander, einzeln angewendet werden.

Hierbei liegt die Genauigkeit des Ergebnisses bei drei Dezimalstellen, mit anderen Worten, der berechnete Winkel ist auf drei Stellen hinter dem Komma genau. Diese Genauigkeit reicht für die Praxis in der Regel völlig aus; oft ist hier lediglich eine Nachkommastelle von Bedeutung.

Beide Programme können auch als Unterprogramme in andere, größere Programme integriert werden und stellen damit eine erhebliche Wertsteigerung des »Joyce« dar.

Das Programm »sincos1« löst das Problem iterativ und ist damit auch für Rechner geeignet, die nicht über eine Arcus-Tangens-Funktion verfügen.

Das Programm »sincos2« ist das für den »Joyce« besser geeignete, da es kürzer und schneller ist, als das Erstere. Es wird hierbei die beim »Joyce« vorhandene Arcus-Tangens-Funktion benutzt.

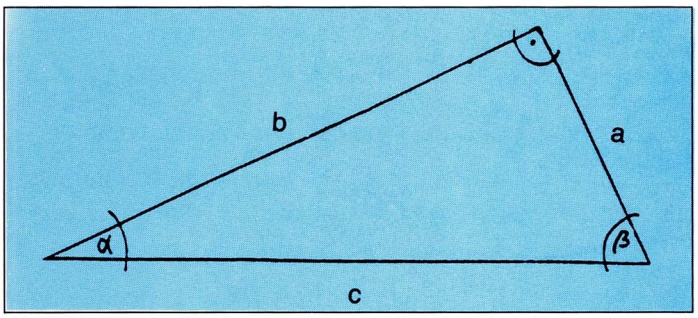


Bild 1: Das rechtwinklige Dreieck

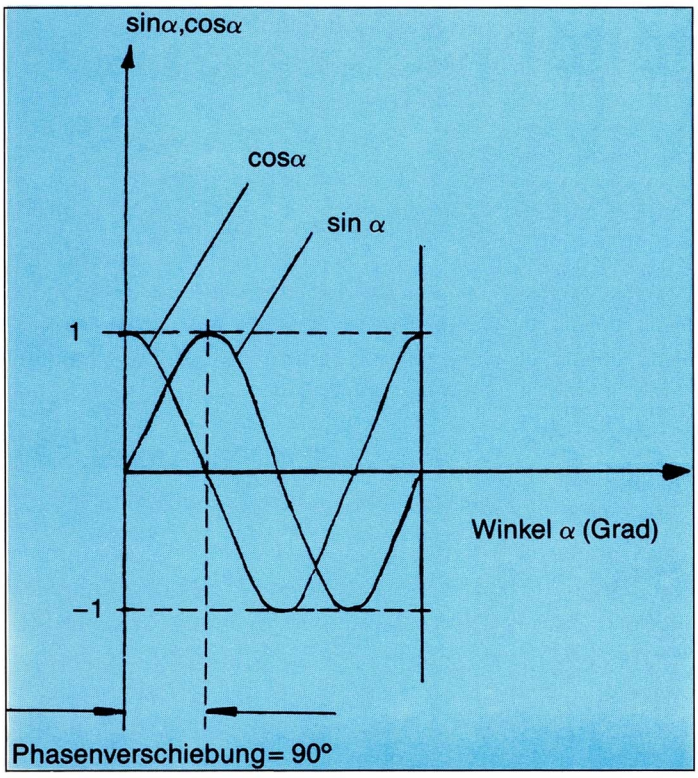


Bild 2: SIN/COS-Phasenverschiebung

Beide Programme stellen sinnvolle Möglichkeiten zur Lösung des Problems dar und ermöglichen Direktvergleiche der zwei Lösungswege.

b) Programm »sincos1«

Wie bereits erwähnt, geht dieses Programm nicht davon aus, daß der Rechner über einen »Arcus-Tangens«-Befehl verfügt. Das Programm ist jedoch fÜr den »Joyce« geschrieben und müßte selbstverständlich für die Anwendung auf anderen Geräten (Betriebssystemen) umgeschrieben werden.

Grundidee von »sincos1« ist ein Iterationsalgorithmus oder einfacher ausgedrückt ein Zählverfahren. Nach dem Programmstart gibt man z.B. den Sinuswert eines Winkels ein, der berechnet werden soll. Das Programm beginnt nun bei einem Winkel von 0 Grad und errechnet den Sinuswert (Zeile 120-140). Dieser wird verglichen mit dem eingegebenen Sinuswert (Zeile 170).

Ist letzterer größer, so wird zu 0 Grad, 1 Grad hinzu addiert und erneut verglichen (Zeile 240, 250). Das läuft solange, bis der eingegebene Wert kleiner als der durch das Zählen (Iterieren oder auch Nähern genannt) entstandene ist. Ist dies der Fall, so wird der letzte durch Iteration ermittelte Winkel um 1 Grad verringert und in Zehntel-Schritten erhöht (Zeile 190−230), wobei nach jeder Erhöhung wieder verglichen wird. Wird der so iterierte Wert erneut größer als der eingegebene, so wird wieder der letzte Winkel erniedrigt und nun stets um ein Hundertstel erhöht. Bei jedem Überschreiten des eingegebenen Sinuswertes nähert sich der iterierte Winkel also um eine Zehnerpotenz weniger an den Endwert, d.h. er tastet sich in immer kleiner werdenden Schritten an diesen Endwert heran. Dieser ist erreicht, wenn der errechnete Sinuswert auf fünf Nachkommastellen genau mit dem eingegebenen Wert übereinstimmt (Zeile 150, 160). Der so ermittelte Winkel wird auf drei Stellen hinter dem Komma gerundet ausgegeben (Zeile 260−310, 380).

Die in Zeile 130 vorgenommene Umrechnung des jeweiligen iterierten Winkels ist notwendig, weil »Joyce« für die Berechnung des Sinuswertes stets das Bogenmaß des Winkels benötigt.

Ein eingegebener, negativer Sinuswert wird wie ein positiver behandelt (Zeile 400, 410).

Soll ein Cosinuswert nach dem »run«-Befehl eingegeben werden, so faßt das Programm ihn als Sinuswert auf (Zeile 80) und errechnet mit dem Endwert des iterierten Winkels und einem Phasenkorrekturterm die endgültige Lösung für den zu ermittelnden Winkel (Zeile 320-360 , 430,440). Dieses Vorgehen ist mathematisch zulässig, da Sinus- und Cosinusfunktion bis auf die Phasenverschiebung von 90 Grad identisch sind (s. Bild 2).

Die Rechenzeit wird kürzer, wenn in Zeile 150 und 160 die Rundung auf weniger als fünf Stellen verringert wird; natürlich beeinflußt eine solche Änderung das Ergebnis ne

KORTEX® Benutzer-Schnittstelle
für JOYCE / JOYCE Plus

KORTEX ist eine Bedienungs-Umgebung, die es Ihnen ermöglicht, eigene Menüs zu definieren und miteinander zu vernetzen. Sie brauchen nur einzugeben, wie die Menüs heißen sollen. Alles andere erledigt KORTEX selbständig.

An jedes Menü können Sie einen beliebigen Satz von Betriebssystem-Kommandos anhängen, der seinerseits automatisch gestartet wird, wenn KORTEX läuft. So wird Ihr täglicher Bedienungsbedarf weitgehend vom Betriebssystem unabhängig, damit durchsichtiger, komfortabler und treffsicher.

KORTEX bietet Ihnen sämtliche Features, die bei einer modernen Menüsteuerung nicht fehlen dürfen: Fenstertechnik, Schreibtisch-Effekte, Pull-Down-Menüs, Führungstexte, Gestaltungsvariationen der einzelnen Menü- und Textanzeigen. Dazu brauchen Sie keine Programmierkenntnisse, denn alles läuft unter Aufsicht eines voll menügesteuerten Spezialeditors ab.

Fordern Sie kostenlos und unverbindlich weitere Informationen an. Oder bestellen Sie gleich eine Demo-Version, komplett mit deutschem Handbuch, für nur DM 25,–* (wird bei späterem Lizenzerwerb voll angerechnet). Oder für DM 198,–* eine volle Benutzungslizenz.

GERALD KEIL
SOFTWARE-ENTWICKLUNG
UND VERTRIEB
Obere Hochstraße 136 – 6652 BEXBACH/Saar
* inkl. MwSt. Nachnahme (nur Brd) zzgl. DM 3,50.

— WEB —

Die 20 MB WINCHESTER-EXPANSION-BOX für JOYCE
DRIVE ACESS TIME 15 msec. DATA TRANSFER RATE 5 MB/sec.
DIRECTORY ENTRIES 2048 CHANNEL C,D,M 3 OPTIONS
Controller, Netzteil, Konsole, Software

QUICK-DATA WEB
Das erste GSX-Geschäftsprogramm speziell für Festplattenbetrieb
und max. 18 MB Datenbank
für die Probleme des Versandgeschäftes
(NN-Paket/NN-Päckchen/NN-Brief)
mit Standards ITOH Serie 8510 BP, SP BINDER MFP 6000

QUICK-CALC 2.1	Buchhaltung	148 DM
QUICK-CLAC PLUS	Erweiterte Buchhaltung	178 DM
QUICK-BILL 2.0	Angebot, Rechnung	168 DM
QUICK-NAME 2.0	Adressverwaltung	128 DM
QUICK-NAME PLUS	Adressen/Serientexte	159 DM
QUICK-CINE 2.0	Videoarchiv	98 DM
QUICK-DATA 3.0	Geschäftssoftware GSX™	228 DM
QUICK-DATA WEB	nur Festplattenbetrieb	498 DM
KIT PC8256/TMS	512 KB RAM-Erweiterung	99 DM
DISK DRIVE 1MB	Laufwerk für JOYCE	564 DM
FILTER	Bildschirmfilter	79 DM
PAPIERFÜHRUNG	für JOYCE Drucker	39 DM
SUPERTYPE	Schriftarten f.LocoScript	p.A.
LocoMail	Mail Merge f. V 1.21	p.A.
LIGHT-PEN	incl. Software	p.A.
MOUSE	komp. DIGITISER/DR DRAW	p.A.
DIGITISER	komp. MOUSE/PEN	p.A.
WINCHESTER	Expansion Box, 20 MB	p.A.
FAX-ADAPTOR	Telefax, Telex	p.A.

weitere Soft- und Hardware in unserer Übersicht

Fordern Sie unser Info-Material an!

Fa. WERDER – Nachrichtentechnik
Sprechzeiten Mo-Fr.: 9-11/14-16 Uhr
Bramfelder Ch. 215 / 2 Hamburg 71 / (040) 641 17 79

gativ. Empfehlenswert ist diese Änderung deshalb nur, wenn keine Nachkommastellen beim Ergebnis benötigt werden.

Das Programm und auch das folgende (»sincos2«) sind in der vorliegenden Form, wie bereits erwähnt, auf drei Nachkommastellen genau und liefern die gleichen Werte wie ein Taschenrechner (getestet am »TI 58« der Firma »Texas Instruments«). Allerdings kann es bei manchen eingegebenen Werten zu Ungenauigkeiten in der dritten Nachkommastelle kommen; diese Abweichung ist jedoch nie größer als ein Tausendstel des Winkels.

c) Programm »sincos2«

Dieses Programm ist programmiertechnisch deutlich weniger aufwendig, wie das oben beschriebene »sincos1«, da es die »Arcus-Tangens«-Funktion des »Joyce« benutzt; es ist daher auch schneller.

Zunächst definiert man die Hypothenuse c in Bild 1 als virtuelle Größe vom Betrag »1«; mit anderen Worten, man weist c den Wert »1« zu.

Dann ist analog zu Kapitel 2:

$$\sin \alpha = a/1 = a, \text{ bzw. } \cos \alpha = b/1 = b$$

Im rechtwinkligen Dreieck gilt nach dem Satz des »Pythagoras« und für c=1:

$$a\uparrow 2 + b\uparrow 2 = c\uparrow 2 = 1$$

Dabei bedeutet »↑2« das gleiche wie »hoch zwei«.

Man kann diesen Ausdruck nach a bzw. nach b auflösen und erhält zwei Wurzelausdrücke (»sqr« bedeutet »Wurzel«):

$$a = \text{sqr}(1 - b\uparrow 2); \; b = \text{sqr}(1 - a\uparrow 2)$$

Diese beiden Gleichungen und die ersten beiden kann man nun in den Ausdruck für den Tangenswert von α (s.Kapitel 2) einsetzen; es ergibt sich dann:

$$\tan \alpha = a/b = \sin \alpha = \text{sqr}(1 - a\uparrow 2)$$
$$\tan \alpha = \sin \alpha = \text{sqr}(1 - \sin\uparrow 2\alpha)$$

und:
$$\tan \alpha = a/b = \text{sqr}(1 - b\uparrow 2) = \cos \alpha$$
$$\tan \alpha = \text{sqr}(1 - \cos\uparrow 2\alpha) = \cos \alpha$$

Wir haben nun eine Gleichung, in der der Tangenswert nur durch den Sinuswert ausgedrückt wird sowie eine, in der er nur durch den Cosinuswert ausgedrückt wird. Es braucht jetzt nur noch der Tangenswert von α mittels Arcus-Operation in den Wert des Winkels selbst umgewandelt zu werden. Daraus folgen dann die beiden endgültigen Formeln:

$$\alpha = \arctan(\sin\alpha \; / \; \text{sqr}(1-\sin\uparrow 2\alpha))$$
$$\alpha = \arctan(\text{sqr}(1-\cos\uparrow 2\alpha) \; / \; \cos\alpha)$$

Diese beiden Formeln sind das »Herz« des Programmes »sincos2«. Man findet sie gekoppelt mit der beim »Joyce« notwendigen Gradumrechnung in Bogenmaß in Zeile 100 und 150. Wird ein Sinus- oder Cosinuswert vom Betrag »Eins« eingegeben, so würde in beiden Formeln durch »Null« dividiert; das würde auch passieren, wenn in der zweiten Formel ein Cosinuswert von »Null« eingegeben wird.

Da dies jedoch in der Mathematik nicht definiert ist, sind die Formeln für diese Eingabewerte nicht zu gebrauchen, ein Nachteil, der durch die direkten Winkelzuweisungen in Zeile 250 bis 360 behoben wird.

Wie beim ersten Programm wird der Ausgabewert auf drei Nachkommastellen genau ausgegeben; dies reicht für die Praxis allemal aus. Wer trotzdem mehr Stellen möchte, kann dies durch Veränderung von Zeile 110 bzw. 160 erreichen. Man schreibt dann z.B. einfach:

y=ROUND (y,5)

Zeile 200-240 beinhaltet, ähnlich wie das erste Programm, Phasenkorrektur-Operationen, die dazu dienen, daß z.B. auch negative eingegebene Sinusoder Cosinuswerte, Winkel ergeben, die mit den Ergebnissen eines Taschenrechners korrelieren . Hier wurde als »Referenz«, wie schon beim Programm »sincos1« der »TI 58« benutzt. Eine genaue Beschreibung dieser Phasenkorrekturterme würde den Rahmen dieses Aufsatzes sprengen; es sei daher der Kreativität eines jeden Lesers überlassen, diese Thematik (mit Hilfe der Programme und u.U. auch eines Mathematikbuches) zu durchdenken.

Wer möchte, kann sich das Ergebnis auch in Minuten-Sekunden-Darstellung ausgeben lassen. Das ist mittels eines einfachen Dreisatzes möglich, der dann auf die Umrechnungsformel führt (1 Grad = 60 Minuten = 3600 Sekunden).

Wichtig:

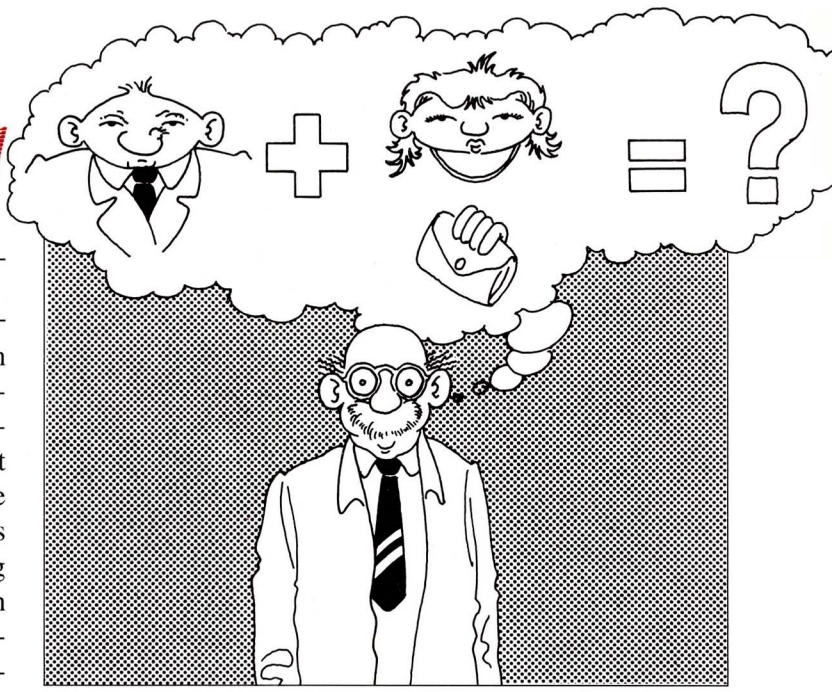
Es ist nur erlaubt, Sinus- bzw. Cosinuswerte zwischen »minus eins« und »plus eins« einzugeben. Das liegt einfach daran, daß beide Funktionen dort ihre Minima bzw. Maxima besitzen.

Wie ganz oben schon erwähnt, wird man die Programme wohl kaum isoliert laufen lassen, sondern vielmehr als Unterprogramme in größeren benutzen. Hier muß darauf geachtet werden, daß die Variablen nicht durch das Hauptprogramm mit Werten belegt sind, die den Ablauf des Unterprogrammes beeinträchtigen; notfalls muß man umbenennen.

(R.Bourdon)

```
LISTING >SINCOS1 <, REMARK = >REM<.

<45> 10 REM. PROGRAMMNAME: "SINCOS1"
<71> 20 ZONE 1
<69> 30 INPUT "ZUM BERECHNEN EINES WINKELS MITTELS SEIN
     ES SINUSWERTES (COSINUSWERTES),DRÜCKEN. SIE RETURN
      (GEBEN SIE BITTE EINE 1 EIN UND DRÜCKEN RETURN):"
     ,p.....
     ...
<69> 40 IF p=1 THEN GOTO 70
<37> 50 INPUT "BITTE GEBEN SIE DEN SINUS-WERT EIN : ",s
     inus
<63> 60 GOTO 90
<91> 70 INPUT "BITTE GEBEN SIE DEN COSINUS-WERT EIN:",c
     osinus
<16> 80 sinus=cosinus
<49> 90 IF sinus<0 THEN GOTO 400
<67> 100 i=1
<74> 110 j=1
<41> 120 Alpha = 0
<32> 130 Alphabog=Alpha*3.141592654#/180
<81> 140 w=SIN(Alphabog)
<63> 150 sinus=ROUND (sinus,5)
<61> 160 w=ROUND (w,5)
<51> 170 IF w=sinus THEN GOTO 260
< 4> 180  IF w<sinus THEN GOTO 240
<13> 190 Alpha=Alpha-(1/j)
<78> 200 k=1
<48> 210 j=j*10
<26> 220 i=k/j
<32> 230 GOTO 130
<83> 240 Alpha=Alpha+i
<36> 250 GOTO 130
<42> 260 Alpha=ROUND (Alpha,3)
<43> 270 IF q=0 THEN GOTO 290
<84> 280 Alpha=-Alpha
< 9> 290 IF p=1 THEN GOTO 320
<27> 300 PRINT "DER ZUM EINGEGEBENEN SINUSWERT ZUGEHÖRI
     GE WINKEL BETRÄGT : "
< 5> 310 GOTO 380
<79> 320 IF ABS(Alpha)>45 THEN GOTO 350
<40> 330 Alpha=ABS((Alpha-45))+45
< 0> 340 GOTO 370
<75> 350 Alpha=45-ABS((Alpha-45))
<18> 360 IF Alpha<0 THEN GOTO 430
<91> 370 PRINT "DER ZUM EINGEGEBENEN COSINUSWERT ZUGEHÖ
     RIGE WINKEL BETRÄGT :"
<61> 380 PRINT. Alpha ,"Grad"
<43> 390 END
<11> 400 q=1
<87> 410 sinus=-sinus
<98> 420 GOTO 100
<64> 430 Alpha=90-Alpha
< 1> 440 GOTO 370



LISTING >SINCOS2<, REMARK = >REM<.

<89> 10 REM BERECHNUNG EINES WINKELS MITTELS SEINES (BE
     KANNTEN)..... SIN
     US-WERTES BZW. COSINUS-WERTES.
<90> 20 REM PROGRAMMNAME: "SINCOS2"
< 8> 30 INPUT "GEBEN SIE DEN SINUS-WERT EIN. MÖCHTEN SI
     E DEN COSINUS-WERT EINGEBEN,. SO TIPPEN. SIE EINE
     2 EIN:",sinus
<56> 40 IF sinus=2 THEN GOTO 50 ELSE GOTO 90
<96> 50 INPUT "GEBEN SIE DEN COSINUS-WERT EIN :",cosinu
     s
<36> 60 IF cosinus = 0 THEN GOTO 350
<47> 70 IF ABS(cosinus)=1 THEN GOTO 300
<76> 80 GOTO 150
<54> 90 IF ABS(sinus)=1 THEN GOTO 250
< 1> 100 Alpha=(ATN(sinus/SQR(1-sinus^2)))*180/3.141592
     654#
<45> 110 Alpha=ROUND(Alpha,3)
<51> 120 PRINT " DER DEM EINGEGEBENEN SINUS-WERT ENTSPR
     ECHENDE WINKEL BETRÄGT:"
<57> 130 PRINT "ALPHA=",Alpha,"GRAD"
<31> 140 END
<26> 150 Alpha=(ATN(SQR(1-cosinus^2)/cosinus))*180/3.14
     1592654#
<55> 160 Alpha=ROUND(Alpha,3)
<15> 170  IF Alpha<0 THEN GOTO 200
<76> 180 PRINT " DER DEM EINGEGEBENEN COSINUS-WERT ENTS
     PRECHENDE WINKEL BETRÄGT:"
<63> 190 GOTO 130
<22> 200 IF ABS(Alpha)<45 GOTO 230
<83> 210 Alpha=(90-ABS(Alpha))+90
<85> 220 GOTO 180
<72> 230 Alpha=180- ABS(Alpha)
<89> 240 GOTO 180
< 6> 250 IF sinus=-1 THEN GOTO 280
<62> 260 Alpha=90
<95> 270 GOTO 180
<72> 280 Alpha=-90
< 0> 290 GOTO 180
<81> 300 IF cosinus=-1 THEN GOTO 330
<71> 310 Alpha=0
<86> 320 GOTO 180
<57> 330 alpha=180
<90> 340 GOTO 180
<46> 350 Alpha = 90
<94> 360 GOTO 180
```

CHI-QUADRAT-TEST



1. Was ist der CHI² Test

Der CHI² Test (x^2) ist ein Verfahren in der Statistik, um zu untersuchen ob zwischen zwei Beobachtungen und ihren in einer Stichprobe ermittelten Ausprägungen ein Zusammenhang besteht. Das Ergebnis des Tests liefert mit einer vorher gewählten Wahrscheinlichkeit entweder die Bestätigung einer Hypothese oder deren Widerlegung. Das heißt mit anderen Worten, daß man nach Durchführung des Tests mit soundsoviel prozentiger Sicherheit sagen kann, daß eine Aussage zutrifft, bzw. nicht zutrifft. (Statistisch wird das alles etwas anders ausgedrückt, oben Gesagtes hilft aber zum Verständnis bei der Anwendung).

2. Wo wird er eingesetzt

Zum Einsatz kann der Test überall dort kommen, wo die Prüfung einer bestimmten Hypothese eine statistische Auswertung von Zahlenmaterial nötig macht, und die Fragestellung es zuläßt.

Ein großes Anwendungsgebiet findet sich in der Arbeit von Marketingabteilungen in Unternehmen, genauer gesagt im Rahmen der dort durchgeführten Marktforschung.

Hier sehen sich die Verantwortlichen immer wieder vor die Aufgabe gestellt (richtige!) Entscheidungen zu treffen, die dem Unternehmen / den Produkten helfen sollen die erreichte Marktposition zu halten oder wenn möglich zu verbessern. Dazu ist es meistens nötig Informationen über den Markt einzuholen und auszuwerten. Hierzu werden (im Rahmen der Marktforschung) aus Kosten- und Zeitgründen Teilerhebungen (repräsentative Stichproben) durchgeführt, von deren Ergebnissen man Rückschlüsse auf eine Grundgesamtheit ziehen kann.

Bei der Untersuchung solcher Stichprobenergebnisse können nun aber auch neue Fragen auftauchen auf die man eine Antwort braucht, und hier kann z.B. der x^2 Test angewendet werden.

Beispiele sind

Bei einer Untersuchung für ein neues Produkt wird eine bestimmte Anzahl potentieller Verbraucher danach gefragt, wieviel sie für dieses Produkt zu zahlen bereit wären (dies dient zur Preisfindung). Die Antworten werden dann nach den verschiedenen Einkommenshöhen der Befragten unterteilt (klassifiziert) und in einer Tabelle dargestellt (dies ist die übersichtlichste Art komplexes Zahlenmaterial darzustellen). Nun kann sich die Frage stellen ob die genannten Preisvorschläge vom Einkommen der Befragten abhängen (oder der genannte Preis tatsächlich den Wert darstellt den das Produkt in den Augen des Befragten hat). Die sich hier ableitende Hypothese die mit Hilfe des x^2 Tests geprüft werden soll könnte lauten:

»Besteht eine Abhängigkeit zwischen Preisnennung und Höhe des Einkommens?«

Oder:

Im Rahmen einer Anzeigenanalyse soll untersucht werden ob eine Abhängigkeit besteht zwischen dem (besseren) Verständnis der Werbebotschaft und der Produktverwendung, d.h. man will herausfinden ob z.B. Produktverwender die Botschaft besser verstehen als die, die das Produkt nur kennen.

Hier kann die Frage lauten:

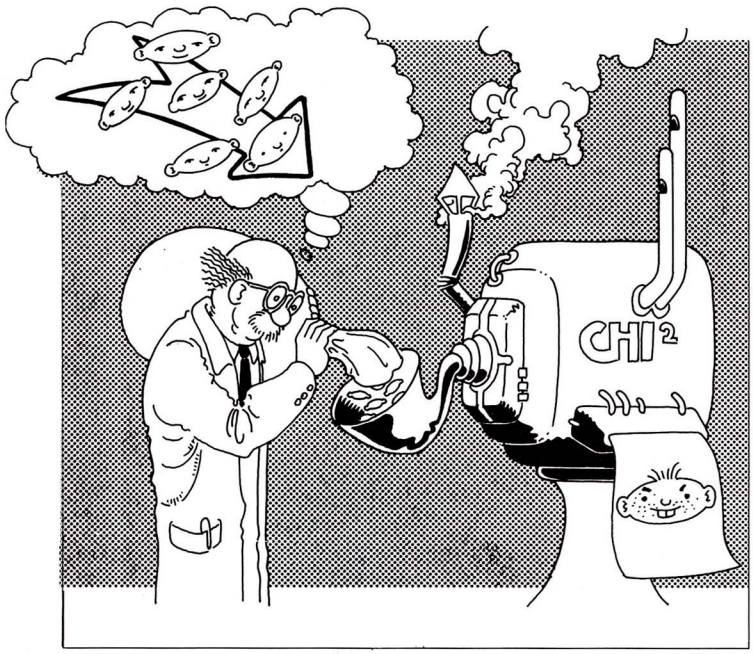
»Besteht Abhänigkeit zwischen Botschaftsverständnis und Produktverwendung (oder nicht)?«

(Das Ergebnis der Untersuchung könnte dann Einfluß auf die zukünftige Aussage in der Anzeige nehmen).

Ein anderes Beispiel wäre, nach einer Erhebung in der Männer und Frauen danach gefragt wurden ob sie viel oder wenig lesen festzustellen, ob man von dieser Stichprobe ausgehend allgemein sagen kann ob eine Abhängigkeit zwischen Leseverhalten und Geschlecht besteht.

3. Wie läuft er ab

Die Aufgabe des Tests ist die Überprüfung der Häufigkeitsverteilung eines Merkmals, die auf der Basis einer Stichprobe ermittelt wurde, gegenüber einer vorgegebenen /



erwarteten theoretischen Häufigkeitsverteilung (Überprüfung von Gesamtheitsstrukturen).

Anders: Es werden Verteilungen (als Stichproben) einander gegenübergestellt, und geprüft, ob sie der gleichen Grundgesamtheit mit einer bestimmten Wahrscheinlichkeit entstammen.

Dazu gibt es eine Prüfgröße, den x^2 Wert. Genauer gesagt sind es zwei Werte, zum einen das kritische x^2, zum anderen gibt es den x^2 Prüfwert, dessen Ermittlung im folgenden vorgestellt wird.

Dazu ein Beispiel:
Eine Stichprobe hat ergeben, daß von 700 befragten Männern 540 viel und 160 wenig lesen. Von den 300 befragten Frauen lesen 260 viel und 40 wenig. Nun stellt sich die Frage ob eine (UN-) Abhängigkeit zwischen Leseverhalten und Geschlecht besteht.

Als erstes werden die Ergebnisse in eine Tabelle eingetragen (Tabelle 1).

Diese Tabelle (hier der Spezialfall einer 4-FelderTafel) lässt sich ohne explizite Werte wie Tabelle 2 darstellen (Tabelle nicht ganz vollständig).

Hierbei bedeutet:

n = Größe der Stichprobe (meistens vorgegeben)

$n_{i,j}$ = absolute Häufigkeit für das Ereignis in der Stichprobe

$n_{i.}$ = Zeilensumme = Häufigkeit der Randverteilung der Randzeile

$n_{.j}$ = Spaltensumme = Haüfigkeit der Randverteilung (Randspalte)

i. = Zeilenindex, wobei der Punkt bedeutet daß j nicht besetzt ist

.j = Spaltenindex (analog i.)

Nun muß die zu überprüfende Hypothese aufgestellt werden Sagen wir einfach:

»Es besteht keine Abhängigkeit zwischen Leseverhalten und Geschlecht.«

Das ist unsere »Nullhypothese« (H_0), die wir immer so stellen wollen als wollten wir Unabhängigkeit nachweisen, so geschieht es auch im Programm; die Gegenhypothese (H_1) wäre:

»Es besteht Abhängigkeit zwischen Leseverhalten und Geschlecht.«

Als nächstes sollte man festlegen mit welcher Sicherheit die Aussage getroffen werden soll (Sicherheitsgrad). Dies geschieht meistens indem man den Wert nimmt, mit dem die Aussage nicht zutreffen soll (Fehlerwahrscheinlichkeit). [Alles in Prozent]. Wenn also die Aussage mit 95.00 %-iger Wahrscheinlichkeit zutreffen soll, sagt man die Fehlerwahrscheinlichkeit (genannt Alpha [α]) soll 5% betragen. Statistisch:

$\alpha = 0{,}05 \Rightarrow 1 - \alpha = 0.9500 = 1 - 0.05 = 0.9500$

Man geht von α aus, weil der Wert für das kritische x^2 von α und dem Freiheitsgrad abhängig tabelliert ist, und er sich bei Änderung dieser Werte ändert (Das liegt an der Art wie sich diese Werte statistisch verteilen).

Wir brauchen nun noch den Freiheitsgrad (r) um unser kritisches x^2 aus einer Tabelle ablesen zu können. Dies geschieht so:

$r = (\text{Spaltenzahl} - 1) * (\text{Zeilenzahl} - 1)$

Spalten- und Zeilenzahl beziehen sich auf die Tabelle (siehe unten). Hier: $r = (2-1)*(2-1) = \underline{1}$

Aus der Tabelle lesen wir nun $x^2_{5\%,1} = 3{,}84$ ab.

Der x^2 Prüfwert wird nach folgender Formel berechnet:

$$\Sigma\,\Sigma \frac{(n_{i,j} - e_{i,j})^2}{e_{i,j}}$$

Tabelle 1

Geschlecht → / Leseverh. ↓	Männer	Frauen	Σ
viel	540	260	800
wenig	160	40	200
Σ	700	300	1000
	Spaltensummen		

die invers dargestellten Werte sind Ergebnisse von Berechnungen

Zeilensummen

Tabelle 2

\ j / i \	1	2	i.
1	$n_{i,j}$	$n_{i,j}$	$n_{i.}$
	$e_{i,j}$	$e_{i,j}$	
2	$n_{i,j}$	$n_{i,j}$	$n_{i.}$
	$e_{i,j}$	$e_{i,j}$	
.j	$n_{.j}$	$n_{.j}$	n

Die Tabelle kann natürlich mehr Spalten und Zeilen haben

wobei die $n_{i,j}$ die absoluten Häufigkeiten sind, die wir in der Tabelle stehen haben und die $e_{i,j}$ Erwartungswerte sind die sich ergeben aus

$$e_{i,j} = \frac{n_{i.} * n_{.j}}{n}$$

Wenn man alle Werte in den Feldern der Tabelle ermittelt hat, braucht man nur noch die Werte der Spalte ganz rechts zu addieren um den x^2 Prüfwert zu erhalten, und dann müssen die beiden x^2-Werte miteinander verglichen werden. Dabei gilt:

Wenn der x^2 Prüfwert größer als der kritische x^2 Wert ist, dann wird die Hypothese der Unabhängigkeit verworfen. (Wegen dieser Bedingung wird in der Nullhypothese auch immer von einer Unabhängigkeit ausgegangen). Wäre es andersrum so könnte die Hypothese aufrecht erhalten werden.In unserem Fall ist

x^2 Prüf.} 11.904 größer als x^2 krit. = 3.84

deswegen müssen wir die oben aufgestellte Hypothese der Unabhängigkeit verwerfen, es besteht Abhängigkeit.

(Dieser Abschnitt erhebt keinen Anspruch auf absolute Vollständigkeit, sondern sollte nur einen kurzen Einblick in die Materie geben!)

4. Hinweise zum Programm und zur Benutzung

Das Programm leistet die Berechnung einer Tabelle mit bis zu 25 Zeilen und 8 Spalten, d.h. der eine Beobachtungspunkt kann maximal 25, der andere maximal 8 Unterteilungen haben. Wobei aber Berechnungen mit einem größeren Freiheitsgrad als 30 wegen der tabellierten x^2 Werte nicht durchgeführt werden können. Folgende Zeilen- und Spaltenkombinationen sollten beachtet werden:

Bei 8 Spalten max. 5 Zeilen
Bei 7 Spalten max. 6 Zeilen
Bei 6 Spalten max. 7 Zeilen
Bei 5 Spalten max. 8 Zeilen
Bei 4 Spalten max. 11 Zeilen
Bei 3 Spalten max. 16 Zeilen
Bei 2 Spalten max. 25 Zeilen

Es werden unter den gewählten Bedingungen (Fehlerwahrscheinlichkeit) die nötigen Berechnungen durchgeführt, und eine Tabelle sowie die Ergebnisse ausgegeben.

Dabei erscheint die Tabelle nach der Berechnung automatisch auf dem Bildschirm und die Ausgabe der Ergebnisse kann angewählt werden.

Unter der Ergebnisanzeige, (die die x^2 Werte, die Fehlerwahrscheinlichkeit, den Sicherheitsgrad, die Freiheitsgrade und einen entsprechenden Antwortsatz enthält) erscheint ein Menue, das weiter durch das Programm führt. Es enthält die Optionen Tabelle zeigen, Tabelle drucken, Ergebnisse drucken, Ergebnisse nochmal zeigen, neue Berechnung und Abbruch.

Die Druckausgabe der Tabelle bzw. der Ergebnisse erfolgt in Kompaktschrift (kann unabhängig voneinander erfolgen). Wenn erst die Tabelle und dann die Ergebnisse gedruckt werden erfolgt ein automatischer Seitenvorschub.

Wenn das Programm ordentlich beendet wird erfolgt ein Druckerreset. Die erforderlichen Eingaben sind:

die Beobachtungspunkte,
die Anzahl der Zeilen und Spalten der Tabelle,
die Fehlerwahrscheinlichkeit bzw. der Sicherheitsgrad,
die Unterteilungen der Beobachtungspunkte,
die Werte für die Felder der Tabelle.

Programmbenutzung (mit Beispiel)

Als erstes BASIC laden und dann das Programm mit >run "CHITEST< aufrufen.

Das Programm fragt dann nach dem ersten Beobachtungspunkt (seine Unterteilungen stehen in der Kopfspalte der Tabelle). Darauf wird nach dem zweiten Beobachtungspunkt gefragt (seine Unterteilungen stehen in der Kopfzeile der Tabelle).

Als nächstes werden die Anzahl der Zeilen (= Anzahl der Unterteilungen des ersten Beobachtungspunktes), und die Anzahl der Spalten (= Anzahl der Unterteilungen des zweiten Beobachtungspunktes) eingegeben.

Nun muß eingegeben werden wie groß die Fehlerwahrscheinlichkeit sein darf (bzw. wie groß der Sicherheitsgrad sein soll) mit der die Aussage getroffen wird. Dazu zeigt des Programm jeweils eine Liste mit den möglichen Werten die zur Auswahl stehen an.

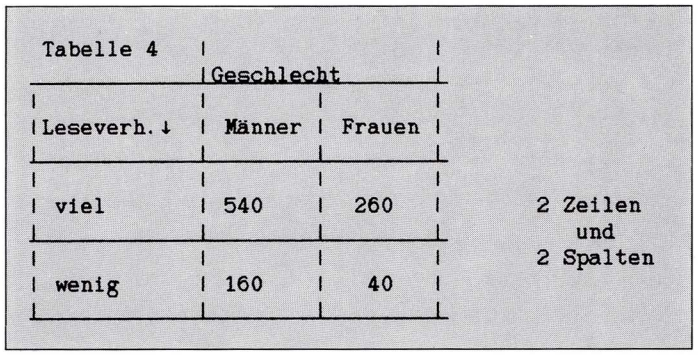
Als nächstes werden die Unterteilungen des Beobachtungspunktes der in der Kopfzeile steht eingegeben (das Programm fragt sie der Reihenfolge nach ab) [= Kopfzeile der Tabelle].

Wenn alle eingegeben sind fragt das Programm nach der ersten Unterteilung des zweiten Beobachtungspunktes (Kopfspalte).

Daruf werden die Werte eingegeben die im ensprechenden Feld stehen, wo sich die Zeile und die Spalte kreuzen

Tabelle 3

i,j	$n_{i,j}$	$\frac{n_{i.} * n_{.j}}{n} = e_{i,j}$	$(n_{i,j}-e_{i,j})$	$(n_{i,j}-e_{i,j})^2$	$\frac{(n_{i,j}-e_{i,j})^2}{e_{i,j}}$
1,1	540	800*700 / 1000 = 560	-20	-20*-20= 400	400 / 560 = 0,714
1,2	260	800*300 / 1000 = 240	20	20*20 = 400	400 / 240 = 1,667
2,1	160	200*700 / 1000 = 140	20	20*20 = 400	400 / 140 = 2,857
2,2	40	200*300 / 1000 = 60	-20	-20*-20= 400	400 / 60 = 6,667
					Σ = 11,904



(erste Zeile d. Tabelle 4). Danach fragt das Programm nach der nächsten Unterteilung des ersten Beobachtunspunktes, und darauf nach den Werten in der Zeile (zweite Zeile der Tabelle 4).

So geht es weiter bis alle Werte eingegeben sind. Dann rechnet das Programm einen Moment (je nach Datenmenge) und gibt die Tabelle die wir vorher eingegeben haben auf dem Bildschirm aus. Von hier an braucht man nur noch den Menues zu folgen um weiterzukommen.

Das klingt verwirrend ist aber ganz einfach, weil alles in der Reihenfolge eingegeben wird, wie es in der Tabelle steht. Das wird gleich im Beispiel deutlich.

Nehmen wir unser Beispiel von vorhin:
Eine Stichprobe hat ergeben, daß von 700 befragten Männern 540 viel und 160 wenig lesen. Von den 300 befragten Frauen lesen 260 viel und 40 wenig.

Als erstes werden die Ergebnisse in die Tabelle 4 eingetragen.

Unsere Hypothese lautet:
»Es besteht UNabhängigkeit zwischen Leseverhalten und Geschlecht.«
(Das Programm geht immer von UNabhängigkeit aus!!)

Nun fragt das Programm nach dem ersten Beobachtungspunkt und wir geben »Leseverhalten« ein. (Groß- oder Kleinschreibung ist egal, es werden alle Buchstaben in Großbuchstaben umgewandelt).

Danach wird gefragt:
Beobachtungspunkt 2 : ? und wir geben »Geschlecht« ein.

Als nächstes folgt die Frage:
Zeilenzahl der Tabelle (Unterteilung für LESEVERHALTEN) ?? : 2

Hier geben wir 2 ein, da wir zwei Unterteilungen (viel, wenig) und somit zwei Zeilen haben.

Analog bei der Spaltenzahl (Männer, Frauen):
Spaltenzahl der Tabelle (Unterteilungen für GESCHLECHT) ?? : 2

Nun fragt das Programm nach der Irrtumswahrscheinlichkeit (bzw. dem Sicherheitsgrad) und bietet eine Liste mit den möglichen Werten an. Wir wählen die Irrtumswahrscheinlichkeit »5%« (das entspräche alternativ dem Sicherheitsgrad 95 % in der anderen Auswahlliste)

Ab jetzt schreiben wir zeilenweise die Tabelle in das Programm.

Die erste Aufforderung lautet:

> Bitte 1. Unterteilung für GESCHLECHT eingeben:
> und wir geben aus der Tabelle »Männer« ein.

Dann folgt:

> Bitte die 2. Unterteilung für GESCHLECHT eingeben:
> »Frauen«

Jetzt haben wir die Kopfzeile eingegeben, und befinden uns in der letzten Spalte der Tabelle. Die nächste Aufforderung des Programms bezieht sich auf die nächste Zeile der Tabelle:

> Bitte 1. Unterteilung für LESEVERHALTEN eingeben:
> »viel«

Nun wird diese Zeile weiter abgefragt mit:

> Wert für VIEL/MÄNNER eingeben: ? : 540
> Wert für VIEL/FRAUEN eingeben: ? : 260

Bisher wurde nach der Eingabe der Werte der Bildschirm immer gelöscht. Wir sind hier aber wieder mit einer Zeile fertig und die letzte Aufforderung bleibt stehen. (Das ist günstig, da man sonst bei einer größeren Tabelle mal durcheinanderkommen könnte)

Darunter erscheint invers:

> Bitte 2. Unterteilung für LESEVERHALTEN eingeben:?
> »wenig«

(Wir befinden uns in der nächsten Zeile). Der Bildschirm wird wieder gelöscht und die Abfragen gehen weiter:

> Wert für WENIG/MÄNNER eingeben: ? : 160
> Wert für WENIG/FRAUEN eingeben: ? : 40

Nun sind alle Eingaben getätigt und der Rechner beginnt seine Arbeit. Kurz darauf erscheint die Tabelle und darunter die erste Auswahlmöglichkeit.

Die Ergebnisse lauten:

Stichprobe	=	1000 Elemente (Elemente deswegen da es nicht immer um Personen geht).
Irrtumswahrscheinlichkeit Alpha	=	5%
Sicherheitsgrad	=	95%
Freiheitsgrad	=	1
x^2 Prüf.	=	11.90476
x^2 krit.	=	3.84
Antwort	:	Die Hypothese der Unabhängigkeit muß verworfen werden, es besteht Abhängigkeit zwischen Leseverhalten und Geschlecht.

Probieren Sie ruhig mal; wenn man sich ein wenig mit dem Programm auseinandersetzt, ist es recht einfach zu handhaben.

5. Zum Programm

Ab Zeile 120 Eingabe der Beobachtungspunkte, Umwandlung in Großbuchstaben.

Ab Zeile 160 Eingabe der Zeilen- u. Spaltenzahl (Anzahl der Unterteilungen.

Ab Zeile 220 Ermittlung der Freiheitsgrade und Meldung zur Zeilen-, Spaltenkombination wenn 100 überschritten.

Ab Zeile 290 Dimensionierungen

Ab Zeile 330 Auswahl der Fehlerwahrscheinlichkeit bzw. Verzweigung zur Auswahl des Sicherheitsgrades (ab Zeile 1440) aus dem dann die Fehlerwahrscheinlichkeit ermittelt wird.

Ab Zeile 440 Eingabe der Tabelle. Dabei wird das erste Feld mit dem 1. Beobachtungspunkt belegt und auf 8 Zeichen + "." gekürzt.

Bei der Eingabe der Unterteilungen geschieht die Kürzung ebenfalls. Aus diesem Grund sollten bei den Unterteilungen sinnvolle Abkürzungen gewählt werden.

Die Eingabe der Zahlenwerte ist auf 8-stellige Ziffern begrenzt (zum einen bedingt durch die sonst nicht mögliche Bildschirmausgabe bei 8 Spalten, zum anderen ist das in der Praxis mehr als genug])

Ab Zeile 610 Berechnungen

Ab Zeile 660 Spaltensummen

Ab Zeile 750 Stichprobengröße aus Spaltensummen ermitteln.

Ab Zeile 810 Zeilensummen

Ab Zeile 900 Berechnung des x^2-Prüfwertes nach der oben gezeigten Formel.

Ab Zeile 1050 Tabelle auf Bildschirm

Ab Zeile 1190 Ergebnisse auf Bildschirm, und Menue (ab Zeile 1330)

Ab Zeile 1400 Enderoutine

Ab Zeile 1440 Auswahl des Sicherheitsgrades (⇒ Ermittlung der Fehlerwahrscheinlichkeit) als Alternative zu Zeile 330 ff.

Ab Zeile 1510 Dimensionierung der Tabelle mit den x^2-kritisch- Werten, und Datazeilen mit den Werten.

Ab Zeile 1910 x^2 kritisch lesen.

Ab Zeile 2020 Tastaturabfrage.

Ab Zeile 2080 Tabelle drucken.

Ab Zeile 2250 Ergebnisse drucken.

Ab Zeile 2350 Bildschirmkomandos.

Ab Zeile 2470 Druckerkomandos.

(T. Metzler)

```
LISTING >CHITEST<, REMARK = >REM<.

< 1> 10 '***********************************
< 8> 20 '* Programm zum Chi-Quadrat-Test *
<47> 30 '* (c) Thomas Metzler 1986..... *
< 4> 40 '***********************************
<86> 50 m=0
<17> 60 ZONE 10:GOSUB 2330
<55> 70 PRINT cls$;FNrahmeno$(30):PRINT seite$"Programm
     zum Chi-Quadrat-Test.";seite$:PRINT seite$;".....
     .... Thomas Metzler 1986 ";seite$:PRINT FNrahmenu$
     (30):FOR i=1 TO 4000:i=i+1:NEXT i
<82> 80 PRINT cls$ :PRINT " Chi-Quadrat-Test " :PRINT :
     PRINT "Mit einer Sicherheit von X % soll geprüft
     werden, ob ein Zusammenhang zwischen zwei " :PRINT
     : PRINT "Beobachtungen und ihren Ausprägungen bes
     teht."
<81> 90 PRINT : PRINT "(Z.B. Leseverhalten und Geschlec
     ht)": FOR i= 1 TO 3000:i = i +1 :NEXT i
<64> 100 GOSUB 1510
< 7> 110 IF m = 1 THEN PRINT cls$
<12> 120 PRINT cls$: INPUT "Beobachtungspunkt 1 :";b$(1
     ) :b$(1)=UPPER$(b$(1)):PRINT FNcp$(1,21);b$(1);era
     zeilrest$: PRINT :. INPUT "Beobachtungspunkt 2 :";
     b$(2):b$(2)=UPPER$(b$(2)):PRINT FNcp$(3,21);b$(2);
     erazeilrest$
<20> 130 :
<46> 140 '*************** Werte einlesen ***************
     **
<24> 150 :
<51> 160 PRINT :PRINT "Zeilenanzahl der Tabelle (Untert
     eilung für ";b$(1);") ?";:INPUT zeile:IF zeile >25
      THEN PRINT cls$;FN rahmeno$(38):PRINT seite$;"Zei
     lenanzahl zu groß, bitte maximal 25";seite$:PRINT
     FN rahmenu$(38):GOTO 160
```

```
<87> 170 PRINT :PRINT "Spaltenzahl der Tabelle (Unterte
     ilung für ";b$(2);") ?";:INPUT spalte: IF spalte >
      8 THEN PRINT cls$;FN rahmeno$(36):PRINT seite$;"S
     paltenzahl zu groß, bitte maximal 8";seite$:PRINT
     FN rahmenu$(36):GOTO 170
<52> 180. d=spalte
<32> 190 :
<93> 200 '******** Freiheitsgrade ermitteln ********
<17> 210 :
<64> 220 freigrad = (spalte - 1) * (zeile - 1):IF freig
     rad > 30 THEN GOTO 230 ELSE GOTO 290
<64> 230 PRINT:PRINT STRING$(50,"=");freigrad:PRINT "De
     r Freiheitsgrad hat 30 überschritten:":PRINT "Die
     Kombination von Zeilen und Spalten ist zu groß." :
     PRINT "Bei 8 Spalten max. 5 Zeilen"
<94> 240 PRINT "bei 7 Spalten max. 6 Zeilen":PRINT "bei
      6 Spalten max. 7 Zeilen":PRINT "bei 5 Spalten max
     . 8 Zeilen"
<72> 250 PRINT "bei 4 Spalten max. 11 Zeilen":PRINT "Be
     i 3 Spalten max. 16 Zeilen ":PRINT "Bei 2 Spalten
     25 Zeilen möglich": PRINT:PRINT "<W>= Weiterarbeit
     en, <E>= Ende":prompt$= "WwEe":GOSUB 2020:ON antwo
     rt GOTO 160,160,1380,1380
<27> 260 :
<19> 270 ' ####### Dimensionierung der Felder #########
     ###################
<31> 280 :
<93> 290 DIM feld$(zeile,spalte):DIM feld(zeile,spalte)
     :DIM summez(zeile):DIM erwartungswert(zeile,spalte
     ) :DIM summes(spalte)
<16> 300 :
<45> 310 '####### Auswahl der Irrtumswahrscheinlichkeit
      => Sicherheitsgrad mit dem die ########.... #####
     ## Hypothese zutreffen soll.....................
     ..... ########
```

```
<20> 320 :
<91> 330 PRINT cls$
<20> 340 PRINT : PRINT "Bitte die Irrtumswahrscheinlich
     keit Alpha wählen :" : PRINT "Alpha = 50.. % ": PR
     INT "Alpha = 30.. % " : PRINT "Alpha = 10.. % " :
     PRINT "Alpha =. 5.. % " : PRINT "Alpha =. 2.5 % "
     : PRINT "Alpha =. 1.. % " : PRINT "Alpha =. 0.1 %
     "
< 4> 350 INPUT "Falls Alpha unbekannt ist bitte die <RE
     TURN-Taste> drücken "; alpha$ :IF alpha$ = "" THEN
      GOSUB 1440 ELSE alpha = VAL (alpha$)
<81> 360 '.............. ##----> Sicherheitsgrad aus F
     ehlerwahrscheinlichkeit ermitteln ######
<19> 370 IF alpha = 50 OR alpha = 30 OR alpha = 10 OR a
     lpha = 5 OR alpha = 2.5 OR alpha = 1 OR alpha = 0.
     1 THEN GOTO 390 ELSE GOTO 380
<45> 380 PRINT cls$ : PRINT "Diese Eingabe ist ungültig
     !!" : PRINT. : GOTO 340
< 5> 390 PRINT : PRINT "Alpha = ";alpha : PRINT
< 1> 400 IF c=0 THEN sich.grad = 100-alpha ELSE sich.gr
     ad = 99.9
<19> 410 :
<39> 420 ' ### Eingabe Kopfzeile, Kopfspalte und Werte
     n der Tabelle ######################
<23> 430 :
<68> 440 FOR a = 0 TO zeile
<45> 450 FOR i = 0 TO spalte
<83> 460 IF a = 0 AND i = 0 THEN GOTO 500
<46> 470 IF a = 0 AND i >= 1 THEN GOTO 520
<40> 480 IF i = 0 AND a >= 1 THEN GOTO 540
<59> 490 IF a >= 0 AND i <= spalte THEN GOTO 560
<81> 500 PRINT cls$:feld$(a,i)=b$(1) :feld$(a,i)=UPPER$
     (feld$(a,i)) :feld$(a,i)=LEFT$(feld$(a,i),8)+"."
<27> 510 GOTO 580
<39> 520 PRINT "Bitte ";i;". Unterteilung für ";b$(2);"
      eingeben: ";:INPUT feld$(a,i) : feld$(a,i)=UPPER$
     (feld$(a,i)):PRINT FNcp$(i,39+LEN(b$(2)));feld$(a,
     i);erazeilrest$:IF LEN(feld$(a,i))>8 THEN feld$(a,
     i)=LEFT$(feld$(a,i),8)+"."
<31> 530 GOTO 580
<46> 540 PRINT invan$;:PRINT "Bitte ";a;". Unterteilung
      für ";b$(1);" eingeben: ";:INPUT feld$(a,i) :feld
     $(a,i)=UPPER$(feld$(a,i)):PRINT invaus$:IF LEN(fel
     d$(a,i))>8 THEN feld$(a,i)=LEFT$(feld$(a,i),8)+"."
<35> 550 GOTO 580
< 2> 560 PRINT cls$: PRINT "Wert für ";feld$(a,0);" / "
     ;feld$(0,i);" eingeben"; : INPUT feld$(a,i) :IF LE
     N(feld$(a,i))>8 THEN PRINT "Zahl zu groß, max. 8 S
     tellen! Moment!":FOR q=1 TO 1500:q=q+1:NEXT q:GOTO
      560
<39> 570 GOTO 580
< 6> 580 NEXT i
<27> 590 NEXT a
<19> 600 :
< 6> 610 '##### Berechnung der Randsummen ############
     ##########
<23> 620 :
<25> 630 :
<73> 640 '########### Spaltensummen. ##################
     **
<29> 650 :
<62> 660 FOR i = 1 TO spalte
<89> 670 FOR a = 1 TO zeile
< 7> 680 feld(a,i) = VAL (feld$(a,i))
<70> 690 summes(i) = summes(i) + feld(a,i)
<11> 700 NEXT a
<93> 710 NEXT i
<24> 720 :
<36> 730 '###### Stichprobengröße ermitteln ###########
     ####
<28> 740 :
<37> 750 FOR i = 1 TO spalte
<61> 760 sumstp = sumstp + summes(i)
< 6> 770 NEXT i
<36> 780 :
<42> 790 '########## Zeilensummen ####################
     #########
<21> 800 :
<79> 810 FOR a = 1 TO zeile
<56> 820 FOR i = 1 TO spalte
<71> 830 feld (a,i) = VAL (feld$(a,i))
<94> 840 summez(a) = summez(a) + feld (a,i)
< 3> 850 NEXT i
<24> 860 NEXT a
<35> 870 :
<44> 880 '###### Berechnung des Chiquadrat-Prüfwertes #
     #########
<39> 890 :
<78> 900 FOR a = 1 TO zeile
<55> 910 FOR i = 1 TO spalte
<70> 920 feld (a,i) = VAL (feld$(a,i))
<52> 930 erwartungswert(a,i) = (summez(a) * summes(i))
     / sumstp
<70> 940 differenz = feld (a,i) - erwartungswert(a,i)
<95> 950 differenzquadrat = differenz^2
<42> 960 quotient = differenzquadrat / erwartungswert (
     a,i)
<71> 970 chipruef = chipruef + quotient
<10> 980 NEXT i
<31> 990 NEXT a
<98> 1000 :
< 2> 1010 :
< 5> 1020 '###### Tabelle ausgeben ########
< 8> 1030 :
<11> 1040 :
< 8> 1050 spalte = d:ZONE 10:PRINT cls$;"Unabhängigkeit
     suntersuchung zwischen ";b$(1);" und ";b$(2):PRINT
      STRING$(9,CHR$(138))+b$(2)+STRING$(90-(9+LEN(b$(2
     ))),CHR$(138))
<12> 1060 FOR a = 0 TO zeile
<11> 1070 FOR i = 0 TO spalte
<14> 1080 PRINT feld$ (a,i),
<74> 1090 IF i = spalte THEN PRINT FN cp$(a+2,1)
<62> 1100 NEXT i
<88> 1110 NEXT a
<57> 1120 PRINT STRING$(90,CHR$(138)):PRINT "Ergebnisse
      ausgeben = E, ";:PRINT "Neue Berechnung = N ":pro
     mpt$="EeNn":GOSUB 2020
<59> 1130 ON antwort GOTO 1190,1190,1140,1140
<98> 1140 PRINT cls$:CLEAR:m=1:GOTO 60
<47> 1150 GOTO 1120
<19> 1160 :
<82> 1170 :'###### Ausgabe der Ergebnisse auf dem Bilds
     chirm ########################
<25> 1180 :
<97> 1190 PRINT invaus$;cls$ :PRINT FN rahmeno$(88):PRI
     NT seite$;"Unabhängigkeitsuntersuchung zwischen ";
     b$(1);" und ";b$(2) TAB(90)seite$:PRINT FN rahmenu
     $(88)
<91> 1200 PRINT FNcp$(7,0);"Stichprobe = ";sumstp;" Ele
     mente":IF sumstp = 0 THEN chipruef = 0
<78> 1210 PRINT FN cp$(7,50);"CHI-quadrat.... = ";chipr
     uef
<31> 1220 PRINT FN cp$(8,54),"prüf"
<10> 1230 PRINT FN cp$(8,0):PRINT "Irrtumswahrscheinlic
     hkeit Alpha = ";alpha;"%": PRINT "Sicherheitsgrad
     = ";sich.grad;"%" : PRINT "Freiheitsgrad =";freigr
     ad
<64> 1240 GOSUB 1910:' ########..................... -->
      CHI-Kritisch lesen..... #######
<56> 1250 PRINT FN cp$(10,50);"Chi..... = ";chikrit
     (freigrad,spalte)
< 4> 1260 PRINT FN cp$(11,51);".. kritisch"
< 8> 1270 PRINT:IF chikrit(freigrad,spalte) < chipruef
     THEN PRINT invan$;FN rahmeno$(88):PRINT seite$;"Di
     e Hypothese das Uabhängigkeit besteht muß verworfe
     n werden." TAB(90)seite$ :ELSE GOTO 1290
<64> 1280 PRINT seite$;"Es besteht ABHÄNGIGKEIT zwische
     n ";b$(1);" und ";b$(2) TAB(90)seite$:PRINT FN rah
     menu$(88);invaus$:GOTO 1330
<24> 1290 PRINT invan$;FN rahmeno$(88):PRINT seite$;" D
     ie Hypothese kann bei diesem Sicherheitsgrad aufre
     cht erhalten werden."TAB (90)seite$:PRINT seite$;"
     Es besteht UNABHÄNGIGKEIT zwischen ";b$(1);" und "
     ;b$(2) TAB(90)seite$:PRINT FN rahmenu$(88);invaus$

< 5> 1300 :
<85> 1310 '####### Menue ##############################
     ########################################
<11> 1320 :
<13> 1330 PRINT FN cp$(20,0) : PRINT invaus$;FN rahmeno
     $(22):PRINT seite$;"Neue Berechnung = N"TAB(24)sei
     te$:PRINT seite$;"Tabelle zeigen = T"TAB(24)seite$
     :PRINT seite$;"Ergebnisse zeigen = E" TAB(24)seite
     $
< 8> 1340 PRINT seite$;"Tabelle drucken = D" TAB(24)sei
     te$:PRINT seite$;"Ergebnisse drucken = P";seite$:P
     RINT seite$;invan$;" Ende = A ";invaus$ TAB(26)sei
     te$:PRINT FN rahmenu$(22):prompt$="NnTtDdAaPpEe":G
     OSUB 2020
<49> 1350 ON antwort GOTO 1360,1360,1050,1050,2060,2060
     ,1400,1400,2250,2250,1190,1190
< 9> 1360 PRINT cls$:CLEAR:m=1:GOTO 60
```

```
<26> 1370 :
<87> 1380 ' ######## Programmende ####################
     ########################################
<32> 1390 :
<23> 1400 PRINT cls$;invan$;FN cp$(25,10);FN rahmeno$(6
     ):PRINT FNcp$(26,10);seite$;" ENDE ";seite$:PRINT
     FNcp$(27,10);FNrahmenu$(6);invaus$;:LPRINT reset$;
     :END
<10> 1410 :
<93> 1420 ' ###### Irrtumswahrscheinlichkeit aus dem Si
     cherheisgrad ermitteln, umgekehrt ########.... ###
     ### zu Zeile.. ff. ########
<16> 1430 :
< 5> 1440 PRINT cls$ : PRINT. " Dann bitte den Sicherhe
     itsgrad eingeben ": PRINT "Der Sicherheitsgrad kan
     n 99.9 % " : PRINT TAB(26)"99.0 % ": PRINT TAB(26)
     "97.5 %":PRINT TAB(26)"95.0 %" : PRINT TAB(26)"90.
     0 %":PRINT TAB(26)"70.0 %" : PRINT TAB(26)"50.0 %
     "
< 3> 1450 PRINT TAB(26)"betragen", : INPUT sich.grad
<72> 1460 IF sich.grad = 99.9 THEN alpha=0.1 :c=1:RETUR
     N
<12> 1470 alpha = 100 - sich.grad: c = 0 :RETURN
<31> 1480 :
<46> 1490 ' ##### Tabelle mit den CHI-Quadrat Prüfwerte
     n ####################################
< 9> 1500 :
<43> 1510 DIM chikrit(30,7) : RESTORE 1520
<28> 1520 DATA .455,1.07,2.71,3.84,5.02,6.64,10.8
<75> 1530 DATA 1.39,2.41,4.61,5.99,7.38,9.21,13.8
<76> 1540 DATA 2.37,3.67,6.25,7.81,9.35,11.3,16.3
<23> 1550 DATA 3.36,4.88,7.78,9.49,11.1,13.3,18.5
<39> 1560 DATA 4.35,6.06,9.24,11.1,12.8,15.1,20.5
<77> 1570 DATA 5.35,7.23,10.6,12.6,14.4,16.8,22.5
< 9> 1580 DATA 6.35,8.38,12.0,14.1,16.0,18.5,24.3
<89> 1590 DATA 7.34,9.52,13.4,15.5,17.5,20.1,26.1
<53> 1600 DATA 8.34,10.7,14.7,16.9,19.0,21.7,27.9
<98> 1610 DATA 9.34,11.8,16.0,18.3,20.5,23.2,29.6
< 6> 1620 DATA 10.3,12.9,17.3,19.7,21.9,24.7,31.3
< 2> 1630 DATA 11.3,14.0,18.5,21.0,23.3,26.2,32.9
< 4> 1640 DATA 12.3,15.1,19.8,22.4,24.7,27.7,34.5
<48> 1650 DATA 13.3,16.2,21.1,23.7,26.1,29.1,36.1
<95> 1660 DATA 14.3,17.3,22.3,25.0,27.5,30.6,37.7
<14> 1670 DATA 15.3,18.4,23.5,26.3,28.8,32.0,39.3
<19> 1680 DATA 16.3,19.5,24.8,27.6,30.2,33.4,40.8
<73> 1690 DATA 17.3,20.6,26.0,28.9,31.5,34.8,42.3
<14> 1700 DATA 18.3,21.7,27.2,30.1,32.9,36.2,43.8
<33> 1710 DATA 19.3,22.8,28.4,31.4,34.2,37.6,45.3
<16> 1720 DATA 20.3,23.9,29.6,32.7,35.5,38.9,46.8
< 3> 1730 DATA 21.3,24.9,30.8,33.9,36.8,40.3,48.3
<79> 1740 DATA 22.3,26.0,32.0,35.2,38.1,41.6,49.7
<41> 1750 DATA 23.3,27.1,33.2,36.4,39.4,43.0,51.2
<63> 1760 DATA 24.3,28.2,34.4,37.7,40.6,44.3,52.6
< 6> 1770 DATA 25.3,29.2,35.6,38.9,41.9,45.6,54.1
< 5> 1780 DATA 26.3,30.3,36.7,40.1,43.2,47.0,55.5
<24> 1790 DATA 27.3,31.4,37.9,41.3,44.5,48.3,56.9
<87> 1800 DATA 28.3,32.5,39.1,42.6,45.7,49.6,58.3
<38> 1810 DATA 29.3,33.5,40.3,43.8,47.0,50.9,59.7
<47> 1820 FOR freigrad = 1 TO 30
<27> 1830 FOR spalte = 1 TO 7
<70> 1840 READ chikrit(freigrad,spalte)
<63> 1850 NEXT spalte
<67> 1860 NEXT freigrad
<16> 1870 RETURN
<39> 1880 :
<72> 1890 ' ###### Lesen Chi-Kritisch bei bestimmten Fr
     eiheitsgrad und bestimmter Fehler- #####..... ###
     ### wahrscheinlichkeit. (Angesprungen von 1180 )..
     ..... #####
<17> 1900 :
<64> 1910 IF alpha = 50. THEN spalte = 1
<68> 1920 IF alpha = 30. THEN spalte = 2
<72> 1930 IF alpha = 10. THEN spalte = 3
<43> 1940 IF alpha = 5.. THEN spalte = 4
<64> 1950 IF alpha = 2.5 THEN spalte = 5
<51> 1960 IF alpha = 1.. THEN spalte = 6
<93> 1970 IF alpha = 0.1 THEN spalte = 7
<21> 1980 RETURN
<44> 1990 :
<28> 2000 '###### Tastaturabfrage ####################
     ########################################
< 3> 2010 :
<66> 2020 antwort$ = INKEY$:WHILE antwort$="" :antwort$
     =INKEY$:WEND
<76> 2030 antwort=INSTR(prompt$,antwort$):IF antwort=0
     THEN GOTO 2020
<91> 2040 RETURN
<15> 2050 :
<57> 2060 '####### Tabelle drucken ###################
     ######################################
<21> 2070 :
<15> 2080 PRINT cls$;"Tabelle wird gedruckt"
<72> 2090 LPRINT :spalte = d:
<35> 2100 ZONE 10:PRINT invan$;STRING$(90,CHR$(138))
<76> 2110 LPRINT kompaktan$;FN links$(30)
<32> 2120 LPRINT STRING$(9,"=")+b$(2)+" -> "STRING$(90-
     (12+LEN(b$(2))),"=")
< 6> 2130 FOR a = 0 TO zeile
< 5> 2140 FOR i = 0 TO spalte
<60> 2150 LPRINT feld$ (a,i),
<56> 2160 IF i = spalte THEN LPRINT
<84> 2170 NEXT i
<11> 2180 NEXT a
<42> 2190 LPRINT STRING$(90,"=")
<74> 2200 LPRINT :LPRINT :LPRINT kompaktaus$:PRINT inva
     us$
< 7> 2210 p=1:GOTO 1330
<10> 2220 :
<74> 2230 '###### Ergebnisse drucken ##################
     ######################################
<16> 2240 :
<81> 2250 PRINT invaus$ : PRINT cls$;"Ergebnisse werden
      gedruckt": LPRINT kompaktan$;FN links$(30);"Unabh
     ängigkeitsuntersuchung zwischen ";b$(1);" und ";b$
     (2):LPRINT:LPRINT:LPRINT "Stichprobe = ";sumstp;"
     Elemente " TAB (50)"Chi-quadrat.... = ";chipruef
<74> 2260 LPRINT TAB(50)"..... prüf":PRINT STRING$
     (90,CHR$(138))
<96> 2270 GOSUB 1910 :LPRINT:LPRINT "Irrtumswahrscheinl
     ichkeit Alpha = ";alpha;"%" TAB(50)"Chi..... = "
     ;chikrit(freigrad,spalte): LPRINT "Sicherheitsgrad
      = ";sich.grad;"%" TAB(50)".. kritisch": LPRINT "F
     reiheitsgrad =";freigrad
<62> 2280 LPRINT : IF chikrit(freigrad,spalte) < chipru
     ef THEN LPRINT "Die Hypothese das Uabhängigkeit be
     steht muß verworfen werden.":LPRINT "Es besteht. A
     BHÄNGIGKEIT. zwischen ";b$(1);" und ";b$(2) :LPRIN
     T kompaktaus$: GOTO 2300
<97> 2290 LPRINT " Die Hypothese kann bei diesem Sicher
     heitsgrad aufrecht erhalten werden.":LPRINT "Es be
     steht UNABHÄNGIGKEIT zwischen ";b$(1);" und ";b$(2
     ) :LPRINT kompaktaus$:GOTO 2300
<14> 2300 IF p=1 THEN LPRINT form$
<42> 2310 PRINT cls$ :GOTO 1330
<12> 2320 :
<61> 2330 '######## BILDSCHIRMKOMANDOS ################
     ######################################
<18> 2340 :
<45> 2350 esc$=CHR$(27)
<91> 2360 cls$=esc$+"E"+esc$+"H":'Bildschirm löschen, C
     ursor oben links
<21> 2370 invan$=esc$+"p":'invers an
<49> 2380 invaus$=esc$+"q":' invers aus
<82> 2390 erazeilrest$=esc$+"K":' zeilenrest löschen
<90> 2400 seite$=CHR$(133):' ergibt
<94> 2410 DEF FN rahmeno$(x)=CHR$(134)+STRING$(x,CHR$(1
     38))+CHR$(140):' rahmenoberteil mit längex zwische
     n den ecken
<60> 2420 DEF FN rahmenu$(x)=CHR$(131)+STRING$(x,CHR$(1
     38))+CHR$(137):' rahmenunterteil analog.. oberteil
<96> 2430 DEF FN cp$(z,s)=esc$+"Y"+CHR$(32+z)+CHR$(32+s
     ):' cursor in zeile z und spalte s setzen
<20> 2440 :
<20> 2450 ' ###### DRUCKERKOMANDOS ##################
     ######################################
<26> 2460 :
<77> 2470 DEF FN links$(x)=CHR$(27)+"l"+CHR$(x):'liker
     rand x zeichen momentaner schrift
<55> 2480 form$= CHR$(12):' seitenvorschub
<28> 2490 reset$=CHR$(27)+"§"
<71> 2500 kompaktan$=CHR$(15):'17 zeichen/zoll an
<20> 2510 kompaktaus$=CHR$(18):'17 zeichen/zoll aus
<40> 2520 unterstran$=CHR$(27)+"-"+"1":' unterstreichen
      an
<20> 2530 unterstraus$=CHR$(27)+"-"+"0":' unterstreiche
     n aus
<87> 2540 IF m=0 THEN RETURN ELSE PRINT "Moment bitte":
      GOTO 100
```

JOYCE DATABOX

— Das ist die Software zum Sonderheft —

DATABOX:

— mehr als der übliche Softwareservice

— bringt alle Programme des Joyce-Sonderheftes auf 3" -Diskette

— trägt das Titelbild des gleichzeitig erscheinenden Sonderheftes.



Für alle PCW's als 3"-Diskette.

Inhalt der Databox zu Joyce-Sonderheft 1/87

Seite A:
Assembler
Disassembler
Chi-Quadrat-Test
Differential/Integralrechnung
Diskettenordner
Testprogramm
für Netzteilumbau
Maskengenerator
Sinus/Cosinusberechnung
Kurvendiskussion
Druckereinstellung
Prüfsummenprogramm
Umwandlung in ASCII-Dateien
XBIOS (12 COM-Dateien)

Seite B:
Mau-Mau
LOGO-Prozeduren
u.a. Diagramm, Torte

Einzelbezugspreise für DATABOX:
Diskette 3'' 30,– DM zuzüglich 3,– DM Porto/Verpackung (im Ausland zuzüglich 5,– DM Porto/Verpackung).

Bitte benutzen Sie den Bestellcoupon!

DMV

Daten & Medien Verlagsges. mbH

Postfach 250

3440 Eschwege

Name:.....

Anschrift: .....

Hiermit bestelle ich die Databox zum Joyce-Sonderheft 1/87

☐ Diskette 30,– DM

zuzüglich 3,– DM Porto/Verpackung
(im Ausland zuzüglich 5,– DM Porto/Verpackung)

☐ bitte liefern Sie per NN (nur innerhalb der BRD)

☐ V-Scheck liegt bei

ASSEMBLER/ DISASSEMBLER

Der (die?) Schneider JOYCE wird mit zwei Assemblern ausgeliefert. MAC.COM und RMAC.COM auf Seite drei der Systemdisketten. Warum also noch einen Assembler schreiben? Das Betriebssystem CP/M wurde für den Prozessor 8080 geschrieben, also müssen sich die unter CP/M laufenden Programme an den Befehlssatz des 8080 halten. Im JOYCE ist aber eine Z80-CPU eingebaut, diese hat einen gegenüber dem 8080 stark erweiterten Befehlssatz, der aber alle Befehle des 8080 beinhaltet. So können Programme, die für den 8080 geschrieben wurden auch auf dem Z80 laufen. Die Nutzung der zusätzlichen Befehle des Z80 mit Hilfe eines 8080-Assemblers ist, wie man sich sicher denken kann, nur auf Umwegen möglich, weiterhin werden die Befehle die beide Prozessoren kennen von 8080- und Z80-Assemblern unterschiedlich benannt. Es gibt also einige Hürden zu überwinden, wenn man ein Z80-Programm mit einem 8080-Assembler übersetzen will. Ein Weg diese Hürden zu umgehen ist, einen Assembler für den Z80 zu schreiben.

Es gibt schon einige in BASIC geschriebene Assembler für den Z80-Prozessor, diese haben aber oft den Nachteil, daß sie für das eigentliche Maschinenprogramm und den zugehörigen Quelltext verhältnismäßig wenig Platz lassen. Auf dem CPC 464 ist es sicher auch der einzig vernünftige Weg Programm, Quelltext, erzeugtes Maschinenprogramm und Tabellen gleichzeitig im Speicher zu haben, vor allem dann, wenn man mit dem Datenrecorder und nicht mit einem Diskettenlaufwerk arbeitet. Auf dem JOYCE wird aber ein Großteil des Speichers schon durch den BASIC-Interpreter belegt, der ja im Gegensatz zu den CPC's von Diskette geladen werden muß. Für größere Quelltexte bleibt da kein Platz mehr. Also lege man Quelltext, Maschinenprogramm und Tabellen auf der Diskette ab, mit der Verwaltung von Direktzugriffsdateien wie das Mallard-BASIC sie bietet kein großes Problem. Zumindest für die Zwischendateien, die am Programende sowieso wieder gelöscht werden nimmt man natürlich zweckmäßigerweise die RAM-Disk (Laufwerk M:).

Den Quelltext für den Assembler erstellt man, sofern er 200 Zeilen nicht überschreitet, am einfachsten mit dem BASIC-Editor RPED.BAS. Ist der Text länger, dann gehts am einfachsten mit LocoScript. Da der Assembler einen LocoScript-Text aber nicht direkt verarbeiten kann, muß dieser als ASCII-Text gespeichert werden. (Option f7 = Modus in der Diskverwaltung). Wer will kann natürlich auch den CP/M Editor ED.COM benutzen, das spart zwar das lästige Rücksetzen des Computers (SHIFT/EXTRA/EXIT) beim Wechsel von LocoScript zu CP/M, aber ED ist halt nicht sehr komfortabel. Jeder andere Texteditor ist natürlich auch geeignet, soweit er eine ASCII-Datei erzeugen kann.

```
run"assem

---------------------------
  Z 8 0 - A S S E M B L E R
 ©1986 by AndreB Freystatzky
---------------------------

Bitte Dateinamen (,DIR oder #) eingeben : #
Bildschirmlisting : Ja
        Listdatei : Ja
        Zieldatei : .COM
     Ziellaufwerk : M:
Bitte Dateinamen (,DIR oder #) eingeben : asstest.asc
***** Fehler Nr.: 15 - Nicht erlaubter Vorwärtsbezug
  10 0100 00000000 10 ORG A000H
  20 0104 00          NOP
  30 0105 00          NOP
  40 0106 00          NOP
  50 0107 C9          RET


Labeltabelle :


Programm : ASSTEST
  Länge  : 9 Bytes
  Start  : 0100H

 1 Fehler
Ok
```

Der Assembler erwartet die Zeilen des Quelltextes in folgender Form:

nr label befehl operand ;kommentar

Hierbei bedeuten

nr:
Eine Zeilennummer, die der Assembler z.B. bei einer Fehlermeldung mit ausgibt, um das Auffinden des Fehlers zu erleichtern. Die Angabe einer Zeilennummer ist optional, das heißt sie muß nicht angegeben werden. Fehlt sie, so numeriert der Assembler die Zeilen fortlaufend durch.

label:
Ein symbolischer Name für die aktuelle Programmadresse. Die angabe eines Labels ist ebenfalls optional. Ein Label wird normalerweise benutzt, um das Ziel eines Sprunges oder Unterprogrammaufrufes zu kennzeichnen.

befehl:
Ein gültiger Z80-Befehl oder eine Assembler-Pseudo-Anweisung (s.u.).

operand:
Ein oder mehrere durch Komma getrennte Operanden, soweit der vorstehende Z80-Befehl oder die Assembler-Pseudo-Anweisung diese benötigt.

kommentar:
Der mit einem Semikolon (»;«) eingeleitete Kommentar dient dazu, dem Programmierer die Maschinenbefehle zu erläutern, vom Assembler selbst wird der Kommentar überlesen.
Als Trennzeichen zwischen den einzelnen Elementen der Zeile sind beliebig viele Lerrzeichen und/oder TAB's erlaubt. Am Zeilenanfang ist kein Leerzeichen nötig.
Benötigt ein Z80-Befehl zwei Operanden, so sind diese durch ein Komma getrennt einzugeben. Zwischen den

```
run
----------------------------
     Z80-DISASSEMBLER
©1986  by AndreE Freystatzky
----------------------------
RAM oder Disk ... : RAM
Bildschirmlisting : JA
Listdatei ....... : JA
Name der Datei .. : test
Druckerlisting .. : NEIN
Startadresse .... : &Hc000
Endadresse ...... : &Hcfff
Datei : --RAM--   Start : &HC000   Ende : &HCFFF

C000  00          NOP
C001  7A          LD    A,D
C002  65          LD    H,L
C003  69          LD    L,C
C004  6C          LD    L,H
C005  E5          PUSH  HL
C006  2C          INC   L
C007  18 19       JR    $+27 )&HC022
C009  29          ADD   HL,HL
C00A  00          NOP
C00B  0D          DEC   C
C00C  00          NOP
C00D  00          NOP
C00E  3C          INC   A
C00F  09          ADD   HL,BC
C010  20 20       JR    NZ,$+34 )&HC032
C012  9C          SBC   H
```

den Operanden und dem Komma dürfen keine(!) Leerzeichen stehen.

Ob die Befehle und Anweisungen in Groß- oder in Kleinbuchstaben eingegeben werden bleibt dem Anwender überlassen. Das Programm unterscheidet nicht zwischen Groß- und Kleinschreibung und wandelt intern alle Buchstaben in Großbuchstaben um.

Der Assembler verarbeitet den gesamten Umfang der Z80-Befehle, sowie einige Pseudo-Anweisungen. Das Befehlsformat entspricht im wesentlichen dem von Zilog angegebenen, mit einer Ausnahme, die allerdings nicht sehr gravierend ist. Der Befehl > EX AF,AF' < muß in der Form > EX AF,AF' < eingegeben werden, der Apostroph(Shift #) am Ende muß also weggelassen werden. Dies vereinfacht für das Programm die Erkennung von Text der als Zeichenkette verarbeitet werden soll und durch eben diesen Apostroph am Anfang und am Ende dieses Textes kenntlich gemacht wird.

Nach dem Start des Assemblers meldet sich dieser nach einer kurzen Wartezeit mit der Aufforderung:

Bitte Dateinamen (DIR, oder #) eingeben :

Die Eingabe von »DIR« gibt das Inhaltsverzeichnis der Diskette aus. Falls gewünscht, kann die Anzeige durch Eingabe eines nachfolgenden Dateinamens, der die CP/M-üblichen Wildcards enthalten darf, eingeschränkt werden. z.B.:

'dir prog??.*'

Die Eingabe von »#« führt in ein Unterprogramm, in dem die Grundeinstellungen des Assemblers geändert werden können. Ändern kann man:

1) Bildschirmlisting	Ja/Nein
2) Listdatei	Ja/Nein
3) Zieldatei	.COM/.DAT
4) Ziellaufwerk	A:/M:

Bei den Optionen mit Ja/Nein Antwort wird durch die Taste »+« oder »[+]« das »Ja« gesetzt, durch »−« oder »[−]« das »Nein« und mit allen anderen Tasten bleibt die vorgegebene Antwort unverändert. Bei den Optionen 3) und 4) wird durch die Taste »−« oder »[−]« die jeweils andere Möglichkeit ausgewählt, alle anderen Tasten bewirken keine Änderung.

Wird bei Punkt 3) eine ».COM«-Datei als Zieldatei gewählt, so erzeugt der Assembler eine Datei, die so wie sie ist direkt ausgeführt werden kann (wie es sich für eine .COM-Datei gehört). Wählt man eine ».DAT«-Datei, so erzeugt der Assembler eine Datei, in der durch Kommata getrennt die Werte der einzelnen Bytes in BASIC-lesbarer Form angegeben sind. Diese Datei kann z.B. von einem BASIC-Programm per »INPUT #« gelesen und in den Speicher »gepokt« werden.

Wird nach der Eingabeaufforderung ein Dateiname eingegeben, so sucht das Programm diese Datei und beginnt mit der Assemblierung. Wird der Dateiname ohne Extension (das ist der Teil nach dem Punkt) eingegeben, dann wird der Name automatisch durch ».Z80« ergänzt.

Für die Angabe von Operanden im Quelltext bestehen folgende Möglichkeiten:

– 1234
Dezimalzahlen, keine besondere Kennzeichnung nötig.

– &H12F oder – 0AB1H
Hexadezimalzahlen. In der zweiten Schreibweise ist zu beachten, daß das erste Zeichen eine Ziffer sein muß, ggf. ist eine führende Null anzugeben.

– 'A'
ASCII-Zeichen, das oder die Zeichen müssen in Apostrophe eingeschlossen werden. Die Zeichen werden durch den ihnen in der ASCII-Zeichentabelle zugeordneten Wert ersetzt.

– name
Ein beliebiger Name mit bis zu sechs Zeichen Länge. Der Name darf nicht mit dem Name eines Z80-Befehls übereinstimmen. Dem Namen muß irgendwann innerhalb des Quelltextes ein Wert zugeordnet werden, aber nicht unbedingt bevor er benutzt wird.

– $
Das $-Zeichen zur Kennzeichnung der aktuellen Programmadresse.

–
Eine beliebige Summe oder Differenz aus den oben genannten Möglichkeiten z.B. > &H80+'z'-offset <. Einschränkend gilt hier, daß ein in einer Summe auftretender Name schon zum Zeitpunkt der Übersetzung einen definierten Wert haben muß, die Zuweisung eines Wertes zu diesem Namen (im Bsp. offset) muß also im Quelltext vor der Summenbildung erfolgen.

Die zur Verfügung stehenden Assembler-Pseudo-Anweisungen

– ORG wert
Dient dazu den Programmzähler am Programmbeginn auf einen bestimmten Wert zu setzen, die Startadresse des Programms. Wird keine ORG-Anweisung gegeben, so nimmt dieser Assembler eine Startadresse von &H100 an, die Startadresse von CP/M-Programmen.

– EQU
In der Form > name EQU 1234 <. d.h. Dem Namen »name« wird ein bestimmter Wert zugeordnet (in diesem Fall 1234). Der Wert kann wie alle Operanden in einer der oben genannten Formen angegeben werden, kann also eventuell auch ein Name sein (dieser müßte dann allerdings schon einen festen Wert haben).

– DEFB 0FH oder – DB 122
Damit wird dem nächsten Byte im Programm ein bestimmter Wert zugewiesen (hier 0FH=15 bzw. 122). Der Operand kann natürlich auch hier in jeder der oben genannten Formen angegeben werden.

– DEFW &H0F3C oder – DW wert
Hiermit wird dem nächsten Wort (zwei Byte) ein Wert zugewiesen, dabei wird das niederwertige Byte zuerst im Speicher abgelegt, wie dies auch bei allen Z80-Befehlen die mit 2Byte-Worten arbeiten üblich ist.

– DEFM 'Text' oder – DM 'Hallo'
Um einen Text im Speicher abzulegen, den man z.B. von einem Maschinenprogramm aus ausgeben will.

– DEFS 17 oder – DS 0FH
Dieser Befehl schreibt nicht die dahinter angegebene Zahl in den Speicher, sondern reserviert an dieser Stelle im Programm die angegebene Zahl an Speicherzellen (im Bsp. 17 bzw. 0FH=15 Byte). Dies ist z.B. wichtig um im Programm Platz für Daten zu schaffen, denen man aber noch keinen bestimmten Wert zuweisen will.

– END
Dieser Befehl sagt dem Assembler, daß er das Ende des Programms erreicht hat. Text der jetzt eventuell noch folgt braucht nicht beachtet zu werden. Achtung! Dieser Befehl entspricht nicht dem END in BASIC, was danach folgt wird wirklich nicht beachtet und auch nie übersetzt.

Die wichtigsten Variablen des Programms in einer Übersicht.

esc$,bel$:
Bildschirmsteuerzeichen

bf$(5):
In diesen Textvariablen stehen die Namen sämtlicher Z80-Befehle und Pseudo-Anweisungen drin.

befi(12):
Die Werte der 12 Befehle die in bf$(2) stehen, diese erzeugen alle nur ein Byte und benötigen keine Operanden.

befii(20):
Die Werte der Befehle in bf$(3). : Das zweite Byte dieser Befehle das erste Byte ist bei allen &HED.

reg$(7):
Die Namen der 8-bit-Register.

red$(3):
Die Namen der 16-bit-Register (außer IX,IY,SP).

bed$(7):
Die Namen der Sprungbedingungen.

femax:
Die Anzahl der verschiedenen Fehlermeldungen.

fe$(..):
Ebendiese.

vlist,dlist:
Bildschirm-, Datei- Listing Ja/Nein.

comdat:
Gibt an, ob eine ».COM«- oder eine ».DAT«-Datei gewünscht wurde.

lwz$:
Das Ziellaufwerk.

lwtemp$:
Das Laufwerk, auf das die temporären Dateien sollen, die am Programmende wieder gelöscht werden.

feanz:
Anzahl der aufgetretenen Fehlermeldungen.

zeile:
Die Zeilennummer im Quelltext.

lzeil:
Die Zeilennummer in der Listdatei, falls eine da ist.

adr;:
Die aktuelle Adresse im Programm.

start;:
Die Startadresse des Programms

a$(3):
Da kommt die gelesene Zeile rein, wenn sie aufgespalten wird.

byte(4):
Da kommen die bis zu vier Bytes rein, aus denen ein einzelner Z80Befehl bestehen kann.

datq,datz...:
Die Nummern der Dateien.

datq$...:
Deren Namen.

lbez$:
Labeldateipuffer für Labelnamen.

lwrt$:
-- " -- für Labelwert.

lprv$:
-- " -- Zeiger auf den vorhergehenden Eintrag.

lnxt$:
-- " -- Zeiger auf den folgenden Eintrag.

u...:
So ähnlich für die Datei der undefinierten Label.

lang:
Befehlslänge in Byte.

ds:
Anzahl der freizuhaltenden Bytes im DEFS-Befehl.

zei$,zeile$:
Inhalt der gelesenen Zeile.

bfnr:
Die Nummer des aktuellen Befehlsstrings bf$(bfnr).

zeiger:
Zeiger auf den Befehl innerhalb von bf$(..).

xypos:
Zeiger auf die Position von »IX« bzw. »IY« innerhalb des Operanden.

op1$:
Erster Operand, falls der Befehl zwei hat.

op2$:
Dann der zweite Operand.

Dies sind noch lange nicht alle Variablen im Programm. Ich hoffe aber, daß die anderen sich durch ihre Namen teilweise selbst erklären, oder der Sinn der Variablen aus dem Zusammenhang deutlich wird. Die kurzen Variablennamen, die aus ein oder zwei (oft gleichen) Buchstaben bestehen haben oft nur Zwischenspeicheraufgaben und sind nicht im ganzen Programm gültig.

Der vorliegende Disassembler erlaubt es, den Speicherinhalt des Rechners, oder den Inhalt einer Programmdatei zu disassemblieren. Es bestehen mehrere Möglichkeiten, das Listing auszugeben. Die Ausgabe kann auf dem Bildschirm, in eine Datei und/oder auf den Drucker erfolgen. Das Programm fragt die gewünschte Konfiguration zu Beginn ab. Zu den einzelnen Fragen gibt das Programm jeweils eine mögliche Antwort vor. Die Antwort kann entweder durch eine der Tasten »+«, »[+]«, »J«, »Y«, <RETURN>' oder Leerzeichen angenommen, oder durch »−«, »[−]« oder »N« abgelehnt werden.

Bei Adresseingaben kann entweder eine Adresse angegeben werden, oder der Vorschlag des Programms durch drücken der <RETURN>-Taste angenommen werden.

Der Abbruch der laufenden Ausgabe ist jederzeit mit Hilfe der <EXIT>-Taste möglich. Beim Abbruch, oder wenn der angegebene Bereich abgearbeitet ist, fragt das Programm, wie es weitergehen soll. Abhängig von der Antwort geschieht folgendes:

− 1) neuer Bereich
Das Programm erfragt einen neuen Adressbereich. Die Ausgabe auf Diskette erfolgt, wenn eine Listdatei gewünscht ist, in dieselbe Datei.

− 2) neue Datei
Eine eventuell vorhandene Listdatei wird geschlossen, das Programm erfragt alle Optionen neu. Achtung! Wenn für die nächste Listdatei ein schon bestehender Name angegeben wird, so wird die alte Datei gelöscht.

− 3) Programmende
Eine eventuell vorhandene Listdatei wird geschlossen und das Programm wird beendet.

Bei der Eingabe des Listing ist in den DATA-Zeilen auf korrekte Groß- und Kleinschreibung zu achten, da das Programm an den Kleinbuchstaben »nn«, »n« und »d« erkennt, ob der Z80-Befehl noch Zahlenwerte als Operanden benötigt. Hierbei kennzeichnet »nn« einen 16-bit-Wert, »n« einen 8-bit-Wert und »d« eine Sprungdistanz.

Bei relativen Sprüngen (JR, DJNZ) gibt der Disassembler zusätzlich zur Sprungdistanz auch noch die absolute Sprungadresse an.

Bis auf die Ausgabe der Sprungdistanzen und der Displacements bei den Indexregistern sind alle Ein- und Ausgaben Hexadezimal. Bei den Adressen und den Byte-Inhalten im Listing wird dies nicht extra angezeigt. Bei den Operanden der Z80-Befehle sind Hexadezimalwerte durch ein vorangestelltes »&H« kenntlich gemacht.

Bei der Eingabe der Datenamen der Quell- und der Listdatei ist die Angabe einer Extension nicht nötig. Der Name einer Quelldatei wird falls nötig automatisch durch ».COM« ergänzt, der Name einer Listdatei durch ».LST«. Außerdem wird eine Listdatei, wenn nicht explizit anders angegeben, auf Laufwerk »M:« gespeichert. Will man die Listdatei behalten, so muß man also entweder Laufwerk »A:« explizit angeben, oder die Datei vor dem Ausschalten des Rechners auf Laufwerk »A:« kopieren.

Variablenliste:

esc$,bel$,bs$,crlf$:	Bildschirmsteuerzeichen
invon$,invoff$:	Bildschirmsteuerzeichen
true,false:	Wahrheitswerte
ok$,ko$:	Gültige Tasten für Ja/Nein-Abfragen
a,a$:	Diverse Zwischenspeicheraufgaben
i,i$:	Diverse Zwischenspeicheraufgaben
n:	Diverse Zwischenspeicheraufgaben
dateiende:	Flag Dateiende
dlist:	Flag Listdatei
index:	Flag Indexregister »IX«, »IY«

ok:	Flag Rückgabewert der Ja/Nein-Abfrage
plist:	Flag Druckerlisting
ram:	Flag RAM oder Disk
vlist:	Flag Bildschirmlisting
arith$:	Feld Die Arithmetik & LogikBefehle
aus$:	Ausgabetext
bef$:	Befehlsname
befehl$:	Feld Die restlichen Z80-Befehle
bitbef$:	Feld Die Einzelbitbefehle
nam$:	Name der Quelldatei
register$:	Feld Die Z80-Register
shift$:	Feld Die Rotations & SchiebeBefehle
xinh$:	Dateipuffer der Quelldatei
anfang:	Anfangsadresse des zu disassemblierenden Bereichs
bit7:	bit7 und bit6 des Bytes
bit5:	bit5 bis bit3 des Bytes
bit2:	bit2 bis bit0 des Bytes
dis:	Displacement für die Indexregister
ende:	Endadresse des zudisassemblierenden Bereichs
ist:	Ist-Adresse im zu disassemblierenden Bereichs
offset:	Startadresse der Quelldatei (&H100 bei ».COM«-Dateien)
wert:	16-bit-Wert
w:	Wert des aktuellen Bytes
ww:	Zwischenspeicher bei Berechnung von 16-bit-Werten
xblock:	Aktueller 128-Byte-Block in der Quelldatei
xbyte:	Aktuelles byte in dem Block
z:	Zeilenzähler für Bildschirmausgabe

(A. Freystatzky)

```
LISTING >ASSEM   <, REMARK = >REM<,

<20> 10000 ' Z80-Assembler Version vom 24.08.86
<43> 10010 '---------- Hauptprogramm ----------
     --
<56> 10020 GOSUB 10180 :' --> Startmeldung
<29> 10030 GOSUB 10260 :' --> Initialisierungen
<60> 10040 GOSUB 10690 :' --> Eingaben
<91> 10050 GOSUB 11190 :' --> Dateien öffnen
< 0> 10060 GOSUB 11610 :' --> Zeile lesen und aufspalte
     n
<14> 10070 WHILE NOT dateiende
<46> 10080.. GOSUB 11930 :' --> Zeile interpretieren
<35> 10090.. GOSUB 15250 :' --> Ausgaben
<15> 10100.. GOSUB 11610 :' --> Zeile lesen und aufspal
     ten
<97> 10110 WEND
<88> 10120 GOSUB 15870 :' --> Labelliste
<10> 10130 GOSUB 16210 :' --> Fertigmeldungen
<97> 10140 GOSUB 16380 :' --> Dateien schließen
<81> 10150 END
<88> 10160 '-----------------------------------------
     --
<45> 10170 '---------- Startmeldungen ----------
     --
<70> 10180 PRINT
<20> 10190 PRINT"-----------------------------"
< 8> 10200 PRINT". Z 8 0 - A S S E M B L E R. "
<73> 10210 PRINT" ©1986 by Andreß Freystatzky "
<94> 10220 PRINT"-----------------------------"
<53> 10230 PRINT
<60> 10240 RETURN
<48> 10250 '---------- Initialisierungen ----------
     --
<94> 10260 DEFINT a-z
<46> 10270 esc$=CHR$(27):bel$=CHR$(7):bs$=CHR$(8):tab$=
     CHR$(9):crlf$=CHR$(13)+CHR$(10)
<13> 10280 invon$=esc$+"p":invoff$=esc$+"q"
<75> 10290 ja$="J"+invon$+"a"+invoff$+bs$+bs$
<37> 10300 nein$="N"+invon$+"ein"+invoff$+bs$+bs$+bs$+b
     s$
<11> 10310 '
<39> 10320 DIM bf$(5),befi(12),befii(20),reg$(7),red$(3
     ),bed$(7)
<19> 10330 '
<16> 10340 bf$(1)="LD. JR. DJNZCALLRET JP. INC DEC PUSH
     POP RST IN. OUT IM. EX. ADD ADC SUB SBC AND XOR OR
     . CP. RLC RRC RL. RR. SLA SRA SLL SRL BIT RES SET
     "
<89> 10350 bf$(2)="CCF CPL DAA DI. EI. EXX HALTNOP RLA
     RLCARRA RRCASCF "
<30> 10360 bf$(3)="CPD CPDRCPI CPIRIND INDRINI INIRLDD
     LDDRLDI LDIRNEG OTDROTIROUTDOUTIRETIRETNRLD RRD "
< 4> 10370 bf$(4)="ORG END DB. DW. DM. DS. DEFBDEFWDEFM
     DEFS"
<28> 10380 bf$(5)="EQU "
<43> 10390 '
<46> 10400 RESTORE 10420
<14> 10410 '
<35> 10420 DATA 3F,2F,27,F3,FB,D9,76,00,17,07,1F,0F,37
<15> 10430 DATA A9,B9,A1,B1,AA,BA,A2,B2,A8,B8,A0,B0,44,
     BB,B3,AB,A3,4D,45,6F,67
<45> 10440 FOR i=0 TO 12:READ i$:befi(i)=VAL("&H"+i$):N
     EXT
<90> 10450 FOR i=0 TO 20:READ i$:befii(i)=VAL("&H"+i$):
     NEXT
<34> 10460 '
<39> 10470 DATA B,C,D,E,H,L,(HL),A
<46> 10480 DATA BC,DE,HL,SP
<62> 10490 DATA NZ,Z,NC,C,PO,PE,P,M
<66> 10500 FOR i=0 TO 7:READ reg$(i):NEXT
<16> 10510 FOR i=0 TO 3:READ red$(i):NEXT
<88> 10520 FOR i=0 TO 7:READ bed$(i):NEXT
<25> 10530 '
<59> 10540 femax=17:DIM fe$(femax)
<15> 10550 DATA "Syntaxfehler","Operand zuviel","Unbeka
     nnter Fehler","Operand fehlt","Syntaxfehler im Ope
     randen","Ungültiger Labelname","Label zu lang","Do
     ppelt definiertes Label","Zahl zu groß","Nicht erl
     aubtes Zeichen","Wertebereich überschritten"
<73> 10560 DATA "Ungültiger Dateiname","Datei nicht gef
     unden","Sprungdistanz zu groß","Nicht erlaubter Vo
     rwärtsbezug","ORG nicht erlaubt","END erwartet"
<61> 10570 FOR i=1 TO femax:READ fe$(i):NEXT
<45> 10580 '
```

```
<33> 10590 true=-1:false=0
<56> 10600 vlist=true:dlist=true:dateiende=false:comdat
     =true:lwz$="M:":lwtemp$="M:"
<20> 10610 '
< 1> 10620 feanz=0:zeile=0:lzeil=0
<28> 10630 '
<32> 10640 adr!=&H100:start!=adr!
<53> 10650 DIM a$(3),byte(4)
<40> 10660 '
<84> 10670 RETURN
<75> 10680 '---------- Eingaben ----------------------
     --
<36> 10690 ok=false
< 9> 10700 WHILE NOT ok
< 9> 10710.. INPUT"Bitte Dateinamen (,DIR oder #) einge
     ben : ",i$:i$=UPPER$(i$)
<52> 10720.. j$=LEFT$(i$+"...",4)
<48> 10730.. IF j$="#.. " THEN GOSUB 10770 ELSE IF j$="
     DIR " THEN GOSUB 10990 ELSE GOSUB 11090
<28> 10740 WEND
<79> 10750 RETURN
< 8> 10760 '---------- #, Grundeinstellungen ändern ---
     --
<48> 10770 PRINT"Bildschirmlisting : ";:IF vlist THEN P
     RINT ja$; ELSE PRINT nein$;
<67> 10780 GOSUB 10920:IF i>0 THEN vlist=true ELSE IF i
     <0 THEN vlist=false
<20> 10790 IF vlist THEN PRINT"Ja. " ELSE PRINT"Nein"
<29> 10800 PRINT"....... Listdatei : ";:IF dlist THEN P
     RINT ja$; ELSE PRINT nein$;
< 6> 10810 GOSUB 10920:IF i>0 THEN dlist=true ELSE IF i
     <0 THEN dlist=false
<13> 10820 IF dlist THEN PRINT"Ja. " ELSE PRINT"Nein"
< 1> 10830 PRINT"....... Zieldatei : ";:IF comdat THEN
     PRINT ".COM"; ELSE PRINT".DAT";
< 9> 10840 PRINT bs$+bs$+bs$+bs$;
<16> 10850 GOSUB 10920:IF i<0 THEN comdat=NOT comdat
< 8> 10860 IF comdat THEN PRINT".COM" ELSE PRINT".DAT"
< 1> 10870 PRINT".... Ziellaufwerk : "lwz$+bs$+bs$;
<67> 10880 GOSUB 10920:IF i<0 THEN IF lwz$="M:" THEN lw
     z$="A:" ELSE lwz$="M:"
<15> 10890 PRINT lwz$
<65> 10900 RETURN
<34> 10910 '---------- Taste holen und auf + - testen -
     --
<66> 10920 i$=""
<15> 10930 WHILE i$="":i$=INKEY$:WEND
<37> 10940 i=0
<39> 10950 IF i$=CHR$(22) OR i$="+" THEN i=1 :' --> Plu
     s-Taste
<40> 10960 IF i$=CHR$(28) OR i$="-" THEN i=-1 :' --> Mi
     nus-Taste
<93> 10970 RETURN
<87> 10980 '---------- DIR ---------------------------
     --
< 3> 10990 j$=MID$(i$,5):IF j$="" THEN j$="*.*"
<22> 11000 ON ERROR GOTO 11050
<13> 11010 FILES j$:PRINT
< 8> 11020 ON ERROR GOTO 0
<52> 11030 RETURN
<73> 11040 '---------- Fehlermeldungen abfangen -------
     --
```

```
<85> 11050 IF ERR=53 THEN PRINT fe$(13)bel$:RESUME NEXT
<85> 11060 IF ERR=64 THEN PRINT fe$(12)bel$:RESUME NEXT
<13> 11070 PRINT"Fehler "ERR"in"ERL":end
<83> 11080 '---------- Datei suchen -------------------
     --
<46> 11090 i=INSTR(i$,":")
<95> 11100 IF i=0 THEN lw$="" ELSE lw$=LEFT$(i$,i):i$=M
     ID$(i$,i+1)
<86> 11110 i=INSTR(i$,".")
<45> 11120 IF i=0 THEN nam$=i$:typ$=".Z80" ELSE nam$=LE
     FT$(i$,i-1):typ$=MID$(i$,i)
<37> 11130 ON ERROR GOTO 11050
<33> 11140 i$=FIND$(lw$+nam$+typ$)
<23> 11150 ON ERROR GOTO 0
<23> 11160 IF i$="" THEN PRINT fe$(13)bel$ ELSE IF STRI
     P$(i$)=LEFT$(nam$+SPACE$(8),8)+typ$ THEN ok=true
<71> 11170 RETURN
<83> 11180 '---------- Dateien öffnen -----------------
     --
<35> 11190 MEMORY ,,6
<38> 11200 ' --------- Quelldatei zum Lesen öffnen ----
     -
<73> 11210 datq=1:datq$=lw$+nam$+typ$:OPEN "I",datq,dat
     q$
<73> 11220 ' --------- Zieldatei öffnen ---------------
     -
< 4> 11230 datz=2
<12> 11240 IF comdat THEN datz$=".COM" ELSE datz$=".DAT
     "
< 4> 11250 datz$=lwz$+nam$+datz$:IF FIND$(datz$)<>"" TH
     EN KILL datz$
<72> 11260 IF comdat THEN OPEN "R",datz,datz$,1:FIELD d
     atz,1 AS byte$
< 4> 11270 IF NOT comdat THEN OPEN "R",datz,datz$,5:FIE
     LD datz,5 AS byte$
<59> 11280 ' --------- Listdatei öffnen falls nötig ---
     -
<69> 11290 datt=3:datt$=lwz$+nam$+".LST"
<51> 11300 IF dlist THEN IF FIND$(datt$)<>"" THEN KILL
     datt$
<39> 11310 IF dlist THEN OPEN "R",datt,datt$,90:FIELD d
     att,88 AS lst$,2 AS ez$
<34> 11320 IF NOT dlist THEN lst$=SPACE$(88)
<60> 11330 ' --------- Fehlerdatei öffnen -------------
     -
<33> 11340 datf=4:datf$=lwz$+nam$+".ERR":OPEN "O",datf,
     datf$
<61> 11350 '---------- Labeldatei öffnen --------------
     -
<22> 11360 datl=5:datl$=lwtemp$+nam$+".LBL":IF FIND$(da
     tl$)<>"" THEN KILL datl$
<28> 11370 OPEN "R",datl,datl$,14:FIELD datl,6 AS lbez$
     ,4 AS lwrt$,2 AS lprv$,2 AS lnxt$
<22> 11380 '---------- Datei der undefinierten Label --
     -
<94> 11390 datu=6:datu$=lwtemp$+nam$+".UDF":IF FIND$(da
     tu$)<>"" THEN KILL datu$
<21> 11400 OPEN "R",datu,datu$,20:FIELD datu,6 AS ubez$
     ,4 AS uadr$,2 AS u816$,2 AS ulst$,2 AS uzei$,2 AS
     uprv$,2 AS unxt$
<95> 11410 '---------- Labeldatei initialisieren ------
     -
```


THE PROGRAMMER'S TOOLBOX

PROFESSIONELLE ENTWICKLUNGSWERKZEUGE

Prospero Software = Compiler für höchste Ansprüche

LANGUAGES FOR MICROCOMPUTER PROFESSIONAL

Pro Pascal: PC und Compatiblen	DM 999,–
Atari ST	DM 429,–
Joyce PCW	DM 239,–
Pro Fortran: Joyce PCW	DM 239,–
Pro Fortran 77: PC und Compatiblen	DM 1.439,–

alle Preise incl.

Pieter Wade: 6246 Glashütten, Zum Talblick 44

JOYCE SOFTWARE

MARTIN KEMPENICH
COMPUTER HARD- U. SOFTWARE
ZETASTRASSE 13
6220 RÜDESHEIM 4

ADRESSENVERWALTUNG	1200/2400 Adressen	DM 49.00
HAUSVERWALTUNG	mit Graphik	DM 59.00
KFZ - ABRECHNUNG	mit Verbauchanalyse	DM 59.00
TERMINKALENDER		DM 59.00
GRAPHOFOX	Erstellen und Drucken von Balken-Kuchendiagrammen, dreidimensional, DIN A 4 Hardcopys.	DM 69.00
FIBUFOX	Finanzbuchhaltung, 1000 Einträge, mit Bilanz, Ust.-Voranmeldung u.a.	DM 198.00

Branchensoftware für:

- Zahntechnische Labore:	DENTAFOX	DM 998.00
- Zahnärzte	DENTIST	DM a.A.
- Schreinereien	SSP-SOFT	DM 1990.00

Versand Vorkasse oder Nachnahme + 5 DM Versandkosten
Infos anfordern / Händleranfragen erwünscht

```
<63> 11420 LSET lwrt$=MKS$(0):LSET lprv$=MKI$(1):LSET l
     nxt$=MKI$(2)
<40> 11430 LSET lbez$="..... ":PUT datl,1
<28> 11440 LSET lnxt$=MKI$(1)
<27> 11450 LSET lbez$="zzzzzz":PUT datl,2
<82> 11460 lmax=2
< 9> 11470 '---------- Undefinierte Labeldat. init. ---
     -
<61> 11480 LSET uadr$=MKS$(0):LSET u816$=MKI$(0):LSET u
     lst$=MKI$(0):LSET uzei$=MKI$(0):LSET uprv$=MKI$(1)
     :LSET unxt$=MKI$(2)
<82> 11490 LSET ubez$="..... ":PUT datu,1
<24> 11500 LSET unxt$=MKI$(1)
<32> 11510 LSET ubez$="zzzzzz":PUT datu,2
<33> 11520 umax=2
<45> 11530 '---------- Titelzeile in die Listdatei ----
     -
<58> 11540 aus1$=SPACE$(10):aus2$=SPACE$(8):aus3$=SPACE
     $(14)
<70> 11550 IF NOT dlist THEN RETURN
<14> 11560 LSET lst$="--- Programm : "+datq$+" ---":LSE
     T ez$=crlf$
<16> 11570 PUT datt,1
<95> 11580 lzeil=2
<91> 11590 RETURN
<71> 11600 '---------- Zeile lesen und aufspalten -----
     --
<76> 11610 lang=0:ds=0:fe=0:fehler=false
<91> 11620 label$="":bef$="":op$="":kom$=""
<44> 11630 IF EOF(datq) THEN dateiende=true:RETURN
<14> 11640 LINE INPUT #datq,zeile$
<14> 11650 zei$=zeile$
<30> 11660 '---------- Zeilennummer lesen, falls da ---
     --
<52> 11670 IF VAL(zei$)=0 THEN zeile=zeile+1:GOTO 11740
<38> 11680.. 'else
<88> 11690.. zeile=VAL(zei$)
<11> 11700.. i=LEN(STR$(zeile))
< 3> 11710.. zei$=MID$(zei$,i)
<58> 11720 'endif
<12> 11730 '---------- Zeile zerlegen -----------------
     --
<84> 11740 FOR i=0 TO 3:a$(i)="":NEXT
<12> 11750 j=-1
< 7> 11760 WHILE zei$<>"" AND j<3
<75> 11770.. a$=LEFT$(zei$,1)
<66> 11780.. WHILE a$=" " OR a$=tab$
<17> 11790.... zei$=MID$(zei$,2)
<60> 11800.... a$=LEFT$(zei$,1)
< 0> 11810.. WEND
<75> 11820 IF a$="" THEN 11890
<52> 11830.. IF a$=";" THEN SWAP kom$,zei$:GOTO 11890
<80> 11840.. i=INSTR(zei$," "):ii=INSTR(zei$,tab$)
< 5> 11850.. IF i=0 OR ii=0 THEN i=MAX(i,ii) ELSE i=MIN
     (i,ii)
<63> 11860.. j=j+1
<65> 11870.. IF i=0 THEN SWAP a$(j),zei$ ELSE a$(j)=LEF
     T$(zei$,i-1):zei$=MID$(zei$,i)
<70> 11880.. IF a$<>"'" THEN a$(j)=UPPER$(a$(j)) ELSE i
     =INSTR(zei$,"'"):a$(j)=a$(j)+LEFT$(zei$,i):zei$=MI
     D$(zei$,i+1)
<53> 11890 WEND
< 1> 11900 IF j>2 THEN fe=1:GOSUB 18960
<71> 11910 RETURN
<72> 11920 '---------- Zeile interpretieren -----------
     --
<42> 11930 j=0
<72> 11940 bfnr=0:IF a$(j)="" THEN RETURN
<83> 11950 IF LEN(a$(j))<2 OR LEN(a$(j))>4 THEN bfnr=5
<44> 11960 i$=LEFT$(a$(j)+". ",4)
<37> 11970 zeiger=0
<75> 11980 WHILE zeiger=0 AND bfnr<5
<71> 11990.. bfnr=bfnr+1
< 6> 12000.. zeiger=INSTR(bf$(bfnr),i$)
<98> 12010 WEND
<94> 12020 IF zeiger>0 THEN SWAP bef$,i$
<56> 12030 ON bfnr GOSUB 12070,14700,14770,14850,17460
<30> 12040 IF label THEN label=false:j=1:GOTO 11940
<62> 12050 RETURN
<49> 12060 '---------- Befehle mit Operanden ----------
     --
<39> 12070 j=j+1
<51> 12080 op$=a$(j)
<33> 12090 IF op$="" THEN IF bef$<>"RET " THEN fe=4:GOS
     UB 18960:RETURN
<22> 12100 GOSUB 12540
```

```
<58> 12110 xypos=INSTR(op$,"IX"):IF xypos<>0 THEN xybyt
     e=&HDD:ireg$="IX":GOTO 12170
<65> 12120 xypos=INSTR(op$,"IY"):IF xypos<>0 THEN xybyt
     e=&HFD:ireg$="IY":GOTO 12170
<88> 12130 GOTO 12310
<21> 12140 '
<24> 12150 '---------- falls es ein Indexregister ist--
     --
<29> 12160 '
<40> 12170       index=true:opp$=op$
<31> 12180       IF (NOT klammer)OR(xypos-klauf<>1) THEN op$=
     LEFT$(op$,xypos-1)+"HL"+MID$(op$,xypos+2):GOTO 122
     70
< 6> 12190       temp$=MID$(klinh$,3,1):IF temp$<>"+" AND tem
     p$<>"-" THEN IF bef$="JP. " THEN 12260 ELSE fe=5:G
     OSUB 18960:RETURN
<10> 12200       a$=MID$(klinh$,4)
<96> 12210       dis$=a$:distanz=true
<56> 12220       GOSUB 17020:lang=lang-1
<31> 12230       IF fehler THEN fe=5:GOSUB 18960:RETURN
< 3> 12240       distanz=true
<46> 12250       disw=wert!:IF temp$="-" THEN disw=(disw XOR
     255)+1
<72> 12260       op$=LEFT$(op$,klauf)+"HL"+MID$(op$,klzu)
<69> 12270       IF (INSTR(op$,"IX")=0)AND(INSTR(op$,"IY")=0)
      THEN 12290
<27> 12280       IF (op$=("HL,"+ireg$))AND(bef$="ADD ") THEN
     op$="HL,HL" ELSE fe=5:GOSUB 18960:RETURN
< 0> 12290       GOSUB 12540:GOTO 12310
<11> 12300 '
<49> 12310 GOSUB 12460
<52> 12320 IF NOT index THEN 12420
<12> 12330       op$=opp$
<31> 12340       FOR j=lang TO 1 STEP -1
<71> 12350       . byte(j+1)=byte(j)
<77> 12360       NEXT
<48> 12370       byte(1)=xybyte:lang=lang+1
<62> 12380       IF NOT distanz THEN 12410
<21> 12390       . IF lang=3 THEN byte(4)=byte(3)
<16> 12400       . byte(3)=disw:lang=lang+1
<28> 12410       index=false:distanz=false
<62> 12420 RETURN
<26> 12430 '
<63> 12440 '---------- Verteilerleiste ----------------
     --
<34> 12450 '
<54> 12460 temp=(zeiger+3)/4
<50> 12470 ON temp GOTO 12700,13200,13320,13360,13420,1
     3480,13660,13700,13810,13850,13930,14020,14060,141
     80,14250
<77> 12480 IF temp<24 THEN 14330 :' --> Arithmetik & Lo
     gikbefehle
< 7> 12490 IF temp<32 THEN 14530 :' --> Rotations & Sch
     iebebefehle
< 1> 12500 GOTO 14620 :' --> Einzelbitbefehle
<21> 12510 '
<10> 12520 '---------- Operanden in op1$, op2$ aufteile
     n
<29> 12530 '
<78> 12540 IF LEFT$(op$,1)="'" THEN komma=false:op1$=op
     $:RETURN
<15> 12550 i=INSTR(op$,",")
<79> 12560 IF i=0 THEN komma=false:op1$=op$:GOTO 12600
<46> 12570.. komma=true
< 8> 12580.. op1$=LEFT$(op$,i-1)
<49> 12590.. op2$=MID$(op$,i+1)
<51> 12600 klauf=INSTR(op$,"(")
<10> 12610 klzu=INSTR(op$,")")
<32> 12620 IF klauf=0 THEN klammer=false:RETURN
<37> 12630 IF klauf>klzu THEN fe=5:GOSUB 18960:RETURN
<13> 12640 klammer=true
<40> 12650 klinh$=MID$(op$,klauf+1,klzu-klauf-1)
<84> 12660 RETURN
<48> 12670 '
<50> 12680 '----- LD
<56> 12690 '
<22> 12700 IF NOT komma THEN fe=5:GOSUB 18960:RETURN
<13> 12710 a$=op1$:GOSUB 16490
<60> 12720 IF register THEN 12920
<89> 12730.. GOSUB 16570
<72> 12740.. IF register THEN 12820
<33> 12750.... a$=op2$:GOSUB 16570
<34> 12760.... IF register THEN 12800
<87> 12770..... SWAP op1$,op2$
< 4> 12780..... a=0:GOSUB 13010
<39> 12790..... IF zahl THEN fe=5:GOSUB 18960:RETURN E
     LSE RETURN
```

```
<67> 12800.... IF NOT klammer THEN fe=5:GOSUB 18960:RET
     URN
<73> 12810.... temp$=op2$:qtemp=1:GOTO 12860
<48> 12820.. IF op$="SP,HL" THEN lang=1:byte(1)=&HF9:RE
     TURN
<71> 12830.. IF klammer THEN temp$=op1$:qtemp=false:GOT
     O 12860
<97> 12840.... a$=op2$
<15> 12850.... lang=1:code=1:GOTO 12890
<32> 12860.. a$=klinh$
<51> 12870.. IF temp$="HL" THEN lang=1:code=&HA:GOTO 12
     890
< 3> 12880.. lang=2:byte(1)=&HED:code=&H4B
<66> 12890.. code=code AND NOT(qtemp*8)
<51> 12900.. byte(lang)=code OR (dd*16)
<51> 12910.. GOSUB 16890:RETURN
<70> 12920 zzz=rrr:a$=op2$:GOSUB 16490
<51> 12930 IF NOT register THEN 12970
<18> 12940.. lang=1:byte(1)=64 OR (zzz*8) OR rrr
<91> 12950.. IF byte(1)=&H76 THEN fe=5:GOSUB 18960
<32> 12960.. RETURN
<45> 12970 a=1:GOSUB 13010
<40> 12980 IF NOT zahl THEN RETURN
<33> 12990 lang=1:byte(1)=&H6 OR (rrr*8)
<97> 13000 a$=op2$:GOSUB 17020:RETURN
< 5> 13010 zahl=false
<95> 13020 IF op1$<>"A" THEN zahl=true:RETURN
<26> 13030 IF klammer THEN 13090
<18> 13040 IF op2$="I" THEN temp=0:GOTO 13070
<96> 13050 IF op2$="R" THEN temp=1:GOTO 13070
< 3> 13060 zahl=true:RETURN
<88> 13070 code=&H47:lang=2:byte(1)=&HED
<45> 13080 pp=(a+a) OR temp:GOTO 13150
<74> 13090 IF klinh$="BC" THEN temp=0:GOTO 13140
<48> 13100 IF klinh$="DE" THEN temp=1:GOTO 13140
<24> 13110 lang=1:byte(1)=&H32 OR (a*8)
<66> 13120 a$=klinh$:GOSUB 16890
<59> 13130 RETURN
<43> 13140 code=&H2:lang=1:pp=(temp+temp)OR a
<10> 13150 byte(lang)=code OR (8*pp)
<71> 13160 RETURN
<35> 13170 '
<21> 13180 '----- JR
<43> 13190 '
<93> 13200 lang=1
<95> 13210 IF NOT komma THEN ccc=&H3:a$=op$:GOTO 13260
<15> 13220.. a$=op1$:GOSUB 16650
<77> 13230.. IF (NOT bedingung)OR(ccc>3) THEN fe=5:GOSU
     B 18960:RETURN
<87> 13240.. ccc=ccc OR 4
<15> 13250.. a$=op2$
<81> 13260 byte(1)=ccc*8
<42> 13270 GOSUB 16740
<82> 13280 RETURN
<46> 13290 '
<94> 13300 '----- DJNZ
<17> 13310 '
<89> 13320 lang=1:byte(1)=&H10:a$=op$:GOTO 13270
<25> 13330 '
<87> 13340 '----- CALL
<33> 13350 '
<46> 13360 temp=1:lang=1
<11> 13370 code=&H4
<58> 13380 GOTO 13510
<49> 13390 '
<71> 13400 '----- RET
<20> 13410 '
<33> 13420 temp=1:lang=1
<38> 13430 code=0
<45> 13440 GOTO 13510
<36> 13450 '
<91> 13460 '----- JP
<44> 13470 '
<71> 13480 IF op$="(HL)" THEN lang=1:byte(1)=&HE9:RETUR
     N
<93> 13490 code=&H2
<16> 13500 temp=0:lang=1
<84> 13510 IF bef$="RET " THEN IF op$="" THEN 13530 ELS
     E 13560
<20> 13520.. IF komma THEN 13560
<97> 13530.... byte(1)=192 OR code OR 1 OR (temp*8)
<46> 13540.... a$=op$
<78> 13550.... GOTO 13600
<78> 13560 a$=op1$:GOSUB 16650
<71> 13570 IF NOT bedingung THEN fe=5:GOSUB 18960:RETUR
     N
<46> 13580 byte(1)=192 OR code OR (ccc*8)
```


Lauter Anwendungsprogramme

PROFIREM v 2.1

Das Geschäftssoftwarepaket
Fakturierung, Adressendatei, Lagerdatei

- universelle Einsatzmöglichkeiten
- Einlesen von Adressen in die Faktura
- Einlesen von Lagerartikeln in die Rechnung, mit automatischer Bestandskorrektur
- Speicherung von Rechnungen
- Einfaches nachträgliches Ändern
- komfortable Eingabemasken
- 9 verschiedene Zahlungsbedingungen
- Ausdruck sortierter Adressenlisten
- Berechnung des Lagerwertes
- **3" Diskette nur 178,– DM**

ETATGRAF

Das Haushaltsbuch mit Grafik

Verwalten Sie Ihre Ausgaben mit dem Computer. Die grafischen Auswertungsmöglichkeiten verschaffen Ihnen jederzeit einen guten Gesamtüberblick

- bis zu 18 verschiedene Kostenarten
- Bearbeitung von 12 Monaten
- Tabellen, Balkengrafik
- **3" Diskette nur 58,– DM**

FIBUKING v 2.0

Die preiswerte Finanzbuchführung

- 60 Konten, einfache oder doppelte Buchführung
- bis zu 4-stellige Kontennummern
- automatische Bilanzierung
- **3" Diskette nur 136,– DM**

VOKABI

Der universelle Vokabeltrainer

- **3" Diskette nur 58,– DM**

DATENREM

Die einfache Dateiverwaltung

- **3" Diskette nur 68,– DM**

LAGDAT

menuegesteuerte Lagerverwaltung

- **3" Diskette nur 68,– DM**

ADRESCOMP

praktisches Adressenprogramm

- **3" Diskette nur 58,– DM**

VAN DER ZALM SOFTWARE

Programm-Entwicklung und Vertrieb
Elfriede van der Zalm

Schieferstätte, 2949 Wangerland 3
Telefon (0 44 61) 55 24

Versand erfolgt per Vorkasse (portofrei), Nachnahme (zzgl. 5,– DM)

```
<55> 13590 a$=op2$
<70> 13600 IF bef$="RET " THEN RETURN
<26> 13610 GOSUB 16890
<70> 13620 RETURN
<34> 13630 '
<28> 13640 '----- INC
<42> 13650 '
<88> 13660 temp=0:GOTO 13710
<50> 13670 '
<37> 13680 '----- DEC
<58> 13690 '
<79> 13700 temp=1
<51> 13710 IF komma THEN fe=5:GOSUB 18960:RETURN
<61> 13720 lang=1:a$=op$:GOSUB 16490
< 9> 13730 IF register THEN byte(1)=&H4 OR (rrr*8) OR t
     emp:RETURN
<63> 13740 GOSUB 16570
<77> 13750 IF NOT register THEN fe=5:GOSUB 18960:RETURN
<68> 13760 byte(1)=&H3 OR (dd*16) OR (temp*8)
<93> 13770 RETURN
<57> 13780 '
<51> 13790 '----- PUSH
<28> 13800 '
<52> 13810 code=&HC5:GOTO 13860
<36> 13820 '
<54> 13830 '----- POP
<44> 13840 '
<72> 13850 code=&HC1
<39> 13860 a$=op$:red$(3)="AF":GOSUB 16570:red$(3)="SP"
<88> 13870 IF NOT register THEN fe=5:GOSUB 18960:RETURN
<87> 13880 lang=1:byte(1)=code OR (dd*16)
< 5> 13890 RETURN
<31> 13900 '
< 3> 13910 '----- RST
<39> 13920 '
< 1> 13930 a$=op$
<27> 13940 GOSUB 17020
<21> 13950 temp=wert1/8
<98> 13960 IF temp<>INT(temp) OR temp>7 THEN fe=5:GOSUB
      18960:RETURN
<47> 13970 lang=1:byte(1)=&HC7 OR (temp*8)
< 4> 13980 RETURN
<67> 13990 '
<13> 14000 '----- IN
<10> 14010 '
<61> 14020 temp=0:GOTO 14070
<18> 14030 '
<77> 14040 '----- OUT
<26> 14050 '
<13> 14060 temp=1:SWAP op1$,op2$
<83> 14070 IF NOT (komma AND klammer) THEN fe=5:GOSUB 1
     8960:RETURN
<36> 14080 IF klinh$<>"C" THEN 14120
<93> 14090.. a$=op1$:GOSUB 16490
<70> 14100.. IF (NOT register)OR(klinh$<>"C") THEN fe=5
     :GOSUB 18960:RETURN
<35> 14110.. lang=2:byte(1)=&HED:byte(2)=64 OR (rrr*8)
     OR temp:RETURN
<49> 14120 lang=1:a$=klinh$:GOSUB 17020
<73> 14130 byte(1)=&HDB XOR (temp*8)
<65> 14140 RETURN
<29> 14150 '
<25> 14160 '----- IM
<37> 14170 '
<84> 14180 IF op$<>"0" AND op$<>"1" AND op$<>"2" THEN f
     e=5:GOSUB 18960:RETURN
<28> 14190 lang=2:byte(1)=&HED
<14> 14200 byte(2)=&H46 OR ((VAL(op$)-(op$<>"0"))*8)
<56> 14210 RETURN
<20> 14220 '
<26> 14230 '----- EX
<28> 14240 '
<16> 14250 lang=1
<61> 14260 IF op$="(SP),HL" THEN byte(1)=&HE3:RETURN
<69> 14270 IF op$="AF,AF" THEN byte(1)=&H8:RETURN
<88> 14280 IF op$="DE,HL" THEN byte(1)=&HEB:IF NOT inde
     x THEN RETURN
<55> 14290 fe=5:GOSUB 18960:RETURN
<15> 14300 '
<40> 14310 '----- Arithmetik- und Logikbefehle
<23> 14320 '
<84> 14330 IF NOT komma THEN a$=op1$:GOTO 14350
<90> 14340.. IF op1$<>"A" THEN 14400 ELSE a$=op2$
<86> 14350 lang=1:code=temp-16
<78> 14360 GOSUB 16490
<85> 14370 IF register THEN byte(1)=128 OR (code*8) OR
     rrr:RETURN
<72> 14380 byte(1)=&HC6 OR (code*8)
<20> 14390 GOSUB 17020:RETURN
<41> 14400 IF op1$<>"HL" THEN fe=5:GOSUB 18960:RETURN
<22> 14410 a$=op2$
<48> 14420 GOSUB 16570
<62> 14430 IF NOT register THEN fe=5:GOSUB 18960:RETURN
<34> 14440 IF bef$="ADD " THEN code=&H9:lang=1:GOTO 144
     80
<59> 14450.. byte(1)=&HED:lang=2
<92> 14460.. IF bef$="ADC " THEN code=&H4A:GOTO 14480
<60> 14470.... IF bef$="SBC " THEN code=&H42 ELSE fe=1:
     GOSUB 18960:RETURN
<46> 14480 byte(lang)=code OR (dd*16)
<94> 14490 RETURN
<21> 14500 '
<29> 14510 '----- Rotations- und Schiebebefehle
<29> 14520 '
<17> 14530 lang=2:byte(1)=&HCB
<59> 14540 IF komma THEN fe=5:GOSUB 18960:RETURN
<55> 14550 a$=op$:GOSUB 16490
<77> 14560 IF NOT register THEN fe=5:GOSUB 18960:RETURN
<69> 14570 byte(2)=(8*(temp-24)) OR rrr
<93> 14580 RETURN
<57> 14590 '
<53> 14600 '----- Einzelbitbefehle
<28> 14610 '
<16> 14620 lang=2:byte(1)=&HCB
<34> 14630 a$=op2$:GOSUB 16490
<72> 14640 IF NOT register THEN fe=5:GOSUB 18960:RETURN
<65> 14650 bbb=ASC(op1$)-48
<52> 14660 IF (0>bbb) OR (7<bbb) OR (LEN(op1$)<>1) THEN
      fe=5:GOSUB 18960:RETURN
<68> 14670 byte(2)=(64*(temp-31))OR(bbb*8) OR rrr
<96> 14680 RETURN
<82> 14690 '---------- 1-Byte-Befehle ohne Operanden --
     --
<11> 14700 lang=1
<94> 14710 IF a$(j+1)<>"" THEN fe=2:GOSUB 18960
< 3> 14720 byte(1)=befi((zeiger-1)/4)
<79> 14730 RETURN
<43> 14740 '
<25> 14750 '---------- 2-Byte-Befehle ohne Operanden --
     --------
<51> 14760 '
<51> 14770 lang=2
<29> 14780 byte(1)=&HED
<27> 14790 IF a$(j+1)<>"" THEN fe=2:GOSUB 18960
<36> 14800 byte(2)=befii((zeiger-1)/4)
<74> 14810 RETURN
<38> 14820 '
<73> 14830 '---------- Assembler-Pseudo-Befehle außer E
     QU ----------
<46> 14840 '
<59> 14850 j=j+1
<71> 14860 op$=a$(j)
<44> 14870 a$=op$:IF a$(j+1)<>"" THEN fe=2:GOSUB 18960
<63> 14880 ON (zeiger+3)/4 GOTO 14920,14990,15030,15070
     ,15110,15200,15030,15070,15110,15200
<53> 14890 fe=3:GOSUB 18960:RETURN
<33> 14900 '
<32> 14910 '----- ORG -----
<47> 14920 IF a$="" THEN fe=4:GOSUB 18960:RETURN
<74> 14930 IF adr!<>start! THEN fe=16:GOSUB 18960
<61> 14940 GOSUB 17090:IF undef THEN fe=15:GOSUB 18960:
     wert!=&H100
<13> 14950 adr!=wert!:start!=adr!:lang=0
<97> 14960 RETURN
<61> 14970 '
<10> 14980 '----- END -----
<75> 14990 dateiende=true
<48> 15000 RETURN
<12> 15010 '
<14> 15020 '----- DB,DEFB -----
< 0> 15030 GOSUB 17020
<64> 15040 RETURN
<28> 15050 '
<72> 15060 '----- DW,DEFW -----
<36> 15070 GOSUB 16890
<80> 15080 RETURN
<44> 15090 '
<97> 15100 '----- DM,DEFM -----
< 2> 15110 IF LEFT$(a$,1)<>"'" OR RIGHT$(a$,1)<>"'" THE
     N fe=5:GOSUB 18960:RETURN
<50> 15120 lang=LEN(a$)-2
<41> 15130 ERASE byte:DIM byte(MAX(lang,4))
<86> 15140 FOR i=1 TO lang
```

```
<98> 15150       byte(i)=ASC(MID$(a$,i+1,1))
<17> 15160 NEXT
<79> 15170 RETURN
<43> 15180 '
<46> 15190 '----- DS,DEFS -----
<54> 15200 GOSUB 17090:IF undef THEN fe=15:GOSUB 18960:
     wert!=1
<29> 15210 ds=wert!
<93> 15220 lang=0
<66> 15230 RETURN
<29> 15240 '---------- Ausgaben ----------------------
     --
<33> 15250 IF NOT fehler THEN 15300
<75> 15260       . lang=4
<15> 15270       . FOR i=1 TO lang
<14> 15280       ... byte(i)=0
<21> 15290       . NEXT
<61> 15300 IF NOT vlist AND NOT dlist THEN 15750
<71> 15310.. IF fehler THEN 15320 ELSE 15380
<20> 15320       . MID$(aus3$,2)=LEFT$(zeile$,13)
<73> 15330       . op$=MID$(zeile$,14,10)
<91> 15340       . sp$=MID$(zeile$,24,1)
<45> 15350       . kom$=MID$(zeile$,25)
<15> 15360       . GOTO 15410
<87> 15370       'else
<37> 15380       . MID$(aus3$,2)=label$
<77> 15390       . MID$(aus3$,10)=bef$
<19> 15400       . sp$=" "
<22> 15410       MID$(aus1$,1)=RIGHT$("... "+STR$(zeile),4)
<33> 15420       MID$(aus1$,6)=HEX$(adr!,4)
<33> 15430       n=1
<65> 15440       WHILE lang>n+4
<51> 15450       . FOR i=0 TO 3
<87> 15460       ... MID$(aus2$,i+i+1)=HEX$(byte(i+n),2)
<19> 15470       . NEXT
<86> 15480       . LSET lst$=aus1$+aus2$
<98> 15490       . IF vlist THEN PRINT lst$
<19> 15500       . IF dlist THEN PUT datt,lzeil:lzeil=lzeil+1
< 8> 15510       . LSET aus1$=""
<42> 15520       . n=n+4
<88> 15530       WEND
<59> 15540       LSET aus2$=""
<45> 15550       FOR i=0 TO 3
<81> 15560       . IF n+i<=lang THEN MID$(aus2$,i+i+1)=HEX$(b
     yte(i+n),2)
<93> 15570       NEXT
< 8> 15580       a=LEN(op$)
<27> 15590       IF a<10 THEN op$=op$+SPACE$(10-a)
<11> 15600       WHILE LEN(op$)+1+LEN(kom$)>56
<41> 15610       . LSET lst$=aus1$+aus2$+aus3$+op$+sp$+kom$
<76> 15620       . IF vlist THEN PRINT lst$
<34> 15630       . IF dlist THEN PUT datt,lzeil:lzeil=lzeil+1
< 9> 15640       . kom$=MID$(kom$,57-LEN(op$)-1)
<45> 15650       . sp$=" "
<14> 15660       . op$=SPACE$(10)
<35> 15670       . LSET aus1$=""
<56> 15680       . LSET aus2$=""
<77> 15690       . LSET aus3$=""
<82> 15700       WEND
<47> 15710       LSET lst$=aus1$+aus2$+aus3$+op$+sp$+kom$
<41> 15720       IF vlist THEN PRINT lst$
<13> 15730       IF dlist THEN PUT datt,lzeil:lzeil=lzeil+1
<51> 15740       LSET lst$="":LSET aus1$="":LSET aus2$="":LSE
     T aus3$=""
< 2> 15750 FOR i=1 TO ds
<52> 15760       IF comdat THEN LSET byte$=CHR$(0) ELSE LSET
     byte$="&H00,"
<72> 15770       PUT datz,adr!-start!+1
<25> 15780       adr!=adr!+1
<47> 15790 NEXT
<91> 15800 FOR i=1 TO lang
<84> 15810       IF comdat THEN LSET byte$=CHR$(byte(i)) ELSE
      LSET byte$="&H"+HEX$(byte(i),2)+","
<71> 15820       PUT datz,adr!-start!+1:adr!=adr!+1
<26> 15830 NEXT
<62> 15840 IF NOT vlist THEN PRINT".";
<92> 15850 RETURN
<81> 15860 '---------- Labelliste ---------------------
     --
<76> 15870 IF vlist THEN PRINT
<35> 15880 IF dlist THEN LSET lst$="":PUT datt,lzeil:lz
     eil=lzeil+1
<62> 15890 GET datu,1
<80> 15900 WHILE CVI(unxt$)<>2
<23> 15910       GET datu,CVI(unxt$)
<85> 15920       feanz=feanz+1
<57> 15930       LSET lst$="Undefiniertes Label "+ubez$+" in
     Zeile"+STR$(CVI(uzei$))
<63> 15940       PRINT #datf,lst$
<59> 15950       IF vlist THEN PRINT lst$
<31> 15960       IF dlist THEN PUT datt,lzeil:lzeil=lzeil+1
<56> 15970 WEND
<67> 15980 '
<34> 15990 '---------- Label auflisten ----------------
     --
<10> 16000 '
<20> 16010 IF NOT vlist AND NOT dlist THEN 16190
<38> 16020.. IF vlist THEN PRINT:PRINT"Labeltabelle :":
     PRINT
<33> 16030.. IF dlist THEN LSET lst$="":PUT datt,lzeil:
     PUT datt,lzeil+2:LSET lst$="Labeltabelle :":PUT da
     tt,lzeil+1:lzeil=lzeil+3
<17> 16040.. GET datl,1
<68> 16050.. LSET lst$=""
<45> 16060.. FOR i=1 TO lmax-2
<63> 16070       . GET datl,CVI(lnxt$)
<47> 16080       . ii=((i-1)MOD 7)*12+1
<93> 16090       . MID$(lst$,ii,6)=lbez$
<43> 16100       . MID$(lst$,ii+7,4)=HEX$(CVS(lwrt$),4)
<96> 16110       . IF i MOD 7<>0 THEN 16150
<63> 16120       . IF vlist THEN PRINT lst$
<21> 16130       . IF dlist THEN PUT datt,lzeil:lzeil=lzeil+1
<43> 16140       . LSET lst$=""
<28> 16150.. NEXT
<74> 16160.. IF i MOD 7=1 THEN 16190
< 8> 16170.. IF vlist THEN PRINT lst$
<65> 16180.. IF dlist THEN PUT datt,lzeil:lzeil=lzeil+1
<89> 16190 RETURN
<81> 16200 '---------- Fertigmeldungen ----------------
     --
<22> 16210 IF dlist THEN LSET lst$="":PUT datt,lzeil:PU
     T datt,lzeil+4
<61> 16220 PRINT
<93> 16230 LSET lst$="Programm : "+nam$
<41> 16240 PRINT lst$
<31> 16250 IF dlist THEN PUT datt,lzeil+1
<24> 16260 LSET lst$=". Länge. :"+STR$(adr!-start!+1)+"
      Bytes"
<53> 16270 PRINT lst$
<79> 16280 IF dlist THEN PUT datt,lzeil+2
<84> 16290 LSET lst$=". Start. : "+HEX$(start!,4)+"H"
<28> 16300 PRINT lst$
<90> 16310 IF dlist THEN PUT datt,lzeil+3
<64> 16320 PRINT
<53> 16330 LSET lst$=STR$(feanz)+" Fehler"
<44> 16340 PRINT lst$
<74> 16350 IF dlist THEN PUT datt,lzeil+5:lzeil=lzeil+6
<83> 16360 RETURN
<79> 16370 '---------- Dateien schließen --------------
     --
<86> 16380 CLOSE
<58> 16390 KILL datl$
<61> 16400 KILL datu$
<81> 16410 IF feanz=0 THEN KILL datf$
<30> 16420 '
<15> 16430.... '
<78> 16440 RETURN
< 6> 16450 '-----------------------------------------

<46> 16460 '
<86> 16470 '---------- a$ auf Register testen ---------
     --
<54> 16480 '
<73> 16490 register=false
<39> 16500 FOR i=0 TO 7
< 8> 16510       IF reg$(i)=a$ THEN register=true:rrr=i
<15> 16520 NEXT
<77> 16530 RETURN
<41> 16540 '
<57> 16550 '---------- a$ auf Registerpaar testen -----
     --
<49> 16560 '
<68> 16570 register=false
<98> 16580 FOR i=0 TO 3
<54> 16590       IF red$(i)=a$ THEN register=true:dd=i
<10> 16600 NEXT
<72> 16610 RETURN
<36> 16620 '
<55> 16630 '---------- a$ auf Bedingung testen --------
     --
<44> 16640 '
<14> 16650 bedingung=false
<66> 16660 FOR i=0 TO 7
<71> 16670       IF bed$(i)=a$ THEN bedingung=true:ccc=i
<42> 16680 NEXT
< 5> 16690 RETURN
<94> 16700 '-----------------------------------------
     --
<35> 16710 '
<63> 16720 '---------- Offset holen -------------------
     --
<43> 16730 '
<38> 16740 u816=3
<66> 16750 b$=LEFT$(a$+"..... ",6):SWAP a$,b$
<25> 16760 GOSUB 16900:lang=lang-1
<65> 16770 SWAP a$,b$
<12> 16780 IF NOT found THEN wert!=adr!
<32> 16790 of!=wert!-adr!-2
<69> 16800 IF of!>129 OR of!<-126 THEN fe=4:GOSUB 18960
     :of!=-2
<67> 16810 wert1=of!
<79> 16820 IF wert1<0 THEN wert1=256+wert1
<61> 16830 werth=0
<39> 16840 byte(lang)=wert1
```

```
<94> 16850 RETURN
<58> 16860 '
<24> 16870 '---------- 2-Byte-Wert holen --------------
     --
<66> 16880 '
<49> 16890 u816=2
<30> 16900 GOSUB 17090
< 7> 16910 IF undef THEN GOSUB 18580:IF keinlabel THEN
     fe=5:GOSUB 18960:wert!=0 ELSE GOSUB 18660:wert!=0
<37> 16920 IF wert!<0 THEN wert!=wert!+2^16
<57> 16930 werth=INT(wert!/256)
<44> 16940 wertl=wert!-256*werth
< 0> 16950 lang=lang+2
<52> 16960 byte(lang-1)=wertl
<65> 16970 byte(lang)=werth
<10> 16980 RETURN
<73> 16990 '
<54> 17000 '---------- 1-Byte-Wert holen --------------
     --
<16> 17010 '
<45> 17020 u816=1:GOSUB 16900
<81> 17030 lang=lang-1
<28> 17040 IF werth<>0 THEN fe=9:GOSUB 18960:wert!=0
<72> 17050 RETURN
<36> 17060 '
<95> 17070 '---------- Zahl lesen ---------------------
     --
<44> 17080 '
< 2> 17090 z$="":wert!=0:pm$="+":pm=1:undef=false:zz$=a
     $
<49> 17100 WHILE pm>0
<16> 17110       IF LEFT$(zz$,1)<>"'" THEN 17160
<15> 17120       . i=INSTR(MID$(zz$,3),"'")
< 5> 17130       . IF i>0 THEN ii=ASC(MID$(zz$,2,1)) ELSE ii=
     0
<15> 17140       . IF i<>1 THEN fe=5:GOSUB 18960
< 5> 17150       . zz$=STR$(ii)+MID$(zz$,i+3)
<95> 17160       p=INSTR(zz$,"+"):m=INSTR(zz$,"-")
<60> 17170       IF p=0 OR m=0 THEN pm=MAX(p,m) ELSE pm=MIN(p
     ,m)
< 7> 17180       IF pm=0 THEN SWAP zz$,z$ ELSE z$=LEFT$(zz$,p
     m-1)
< 3> 17190       GOSUB 17270
<92> 17200       IF pm$="+" THEN wert!=wert!+wrt! ELSE wert!=
     wert!-wrt!
<13> 17210       IF pm>0 THEN pm$=MID$(zz$,pm,1):zz$=MID$(zz$
     ,pm+1)
<19> 17220 WEND
<70> 17230 RETURN
<34> 17240 '
<57> 17250 '---------- z$ --> wrt! --------------------
     --
<42> 17260 '
<40> 17270 IF z$="$" THEN wrt!=adr!:RETURN
<20> 17280 SWAP a$,z$
<21> 17290 GOSUB 18580
<90> 17300 SWAP a$,z$
<89> 17310 IF keinlabel THEN 17320 ELSE 17360
<60> 17320       IF RIGHT$(z$,1)<>"H" THEN wrt!=VAL(z$) ELSE
     wrt!=VAL("&H"+LEFT$(z$,LEN(z$)-1))
<96> 17330       IF wrt!=0 THEN IF LEFT$(z$,1)<>"&" AND LEFT$
     (z$,1)<>"0" AND RIGHT$(z$,1)<>"H" THEN fe=5:GOSUB
     18960
<38> 17340       RETURN
<94> 17350 'ELSE
<12> 17360       b$=LEFT$(z$+".... ",6)
< 6> 17370       SWAP wert!,wrt!:SWAP a$,b$
<10> 17380       GOSUB 18870
<14> 17390       SWAP wert!,wrt!:SWAP a$,b$
<17> 17400       IF NOT found THEN undef=true
<68> 17410 RETURN
<95> 17420 '-------------------------------------------
     --
<36> 17430 '
<27> 17440 '---------- Name EQU Wert und Label --------
     --
<44> 17450 '
< 2> 17460 IF j>=1 THEN fe=1:GOSUB 18960:RETURN
<12> 17470 IF LEFT$(a$(1)+" ",4)<>bf$(5) THEN 17590
<84> 17480.. IF a$(2)="" THEN fe=4:GOSUB 18960:RETURN
<11> 17490.. a$=a$(2)
< 9> 17500.. label$=a$(0)
<56> 17510.. bef$=a$(1)
< 9> 17520.. op$=a$(2)
<60> 17530.. GOSUB 17090
<63> 17540.. a$=a$(0)
<76> 17550.. GOTO 17650
<98> 17560.. '
<21> 17570 '---------- Labelname und Wert eintragen ---
     --
< 7> 17580.. '
<26> 17590 label=true
<86> 17600 a$=a$(0):label$=a$(0)
<78> 17610 wert!=adr!
<38> 17620 '
<60> 17630 '---------- Label in die Tabelle eintragen -
     --
<46> 17640 '
<17> 17650 GOSUB 18580
<84> 17660 IF keinlabel THEN fe=6:GOSUB 18960:RETURN
<22> 17670 lnam$="":SWAP lnam$,a$
<62> 17680 '
<90> 17690 '----- Stelle zum Einfügen aussuchen -------
     --
< 0> 17700 GET datl,1
<37> 17710 WHILE lnam$>lbez$
<16> 17720       GET datl,CVI(lnxt$)
<38> 17730 WEND
<49> 17740 '
<78> 17750 '----- Doppelt definiertes Label abfangen --
     --
<97> 17760 IF lnam$=lbez$ THEN fe=8:GOSUB 18960:RETURN
<61> 17770 '
<51> 17780 '----- Vorgänger und Nachfolger anpassen ---
     --
<76> 17790 lmax=lmax+1
<18> 17800 lprv=CVI(lprv$)
<21> 17810 GET datl,lprv
< 0> 17820.. lnxt=CVI(lnxt$)
<31> 17830.. LSET lnxt$=MKI$(lmax)
<26> 17840 PUT datl,lprv
<73> 17850 GET datl,lnxt
<11> 17860.. LSET lprv$=MKI$(lmax)
<74> 17870 PUT datl,lnxt
<68> 17880 '
<94> 17890 '----- Neues Label eintragen ---------------
     --
<51> 17900 LSET lbez$=lnam$
<44> 17910 LSET lwrt$=MKS$(wert!)
< 7> 17920 LSET lprv$=MKI$(lprv)
<89> 17930 LSET lnxt$=MKI$(lnxt)
< 7> 17940 PUT datl,lmax
<59> 17950 '
<80> 17960 '---------- Undefinierte Label austragen ---
     --
<67> 17970 '
<65> 17980 GET datu,1
<21> 17990 WHILE CVI(unxt$)<>2
<97> 18000       GET datu,CVI(unxt$)
<35> 18010       IF lnam$<>ubez$ THEN 18230
<70> 18020       . uadr!=CVS(uadr$)
<31> 18030       . u816=CVI(u816$)
<89> 18040       . ON u816 GOSUB 18270,18350,18430
<33> 18050       . IF vlist THEN PRINT"---> Zeile"CVI(uzei$)"
     : "lnam$" = "wh$+wl$
<84> 18060       . IF NOT dlist THEN 18150
<17> 18070       ... ulst=CVI(ulst$)
<83> 18080       ... GET datt,ulst
<14> 18090       ... IF MID$(lst$,6,4)="... " THEN 18150
<38> 18100       ..... addr!=VAL("&H"+MID$(lst$,6,4)):IF addr
     !<0 THEN addr!=addr!+2^16
<93> 18110       ..... a=uadr!-addr!
<67> 18120       ..... a=a+a+11
<53> 18130       ..... MID$(lst$,a)=wl$+wh$
<49> 18140       ..... PUT datt,ulst
<51> 18150       . uprv=CVI(uprv$)
<18> 18160       . unxt=CVI(unxt$)
<89> 18170       . GET datu,unxt
<15> 18180       ... LSET uprv$=MKI$(uprv)
<41> 18190       . PUT datu,unxt
<24> 18200       . GET datu,uprv
<84> 18210       ... LSET unxt$=MKI$(unxt)
<75> 18220       . PUT datu,uprv
<25> 18230 WEND
<76> 18240 RETURN
<40> 18250 '
<98> 18260 '----- 8-bit-Wert
<18> 18270 GOSUB 18520
<91> 18280 wh$=""
<36> 18290 IF wh<>0 THEN fe=9:zzz=zeile:zeile=CVI(uzei$
     ):GOSUB 18960:zeile=zzz:wh=0:wl=0
<58> 18300 IF comdat THEN LSET byte$=CHR$(wl) ELSE LSET
      byte$="&H"+wl$+","
<55> 18310 PUT datz,uadr!-start!+1
<71> 18320 RETURN
<35> 18330 '
<48> 18340 '----- 16-bit-Wert
<13> 18350 GOSUB 18520
<82> 18360 IF comdat THEN LSET byte$=CHR$(wl) ELSE LSET
      byte$="&H"+wl$+","
<79> 18370 PUT datz,uadr!-start!+1
<74> 18380 IF comdat THEN LSET byte$=CHR$(wh) ELSE LSET
      byte$="&H"+wh$+","
<17> 18390 PUT datz,uadr!-start!+2
<66> 18400 RETURN
<30> 18410 '
<25> 18420 '----- relativer Wert
<32> 18430 of!=wert!-uadr!-1
<21> 18440 IF of!>129 OR of!<-126 THEN fe=14:GOSUB 1896
     0:of!=-2
<88> 18450 of=of!
<78> 18460 IF of<0 THEN of=256+of
<93> 18470 wh$=""
<80> 18480 wl$=HEX$(of,2)
```

```
<54> 18490 IF comdat THEN LSET byte$=CHR$(of) ELSE LSET
     byte$="&H"+wl$+","
<57> 18500 PUT datz,uadr!-start!+1
<73> 18510 RETURN
<32> 18520 wh=INT(wert!/256):wl=wert! MOD 256
<70> 18530 wh$=HEX$(wh,2):wl$=HEX$(wl,2)
<85> 18540 RETURN
<49> 18550 '
< 3> 18560 '---------- Test auf gültigen Labelnamen ---
     --
<57> 18570 '
<21> 18580 IF LEFT$(a$,1)<"A" OR LEFT$(a$,1)>"Z" THEN k
     einlabel=true:RETURN
<68> 18590 IF LEN(a$)>6 THEN fe=7:GOSUB 18960
<27> 18600 a$=LEFT$(a$+".... ",6)
<65> 18610 keinlabel=false
<80> 18620 RETURN
<44> 18630 '
<30> 18640 '---------- Undef. Label in Tabelle eintrage
     n
<52> 18650 '
<36> 18660 umax=umax+1
<70> 18670 GET datu,2
<16> 18680       uprv=CVI(uprv$)
<57> 18690       LSET uprv$=MKI$(umax)
<38> 18700 PUT datu,2
<92> 18710 GET datu,uprv
<49> 18720       unxt=CVI(unxt$)
<64> 18730       LSET unxt$=MKI$(umax)
<97> 18740 PUT datu,uprv
<84> 18750 LSET ubez$=a$
<75> 18760 LSET uadr$=MKS$(adr!+lang-index-distanz)
<23> 18770 LSET ulst$=MKI$(lzeil)
<51> 18780 LSET uzei$=MKI$(zeile)
<45> 18790 LSET u816$=MKI$(u816)
<16> 18800 LSET uprv$=MKI$(uprv)
<98> 18810 LSET unxt$=MKI$(unxt)
<70> 18820 PUT datu,umax
<90> 18830 RETURN
<54> 18840 '
<97> 18850 '---------- Labelwert aus Tabelle holen ----
     --
<62> 18860 '
<33> 18870 GET datl,1
< 5> 18880 WHILE CVI(lnxt$)<>1 AND a$<>lbez$
<49> 18890       GET datl,CVI(lnxt$)
<34> 18900 WEND
<89> 18910 IF CVI(lnxt$)=1 THEN found=false:wert!=0:RET
     URN
<57> 18920 found=true
<73> 18930 wert!=CVS(lwrt$)
<97> 18940 RETURN
<33> 18950 '---------- Fehlerroutine ------------------
     --
<25> 18960 fehler=true:feanz=feanz+1
<30> 18970 PRINT #datf,"Zeile :"+RIGHT$(".. "+STR$(zeil
     e),4)+" Fehler Nr.:"+RIGHT$(" "+STR$(fe),2)+" - "+
     fe$(fe)
<75> 18980 IF vlist THEN PRINT"***** Fehler Nr.:"fe"- "
     fe$(fe)
<13> 18990 IF NOT dlist THEN RETURN
<20> 19000 LSET lst$=SPACE$(18)+">>>>> "+fe$(fe)+" <<<<
     <"
<23> 19010 PUT datt,lzeil:lzeil=lzeil+1
<64> 19020 RETURN
```

```
LISTING >DISASS<, REMARK = >REM<.

<53> 10000 ' Z80-DISASSEMBLER.. Version vom 21.08.86
<86> 10010 PRINT"--------------------------"
<58> 10020 PRINT"..... Z80-DISASSEMBLER"
<77> 10030 PRINT"$1986. by Andreß Freystatzky"
<98> 10040 PRINT"--------------------------"
<15> 10050 esc$=CHR$(27):bel$=CHR$(7):bs$=CHR$(8):crlf$
     =CHR$(13)+CHR$(10):invon$=esc$+"p":invoff$=esc$+"q
     "
<71> 10060 false=0:true=-1
<38> 10070 ok$="jJyY+"+CHR$(22)+" "+CHR$(13):ko$="nN-"+
     CHR$(28)
<10> 10080 GOSUB 11950
<34> 10090 '
<87> 10100 PRINT"RAM oder Disk ... : R"+invon$+"AM"+bs$
     +bs$+bs$+invoff$;:GOSUB 11890
<56> 10110 ram=ok
<43> 10120 IF ram THEN 10130 ELSE 10180
<35> 10130.. PRINT"RAM"
<15> 10140.. nam$="--RAM--"
<47> 10150.. offset=0
<31> 10160.. GOTO 10310
<11> 10170 'else
<22> 10180.. PRINT"Disk"
<50> 10190.. INPUT"Dateiname ....... : ",nam$:nam$=UPPE
     R$(nam$)
<81> 10200.. IF INSTR(nam$,".")=0 THEN nam$=nam$+".COM"
<88> 10210.. IF FIND$(nam$)="" THEN PRINT bel$+"Datei "
     +nam$" nicht gefunden !":GOTO 10190
< 5> 10220.. PRINT"Dateibeginn ..... : &H0100"+bs$+bs$+
     bs$+bs$;
<42> 10230.. INPUT "",a$:IF a$="" THEN a$="0100"
<92> 10240.. GOSUB 11630:offset=a
<85> 10250.. xblock=1:xbyte=1
<13> 10260.. OPEN "R",#1,nam$,128
< 3> 10270.. FIELD #1,128 AS xinh$
<97> 10280.. GET #1,xblock
<69> 10290 'endif
<86> 10300 ' ---------- Bildschirmlisting ? ----------
<86> 10310 PRINT"Bildschirmlisting : j"+invon$+"a"+bs$+
     bs$+invoff$;:GOSUB 11890
< 4> 10320 vlist=ok:IF vlist THEN PRINT"JA" ELSE PRINT"
     NEIN"
<25> 10330 ' ---------- Listdatei ? ----------
< 8> 10340 PRINT"Listdatei ....... : j"+invon$+"a"+bs$+
     bs$+invoff$;:GOSUB 11890
<73> 10350 dlist=ok:IF dlist THEN PRINT"JA" ELSE PRINT"
     NEIN":GOTO 10420
<76> 10360.. IF ram THEN INPUT"Name der Datei .. : ",na
     mout$:namout$=UPPER$(namout$)
<79> 10370.. IF NOT ram THEN i=INSTR(nam$,"."):namout$=
     LEFT$(nam$,i-1)
< 4> 10380.. IF INSTR(namout$,".")=0 THEN namout$=namou
     t$+".DIS"
< 2> 10390.. IF INSTR(namout$,":")=0 THEN namout$="M:"+
     namout$
<24> 10400.. OPEN "O",#2,namout$
<82> 10410 ' ---------- Druckerlisting ? ----------
< 9> 10420 PRINT"Druckerlisting .. : j"+invon$+"a"+bs$+
     bs$+invoff$;:GOSUB 11890
<53> 10430 plist=ok:IF plist THEN PRINT"JA" ELSE PRINT"
     NEIN"
<62> 10440 ' ---------- Adressbereich ----------
<12> 10450 PRINT"Startadresse .... : &H"+HEX$(offset,4)
     +bs$+bs$+bs$+bs$;:INPUT"",a$:IF a$="" THEN a$=HEX$
     (offset)
<34> 10460 GOSUB 11630:anfang=a
<29> 10470 IF anfang<offset THEN PRINT bel$+"Startadres
     se "+HEX$(anfang)" ungültig !":GOTO 10450
<42> 10480 '
<28> 10490 PRINT"Endadresse ..... : &HFFFF"+bs$+bs$+bs
     $+bs$;:INPUT"",a$:IF a$="" THEN a$="FFFF"
<64> 10500 GOSUB 11630:ende=a
<78> 10510 IF anfang>ende THEN PRINT bel$+"Falscher Adr
     essbereich !":GOTO 10450
<45> 10520 ' ---------- Kopfzeile ----------
<98> 10530 aus$="Datei : "+UPPER$(nam$)+".. Start : &H"
     +HEX$(anfang,4)+".. Ende : &H"+HEX$(ende,4)
<54> 10540 IF vlist THEN PRINT aus$:PRINT
<82> 10550 IF dlist THEN PRINT #2,aus$:PRINT #2
<10> 10560 IF plist THEN LPRINT aus$:LPRINT
<27> 10570 ' ---------- Vorbelegungen ----------
<65> 10580 ist=anfang
<65> 10590 z=0
<25> 10600 dateiende=false
<16> 10610 ' ---------- Hauptschleife ----------
<21> 10620 WHILE ist<=ende
<43> 10630       adr=ist
<77> 10640       aus$=HEX$(adr,4)+". "
<94> 10650       index=false
<67> 10660       GOSUB 11670
<70> 10670       GOSUB 10950
<75> 10680       IF index THEN GOSUB 11290
<92> 10690       IF INSTR(bef$,"n")<>0 THEN GOSUB 11400
<90> 10700       IF INSTR (bef$,"d")<>0 THEN GOSUB 11540
<13> 10710.. IF bef$="" THEN bef$="???"
<10> 10720.. i=INSTR(bef$," ")
<45> 10730       IF i=0 THEN aus$=LEFT$(aus$+SPACE$(14),18)+b
     ef$:GOTO 10750
<23> 10740       . aus$=LEFT$(aus$+SPACE$(14),18)+LEFT$(LEFT$
     (bef$,i-1)+"..... ",6)+MID$(bef$,i+1)
<43> 10750       IF dateiende THEN IF ist>ende THEN aus$=crlf
     $+"Dateiende !!!"
< 4> 10760.. i$=INKEY$
<25> 10770       IF vlist THEN PRINT aus$:z=z+1:IF z=30 THEN
     z=0:i$="":WHILE i$="":i$=INKEY$:WEND
<31> 10780.. IF dlist THEN PRINT #2,aus$
<24> 10790.. IF plist THEN LPRINT aus$
<83> 10800.. IF NOT vlist AND NOT plist THEN PRINT".";
<37> 10810 ' ---------- Abbruch mit EXIT ----------
< 2> 10820.. IF i$=CHR$(27) THEN ist=ende+1
<27> 10830 WEND
<96> 10840 ' ---------- Wie geht's weiter ? ----------
<39> 10850 PRINT "1) neuer Bereich"
<65> 10860 PRINT "2) neue Datei"
< 7> 10870 PRINT "3) Programmende"
<33> 10880 a$="":WHILE a$<"1" OR a$>"3":a$=INKEY$:WEND
<40> 10890 IF a$="1" THEN 10450
<60> 10900 CLOSE
<26> 10910 IF a$="2" THEN 10100
<93> 10920 END
<61> 10930 ' ********** Unterprogramme ***********
```

```
      ****
<52> 10940 ' ---------- Byte interpretieren ----------
<53> 10950 IF (w=&HDD OR w=&HFD) THEN GOSUB 11190
<18> 10960 IF w=&HCB THEN 11090
< 4> 10970 IF w=&HED THEN 11150
<15> 10980 GOSUB 11240
<42> 10990 ON bit7 GOTO 11030,11060,11010
<26> 11000 ' ---------- bit7=0, bit7=3 Diverse Befehle
     aus der Tabelle ----------
<86> 11010 bef$=befehl$(w):RETURN
<94> 11020 ' ---------- bit7=1 LD-Befehle und HALT ----
     ------
<86> 11030 IF w=&H76 THEN bef$="HALT":RETURN
<13> 11040 bef$="LD "+register$(bit5)+","+register$(bit
     2):RETURN
<68> 11050 ' ---------- bit7=2 Arithmetik- und Logikbef
     ehle ----------
<31> 11060 IF bit5=0 OR bit5=1 OR bit5=3 THEN a$=" A,"
     ELSE a$=" "
<23> 11070 bef$=arith$(bit5)+a$+register$(bit2):RETURN
<83> 11080 ' ---------- &HCB Rotations-, Schiebe- und E
     inzelbitbefehle ----------
< 3> 11090 GOSUB 11670
<26> 11100 IF index THEN dis=w:GOSUB 11670
<64> 11110 GOSUB 11240
<13> 11120 IF bit7=0 THEN bef$=shift$(bit5)+" "+reg$ EL
     SE bef$=bitbef$(bit7)+STR$(bit5)+","+reg$
<55> 11130 RETURN
<31> 11140 ' ---------- &HED Ein-, Ausgabe & 16-bit-Bef
     ehle ----------
<89> 11150 GOSUB 11670
<62> 11160 IF w<&H40 OR w>&HBF THEN bef$="???" ELSE bef
     $=befehl$(w)
<71> 11170 RETURN
<64> 11180 ' ---------- IX oder IY zu Austausch gegen H
     L bereitstellen ----------
<37> 11190 index=true
<89> 11200 IF w=&HDD THEN i$="IX" ELSE i$="IY"
<76> 11210 GOSUB 11670
<54> 11220 RETURN
<41> 11230 ' ---------- Code zerlegen ----------
<63> 11240 bit7=(w AND &HC0)/64
<70> 11250 bit5=(w AND &H38)/8
<89> 11260 bit2=(w AND &H7)
<74> 11270 RETURN
<18> 11280 ' ---------- Indexregister IX oder IY statt
     HL einsetzen ----------
<90> 11290 i=INSTR(bef$,"HL")
<79> 11300 IF i=0 THEN bef$="???":RETURN
< 2> 11310 IF bef$="EX DE,HL" THEN bef$="???":RETURN
<22> 11320 IF bef$="ADD HL,HL" THEN bef$="ADD "+i$+","+
     i$:RETURN
< 6> 11330 IF INSTR(bef$,"(HL)")=0 THEN 11380
<44> 11340.. IF LEFT$(bef$,2)="JP" THEN 11380
<73> 11350.... IF ist-adr<3 THEN GOSUB 11670:dis=w
<72> 11360.... IF dis>127 THEN dis$=STR$(dis-256) ELSE
     dis$="+"+MID$(STR$(dis),2)
<74> 11370.... i$=i$+dis$
<34> 11380 bef$=LEFT$(bef$,i-1)+i$+MID$(bef$,i+2):RETUR
     N
<57> 11390 ' ---------- n und nn ersetzen ----------
<22> 11400 i=INSTR(bef$,"nn")
<32> 11410 IF i=0 THEN 11430 ELSE 11480
<16> 11420 ' ---------- 8-bit ----------
<14> 11430 i=INSTR(bef$,"n")
<94> 11440 GOSUB 11670
<98> 11450 bef$=LEFT$(bef$,i-1)+"&H"+HEX$(w,2)+MID$(bef
     $,i+1)
<76> 11460 RETURN
<94> 11470 ' ---------- 16-bit ----------
<91> 11480 GOSUB 11670:ww=w
<15> 11490 GOSUB 11670
<45> 11500 wert=w*256+ww
<26> 11510 bef$=LEFT$(bef$,i-1)+"&H"+HEX$(wert,4)+MID$(
     bef$,i+2)
<63> 11520 RETURN
<25> 11530 ' ---------- d ersetzen ----------
< 9> 11540 i=INSTR(bef$,"d")
< 2> 11550 GOSUB 11670
<94> 11560 IF w>127 THEN w=w-256
<73> 11570 w=w+2
<91> 11580 ww$=STR$(w):IF w>=0 THEN ww$="+"+MID$(ww$,2)
<93> 11590 a$="$"+ww$+" >&H"+HEX$(ist+w-2,4)
<79> 11600 bef$=LEFT$(bef$,i-1)+a$+MID$(bef$,i+1)
<62> 11610 RETURN
<93> 11620 ' ---------- Umwandlung hex --> dez --------
     --
<28> 11630 IF a$="" THEN a=0:RETURN
<95> 11640 a=VAL("&H"+a$):IF a<0 THEN a=a+2^16
<78> 11650 RETURN
<52> 11660 ' ---------- Byte lesen ----------
<32> 11670 IF ram THEN 11690 ELSE 11740
<23> 11680 ' ---------- RAM lesen ----------
<44> 11690 w=PEEK(ist)
<69> 11700 aus$=aus$+HEX$(w,2)+" "
<26> 11710 ist=ist+1
<69> 11720 RETURN
<88> 11730 ' ----------Disk lesen ----------
<30> 11740 n=ist-offset
<57> 11750 GOSUB 11820
<73> 11760 xbyte=n-128*(xblock-1)+1
<21> 11770 w=ASC(MID$(xinh$,xbyte,1))
< 2> 11780 aus$=aus$+HEX$(w,2)+" "
<58> 11790 ist=ist+1
<64> 11800 RETURN
< 0> 11810 ' ---------- Neuen Block lesen, falls nötig
     ----------
< 5> 11820 xneu=INT(n/128)+1
<41> 11830 IF xneu=xblock THEN RETURN
<76> 11840 GET #1,xneu
<25> 11850 IF EOF(1) THEN dateiende=true:ist=ende+1:xbl
     ock=xneu:RETURN
<19> 11860 xblock=xneu
<92> 11870 RETURN
<97> 11880 ' ---------- Ja/Nein +/- Abfrage einlesen --
     --------
<37> 11890 a$=""
<46> 11900 WHILE a$="":a$=INKEY$:WEND
<81> 11910 IF INSTR(ok$,a$)<>0 THEN ok=true:RETURN
< 0> 11920 IF INSTR(ko$,a$)<>0 THEN ok=false:RETURN
<89> 11930 GOTO 11890
<34> 11940 ' ---------- Befehlstabellen erstellen -----
     -----
<57> 11950 DIM register$(7),shift$(7),bitbef$(3),arith$
     (7),befehl$(255)
<31> 11960 FOR i=0 TO 7:READ register$(i):NEXT
<60> 11970 FOR i=0 TO 7:READ shift$(i):NEXT
<45> 11980 FOR i=1 TO 3:READ bitbef$(i):NEXT
<21> 11990 FOR i=0 TO 7:READ arith$(i):NEXT
<92> 12000 FOR i=0 TO &H7F:READ befehl$(i):NEXT
<38> 12010 FOR i=&H80 TO &H9F:befehl$(i)="":NEXT
< 6> 12020 FOR i=&HA0 TO &HFF:READ befehl$(i):NEXT
<54> 12030 RETURN
< 6> 12040 ' ---------- DATA's
<23> 12050 DATA B,C,D,E,H,L,(HL),A
<28> 12060 DATA RLC,RRC,RL,RR,SLA,SRA,???,SRL
<20> 12070 DATA BIT,RES,SET
<46> 12080 DATA ADD,ADC,SUB,SBC,AND,XOR,OR,CP
<38> 12090 '
<65> 12100 DATA NOP,"LD BC,nn","LD (BC),A",INC BC,INC B
     ,DEC B,"LD B,n",RLCA
<28> 12110 DATA "EX AF,AF'","ADD HL,BC","LD A,(BC)",DEC
      BC,INC C,DEC C,"LD C,n",RRCA
<14> 12120 DATA DJNZ d,"LD DE,nn","LD (DE),A",INC DE,IN
     C D,DEC D,"LD D,n",RLA
<61> 12130 DATA JR d,"ADD HL,DE","LD A,(DE)",DEC DE,INC
      E,DEC E,"LD E,n",RRA
<68> 12140 DATA "JR NZ,d","LD HL,nn","LD (nn),HL",INC H
     L,INC H,DEC H,"LD H,n",DAA
<63> 12150 DATA "JR Z,d","ADD HL,HL","LD HL,(nn)",DEC H
     L,INC L,DEC L,"LD L,n",CPL
<20> 12160 DATA "JR NC,d","LD SP,nn","LD (nn),A",INC SP
     ,INC (HL),DEC (HL),"LD (HL),n",SCF
<76> 12170 DATA "JR C,d","ADD HL,SP","LD A,(nn)",DEC SP
     ,INC A,DEC A,"LD A,n",CCF
<16> 12180 'Die folgenden Befehle sind nur mit &HED als
      Vorsatzbyte gültig
<16> 12190.. DATA "IN B,(C)","OUT (C),B","SBC HL,BC","L
     D (nn),BC",NEG,RETN,IM 0,"LD I,A"
<58> 12200.. DATA "IN C,(C)","OUT (C),C","ADD HL,BC","L
     D BC,(nn)",,RETI,,"LD R,A"
<93> 12210.. DATA "IN D,(C)","OUT (C),D","SBC HL,DE","L
     D (nn),DE",,,IM 1,"LD A,I"
<19> 12220.. DATA "IN E,(C)","OUT (C),E","ADC HL,DE","L
     D DE,(nn)",,,IM 2,"LD A,R"
<51> 12230.. DATA "IN H,(C)","OUT (C),H","SBC HL,HL","L
     D (nn),HL",,,,RRD
<29> 12240.. DATA "IN L,(C)","OUT (C),L","ADC HL,HL","L
     D HL,(nn)",,,,RLD
<24> 12250.. DATA ,,"SBC HL,SP","LD (nn),SP",,,,
< 0> 12260.. DATA "IN A,(C)","OUT (C),A","ADC HL,SP","L
     D SP,(nn)",,,,
<94> 12270.. DATA LDI,CPI,INI,OUTI,,,,,LDD,CPD,IND,OUTD
     ,,,,
< 7> 12280.. DATA LDIR,CPIR,INIR,OTIR,,,,,LDDR,CPDR,IND
     R,OTDR,,,,
<93> 12290 'Ab hier wieder ohne &HED
<88> 12300 DATA RET NZ,POP BC,"JP NZ,nn",JP nn,"CALL NZ
     ,nn",PUSH BC,"ADD A,n",RST &H00
<93> 12310 DATA RET Z,RET,"JP Z,nn",->,"CALL Z,nn",CALL
      nn,"ADC A,n",RST &H08
< 7> 12320 DATA RET NC,POP DE,"JP NC,nn","OUT (n),A","C
     ALL NC,nn",PUSH DE,"SUB n",RST &H10
<59> 12330 DATA RET C,EXX,"JP C,nn","IN A,(n)","CALL C,
     nn",->,"SBC A,n",RST &H18
<96> 12340 DATA RET PO,POP HL,"JP PO,nn","EX (SP),HL","
     CALL PO,nn",PUSH HL,"AND n",RST &H20
<93> 12350 DATA RET PE,JP (HL),"JP PE,nn","EX DE,HL","C
     ALL PE,nn",->,"XOR n",RST &H28
<35> 12360 DATA RET P,POP AF,"JP P,nn",DI,"CALL P,nn",P
     USH AF,"OR n",RST &H30
<69> 12370 DATA RET M,"LD SP,HL","JP M,nn",EI,"CALL M,n
     n",->,"CP n",RST &H38
```

MAU – MAU

Auch wenn der JOYCE eigentlich mehr der Personal Computer des Geschäftsmannes als die Spielmaschine des Hackers ist, gibt es sicherlich auch jüngere JOYCE-Besitzer, die sich besonders gerne auf diesem Gebiet engagieren. Auch sind für den Anfänger Spiele oft anschaulicher als nüchterne Verwaltungsprogramme (es heißt ja nicht umsonst: spielend lernen!). Da recht selten Spielprogramme für den JOYCE abgedruckt werden, entschloß ich mich eben, selbst eines zu schreiben. Weil ich aber keine Möglichkeit fand, bewegte Bilder für Action-Spiele zu konstruieren, versuchte ich mein Glück mit einem Kartenspiel. Meine Wahl fiel auf das Spiel MAU-MAU (eigentlich spiele ich lieber Skat, aber für einen ersten Versuch erschien mir Skat doch noch zu schwer zu programmieren).

Um etwas mehr Komfort in das Programm zu bringen, habe ich die Möglichkeit, ein kleines Turnier (9 Durchgänge, mit Punkten) zu machen, eingebaut. Die Punktezählung kann in Zeile 380-430 nachgelesen werden, die Spielregeln stehen in Zeile 480-670 und die Bedienung (Karten ziehen usw.) in Zeile 74060. Zu Zeile 630/40 ist folgendes zu sagen: es war nicht nötig, mehr als max. 20 Karten auf dem Bildschirm anzeigen zu können, da der Computer normalerweise schon gewonnen hat, bevor der Gegner überhaupt 20 Karten hat sammeln können (aber das ist ja nicht der Sinn des Spiels...).

Das Programm selbst bedarf wohl keinen weiteren Erläuterungen. Es ist durch »Überschriften« gegliedert und der Einfachheit halber fortlaufend durchnumeriert. Auch wurde viel mit Unterprogrammen gearbeitet, die ab Zeile 1920 untergebracht sind.

Für den interessierten Spieler: als Level 3 ließe sich noch die Möglichkeit einbauen, daß der Computer sich, nachdem er einen Buben gelegt hat, auch eine Farbe wünschen kann!

(B. Breede)

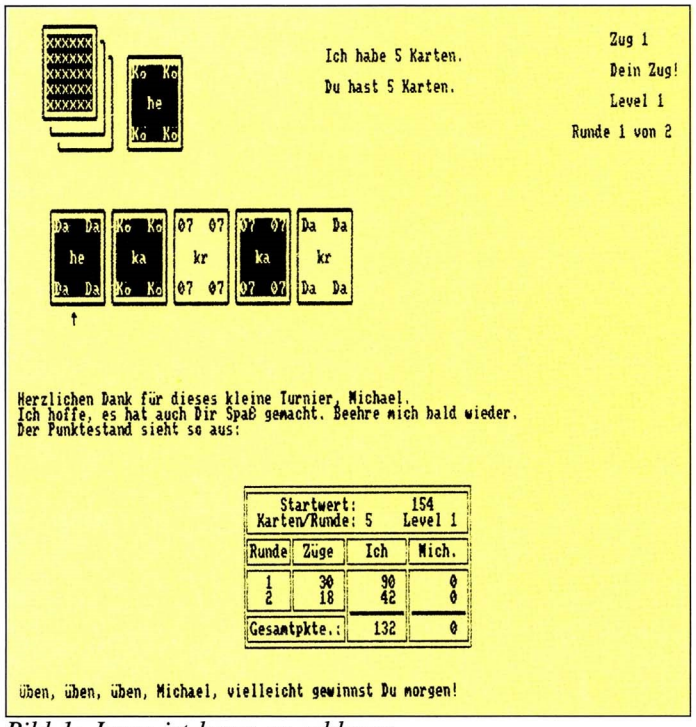


Bild 1: Joyce ist kaum zu schlagen ...

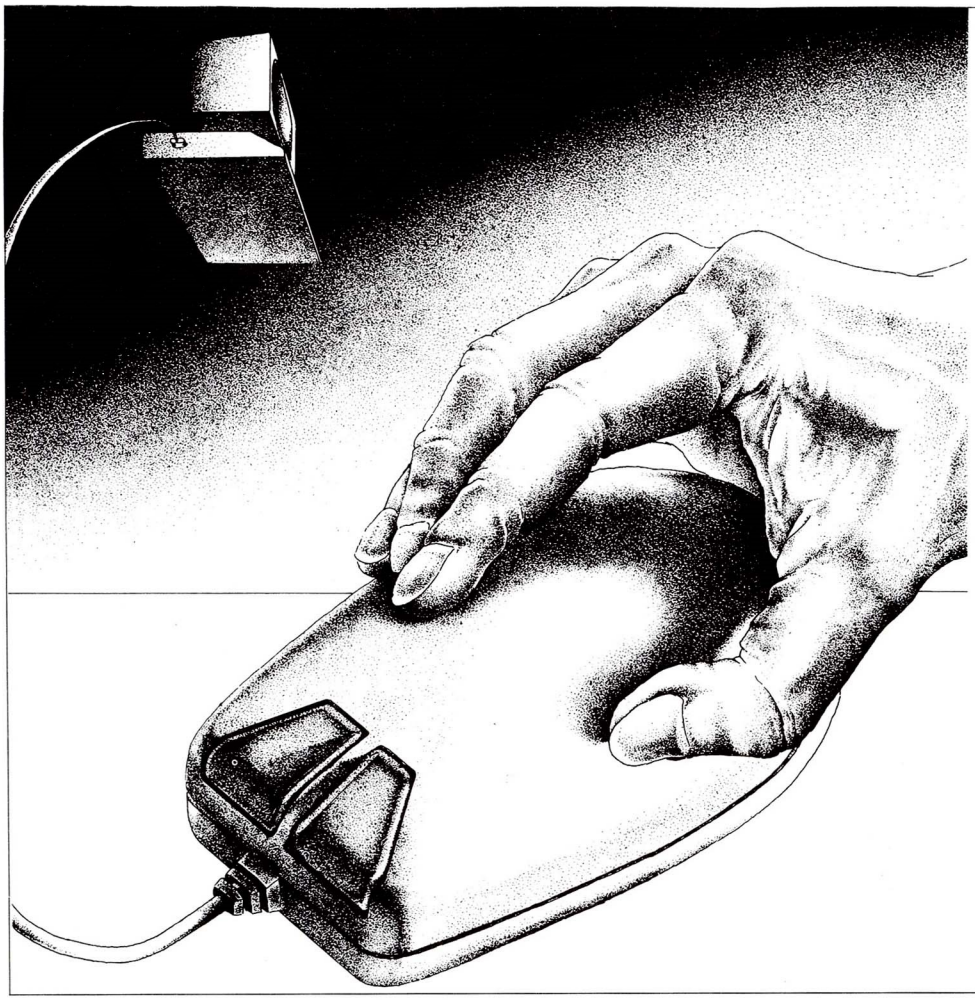
VARIABLENLISTE "MAU-MAU.BAS"

(LV=Laufvar. =HV=Hilfsvar.)

a	: HV
a–	: HV
b	: HV
cgew	: Gewinnpkte. des Computers
cgew–(9)	: Gewinnpkte. des Computers (einzelne Runden)
ckarte–(20)	: Karten des Computers (einzelne Karte)
ckarten	: Anzahl der Karten des Computers
ckarten–	: Karten des Computers (gesamt)
cls–	: Bildschirm löschen
coff–	: Cursor ausschalten
col– (grn./schw.)	: Farbe der Karte auf Bildschirm
con–	: Cursor einschalten
du	: Anzahl der Spieldurchläufe
fa–	: Gewünschte Farbe (bei Bube)
farbe–	: Spielfarbe der Karte (ka/he/pi/kr)
FNl–(x–)	: Spielfarbe von x– bestimmen
FNloc–(x,y)	: Cursor auf Spalte x, Zeile y setzen
FNr–(x–)	: Wert von x– bestimmen
ggew	: Gewinnpkte. des Gegners
gew–(9)	: Gewinnpkte. des Gegners (einzelne Runden)
gkarte–(20)	: Karten des Gegners (einzelne Karte)
gkarten	: Anzahl der Karten des Gegners
gkarten–	: Karten des Gegners (gesamt)
I	: LV
invoff–	: Invers-Modus ausschalten
invon–	: Invers-Modus einschalten
J	: LV
k	: HV
ka, ka–	: Karten/Spieler
karte–	: Vom Stapel genommene Karte
level–	: Schwierigkeitsstufe
M	: LV
n	: HV
name–	: Name des Gegners
name1–	: Name des Gegners, auf 5 Zeichen gekürzt
obkarte–	: offenliegende Karte
s	: HV
stapel1–	: Stapel, von dem genommen wird
stapel2–	: Stapel, auf den abgelegt wird
startwert, startwert–	: Startwert des Zufallszahlengenerators
stoff–	: Statuszeile ausschalten
ston–	: Statuszeile einschalten
taste–	: von der Tastatur eingelesenes Zeichen
v	: HV
wert–	: Wert der Karte
x	: HV
y	: HV
z	: HV
zug(9), zug–(9)	Anzahl der einzelnen Züge pro Runde

```
LISTING >MAU-MAU <, REMARK = >'<.
<78> 10 '********** Initialisierung **********
< 2> 20 '
<84> 30 cls$=CHR$(27)+"E"+CHR$(27)+"H"
< 9> 40 invon$=CHR$(27)+"p"
<65> 50 invoff$=CHR$(27)+"q"
<17> 60 con$=CHR$(27)+"e"
<70> 70 coff$=CHR$(27)+"f"
<55> 80 stoff$=CHR$(27)+"0"
< 3> 90 ston$=CHR$(27)+"1"
<50> 100 DEF FNloc$(x,y)=CHR$(27)+"Y"+CHR$(32+y)+CHR$(3
     2+x)
<60> 110 DEF FNr$(x$)=RIGHT$(x$,2)
< 7> 120 DEF FNl$(x$)=LEFT$(x$,2)
< 0> 130 DIM gkarte$(20),ckarte$(20),ggew$(9),cgew$(9),
     zug(9),zug$(9)
<26> 140 '
< 7> 150 '********** Vorspann **********
<30> 160 '
<53> 170 PRINT cls$;stoff$;
<15> 180 PRINT FNloc$(14,1);CHR$(150);STRING$(59,154);C
     HR$(156)
<43> 190 FOR I=2 TO 8
<11> 200 PRINT FNloc$(14,I);CHR$(149);FNloc$(74,I);CHR$
     (149)
<37> 210 NEXT
<89> 220 PRINT FNloc$(14,9);CHR$(147);STRING$(59,154);C
     HR$(153)
<51> 230 PRINT FNloc$(37,3);invon$;CHR$(134);STRING$(13
     ,138);CHR$(140);invoff$
<18> 240 PRINT FNloc$(37,4);invon$;CHR$(133);"** MAU-MA
     U **";CHR$(133);invoff$
<71> 250 PRINT FNloc$(37,5);invon$;CHR$(131);STRING$(13
     ,138);CHR$(137);invoff$
<21> 260 PRINT FNloc$(22,7);"Version 1.1 $1987 by B. Br
     eede Software & Co."
<11> 270 PRINT:PRINT:PRINT
<78> 280 INPUT"Ihr werter Name (Vorname reicht)?. ",nam
     e$
<51> 290 PRINT
<73> 300 PRINT"Einen wunderschönen guten Tag, ";name$;"
      "
<55> 310 PRINT:PRINT"Wollen wir beide ein kleines Turni
     er machen? ";
<69> 320 taste$=""
<36> 330 WHILE taste$="" OR taste$<>"j" AND taste$<>"n"
     :taste$=INKEY$:WEND:PRINT taste$
< 3> 340 IF taste$="n" THEN du=1:GOTO 470
<76> 350 PRINT:PRINT"Mit wieviel Durchgängen (2-9)? ";
<32> 360 taste$=" "
<10> 370 WHILE taste$=" " OR ASC(taste$)<50 OR ASC(tast
     e$)>57:taste$=INPUT$(1):WEND:PRINT taste$:du=VAL(t
     aste$)
<44> 380 PRINT:PRINT"Sehr Gut! Die Rechenregeln für's T
     urnier:"
<83> 390 PRINT:PRINT TAB(10)"- 7, 8, 9, 10 zählen 7, 8,
      9 bzw. 10 Punkte"
<89> 400 PRINT TAB(10)"- Bube, Dame, König zählen je 10
      Punkte"
<70> 410 PRINT TAB(10)"- As zählt 11 Punkte"
<44> 420 PRINT:PRINT TAB(10)"- Die Punkte des Verlierer
     s werden addiert, mit der Anzahl der Karten des Ge
     g-"
<19> 430 PRINT TAB(10)". ners multipliziert und das Erg
     ebnis wird dem Gewinner gutgeschrieben"
<43> 440 PRINT:PRINT:PRINT TAB(8)"Alles klar? Dann drüc
     k' bitte eine Taste und lies dir die Spielregeln d
     urch!"
<41> 450 PRINT coff$
<20> 460 taste$=INPUT$(1)
<40> 470 PRINT cls$;coff$
<27> 480 PRINT"Ich spiele dieses Spiel mit Dir nach fol
     genden Regeln:"
<53> 490 PRINT
<92> 500 PRINT TAB(10)"- gespielt wird mit einem 32er S
     kat-Blatt
<72> 510 PRINT TAB(10)"- Die erste Karte wird vom Stape
     l genommen und ist (mit Ausnahme einer 7) neutral
<43> 520 PRINT TAB(10)"- Du fängst an
< 6> 530 PRINT TAB(10)"- Gespielt wird durch Bedienen d
     er offenliegenden Karte (auflegen einer Karte
<53> 540 PRINT TAB(10)". gleichen Wertes oder gleicher
     Farbe oder eines Buben, egal welche Farbe)
<10> 550 PRINT TAB(10)"- wird eine 7, eine 8 oder ein B
     ube aufgelegt, gilt folgendes:"
<19> 560 PRINT TAB(15)"7:..... Der Gegner muß zwei Kart
     en vom Stapel nehmen"
<25> 570 PRINT TAB(15)"8:..... Der Gegner muß aussetzen
      (Der Spieler darf eine weitere Karte"
<67> 580 PRINT TAB(15)"..... aufdecken)"
<77> 590 PRINT TAB(15)"Bube:.. Der Spieler der einen Bu
     ben auflegt, darf sich eine Farbe aus-"
<58> 600 PRINT TAB(15)"..... suchen, die der andere b
     edienen muß (Bube auf Bube ist erlaubt)"
<22> 610 PRINT TAB(10)"- Gegenwerfen ist nicht erlaubt!
     "
<31> 620 PRINT TAB(10)"- Gewinner ist, wer zuerst keine
      Karten mehr hat"
<82> 630 PRINT TAB(10)"- Wer zuerst mehr als 20 Karten
     auf der Hand hat, verliert (weil ich nicht mehr"
<18> 640 PRINT TAB(10)". als 20 Karten anzeigen kann, s
     orry!)"
<69> 650 PRINT TAB(10)"- Wenn der Stapel, von dem genom
     men wird, leer ist, wird er mit dem Ablegestapel"
<88> 660 PRINT TAB(10)". vertauscht"
<34> 670 PRINT:PRINT"Garantie: Ich bin so programmiert
     worden, daß ich Deine Karten nicht kenne!!"
<53> 680 PRINT
<16> 690 PRINT FNloc$(30,31);"Weiter mit beliebiger Tas
     te...";
<82> 700 IF INKEY$="" THEN startwert=startwert+1 ELSE G
     OTO 730
<29> 710 IF startwert=99999999# THEN startwert=0
<62> 720 GOTO 700
<86> 730 PRINT cls$;con$
<37> 740 PRINT TAB(10)"- mit den Cursor-Tasten wählst D
     u Deine Karte aus"
< 8> 750 PRINT TAB(10)"- mit <SPACE> ziehst Du die ausg
     ewählte Karte"
<90> 760 PRINT TAB(10)"- mit <K> nimmst Du eine (bzw. 2
     , bei einer 7) Karte vom Stapel"
<47> 770 PRINT:PRINT"In welchem Level möchtest Du spiel
     en, ";name$;" (1/2)?. ";
<90> 780 WHILE level$="" OR level$<>"1" AND level$<>"2"
     :level$=INKEY$:WEND:PRINT level$
<73> 790 PRINT:PRINT"Wieviele Karten soll jeder zu Anfa
     ng haben (3-9)? ";
< 0> 800 ka$=" ":WHILE ASC(ka$)<51 OR ASC(ka$)>57:ka$=I
     NPUT$(1):WEND:ka=VAL(ka$)
<52> 810 RANDOMIZE startwert
<94> 820 PRINT cls$
<83> 830 FOR M=1 TO du
<79> 840 stapel1$="ka07ka08ka09ka10kaBukaDakaKökaAshe07
     he08he09he10heBuheDaheKöheAspi07pi08pi09pi10piBupi
     DapiKöpiAskr07kr08kr09kr10krBukrDakrKökrAs"
<11> 850 stapel2$=""
<90> 860 ckarten$="":gkarten$="":zug(M)=0
<62> 870 FOR I=1 TO 20:gkarte$(I)="":ckarte$(I)="":NEXT
<51> 880 PRINT coff$
<69> 890 IF du>1 THEN PRINT FNloc$(72,8);"Runde";M;"von
     ";du
<26> 900 '
< 1> 910 '********** Kartenzuweisung Gegner **********
<30> 920 '
<18> 930 FOR I=1 TO ka
<26> 940 IF stapel1$="" THEN GOSUB 4020
<82> 950 z=INT(RND*(LEN(stapel1$)/4))*4+1
<11> 960 gkarte$(I)=MID$(stapel1$,z,4)
< 8> 970 stapel1$=LEFT$(stapel1$,z-1)+MID$(stapel1$,z+4
     )
<58> 980 NEXT
<36> 990 gkarten=ka
<54> 1000 FOR I=1 TO gkarten
< 6> 1010 gkarten$=gkarten$+gkarte$(I)
<88> 1020 NEXT
<92> 1030 '
<75> 1040 '********** Kartenzuweisung Computer ********
     **
<98> 1050 '
<34> 1060 FOR I=1 TO ka
<96> 1070 IF stapel1$="" THEN GOSUB 4020
<10> 1080 z=INT(RND*(LEN(stapel1$)/4))*4+1
<62> 1090 ckarte$(I)=MID$(stapel1$,z,4)
<28> 1100 stapel1$=LEFT$(stapel1$,z-1)+MID$(stapel1$,z+
     4)
<87> 1110 NEXT
<43> 1120 ckarten=ka
<96> 1130 FOR I=1 TO ckarten
<35> 1140 ckarten$=ckarten$+ckarte$(I)
< 0> 1150 NEXT
< 4> 1160 '
<47> 1170 '********** Oberste Karte aufdecken *********
     *
<10> 1180 '
```

```
< 5> 1190 IF stapel1$="" THEN GOSUB 4020
<15> 1200 GOSUB 1920:'----- Karte vom Stapel nehmen ---
     --
<87> 1210 obkarte$=karte$
<68> 1220 stapel2$=stapel2$+obkarte$
<96> 1230 '
<51> 1240 '+++++ Stapel anzeigen +++++
< 3> 1250 '
<84> 1260 PRINT FNloc$(3,1);CHR$(150);STRING$(6,154);CH
     R$(156)
<31> 1270 PRINT FNloc$(3,2);CHR$(149);invon$;STRING$(6,
     "x");invoff$;CHR$(149);CHR$(156)
<33> 1280 PRINT FNloc$(3,3);CHR$(149);invon$;STRING$(6,
     "x");invoff$;STRING$(2,149);CHR$(156)
<22> 1290 FOR I=1 TO 3
<94> 1300 PRINT FNloc$(3,3+I);CHR$(149);invon$;STRING$(
     6,"x");invoff$;STRING$(3,149)
<91> 1310 NEXT
< 3> 1320 FOR I=1 TO 3
<59> 1330 PRINT FNloc$(2+I,6+I);CHR$(147);STRING$(6,154
     );CHR$(153);STRING$(3-I,149)
< 1> 1340 NEXT
< 5> 1350 '
<31> 1360 '********** SPIELBEGINN **********
<11> 1370 '
< 4> 1380 GOSUB 2000:'----- offene Karte anzeigen -----
<13> 1390 PRINT FNloc$(77,6);"Level ";level$
<84> 1400 PRINT FNloc$(40,3);"Ich habe";ckarten;"Karten
     ."
<33> 1410 PRINT FNloc$(40,5);"Du hast";gkarten;"Karten.
     "
<61> 1420 GOSUB 2080:'----- Karten Gegner anzeigen ----
     -
<36> 1430 zug(M)=zug(M)+1:PRINT FNloc$(77,2);"Zug";zug(
     M):GOSUB 3070:'----- Zug Gegner -----
<86> 1440 IF gkarten=0 THEN GOSUB 3800:GOTO 1480 ELSE I
     F v=1 THEN v=0:GOSUB 3580:GOTO 1480
< 3> 1450 zug(M)=zug(M)+1:PRINT FNloc$(77,2);"Zug";zug(
     M):GOSUB 2630:'----- Zug Computer -----
<21> 1460 IF ckarten=0 THEN GOSUB 3580:GOTO 1480 ELSE I
     F v=1 THEN v=0:GOSUB 3580:GOTO 1480
< 9> 1470 GOTO 1430
<83> 1480 taste$=INPUT$(1)
< 1> 1490 IF du=1 THEN PRINT con$;cls$;ston$:END
<15> 1500 PRINT cls$
<21> 1510 IF M=du GOTO 1550
<93> 1520 PRINT FNloc$(20,1);"Die letzte Runde brachte
     ";
<93> 1530 IF cgew=0 THEN PRINT"Dir";ggew*ckarten;"Punkt
     e.... "
<71> 1540 IF ggew=0 THEN PRINT"mir";cgew*gkarten;"Punkt
     e.... "
<86> 1550 NEXT M
<33> 1560 PRINT cls$
<22> 1570 PRINT"Herzlichen Dank für dieses kleine Turni
     er, ";name$;"."
<76> 1580 PRINT"Ich hoffe, es hat auch Dir Spaß gemacht
       Beehre mich bald wieder."
<39> 1590 PRINT"Der Punktestand sieht so aus:"
<54> 1600 PRINT:PRINT:PRINT
<92> 1610 startwert$=DEC$(startwert,"#########"):FOR I=
     1 TO du:zug$(I)=DEC$(zug(I),"####"):NEXT
<46> 1620 PRINT TAB(30)CHR$(134);STRING$(28,138);CHR$(1
     40)
<11> 1630 PRINT TAB(30);CHR$(133);"... Startwert: ";sta
     rtwert$;"... ";CHR$(133)
<35> 1640 PRINT TAB(30);CHR$(133);" Karten/Runde: ";ka$
     ;"... Level ";level$;" ";CHR$(133)
<14> 1650 PRINT TAB(30)CHR$(135);STRING$(5,138);CHR$(14
     2);STRING$(6,138);CHR$(142);STRING$(7,138);CHR$(14
     2);STRING$(7,138);CHR$(141)
<96> 1660 IF LEN(name$)<5 THEN name1$=name$+STRING$(5-L
     EN(name$),32)
<39> 1670 IF LEN(name$)>5 THEN name1$=LEFT$(name$,4)+".
     "
<80> 1680 IF LEN(name$)=5 THEN name1$=name$
<37> 1690 PRINT TAB(30)CHR$(133);"Runde";CHR$(133);" Zü
     ge ";CHR$(133);". Ich. ";CHR$(133);" ";name1$;" ";
     CHR$(133)
<31> 1700 PRINT TAB(30)CHR$(135);STRING$(5,138);CHR$(14
```

Joyce-MousePack DM 249.-

Bestellungen oder kostenloses Info (Ihren Computertyp angeben !) bei:
Imperial SOFTware Systems Gerdes KG, Rochus-Center
Lessenicher Str. 9, 5300 Bonn 1, Tel.: (02 28) 61 62 10 oder 25 24 74

```
      3);STRING$(6,138);CHR$(143);STRING$(7,138);CHR$(14
      3);STRING$(7,138);CHR$(141)
<76>  1710 FOR I=1 TO du
< 5>  1720 PRINT TAB(30)CHR$(133);USING "###";I;
<19>  1730 PRINT". ";CHR$(133);" ";zug$(I);" ";CHR$(133)
      ;" ";cgew$(I);" ";CHR$(133);" ";ggew$(I);" ";CHR$(
      133)
<43>  1740 NEXT I
<52>  1750 cgew=0
<79>  1760 ggew=0
<94>  1770 FOR I=1 TO du
<64>  1780 cgew=cgew+VAL(cgew$(I))
<17>  1790 ggew=ggew+VAL(ggew$(I))
<33>  1800 NEXT I
<89>  1810 cgew$=DEC$(cgew,"#####")
<85>  1820 ggew$=DEC$(ggew,"#####")
<52>  1830 PRINT TAB(30)CHR$(135);STRING$(5,138);CHR$(13
      9);STRING$(6,138);CHR$(141);STRING$(7,154);CHR$(13
      3);STRING$(7,154);CHR$(133)
<15>  1840 PRINT TAB(30)CHR$(133);"Gesamtpkte.:";CHR$(13
      3);" ";cgew$;" ";CHR$(133);" ";ggew$;" ";CHR$(133)
<62>  1850 PRINT TAB(30)CHR$(131);STRING$(12,138);CHR$(1
      39);STRING$(7,138);CHR$(139);STRING$(7,138);CHR$(1
      37)
<56>  1860 PRINT:PRINT
<66>  1870 IF ggew>cgew THEN PRINT"Ausgezeichnet gespiel
      t, ";name$;", Du hast gewonnen!!"
<30>  1880 IF ggew<cgew THEN PRINT"üben, üben, üben, ";n
      ame$;", vielleicht gewinnst Du morgen!"
<46>  1890 IF ggew=cgew THEN PRINT"Leiiider unentschiede
      n!! Was für eine Schande für Dich!!"
<33>  1900 PRINT FNloc$(1,28);con$;ston$;
<87>  1910 END
< 8>  1920 '
< 3>  1930 '----- Karte vom Stapel nehmen -----
<14>  1940 '
< 9>  1950 IF stapel1$="" THEN GOSUB 4020
<22>  1960 z=INT(RND*(LEN(stapel1$)/4))*4+1
<94>  1970 karte$=MID$(stapel1$,z,4)
<68>  1980 stapel1$=LEFT$(stapel1$,z-1)+MID$(stapel1$,z+
      4)
<24>  1990 RETURN
<84>  2000 '
<98>  2010 '----- offene Karte anzeigen -----
<90>  2020 '
<15>  2030 farbe$=FN1$(obkarte$):wert$=FNr$(obkarte$)
< 5>  2040 IF farbe$="ka" OR farbe$="he" THEN col$=invon
      $ ELSE col$=""
<30>  2050 x=14:y=3
<74>  2060 GOSUB 2260
< 1>  2070 RETURN
< 9>  2080 '
<61>  2090 '----- Karten Gegner anzeigen -----
<86>  2100 '
<40>  2110 FOR I=1 TO gkarten+1
<57>  2120 IF I>20 GOTO 2250
<46>  2130 farbe$=FN1$(gkarte$(I)):wert$=FNr$(gkarte$(I)
      )
< 7>  2140 IF farbe$="ka" OR farbe$="he" THEN col$=invon
      $ ELSE col$=""
<17>  2150 IF I>10 THEN y=22 ELSE y=13
<79>  2160 x=(4+I*8)-8
<64>  2170 IF I>10 THEN x=x-80
<24>  2180 IF gkarte$(I)<>"" GOTO 2230
<87>  2190 FOR J=0 TO 7
<73>  2200 PRINT FNloc$(x,y+J);SPC(8)
<36>  2210 NEXT J
<90>  2220 GOTO 2240
<69>  2230 GOSUB 2260
< 0>  2240 NEXT
<98>  2250 RETURN
< 7>  2260 '
<72>  2270 '----- Karte malen -----
<13>  2280 '
<46>  2290 PRINT FNloc$(x,y);CHR$(150);STRING$(6,154);CH
      R$(156)
<86>  2300 PRINT FNloc$(x,y+1);CHR$(149);col$;wert$;". "
      ;wert$;invoff$;CHR$(149)
<41>  2310 PRINT FNloc$(x,y+2);CHR$(149);col$;SPC(6);inv
      off$;CHR$(149)
<49>  2320 PRINT FNloc$(x,y+3);CHR$(149);col$;". ";farbe
      $;". ";invoff$;CHR$(149)
<93>  2330 PRINT FNloc$(x,y+4);CHR$(149);col$;SPC(6);inv
      off$;CHR$(149)
<91>  2340 PRINT FNloc$(x,y+5);CHR$(149);col$;wert$;". "
      ;wert$;invoff$;CHR$(149)
<92>  2350 PRINT FNloc$(x,y+6);CHR$(147);STRING$(6,154);
      CHR$(153)
< 4>  2360 RETURN
<12>  2370 '
<27>  2380 '----- Karten Gegner neu ordnen -----
<18>  2390 '
<84>  2400 gkarten$=""
<58>  2410 FOR I=1 TO 20
<18>  2420 gkarten$=gkarten$+gkarte$(I)
<63>  2430 gkarte$(I)=""
< 4>  2440 NEXT
< 2>  2450 gkarten=LEN(gkarten$)/4
<81>  2460 FOR I=1 TO gkarten
<51>  2470 gkarte$(I)=MID$(gkarten$,I*4-3,4)
<16>  2480 NEXT
<15>  2490 RETURN
<94>  2500 '
<37>  2510 '----- Karten Computer neu ordnen -----
< 1>  2520 '
<71>  2530 ckarten$=""
<69>  2540 FOR I=1 TO 20
<47>  2550 ckarten$=ckarten$+ckarte$(I)
<50>  2560 ckarte$(I)=""
<15>  2570 NEXT
<16>  2580 ckarten=LEN(ckarten$)/4
<24>  2590 FOR I=1 TO ckarten
<21>  2600 ckarte$(I)=MID$(ckarten$,I*4-3,4)
<98>  2610 NEXT
<97>  2620 RETURN
< 6>  2630 '
<89>  2640 '----- Zug Computer/Level 1 -----
<12>  2650 '
<64>  2660 n=0
<10>  2670 IF stapel1$="" THEN GOSUB 4020
<83>  2680 PRINT FNloc$(77,4);"Mein Zug!"
<29>  2690 IF FNr$(obkarte$)="07" AND s=0 THEN FOR I=1 T
      O 2 ELSE GOTO 2720
<90>  2700 IF stapel1$="" THEN GOSUB 4020
<56>  2710 GOSUB 1920:IF ckarten+I>20 THEN GOSUB 3800:GO
      TO 1480 ELSE ckarte$(ckarten+I)=karte$:NEXT I:s=1:
      GOTO 2820
< 0>  2720 IF level$="2" GOTO 2880
<10>  2730 FOR I=1 TO ckarten
<13>  2740 IF FN1$(ckarte$(I))=FN1$(obkarte$) OR FNr$(ck
      arte$(I))=FNr$(obkarte$) THEN obkarte$=ckarte$(I):
      ckarte$(I)="":s=0:GOTO 2800
<47>  2750 NEXT I
<19>  2760 FOR I=1 TO ckarten
<57>  2770 IF FNr$(ckarte$(I))="Bu" THEN obkarte$=ckarte
      $(I):ckarte$(I)="":s=0:GOTO 2800
<56>  2780 NEXT I
<78>  2790 IF ckarten=20 THEN GOSUB 3800:GOTO 1480 ELSE
      GOSUB 1920:ckarte$(ckarten+1)=karte$:GOTO 2820
<18>  2800 IF FNr$(obkarte$)="08" THEN n=1 ELSE n=0
<78>  2810 stapel2$=stapel2$+obkarte$
<33>  2820 GOSUB 2500
<70>  2830 GOSUB 2000
<86>  2840 PRINT FNloc$(40,3);"Ich habe";ckarten;:IF cka
      rten=1 THEN PRINT"Karte.. " ELSE PRINT"Karten. "
< 2>  2850 IF n=1 AND ckarten>0 THEN n=0:FOR I=1 TO 400:
      NEXT:GOTO 2720 ELSE n=0
<45>  2860 PRINT FNloc$(77,4);SPC(10)
<17>  2870 RETURN
<25>  2880 '
<41>  2890 '----- Zug Computer/Level 2 -----
< 3>  2900 '
< 8>  2910 FOR I=1 TO ckarten
<53>  2920 IF FNr$(ckarte$(I))="07" AND FN1$(ckarte$(I))
      =FN1$(obkarte$) THEN obkarte$=ckarte$(I):ckarte$(I
      )="":s=0:GOTO 2810
<45>  2930 NEXT I
<17>  2940 FOR I=1 TO ckarten
<72>  2950 IF FNr$(ckarte$(I))="08" AND FN1$(ckarte$(I))
      =FN1$(obkarte$) THEN obkarte$=ckarte$(I):ckarte$(I
      )="":s=0:GOTO 2800
<54>  2960 NEXT I
<26>  2970 FOR I=1 TO ckarten
<81>  2980 IF FNr$(ckarte$(I))=FNr$(obkarte$) THEN obkar
      te$=ckarte$(I):ckarte$(I)="":s=0:GOTO 2800
<63>  2990 NEXT I
<87>  3000 FOR I=1 TO ckarten
<98>  3010 IF FNr$(ckarte$(I))<>"Bu" AND FN1$(ckarte$(I)
      )=FN1$(obkarte$) THEN obkarte$=ckarte$(I):ckarte$(
      I)="":s=0:GOTO 2800
<25>  3020 NEXT I
<96>  3030 FOR I=1 TO ckarten
<35>  3040 IF FNr$(ckarte$(I))="Bu" THEN obkarte$=ckarte
      $(I):ckarte$(I)="":s=0:GOTO 2800
```

```
<34> 3050 NEXT I
<26> 3060 GOTO 2790
< 7> 3070 '
<40> 3080 '----- Zug Gegner -----
<13> 3090 '
<79> 3100 IF stapel1$="" THEN GOSUB 4020
<17> 3110 PRINT FNloc$(77,4);"Dein Zug!"
<63> 3120 k=1:y=20:x=7:I=1:PRINT CHR$(7)
<58> 3130 IF x>80 THEN x=x-80
<89> 3140 PRINT FNloc$(x,y);"^"
<70> 3150 taste$=INPUT$(1)
<30> 3160 PRINT FNloc$(x,y);" "
<79> 3170 IF UPPER$(taste$)="K" AND FNr$(obkarte$)="07"
     AND s=0 THEN FOR I=1 TO 2 ELSE GOTO 3200
< 4> 3180 IF stapel1$="" THEN GOSUB 4020
<81> 3190 IF gkarten=20 THEN v=1:GOTO 3470 ELSE GOSUB 1
     920:gkarte$(gkarten+I)=karte$:NEXT I:PRINT FNloc$(
     30,8);SPC(30):s=1:GOTO 3380
<73> 3200 IF UPPER$(taste$)="K" THEN IF gkarten=20 THEN
      v=1:GOTO 3470 ELSE GOSUB 1920:gkarte$(gkarten+1)=
     karte$:PRINT FNloc$(30,8);SPC(30):GOTO 3380
<19> 3210 IF ASC(taste$)=1 AND x>7 THEN x=x-8:k=k-1:GOT
     O 3130
<23> 3220 IF ASC(taste$)=6 AND x<79 THEN IF gkarte$(k+1
     )<>"" THEN x=x+8:k=k+1:GOTO 3130
<39> 3230 IF y=29 GOTO 3260
<17> 3240 IF ASC(taste$)=30 AND gkarte$(k+10)<>"" THEN
     y=y+9:k=k+10:GOTO 3130
<51> 3250 GOTO 3270
<86> 3260 IF ASC(taste$)=31 THEN y=y-9:k=k-10:GOTO 3130
<73> 3270 IF ASC(taste$)<>32 GOTO 3130
<39> 3280 IF FN1$(gkarte$(k))<>FN1$(obkarte$) AND FNr$(
     gkarte$(k))<>FNr$(obkarte$) AND FNr$(gkarte$(k))<>
     "Bu" THEN PRINT CHR$(7);FNloc$(30,8);"Das ist leid
     er nicht erlaubt!":GOTO 3130
<87> 3290 IF FNr$(obkarte$)="07" AND s=0 THEN PRINT FN1
     oc$(30,8);"Erst 2 Karten nehmen!..... ":FOR I=
     1 TO 2 ELSE GOTO 3320
<83> 3300 IF stapel1$="" THEN GOSUB 4020
<36> 3310 IF gkarten+I>20 THEN v=1:GOTO 3470 ELSE GOSUB
      1920:gkarte$(gkarten+I)=karte$:NEXT I:s=1:GOTO 33
     80
< 1> 3320 obkarte$=gkarte$(k)
<40> 3330 IF FNr$(obkarte$)="08" THEN n=1
<87> 3340 IF FNr$(obkarte$)="Bu" THEN b=1
<81> 3350 stapel2$=stapel2$+obkarte$
<52> 3360 gkarte$(k)=""
<92> 3370 s=0
<15> 3380 GOSUB 2370
<45> 3390 IF gkarten>20 THEN GOSUB 3580:GOTO 1480
<20> 3400 PRINT FNloc$(77,4);SPC(10)
<70> 3410 GOSUB 2080
<60> 3420 GOSUB 2000
<58> 3430 PRINT FNloc$(30,8);SPC(30)
<45> 3440 PRINT FNloc$(40,5);"Du hast";gkarten;:IF gkar
     ten=1 THEN PRINT"Karte.. " ELSE PRINT"Karten. "
<94> 3450 IF b=1 AND gkarten>0 THEN b=0:GOSUB 3480 ELSE
      b=0
<20> 3460 IF n=1 AND gkarten>0 THEN n=0:GOTO 3070 ELSE
     n=0
<10> 3470 RETURN
<18> 3480 '
<78> 3490 '----- Karte wünschen -----
<95> 3500 '
< 4> 3510 PRINT FNloc$(30,8);con$;:INPUT"Welche Farbe (
     ka/he/pi/kr)?. ",fa$
<96> 3520 IF fa$<>"ka" AND fa$<>"he" AND fa$<>"pi" AND
     fa$<>"kr" GOTO 3510
<21> 3530 PRINT FNloc$(30,8);SPC(32)
<13> 3540 obkarte$=fa$+". "
<39> 3550 IF fa$="ka" OR fa$="he" THEN col$=invon$ ELSE
      col$=""
<72> 3560 PRINT coff$
<12> 3570 RETURN
<20> 3580 '
<74> 3590 '----- Ende (Gegner verloren) -----
<97> 3600 '
<76> 3610 PRINT
<93> 3620 PRINT FNloc$(34,12);SPC(22)
<69> 3630 PRINT FNloc$(34,13);" ";STRING$(20,"*");" "
<55> 3640 PRINT FNloc$(34,14);" * Du. hast. leider * "
<62> 3650 PRINT FNloc$(34,15);" * V E R L O R E N. * "
<48> 3660 PRINT FNloc$(34,16);" ";STRING$(20,"*");" "
<11> 3670 PRINT FNloc$(34,17);" << Taste. drücken >> "
<51> 3680 PRINT FNloc$(34,18);SPC(22)
<71> 3690 ggew$(M)="... 0"
<25> 3700 cgew=0:ggew=0
<73> 3710 FOR I=1 TO gkarten
<52> 3720 a$=FNr$(gkarte$(I))
<58> 3730 IF a$="07" OR a$="08" OR a$="09" OR a$="10" T
     HEN a=VAL(a$)
<36> 3740 IF a$="Bu" OR a$="Da" OR a$="Kö" THEN a=10
<50> 3750 IF a$="As" THEN a=11
< 1> 3760 cgew=cgew+a
<54> 3770 NEXT I
<85> 3780 cgew$(M)=DEC$(cgew*gkarten,"#####")
<22> 3790 RETURN
< 2> 3800 '
<81> 3810 '----- ENDE (Gegner gewonnen) -----
< 8> 3820 '
<86> 3830 PRINT
<68> 3840 PRINT FNloc$(34,12)SPC(21)
<19> 3850 PRINT FNloc$(34,13);" ";STRING$(21,"*");" "
< 1> 3860 PRINT FNloc$(34,14);" * Gut gespielt!. Du * "
< 9> 3870 PRINT FNloc$(34,15);" *. hast. gewonnen!. * "
<97> 3880 PRINT FNloc$(34,16);" ";STRING$(21,"*");" "
<89> 3890 PRINT FNloc$(34,17);" <<. Taste drücken. >>"
< 2> 3900 PRINT FNloc$(34,18);SPC(21)
<29> 3910 cgew$(M)="... 0"
< 7> 3920 ggew=0:cgew=0
<15> 3930 FOR I=1 TO ckarten
< 6> 3940 a$=FNr$(ckarte$(I))
<68> 3950 IF a$="07" OR a$="08" OR a$="09" OR a$="10" T
     HEN a=VAL(a$)
<46> 3960 IF a$="Bu" OR a$="Da" OR a$="Kö" THEN a=10
<60> 3970 IF a$="As" THEN a=11
<79> 3980 ggew=ggew+a
<64> 3990 NEXT I
<52> 4000 ggew$(M)=DEC$(ggew*ckarten,"#####")
<84> 4010 RETURN
<92> 4020 '
<53> 4030 '----- Stapel tauschen -----
<98> 4040 '
<65> 4050 stapel1$=LEFT$(stapel2$,LEN(stapel2$)-4)
<71> 4060 stapel2$=RIGHT$(stapel2$,4)
< 3> 4070 RETURN
```


Die absolute Sensation auf dem Software Markt!

TAS **298,–**

Relationale Datenbank/Programmiersprache mit Compiler

TAS Leistungsdaten
max. 16 geöffnete Datendateien
max. 16 Schlüsselfelder pro Datei
max. Feldgröße 254 Zeichen
max. 255 Variablen pro Programm
max. Datensatzgröße 10.254 Zeichen
unlimitierte Anzahl an Feldern pro Datei
unlimitierte Dateigröße
max. 65.535 Datensätze

Mit TAS entwickeln Sie Ihre eigenen menügesteuerten Programme von der einfachsten Adreßverwaltung bis zur Fakturierung oder Finanzbuchhaltung.
Mit TAS erwerben Sie eine Programmiersprache mit vielfältigen Möglichkeiten, einen Compiler und das Anwenderprogramm ADRESS. Mit ADRESS können Sie Adressen verwalten, ausgeben, zwei verschiedene Listenformate, sowie Etiketten ausdrucken.
Ab sofort bei uns erhältlich für die Schneider Rechner CPC 6128 und Joyce.

WOLFGANG HENSCHKE DATENTECHNIK

Generalvertretung für die Schweiz:
Domodul AG, Postfach 204, CH-9450 Altstätten

inklusive 200 seitigem deutschen Handbuch und einer mit TAS entwickelten Adressverwaltung der Spitzenklasse.

Händleranfragen erwünscht.

Aidlinger Weg 6, Tel.: 0 70 34/2 96 10
7034 Gärtringen

Demo-Version DM 35,– (incl. 60-seitigem Übungsheft zuzügl. Versandkosten)

Gutschein für kostenloses Info-Material

Hardcopy ohne Ausstieg

Wird heute der Ausstieg aus manchen Bereichen gefordert (z.B. aus der Kernenergie, aus der Wohlstandsgesellschaft u.ä.), so ist der Ausstieg eines Rechners während des Programmablaufs meist unerwünscht und ärgerlich. Ist dies durch eigenes Verschulden geschehen, so nimmt man das meist nicht so tragisch, denn man hat ja wieder einmal dazu gelernt. Ärgerlicher sind da schon Rechnerabstürze durch schlecht konzipierte Programme oder – womit wir uns im folgenden befassen wollen – durch einen Fehler in der Hardware.

Im Vergleich zu den frühen Homecomputern sind Hardware-Fehler heute selten geworden. Die Automation in der Herstellung und die umfangreiche Ausgangsprüfung jedes Rechners sorgen für eine relativ geringe Ausfallquote in dieser Branche. Trotzdem gelangen immer noch Geräte mit Hardware-Fehlern auf den Markt. Häufig sind es dann Kontaktschwierigkeiten, die erst nach der Endkontrolle aufgetreten sind, oder mangelhafte Rechnerfunktionen, die nicht überprüft wurden.

Auch der JOYCE PCW 8256 wird da nicht ausgenommen. Obwohl rundum ein solides und zuverlässiges Gerät, gibt es bei einigen Exemplaren den »Ausstieg bei der Hardcopy«.

Schon einmal, in Heft 10/86, hatte sich CPC-International mit diesem Fehlverhalten befaßt, und damals einen Fehler im LOCOSCRIPT-Textsystem vermutet. So konnte die in diesem Artikel beschriebene Methode den auftretenden Fehler nicht beseitigen, sondern nur umgehen. Darüber hinaus funktionierte der Ausdruck bei einigen Rechnern trotz Einblendung eines Menuefensters immer noch nicht. Grund genug also, sich dieses Problems erneut anzunehmen.

Zunächst noch einmal die Beschreibung des Fehlers, der bei einigen Joyce-Rechnern immer auftritt, bei manchen nur zeitweise und bei anderen wiederum überhaupt nicht. Ein Fehler ganz besonderer Art also.

Ein Vorzug der Joyce-Anlage ist die Fähigkeit, eine Direktkopie des Bildschirminhaltes, also eine sogenannte Hardcopy erstellen zu können. Dies ist in jeder Betriebsart, in jeder Programmsprache und auch zu jedem Zeitpunkt möglich. So kann man sich z.B. im CP/M- und BASIC-Modus den kompletten Bildschirminhalt ausdrucken lassen, in LOGO die mit Hilfe der Schildkröte gezeichneten Grafiken aufs Papier bringen, oder sich beim Arbeiten mit LOCOSCRIPT die aktuelle Diskverwaltung dokumentieren lassen. Gerade die letzte Möglichkeit ist von großem Interesse, da eine andere Auflistung der Textdateien, z.B. unter CP/M, zwar möglich, jedoch sehr unübersichtlich ist.

Gerade hier jedoch versagt so mancher Joyce. Nach Drücken der Tasten <EXTRA> und <PTR> beginnt der Drucker zwar mit dem Ausdruck, verabschiedet sich aber schon meist nach der Hälfte der ersten Zeile. Dann geht nichts mehr, kein CTRL-C, kein EXIT, nicht einmal ein neues Booten ist möglich. Da hilft nur noch der »Kaltstart«; d.h. Gerät aus- und wieder einschalten, anschließend LOCOSCRIPT neu starten. Ein Trauerspiel also für jeden Programmierer – denn zumeist passiert dieses Mißgeschick ja gerade dann, wenn eine Datei nahezu fertig gestellt, aber leider noch nicht gespeichert wurde.

Die genaue Fehlerwirkung haben sicher schon einige Betroffene experimentell ermittelt: Der Rechner steigt immer dann aus, wenn es gilt, eine lange schwarze Voll-Zeile zu drucken. Genauer gesagt, wenn im Grafik-

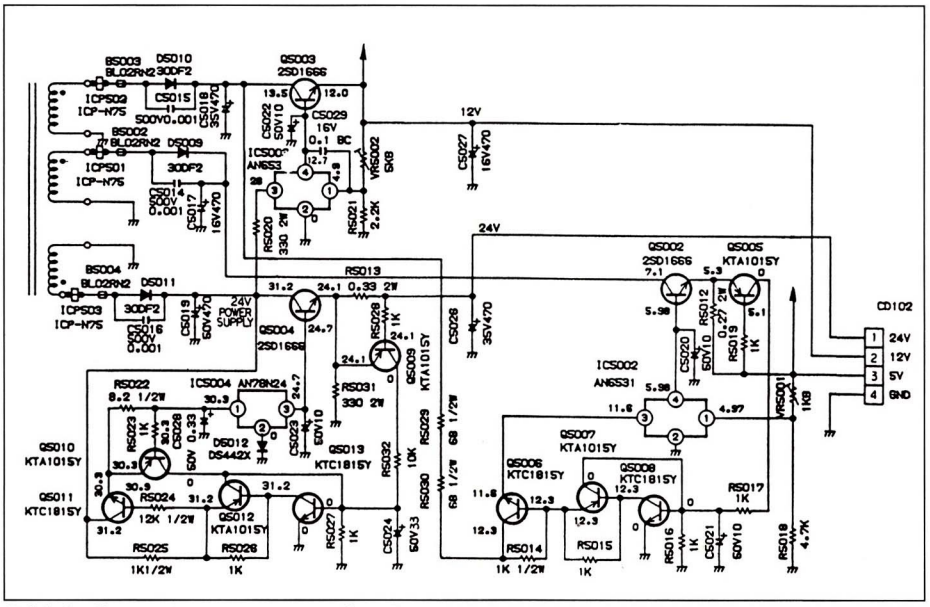


Bild 1: Spannungsversorgung des Joyce

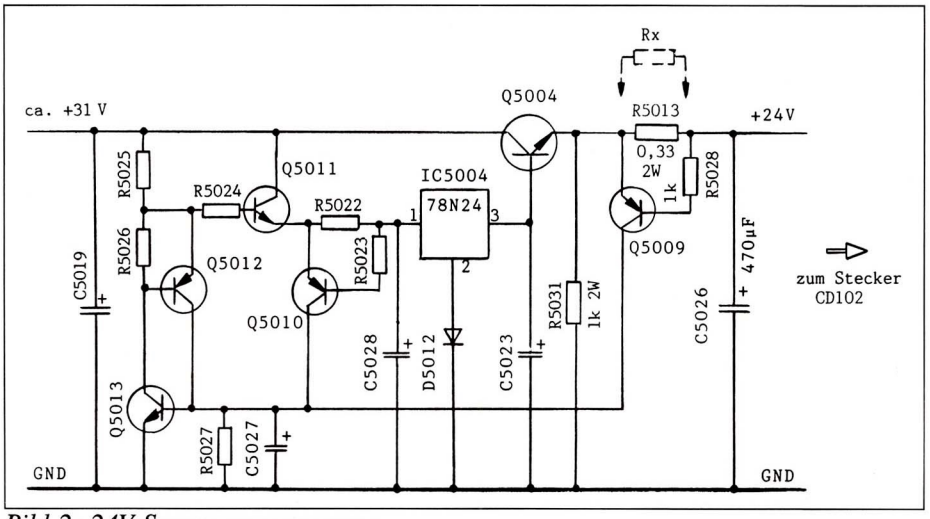


Bild 2: 24V-Spannungsversorgung

Modus alle neun Nadeln des Druckers gleichzeitig und für mehrere Zeichen lang angesteuert werden.

Jetzt erkennt man auch, warum der bereits erwähnte Vorschlag, ein Menuefenster (z.B. das Seitenlayout) aufzurufen, zumindest einen Teilerfolg bringt. Die zunächst »schwarze« erste Zeile des Bildschirms wird bei Einblendung des Menuefensters durch einen Text unterbrochen.

Kein Fehler in der LOCOSCRIPT-Software also, sondern eindeutig ein Hardware-Fehler. Messungen konnten dies bestätigen: Bei gleichzeitiger Ansteuerung aller Druckernadeln fällt bei einigen Rechnern die 24 Volt Spannungsversorgung vollständig aus.

Beschäftigen wir uns also etwas mit der JOYCE-Hardware. Bild 1 zeigt den in Frage kommenden Auschnitt des Stromlaufplans. Für die Elektronik-Interessierten wurde zum besseren Verständnis die 24 Volt-Versorgung noch einmal gesondert herausgezeichnet (Bild 2). Die Spannungsstabilisierung erfolgt durch den integrierten Schaltkreis IC5004; der nachgeschaltete Leistungstransistor Q5004 liefert den erforderlichen Strom. Unser Augenmerk müssen wir jedoch auf den Widerstand R5013 und den Transistor Q5009 richten. Diese bilden eine Strombegrenzung und schalten über Q5013 und Q5011 die 24 Volt-Versorgung ab, wenn der Ausgangsstrom einen Grenzwert von ca. 1,8 Ampere überschreitet.

Bei Ansteuerung aller neun Druckernadeln wird jedoch ein Strom von etwa gleicher Größe benötigt. Und hier liegt die Ursache für den Rechnerausstieg. Bei ungünstigen Toleranzen der Stabilisierungsschaltung kann der Wert der Strombegrenzung geringfügig unter dem benötigten Strom liegen. Die Folge ist bekannt, der Rechner steigt aus.

Die Firma Schneider Computer Division teilte auf Anfrage mit, daß ihr dieser Fehler bekannt sei. Bei ca. 1000 Geräten der ersten Serie sei der Fehler häufiger aufgetreten. Man habe daraufhin eine Änderung der Bauteilauswahl vorgenommen. Die danach ausgelieferten Geräten zeigten diesen Fehler nur noch selten.

Was ist zu tun? Zunächst einmal sollte ein jeder Joyce-Besitzer testen, ob das eigene Gerät diesbezüglich einwandfrei arbeitet. Das kurze BASIC-Programm in Bild 3 ermöglicht eine Prüfung der Hardcopy-Funktion. Die dabei so wichtigen schwarzen Balken werden durch die inverse Darstellung des Leerzeichen auf dem Bildschirm erreicht. Die verwendeten Escape-Sequenzen für die Bildschirmfunktionen (ESC "p" und ESC "q") sind im Anhang 3 des Joyce-Benutzerhandbuches genauer beschrieben.

Bevor der Test durchgeführt wird, sollte jedoch der Rechner gut »warm gelaufen« sein. Die Strombegrenzung ist nämlich temperaturabhängig. Bei höherer Temperatur wird schon bei einem niedrigerem Strom abgeschaltet. (Für die Hardware-Spezialisten ist dies klar: Die Basis-Emitter-Spannung von Q5009 sinkt mit zunehmender Temperatur).

Also: In einem gut geheizten Raum und nach etwa einer halben Stunde »Aufwärmzeit« des Rechners, BASIC-Programm eingeben und sicherheitshalber auf Diskette speichern. Danach kann mit RUN gestartet werden. Auf dem Bildschirm müßte sich dann eine Darstellung wie in Bild 4 zeigen. Nun Papier in den Matrixdrucker einlegen und durch gleichzeitiges Betätigen der Tasten <EXTRA> und <PTR> einen Bildschirm-Ausdruck starten.

Sollte ihr Drucker den Ausdruck bis zum Ende korrekt durchführen, so können Sie aufatmen und sich dem nächsten Artikel der PC-International zuwenden.

Ist der Rechner während des Ausdrucks ausgestiegen, so gehören Sie zu den 1000 Pechvögeln und müssen sich nun entscheiden: Kann ich ohne funktionstüchtige Hardcopy leben, oder lege ich Wert auf diese Funktion. Wird Letzteres bejaht, so steht man vor einer weiteren Entscheidung: Soll die Reparatur in einer Schneider-Service-Werkstatt durchgeführt werden (innerhalb der Garantiezeit sicher

```
LISTING >HARDCOP <, REMARK = >REM<.

<36> 10 'Hardcopy-Testprogramm für JOYCE
< 8> 20 ESC$=CHR$(27)
<62> 30 CLS$=ESC$+"H"+ESC$+"E"
<85> 40 INVERSEIN$=ESC$+"p"
<61> 50 INVERSAUS$=ESC$+"q"
<49> 60 PRINT CLS$
<46> 70 PRINT TAB(26);"H. A. R. D. C. O. P. Y. -. T. E.
     S. T "
<80> 80 FOR I=1 TO 90:PRINT"-";:NEXT I
< 5> 90 PRINT:PRINT
<10> 100 FOR I=5 TO 85 STEP 5
<36> 110 PRINT USING"##";I;
<35> 120 PRINT ": ";
<94> 130 PRINT INVERSEIN$;
< 6> 140 FOR J=1 TO I
<12> 150 PRINT " ";
<84> 160 NEXT J
<96> 170 PRINT INVERSAUS$
<78> 180 NEXT I
<50> 190 PRINT
<71> 210 PRINT "4 Zeilen mit je 90 Zeichen :"
<93> 220 PRINT INVERSEIN$;
<40> 230 FOR I=1 TO 360
<11> 240 PRINT " ";
<73> 250 NEXT I
<95> 260 PRINT INVERSAUS$
<47> 270 PRINT
<36> 280 PRINT "Jetzt bitte mit <EXTRA> und <PTR> eine
     Hardcopy erstellen ";
< 8> 290 INPUT "",A
<25> 300 END
```

Bild 3: Basic-Testprogramm

zu empfehlen), oder kann ich die erforderliche Schaltungsänderung selber vornehmen.
Der Eingriff selber ist relativ problemlos. Eine geringfügige Erhöhung der Strombegrenzung ist unser Ziel. Dies läßt sich am einfachsten durch Verringerung des Widerstand R5013 erreichen. Legen wir einmal eine Erhöhung der Strombegrenzung von 20% zugrunde, so erreicht man dies durch Parallelschalten eines 1,5 Ohm Widerstandes (so wie in Bild 2 bereits als Rx eingezeichnet), oder durch Austausch des Widerstandes R5013 (0,33 Ohm) gegen einen 0,27 Ohm Widerstand.
Bevor Sie jedoch mit Schraubendreher und Lötkolben ans Werk gehen, hier noch einmal der Hinweis, daß durch einen Eingriff in den Rechner eine möglicherweise noch bestehende Garantie erlischt.

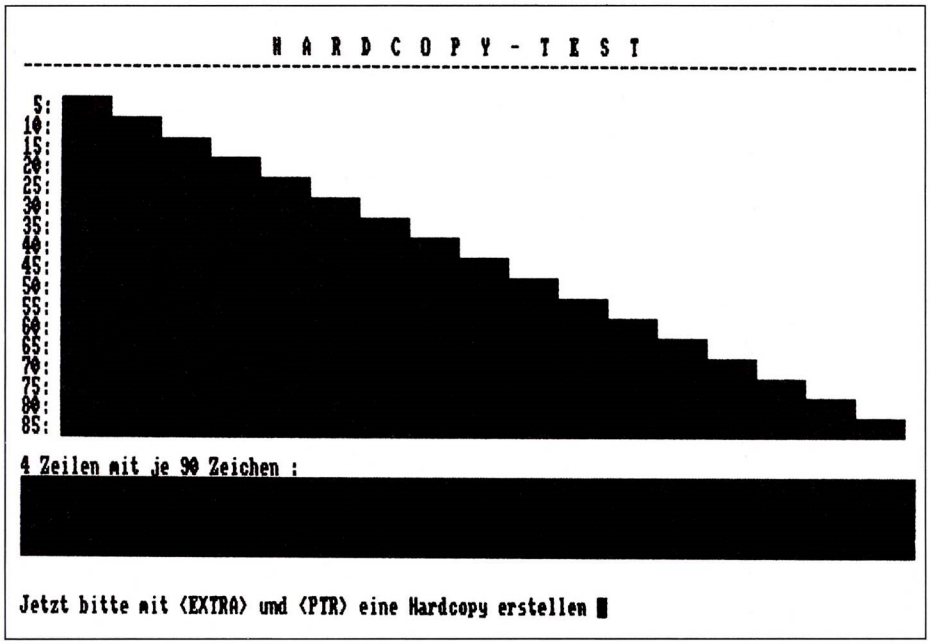


Bild 4: So muß die Hardcopy aussehen.

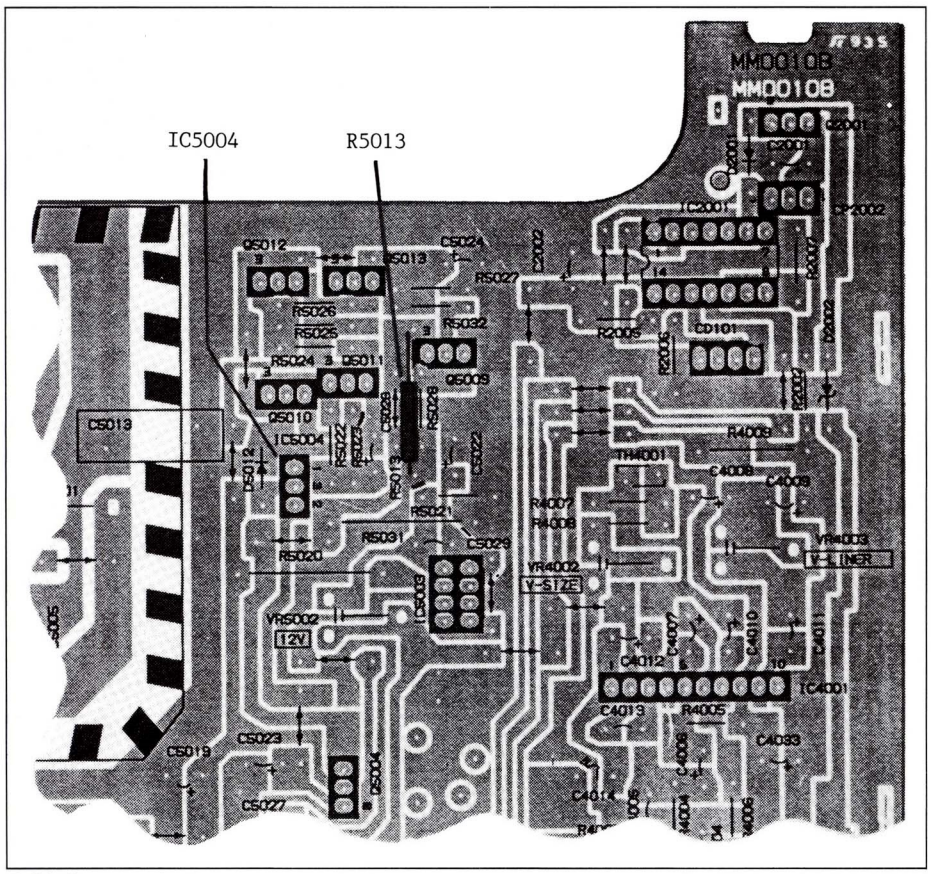


Bild 5: Ausschnitt der Hauptplatine

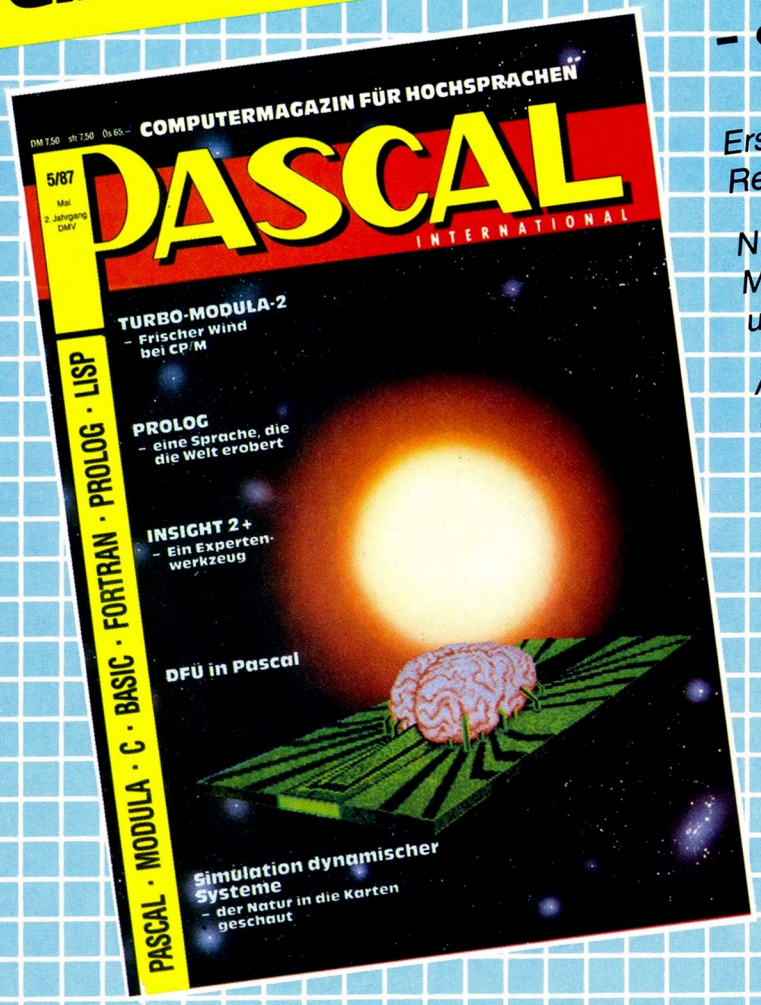
Im Gehäuse des Monitors ist nahezu der gesamte Rechner untergebracht. Es läßt sich leicht durch Lösen der sechs durch Pfeile gekennzeichneten Schrauben öffnen (Achtung: Vorher Netzstecker ziehen!). Auf der unten eingeschobenen Basis-Platine befindet sich die Speisung und Ansteuerung des Monitors, und die gesuchte Spannungsversorgung. Durch Anheben eines Arretierungshebels auf der rechten Seite der Platinenhalterung läßt sich die Basisplatine leicht nach vorne ziehen und ist so besser zugänglich. Der gesuchte Widerstand R5013 befindet sich im linken vorderen Teil der Platine. Gut zu erkennen, da es sich um einen größeren 2 Watt-Widerstand handelt, der den Aufdruck 0,33 trägt (siehe auch Layoutplan Bild 5). Da er auf Abstand gesetzt ist, läßt er sich nun leicht mit einem Widerstand von 1,5 Ohm = 0,5 Watt überbrücken (parallelschalten). Die andere Möglichkeit wäre den R5013 aus der Platine auszulöten und durch einen Widerstand von 0,27 Ohm = 2 Watt zu ersetzen. Dabei ist jedoch die Herausnahme der Platine nach Lösen aller Steckverbindungen zweckmäßig.

Nach Zusammenbau und Anschluß von Tastatur und Drucker kann unter BASIC der Hardcopy-Test erneut gestartet werden. In 90% der Fälle wird dieser nun erfolgreich verlaufen und der Drucker ohne Mucken das Papier mit schwarzen Streifen füllen. Sollte trotz richtig durchgeführter Änderung und korrekter Bedienung des Rechners dieser immer noch aussteigen, so wird empfohlen, auch noch den Stabilisierungsbaustein IC5004 auszuwechseln. Möglicherweise liefert dieser eine etwas zu hohe Ausgangsspannung, und damit einen zu hohen Druckerstrom.

Alles in allem dürfte sich der kleine Eingriff in die JOYCE-Hardware lohnen. Die Hardcopy-Funktion des Joyce ist schon eine feine Sache, besonders dann, wenn man sie ohne Angst vor einem Rechnerabsturz benutzen kann.

(W.Otternberg)

ES GIBT VIELE COMPUTERZEITSCHRIFTEN



– das Konzept von PASCAL ist einmalig.

Erstklassige Programme für alle Rechner, die eine CPU haben!

Nicht nur Pascal – fit auch in Modula-2, C, FORTRAN, PROLOG und BASIC.

Aktuelle News aus allen Bereichen des Computeralltages, Report und Reviews neuester Software.

Buchbesprechungen, Feuilleton und Spiele.

Eine »heiße Hotline« auf vier Kanälen als Leserforum und noch einiges mehr!

Mit PASCAL – keine Angst mehr vor Computersprachen – für Experten, Hobbyisten und Lernende.

PASCAL-International kostet im Abonnement:

Im Inland und Westberlin:	für 1/2 Jahr	40,– DM
	für 1 Jahr	80,– DM
Im europäischen Ausland:	für 1/2 Jahr	50,– DM
	für 1 Jahr	100,– DM
Im außereurop. Ausland:	für 1/2 Jahr	70,– DM
	für 1 Jahr	140,– DM

In diesen Preisen sind Kosten für Porto und Verpackung, sowie die MwSt. enthalten.

DMV GmbH · PASCAL-International
Postfach 250 · 3440 Eschwege · Tel. (0 56 51) 87 02

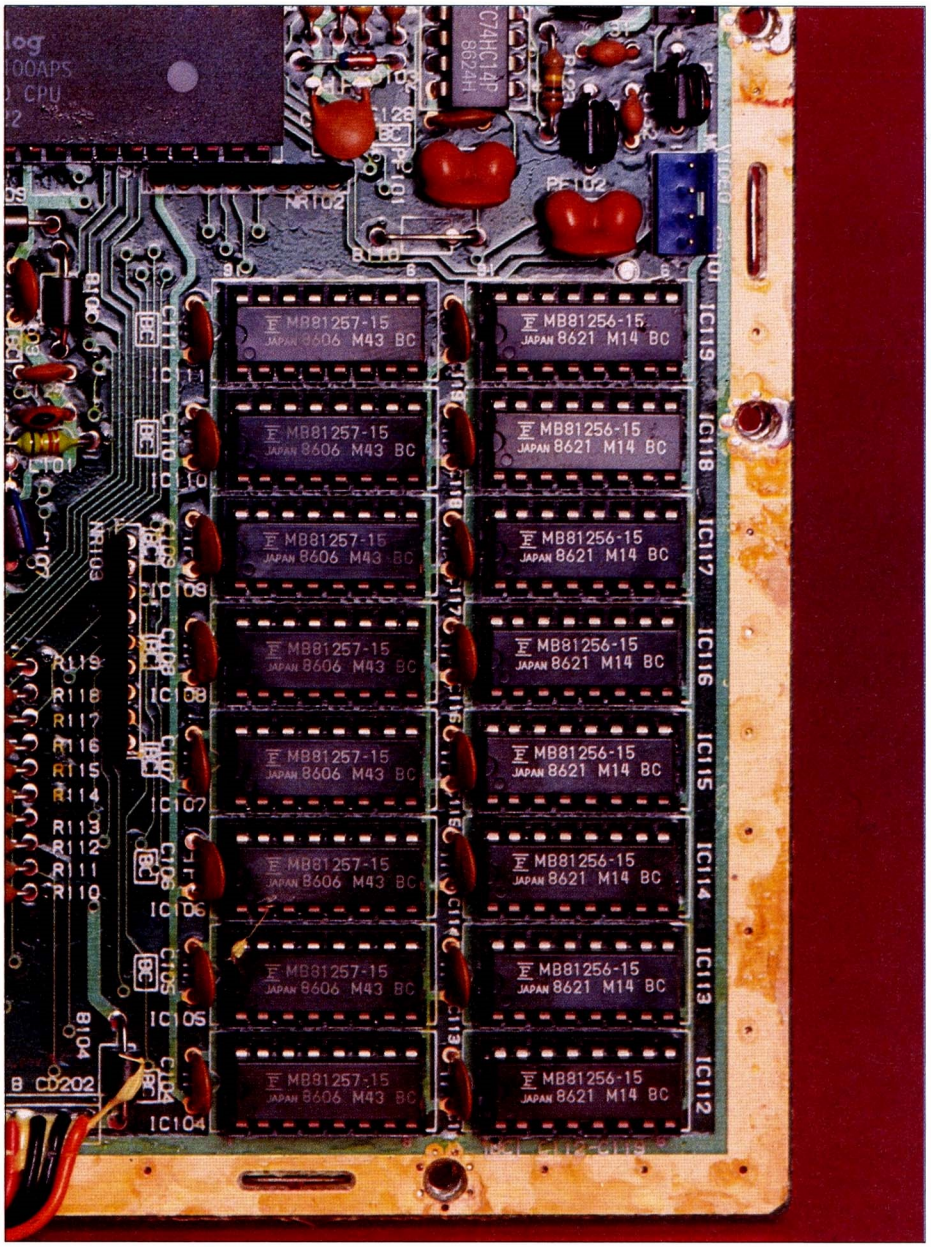
Aufrüstung auf 512 KB RAM für JOYCE

Es gibt mittlerweile eine Reihe von Firmen und Versandhändlern, die ziemlich preisgünstig – bis um die 100 Mark – Chips des Typs TMS 425715NL vertreiben. Mit acht solchen Chips kann man den Schneider PCW 8256 JOYCE zum Schneider PCW 8512 machen. Die RAM-Floppy hat dann unter LocoScript 356 KB und unter CP/M 368 KB frei verwaltbare RAM-Kapazität. Es fehlt nur noch das zweite Laufwerk FD-2.

Nicht schlecht, wird mancher sagen und den Einbau vornehmen wollen. Zum Selbsteinbau hier ein Bericht aus der Praxis:

Als erstes ziehen Sie den Netzstecker und lösen die Verbindungen zum Drucker und zur Tastatur. Legen Sie JOYCE mit der Bildschirm-Seite auf eine Schaumstoffmatte. Sie sorgt dafür, daß JOYCE während des Arbeitens nicht wackelt. Um JOYCE zu öffnen, brauchen Sie einen Kreuz-Schraubendreher um die 6mm Durchmesser mit mindestens 15 cm Klingenlänge. Es sind sechs Gehäuseschrauben zu entfernen (nicht nur vier wie in vielen Einbauanleitungen zu lesen). Die Schrauben sind durch eingeprägte Pfeile auf dem Kunststoff-Chassis leicht zu finden. Die Kunststoff-Abdeckung kann dann abgehoben und beiseite gelegt werden.

Wenn Sie den JOYCE-Standfuß zu sich gerichtet haben, sehen Sie links oben das Floppy-Disc-Laufwerk und rechts daneben senkrecht eingeschoben eine in Metall eingebaute Platine. Das ist die Hauptplatine des JOYCE, an die Sie heran müssen. Dazu ziehen Sie zunächst den blauen und den schwarzen Kabelschuh von der linken Seite der Platine ab. Auf der linken Seite ist das Erdungskabel festgeschraubt. Auch dieses lösen Sie mit dem Kreuzschlitz-Schraubendreher, weil Sie sonst später die Platine nicht bewegen können. Die Metallbox mit der Platine kann jetzt vorsichtig aus



der Einschubleiste herausgezogen werden. Das Steckerleisten-Kabel und die Verbindungen zur Floppy werden nicht gelöst.
Die eigentliche Platine des JOYCE ist mit einem Metallgehäuse abgeschirmt. Es sind acht Schrauben, die mit dem Kreuzschlitz-Schraubendreher gelöst werden müssen, um an das Herz des JOYCE heranzukommen. Achten Sie beim Lösen dieser acht Schrauben darauf, daß die Platine ja noch mit weiteren Kabeln zu anderen Geräteteilen verbunden ist. Die Kabel sollen nicht unnötig geknickt werden und Lötverbindungen nicht verbogen werden. Ein bißchen jonglieren muß man schon, denn die Schrauben sitzen teilweise sehr fest. Wenn Sie alle acht Schrauben gelöst und gut aufbewahrt haben, heben Sie den Metall-Deckel vorsichtig über Kopf über die Kabel an der Steckerleiste ab. Sie haben nun die Hauptplatine des JOYCE vor sich.

Wenn Sie die Platine so betrachten, daß Sie die Aufschrift der Chips lesen können, finden Sie rechts 16 Steckplätze. Acht davon sind schon besetzt und die freien acht werden Sie mit den neu gekauften TMS 425615 NL beschicken. Auf der Oberseite der Chips finden Sie eine Einbuchtung. Die Chips müssen mit dieser Einbuchtung nach links in die Steckplätze gesteckt werden. Sie brauchen sich nur an den bereits vorhanden acht Chips zu orientieren. Deren Einbuchtung ist nach links zu den kakaobraunen Anschlüssen gerichtet.

Nun kommt der schwierigste Teil. Die Beinchen der Chips sind in der Regel so verbogen, daß sie nicht in die Steckplätze passen.

Bei der folgenden Biege-Aktion sollten Sie die blanken Metall-Beinchen nicht mit den Fingern berühren. Fett und andere Hauptartikel können die elektrische Leitfähigkeit verändern und statische Aufladungen können unvorhersehbare Wirkungen auf Ihren Computer haben. Ideal wäre es, wenn Sie feine Handschuhe aus weichem Stoff tragen würden. Saubere Gummihandschuhe tun es aber auch.

Legen Sie das Bauteil wie einen Käfer auf den Rücken. Spreizen Sie die Beinchen mit einer Pinzette oder einem Hölzchen nach außen. Dann biegen sie durch möglichst flächigen, massierenden Druck mit den – behandschuhten! – Fingern die Beinchen möglichst lotrecht zur Grundfläche an. Ein Blick über die Flucht zeigt ihnen, ob die Beinchen in einer Reihe stehen. Zu dieser Biege-Aktion gehört Geduld. Sie müssen mit fünf Minuten pro Chip rechnen. Nicht mit Gewalt biegen. Die Beinchen sind ziemlich steif. Wenn Sie eines abbrechen, ist ihr gekaufter Bausatz unbrauchbar.

Die einzelnen Chips stecken Sie nacheinander in die vorgesehenen Steckplätze. Dazu sollten Sie mit der linken Hand die Platine unterstützen. Denn der Druck, den Sie mit dem Daumen der rechten Hand ausüben müssen, um den Chip in den Steckplatz zu drücken, ist nicht unerheblich. Allerdings sollten Sie auch keine Gewalt anwenden und sich immer vergewissern, daß alle Beinchen genau auf einem der Einschub-Schlitze stehen. Nicht erschrecken: Wenn Sie den Chip einstecken knirscht es ein bißchen. Das hört sich gefährlicher an, als es ist.

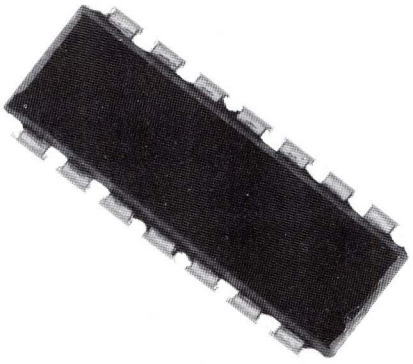
Die Operation am JOYCE ist bald beendet. Nachdem Sie alle acht Chips in die Steckplätze hineinpraktiziert haben, schauen Sie nach rechts auf die Platine. Dort finden Sie ein blau ummanteltes tonnenförmiges Bauteil von nicht einmal einem Zentimeter Länge. Es handelt sich um einen Kondensator. Um sicher zu gehen, daß sie das richtige Bauteil haben, lesen Sie, was darauf steht. Sie werden eine Zahl mit der Einheit μF (Mikro-Farad, d.i. die Einheit für die elektrische Kapazität eines Kondensators) finden. Bei diesem Kondensator befindet sich eine Lötdrahtbrücke, die mit einem »B« gekennzeichnet ist. Diese trennen Sie mit einer Zange oder einer kräftigen Nagelschere durch. Dabei immer die Platine unterstützen, da diese an den Kabeln mehr oder weniger lose herumhängt. Die freien Drahthälften biegen Sie dann vorsichtig auseinander.

Sie können nun in umgekehrter Reihenfolge wie oben beschrieben JOYCE wieder zusammenbauen.

Wenn Sie JOYCE gemäß den Anweisungen im Handbuch wieder betriebsfertig gemacht haben, meldet er ihnen unter LocoScript im Laufwerk M nunmehr 356 KB. Unter CP/M 3.0 sind es 368 KB. Eine Besonderheit ist durch den Umbau entstanden:

Mit SHIFT – EXTRA – EXIT können Sie JOYCE nicht mehr komplett zurücksetzen. Tun Sie dies, so verbleiben im Laufwerk M in der Regel nur etwa 100 KB frei, weil das jeweilige vorherige Betriebssystem nicht gelöscht wurde. Wenn Sie also von LocoScript auf CP/M übergehen und umgekehrt, sollten Sie zwischendurch JOYCE abschalten. Das tut ihnen bei aller Freude über den neuen 512 K-Computer sicherlich sowie ganz gut.

(G. Ott)



Joyce-Professionals

News Desk International
Desk Top Publishing für den Joyce
mit deutschem Handbuch DM 159,00

Wordstar 3.0	189,00
Multiplan 1.06	189,00
dBase II 2.41	189,00
CBasic Compiler	169,00
DR Draw	189,00
DR Graph	189,00
MS Basic	189,00

INFOCOM-Spiele für den Joyce

Enchanter	*standart*	69,90
Sorcerer	*advanced*	69,90
Witness	*standart*	69,90
Suspect	*advanced*	69,90
Deadline	*expert*	69,90
Seastalker	*junior*	69,90
Infidel	*standart*	69,90
Planetfall	*standart*	69,90
Starcross	*expert*	69,90
Suspended	*expert*	69,90

Waldeck-Software
Tulpenstraße 30
2870 Delmenhorst
Tel.: 0 42 21 / 1 64 64

Achtung



Disketten

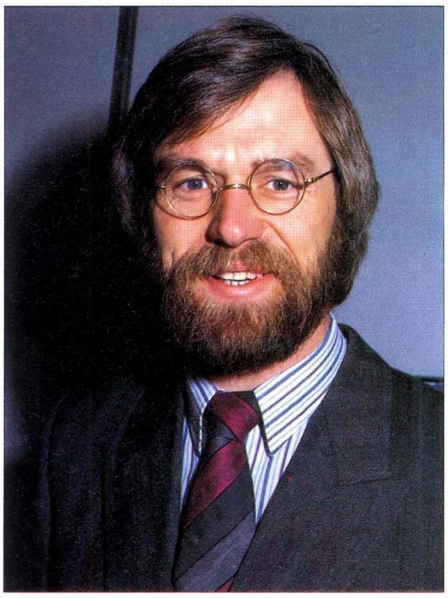
Maxell 3" -Disketten CF2
10er Pack
DM 69,–

Disketten-boxen

Diskettenbox SS 50

für 50 3" - oder
3 1/2"- Disketten,
abschließbar/tragbar,
Rauchglas
DM 15,90

Ferdi's Computer Software
EDV-Service
Inh. Ferdinand Göddeker
Höftestraße 32
4400 Münster
Telefon 02 51 / 61 98 81



»Nachrichten aus Türkheim«

Interview von der CeBIT

Anläßlich der CeBIT'87 hatten wir Gelegenheit, mit Hanns-Werner Laaks, seines Zeichens Leiter der Vertriebsunterstützung, ein Gespräch zum Thema Joyce zu führen. Aufsehenerregende Neuigkeiten sind weiter nicht zu vermelden, doch versicherte uns Herr Laaks, daß Schneider den Joyce auch in Zukunft weiter pflegen wird. Doch lesen Sie selbst.

DMV:
Erst einmal herzlichen Dank, Herr Laaks, daß Sie sich kurzfristig bereit erklärt haben, dies Gespräch im Messe-Stress mit uns zu führen. Zuerst die Standardfrage: In welcher Stückzahl befindet sich der Joyce derzeit auf dem deut schen Markt?

Laaks:
Insgesamt sind ca. 70.000 Exemplare des Joyce von Schneider auf dem deutschen Markt verkauft worden.

DMV:
Wie ist die Aufteilung dieser Stückzahl zwischen Joyce und Joyce Plus?

Laaks:
Der Joyce Plus hat einen Anteil von ca. 35% an der Gesamtverkaufszahl.

DMV: Wann und bei welcher Stückzahl ist die Aufnahmekapazität des Marktes für den Joyce erschöpft?

Laaks:
Das wird wohl erst dann der Fall sein, wenn es auch nicht mehr notwendig ist, sich eine Schreibmaschine anzuschaffen.

DMV:
Die Antwort ist natürlich ungeheuer elegant, doch würden wir es gern ein wenig genauer wissen. Sie sehen also zur Zeit keine Grenzen für den Absatz und es gibt bei Ihnen keine konkreten Überlegungen, ob und wie lange sich der Joyce noch pflegen lassen wird, bzw. wann der Zeitpunkt gekommen ist, daß er vom technischen Fortschritt überholt und deshalb nicht mehr absetzbar sein wird?

Laaks:
Sicher wird das System – wie alle anderen auch – irgendwann einmal technisch überholt sein. Aber das Konzept des Joyce als intelligente Schreibmaschine, die außerdem noch die Möglichkeiten eines leistungsfähigen Computers bietet, dürfte sich auch weiterhin gegenüber anderen Systemen behaupten. Wir sind sicher, daß wir die heutige Position des Joyce auf dem Markt noch beträchtlich ausbauen können. Wir werden an Konzept und Technik dieses Systems auch noch weiter arbeiten und der Joyce wird sicher noch über einen längeren Zeitraum von uns vertrieben werden.

DMV:
Trifft es zu, daß Schneider den Joyce nur noch in der kleinen Version vertreiben will und deshalb in den Anzeigen auch nur noch die kleine Version erscheint?

Laaks:
Nein, das trifft nicht zu. Wir werden auch in Zukunft sowohl den »kleinen« als auch den Joyce Plus vertreiben. Man muß sich natürlich in der Werbung auf ein einheitliches Motiv einigen und unter dieser Prämisse fällt es schwer, in einer Anzeige die Vorzüge beider Versionen differenziert darzustellen.

DMV:
Kann es sein, daß Sie den Joyce lieber als reines Textsystem verkaufen möchten und seine Leistungsfähigkeit als Personal Computer am liebsten ganz verschweigen würden, um dem PC keine Konkurrenz zu machen?»

Laaks:
Joyce wurde von Grund auf als komplettes Textsystem – mit Software und Drucker – konzipiert. Mit diesem System zielen wir auf eine bestimmte Benutzergruppe und haben keinerlei Befürchtungen, daß es hier zu Konflikten oder Konkurrenzsituationen zum Schneider PC kommen wird.

DMV:
Hat Schneider – dies im Hinblick auf zukünftige Entwicklungen gefragt – direkten Einfluß auf das Produktdesign der Geräte, die Amstrad herstellen läßt?

Laaks:
Dies läuft in Absprache mit England ab und bei der Entwicklung haben wir ein Mitspracherecht.

DMV:
Sie sind also nicht per Vertrag gebunden, blind das zu kaufen, was Amstrad Ihnen anbietet?

Laaks:
Nein, das sieht man allein schon daran, daß Amstrad die Rechte an Sinclair gekauft hat und die Sinclair Geräte auch in England vertreibt, wir jedoch die Vertriebsrechte für Sinclair Produkte nicht übernommen haben.

DMV:
Damit hätte man sich sicher auch Konkurrenz im eigenen Haus geschaffen.

Laaks:
Dies sei dahingestellt. Wichtig ist jedenfalls, daß wir die Entscheidungen für den deutschen Markt selbst treffen.

DMV:
Jetzt wird es spannend. Sie wissen, daß die Gerüchte wuchern. Sie ranken sich um einen neuen Joyce zum Jahresende, einen intelligenten Drucker für den Joyce, vielleicht sogar einen Laserdrucker. Was ist an diesen Gerüchten dran? Werden Sie die Politik der Überraschungseier fortsetzen, oder können Sie schon etwas über die Zukunft des Joyce erzählen?

Laaks:
Eine Politik der Überraschungseier war nie unsere Linie. Wir möchten nur eine seriöse Ankündigungspolitik betreiben, das heißt, wir wollen nicht über ungelegte Eier gackern, die dann vielleicht hinterher doch nicht ausgebrütet werden.

DMV:
Sie lassen Sich also heute noch keine Informationen entlocken?

Laaks:
Wir arbeiten auch weiter am Joyce. Der Joyce ist keine Eintagsfliege. Der Joyce wird weiter Bestandteil unseres Angebotes sein und zum gegebenen Zeitpunkt werden wir dann auch beim Joyce mit Neuheiten aufwarten.»

DMV:
Wenn Sie schon nicht sagen können oder wollen, was es an Neuheiten geben wird, so könnten Sie zumindest ausgrenzen und sagen, was es nicht geben wird.

Laaks:
Ich kann also eines schon sagen, ein Laserdrucker für den Joyce ist nicht geplant.

DMV:
Auf der Betriebssystemebene wird es keine Änderungen geben, damit das Gerät vielleicht schneller wird?

Laaks:
Ich kann dazu wirklich nichts sagen, denn ich habe auch in dieser Richtung noch nichts gesehen. Es ist also wirklich noch nicht 'raus, was da passiert. Es wird an am Produkt gearbeitet, um es noch besser an den Markt anzupassen.

DMV:
Okay, wann die großen Neuentwicklungen kommen und wie sie aussehen, ist also noch nicht zu erfahren. Wenden wir uns anderen Dingen zu. Wann kommt ein deutsches Locospell auf den Markt?

Laaks:
Wir suchen seit geraumer Zeit ein deutsches Wörterbuch, das schon in codierter Form vorliegt und in LocoScript eingearbeitet werden kann.

DMV:
Die Schwierigkeiten liegen also nicht darin, das Programm an die relativ komplizierte deutsche Syntax anzupassen? Die Schwierigkeiten liegen darin, das eigentliche Wörterbuch zusammenzustellen?

Laaks:
Ich weiß zwar nicht wie Locomotive das Problem der deutschen Syntax lösen wird, doch bin ich davon überzeugt, daß es hier keine größeren Schwierigkeiten geben wird. Ob es jedoch gelingt, ein deutsches Locospell noch in 1987 auf den Markt zu bringen, vermag ich zum gegenwärtigen Zeitpunkt noch nicht zu sagen.

DMV:
Bringt Schneider in absehbarer Zeit noch zusätzliche Peripheriegeräte für den Joyce auf den Markt, z.B. eine Harddisk oder einen Streamer?

Laaks:
Von unserer Seite ist in dieser Hinsicht nichts geplant. Jedoch versicherte mir die Firma Vortex, daß die Entwicklung ihrer Harddisk abgeschlossen sei und sie damit in Kürze auf den Markt kommt. Von der Entwicklung eines Streamers ist mir derzeit nichts bekannt und ich möchte bezweifeln, daß ein derartiges Gerät für Joyce Besitzer interessant ist.

DMV:
Wird es von Schneider noch neue Anwendersoftware für den Joyce geben?

Laaks:
Wichtig für den Anwender ist, daß gute Programme vorhanden sind. Aus diesem Grund liegt derzeit unser Akzent eher darauf, den Anwendern Informationen über ihre Probleme zu geben, als eigene Software anzubieten. Unsere Aktivitäten im Softwarebereich konzentrieren sich darauf, Händler und Endverbraucher darüber zu informieren, was sich an Software auf dem Markt befindet. Dazu dient unsere von Anfang an erstellte und permanent aktualisierte Softwareliste, die wir unseren Händlern und – auf Anfrage – auch Endverbrauchern zusenden.

DMV:
Herr Laaks, wir bedanken uns für dieses Gespräch. (W. Huske)

Joyce-Angebote

! 5,25" Zweitlaufwerk ! ! Qualitätslaufwerk incl. Gehäuse, ! (Metall oder Plexiglas), ! Adapter und Einbauanleitung ! 1 MB Kapazität, voll FD-2 kompatibel !	569,–
! vortex 20 MB Festplatte WD-2000 für Joyce	1998,– !
! Joyce-Mousepack	228,– !
! Speichererweiterung auf 512 K	79,– !
! FD-2 3" Zweitlaufwerk, 1MB Kapazität	589,– !
Bildschirmfilter	59,–
DR Draw prof. Zeichenprogramm	179,–
DR Graph	179,–
dBase II / Wordstar / Multiplan	je 179,–
Turbo Pascal	215,–
M&T Finanzbuchhaltung	189,–
Vokabeltrainer für Englisch und Latein	59,–
Verbentrainer ENGLISCH I	59,–
Window-Adress	98,–
Carat-Lager	288,–
Carat-Kasse Plus	188,–
! IDA Faktura	298,– !
! IDA Fibu	298,– !

Händleranfragen erwünscht!
Alle Preise zzgl. Versandkostenanteil!
Der Versand erfolgt per Nachnahme oder Vorauskasse
(2 % Skonto).

TG-Soft
Thomas Gmach
Offersdorf 5
8491 Rimbach
Tel. (0 99 41) 37 65

ANWENDUNGSSOFTWARE FÜR JOYCE PCW 8256/8512

Schnelle C-Pprogramme auf 3"Disketten

ADRESS-CONTROL 79,– DM
600 Adressen je Diskette nach 9 Feldern sortierbar, Format für Etikettendruck variabel, Serienbriefe mit LocoScript-Textverarbeitung auch einzeln, nach PLZ-Gebiet oder wahlfreiem Gruppencode.

FAKTURA-STANDARD 94,– DM
Zur komfortablen und schnellen Erstellung von Rechnung/Gutschrift/Lieferschein/Auftragsbest/Angebot.
Datei für 600 Kunden zum Einblenden der Rechnungsanschriften und Rabattsätze, kompatibel zu ADRESSCONTROL (Serienbriefe ohne doppeltes Erfassen).
Rabatt/MwSt./Währung frei wählbar und gespeichert.
Kopf- und Fußtexte, Werbetexte, beliebige Texte zwischen den Rechnungszeilen, Druckwiederholung.

FAKTURA-CONTROL II 169,– DM
Wie FAKTURA STANDARD, jedoch zusätzlich mit Artikeldatei und integrierter Offene-Posten-Buchhaltung mit Mahnprog., 600 Kunden, 700 Artikel, 1000 Rechnungen je Diskette. Arbeitet ohne Umschalten mit u. ohne Artikeldatei, direktes Einblenden der Artikel in die Faktura, Lagerbestandsfortschreibung beim Fakturieren, Umsatz- u. Mindestbestandskontrolle.

BAUFINANZ-CONTROL II 279,– DM
Berechnet Baufinanzierung, Detaillierte Ausdrucke, Neues Steuerrecht (Update-Service), Ein- u. Mehrfamilienhäuser, Hypotheken, Bausparvertr. Zwischenfinanz. und Lebensvers. beliebig kombinierbar.

IMMOBILIENVERMITTLUNGS-DATEI 149,– DM
300 Objekte mit Kurzbeschreibung und 500 Interessenten je Datendiskette, Selektierte Angebotsdrucke mit Nachweis für den Makler.

SPORT-TABELLEN 65,– DM
Verwaltet 26 Mannschaften und 180 Spieler für Statistiken, Spieltage gespeichert und änderbar, Nachträgliche Tabellenberechnung.

Diese Programme wurden speziell auf JOYCE PCW entwickelt und sind ohne umständliche Installation sofort einsetzbar, Arbeitskopien ohne Mehrkosten.

Versand per NN zuzüglich DM 5,– (Ausland 10,–) Porto/Verp.
oder im Fachhandel

Hashagen - EDV
Tel. (0 60 73) 6 19 93
Eckstr. 11, 6113 Babenhausen 3

JOYceSTICK selbstgemacht

So mancher Joyce-Besitzer wird schon bei cursorgesteuerten Programmen angesichts verkrampfter Finger über den Cursortasten und schmerzend vorgebeugtem Rücken einen Joystickanschluß vermißt haben, so etwa bei DR Draw, Hyroglyphos total (CPC 12/86) oder CBASIC-Programmen mit Graphic Input. Das soll sich ändern: Der Joyce wird nachgerüstet, und zwar fast zum Nulltarif! Eine evtl. bestehende Geräte-Garantie erlischt allerdings!

Alles, was der geneigte Leser benötigt, sind:

- 3 Dioden (z.B. 1N4148);
- evtl. eine 9polige MIN-D-Einbaustiftleiste;
- einen aufschraubbaren Joystick oder ein ähnliches Eingabegerät mit vier Richtungs- und zwei Zusatztasten und dem üblichen sechsadrigen Kabel mit MIN-D-Buchsenstecker;
- einige Dezimeter Litze oder Flachbandkabel.

Sechs Taster und sechs Adern – wo bleibt da die gemeinsame Masseleitung?

Der Trick besteht darin, daß wir mit den Joystickanschlüssen einfach die Tastaturmatrix »anzapfen«.

Zur Erinnerung: Die einzelnen Tasten eines Tastenfeldes werden nacheinander abgefragt. Dies erledigt beim Joyce ein Einchipmikrocomputer 8048, eine maskenprogrammierte 8-Bit-CPU. Die 81 Tasten sind in einer 8 x 11-Matrix (s.Abb.1) organisiert, wobei die Spalten X1...X8 ständig an den als Eingänge benutzten Datenleitungen D1...D8 liegen, die ihrerseits über 22kOhm-Widerstände auf H gezogen werden. Dagegen werden die elf Zeilen Y1...Y11 nacheinander kurzzeitig über elf Portleitungen des 8048 auf L gelegt.

	X8	X7	X6	X5	X4	X3	X2	X1
Y1	▦ 07	→ 06	ZCHN 05	PARA 04	EINBLO3	F1/2 02	FORM 01	F3/4 00
Y2	← 15	↑ 14	EZ 13	SEITE12	COPY 11	AUSBL10	PTR 09	EXIT 08
Y3	[+] 23	<> 22	SHIFT21	SUCHE20	# ' 19	RTN 18	+ * 17	DEL← 16
Y4	. : 31	- _ 30	ö 29	ä 28	p 27	ü 26	ß ? 25	↑ ` 24
Y5	, ; 39	m 38	k 37	l 36	i 35	o 34	9) 33	0 = 32
Y6	SPACE47	n 46	j 45	h 44	z 43	u 42	7 / 41	8 (40
Y7	v 55	b 54	f 53	g 52	t 51	r 50	5 % 49	6 & 48
Y8	x 63	c 62	d 61	s 60	w 59	e 58	3 § 57	4 % 56
Y9	y 71	SLOCK70	a 69	TAB 68	q 67	STOP 66	2 " 65	1 ! 64
Y10	←DEL 72	↓ 79	ENTER78	F7/8 77	[-] 76	CAN 75	EXTRA74	F5/6 73
Y11	ALT 80	--	--	--	--	--	--	--

Abb.1: Die Tastaturmatrix des Joyce
Für die Joyce-Besitzer, in deren Handbuch noch die Abbildung mit den Tastennummern (Anhang I, S.7) fehlt, sind diese hinter der jeweiligen Tastenbedeutung angegeben.

Tabelle 1

Taste	Tastennr.	Joystick	Matrixkoordinaten	
↑	14	oben	X7 / Y2	
↓	79	unten	X7 / Y10	
→	06	rechts	X7 / Y1	
←	15	links	X8 / Y2	
FORM	01	fire-2	X2 / Y1	nur bei LocoScript reserviert
▦	07	fire-1	X8 / Y1	nur bei LocoScript reserviert

```
07 N " "        DR Draw 'PICK'-Taste (Space)   Taste 1
01 N "↑M"       DR Draw 'DONE'-Taste (Return)  Taste 2
```

Tabelle 2

```
07 N "°'22'"    Pixel gesetzt (+) Taste 1
01 N "°'28'"    Pixel gelöscht (-) Taste 2
```

Tabelle 3

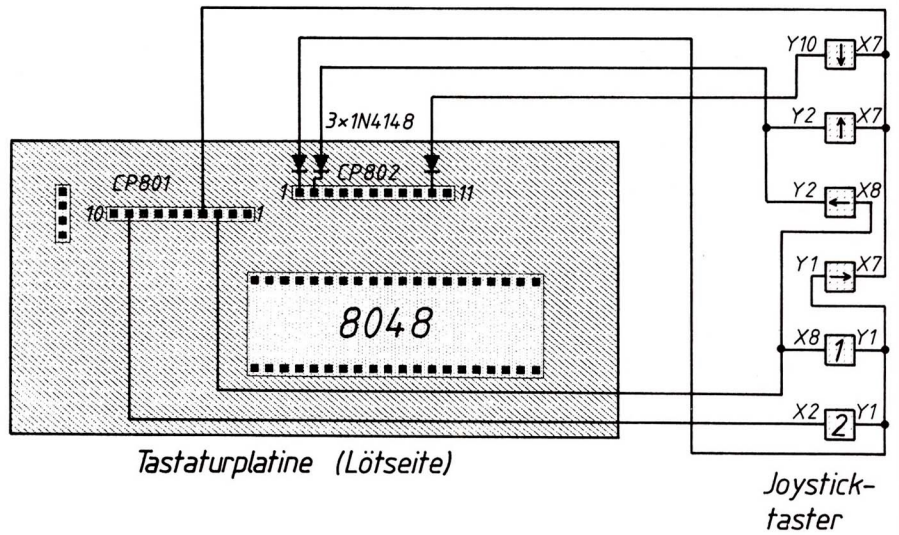


Abb.2: Anschlußplan für den JOYceSTICK

Jede Zeile wird etwa 70 mal in der Sekunde abgetastet. Jede Taste innerhalb einer Zeile, die während des Abtastens gedrückt ist, gibt das L an die entsprechende Datenleitung der CPU weiter. Dort wird die gedrückte Taste identifiziert, ihre Tastennummer codiert und seriell über das vieradrige Spiralkabel der Tastatur an die Zentraleinheit des Joyce gesendet.

Tabelle 1 gibt Auskunft über die für den JOYceSTICK benutzten Tasten.

Wie man sieht, sind nur sechs Leitungen nötig, um alle sechs Tasten zu erfassen. Damit beim gleichzeitigen Drücken mehrerer Joystick-Tasten keine Kurzschlüsse zwischen den drei verwendeten Y-Leitungen (Portausgänge!) entstehen, müssen diese mit Dioden abgesichert werden. Diese sperren, solange die Y-Leitungen auf H liegen, während das Abfrage-L durchgelassen wird.

Nun zur Praxis!

Zuerst müssen die Anschlüsse innerhalb des Joysticks gemäß Anschlußplan umgelötet werden. Notieren Sie sich die Farben der einzelnen Kabeladern und ihre jeweilige Bedeutung mit X1, Y2 usw.

Nach dieser Prozedur ist der Joystick allerdings nur noch ein JOYceSTICK und nicht mehr kompatibel zu normalen Joystick-Ports.

Als nächstes ziehen Sie den Tastaturstecker aus der Buchse im Monitor. Lösen Sie die sechs Kreuzschlitzschrauben auf der Unterseite der Tastatur und heben sie die untere (jetzt oben liegende) Gehäusehälfte ab. Zum Vorschein kommt direkt die Lötseite der CPU-Platine. Orientieren Sie sich entsprechend dem Anschlußplan. Kürzen Sie nun die Anschlußdrähte der drei Dioden auf etwa 5 mm und löten Sie die drei Dioden mit dem Kathodenring zu den Lötpunkten der Platine zeigend an. Dabei sollten Sie die Dioden zwischen Gehäuse und Lötstelle mit einer Pinzette halten und kühlen! Sorgen Sie dafür, daß die freien Anschlußenden der Dioden nicht unkontrolliert mit anderen Lötstellen auf der Platine Kontakt bekommen können.

Sie können den Joystick entweder fest mit der Tastatur verbinden oder – was sicherlich empfehlenswerter ist – den Einbau einer neunpoligen MIN-D-Stiftleiste vorziehen, um den Joystick auch abkoppeln zu können.

Im ersten Fall schneiden Sie den Joystickstecker ab und wählen eine geeignete Stelle für den Gehäusedurchbruch. Nach Ausführung der Bohrung (Kabeldurchmesser) oder Einkerbung (an der Trennlinie der Gehäusehälften) entgraten Sie dieselbe und sichern das genügend lang durchgeführte Joystickkabel mit einer anständigen Zugentlastung!

Nach Abisolieren des Kabelmantels (auf ca. 8 cm) löten Sie die sechs Adern entsprechend Ihren Aufzeichnungen und dem Anschlußplan auf der Platine an.

Haben Sie sich jedoch für die Einbaubuchse entschieden, denken Sie bei der Wahl des Einbauortes an den »Tiefgang« der Buchse, damit ärgerliche Kollisionen mit dem Tastaturchassis oder der CPU-Platine vermieden werden. Nach dem Einbau der Buchse müssen Sie die Anschlußbelegung des Joystick-Buchsensteckers herausfinden, indem Sie mit einem Ohmmeter oder Durchgangsprüfer die Adern durchmessen und sich die Lage der Anschlüsse auf der Stirnseite des Steckers sowie die Bedeutungen (X1, Y2 usw.) notieren.

Danach müssen Sie nur noch die Einbaustiftleiste z.B. mittels sechsadrigem Flachbandkabel mit der CPU-Platine gemäß Anschlußplan und Ihren Notizen über die Anschlußbelegung verbinden

Überprüfen Sie abschließend noch einmal, ob Sie alles korrekt ausgeführt und keine Anschlüsse vertauscht haben. Dann verschrauben Sie das Tastaturgehäuse wieder. Das war's – bis auf die Belegung der Zusatztasten des Joysticks.

Die Richtungstasten müssen nicht definiert werden, da sie parallel zu den Cursortasten geschaltet sind. Die Zusatztasten werden je nach den Erfordernissen des laufenden Programmes mit Hilfe des SETKEYS-Dienstprogrammes und einer von Ihnen zu erstellenden ASCII-Tastendefinitionsdatei belegt (siehe auch Joyce-Handbuch Kap.5, S.33 sowie Anhang 1, S.6 ff).

Für DR Draw z.B. sieht diese ASCII-Datei mit dem Namen »KEYS.DRW« wie in Tabelle 2 aus.

Vor dem Aufruf von DR Draw werden dann mit dem Kommando:

SETKEYS KEYS.DRW

die Zusatztasten definiert. Das Dienstprogramm SETKEYS.COM und die Datei KEYS.DRW müssen natürlich auf dem angemeldeten Laufwerk zu finden sein. Wenn Sie sich eine Startdiskette angelegt haben, mit der nach dem Laden des Betriebssystems über die Stapeldatei PROFILE.SUB z.B. automatisch Dateien auf die Ramdisk (Laufwerk m:) kopiert werden, können Sie das oben angegebene Kommando in die PROFILE.SUB-Datei integrieren und brauchen sich fortan nicht mehr um die Tastaturbelegung zu kümmern.

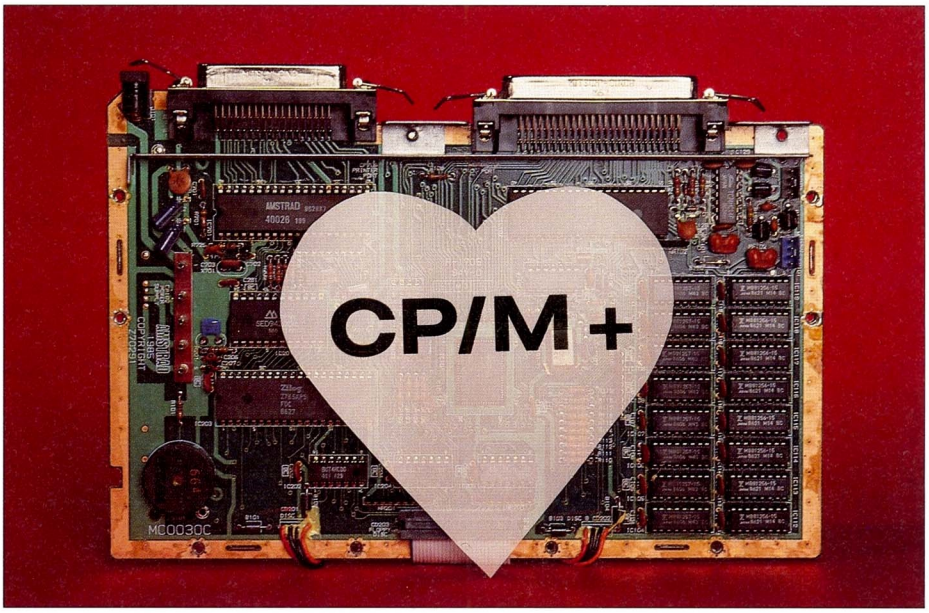
Für das Programm HYROGLYPHOS TOTAL aus CPC 12/86 wäre die Tastendefinition 'KEYS.HYR' (Tabelle 3) denkbar.

Vor dem Aufruf von BASIC wird dann SETKEYS KEYS.HYR eingegeben – und man kann sich beim Editieren eines Buchstabens beqüm zurücklehnen.

(Stephan Stürmer)

Im Herzen von CP/M Plus

- XBIOS zerlegt -



Anfang dieses Jahres »schneite« mir ein Entwurf für eine Serie ins Büro, die detaillierte Informationen über »hautnahe« Programmierung des Betriebssystems CP/M 3.0 speziell auf den PCWs geben wollte. Die Durchsicht der ersten Teile der Serie begeisterte mich derart, daß ich mich entschloß, alle Folgen komplett in einen Beitrag zusammenzufassen und diesen in diesem Heft zu veröffentlichen. Zugegeben, es ist wirklich »hard stuff«, der dem geneigten Leser in dieser konzentrierten Form doch einiges abverlangt. Hier liegt aber eine Sammlung von know-how vor, die in dieser Form bislang in Deutschland noch nicht veröffentlicht wurde; in Form einer Serie wäre dem Thema doch einiges an Aktualität verlorengegangen.

Alsdann möchte ich Sie mit der Überschrift, die mir zu dem Artikel eingefallen ist, in die Materie entlassen: Herr Anton hat gezaubert...

(me)

Floppy-Speeder

Zugegeben, schnell sind sie schon, diese 3 Zoll Schneider Floppies. Aber beispielsweise beim Kopieren umfangreicher Dateien wünscht man sich schon manchmal etwas mehr »Power«. Aber wie kann man das denn bloß machen? – Die Floppy-Drehzahl erhöhen? Das ist vielleicht ein guter Aprilscherz, bringt aber absolut gar nichts. Und doch gibt es noch einen Punkt, an dem man einhaken kann, nämlich die Steprate.

Die Steprate ist die Zeit, die der Lesekopf der Floppy benötigt, um von einem Track auf den nächsten zu gelangen. Oder genauer: die Zeit, die dem Laufwerk vom Rechner zugestanden wird. Tatsächlich räumt der Rechner der Floppy eine gewisse Zeit ein, bis er annimmt, daß der Lesekopf dort ist, wo er sein soll, obwohl die Floppy notfalls auch noch schneller könnte.

Wenn es nun gelänge, diese Zeit soweit wie möglich zu verringern, wäre der Transfer tatsächlich noch etwas schneller (allerdings nur bei Stepintensiven Vorgängen).

Und daß es wirklich geht, beweist das Programm SPEEDER. Assemblieren Sie es mit M80, linken es, und Sie haben einen ebenso einfachen wie wirkungsvollen Diskbeschleuniger. Wenn Sie SPEEDER dann unter CP/M starten, wird die Steprate von standardmäßig 12 Milisekunden auf 8 ms gesenkt. Das machen die Schneider-Floppies auf alle Fälle noch mit. Weitere Senkungen sind nicht sehr zu empfehlen, obwohl mit einem 5,25"-Floppy (NEC FD 1055) auch eine Rate von 4 ms noch funktionierte. (Die 3 Zoll-Floppies spielten da aber nicht mehr mit...).

Die Steprate bleibt solange erhalten, bis entweder das Betriebssystem neu gebootet oder mit UNSPEED der Default von 12 ms eingestellt wird. Bei entsprechender Änderung ist es auch möglich, sehr langsame Floppies anzupassen.

Jetzt werden Sie sich wohl fragen, wie die Sache eigentlich genau funktioniert, denn die Assemblerlistings weisen ja einige Sachen auf, die extrem merkwürdig sind. Diese Fragen werden im Lauf des Artikels eine Antwort finden. (PS: Das Listing hat sowohl auf dem JOYCE als auch auf dem CPC 6128 unter CP/M 3.0 volle Funktionsgarantie, aber auch davon später mehr...)

```
        ;*************************
        ;* FLOPPY-SPEEDER V 1.3 *
        ;*   1986 by M.Anton    *
        ;*************************
;
.Z80  ; (mit M80 assemblieren)
;
ASEG
ORG 0100H
;
        LD    HL,TABLE    ; Parametertabelle
        LD    DE,0C000H   ; in COMMON-Bereich
        LD    BC,0007H    ; ab C000H
        LDIR              ; verschieben.
;
        LD    HL,0C000H   ; Para-Adresse laden
        CALL  0FC5AH      ; X-BIOS aufrufen
        DW    0083H       ; (DD SETUP)
        JP    0000H       ; ...das war alles!
;
TABLE:  DW    320AH       ; Parametertabelle
        DW    1EAFH
        DB    08H         ; <- STEPRATE in ms.
        DW    030FH
;
END
```

SPEEDER.MAC

```
;UNSPEED -> setzt Steprate
;    auf Normalwert zurück
;
.Z80          ; (M80)
ASEG
ORG  0100H
;
CALL 0FC5AH   ; X-BIOS
DW   0080H    ; (DD INIT)
JP   0000H    ; ...OK...
;
END
```

UNSPEED.MAC

- Im Herzen von CP/M Plus - Das erweiterte BIOS von JOYCE und CPC 6128

1. : Grundlagen

Herzlich willkommen zu unserer Exkursion in die tiefsten Abgründe des Betriebssystems CP/M 3.0. In diesem Artikel soll gezeigt werden, wie sich eine Vielzahl von Systemparametern unter – oder besser aus CP/M Plus heraus ändern lassen, auch solche, an die man sonst »normalerweise« gar nicht herankommt.

Vorbemerkungen

Inhalt dieses Artikels ist also CP/M Plus. Allerdings nicht in der »gewohnten Form« von BDOS- und BIOS-Plaudereien, sondern noch einen Schritt näher an der Hardware. So soll z.B. gezeigt werden, wie man aus Anwenderprogrammen heraus die Schnittstellenparameter oder die Tastaturbelegung ändern kann. Manche dieser Dinge gehen natürlich auch mit den üblichen Dienstprogrammen, man kann sie aber auch anders und intelligenter machen. So ist es wohl wesentlich eleganter, wenn sich ein Programm seine Tastaturbelegung selbst erzeugt, als wenn es auf ein SUBMITFile mit SETKEYS angewiesen ist. Und dann sind dort immer noch jene Dinge, die man eigentlich nie zu Gesicht bzw. in den Griff bekommt wie z.B. die Floppy-Steprate oder, beim JOYCE, der Bildschirmzeichensatz (!).

Nicht immer wird sich eine so große Wirkung mit so wenig Aufwand wie im Falle des Floppy-Speeders erreichen lassen, viele der hier vorzustellenden Beispiele entfalten ihre Wirkung erst im Zusammenhang eines größeren Programms. Dies heißt aber nicht, daß sich die Serie ausschließlich an Profi-Programmierer richtet, denn auch für den Nur-Anwender wird das eine oder andere nützliche kleine Programm anfallen.

Zu diesen Programmen noch ein Wort: Sie wurden alle auf einem JOYCE entwickelt und sind nur auf diesem garantiert lauffähig. Die Praxis hat gezeigt, daß einige Dinge auch auf den CPC 6128 unter CP/M 3.0 übertragen werden können, dies hindert mich jedoch nicht daran, nur für den JOYCE zu schreiben. (Schon allein deswegen, weil ich keinen CPC 6128 mein Eigen nenne). Hier sei an die Experimentier- und Mitteilungsfreudigkeit der CPC-Anwender appelliert. Eines kann jedoch schon gesagt werden: Der Floppy-Speeder läuft ohne Einschränkungen auch auf dem CPC. Wie mir allerdings bekannt wurde, besteht dort eher der Bedarf nach einem »Floppy-Slower«. Auch dieses leistet das Programm, wenn das Byte für die Steprate entsprechend geändert wird.

Noch kurz zu der Ausrüstung, die wir für unsere Expedition benötigen. Ich habe möglichst oft die Programmiersprachen BASIC und Turbo Pascal berücksichtigt, letztendlich läuft aber doch alles auf Maschinensprache hinaus. Sie sollten also schon etwas Erfahrung, zumindest Interesse in Sachen Z80-Assembler haben. Und wenn ich Z80 sage, dann meine ich auch Z80, denn die meisten Beispielprogramme wurden auf M80 von Microsoft entwickelt, weil ich es nicht einsehe, auf einer Z80-Maschine mit dem eingeschränkten 8080-Befehlssatz zu programmieren. (»Gnadenhalber« werden für manche Dinge aber auch SID und MAC herangezogen, aber dies wird die Ausnahme sein.

Grundlagen

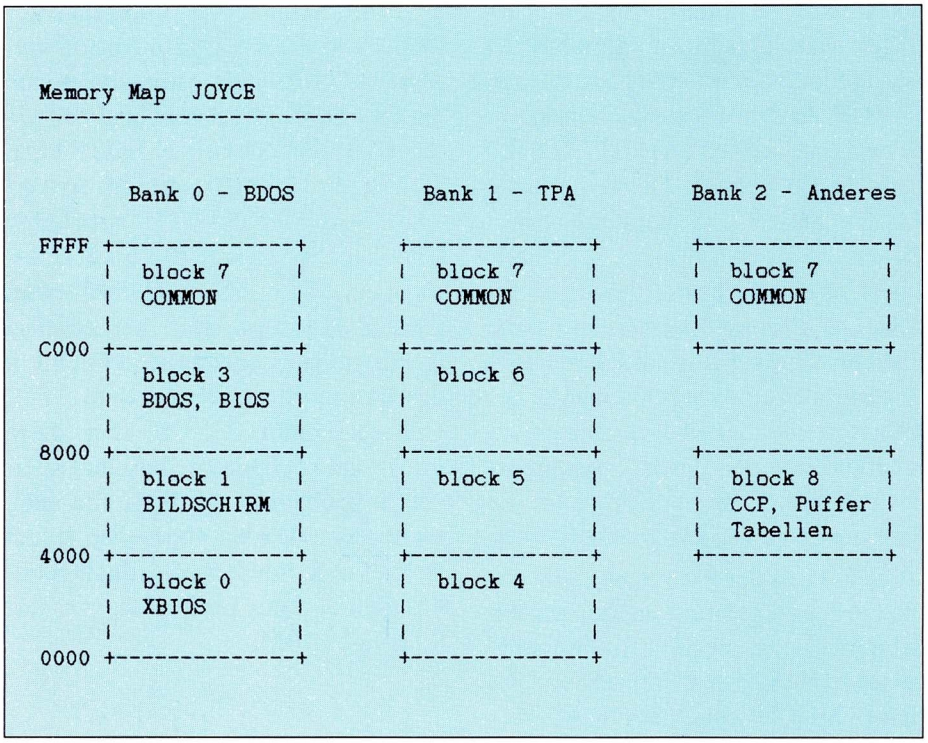
Ich möchte an dieser Stelle voraussetzen, daß Sie sich im »normalen CP/M« einigermaßen auskennen, daß also Begriffe wie BDOS und BIOS zu ihrem täglichen Vokabular gehören. Wenn dem nicht so ist, sollten Sie sich schleunigst irgendwo informieren, z.B. im »CP/M Plus Anwenderhandbuch« von Jürgen Hückstädt. Denn wir werden nun dort weitermachen, wo alle anderen CP/M-Darstellungen aufhören, nämlich bei der BIOS-Funktion Nummer 30...

Diese Funktion, auch USERF genannt, ist im normalen CP/M für Erweiterungen durch den Anwender freigehalten. Nicht so bei CP/M 3.0 auf den Schneider-Rechnern. Hier dient diese Funktion zum Aufruf weiterer BIOS-Routinen, die zusammen das erweiterte BIOS, auf englisch extended BIOS oder kurz XBIOS bilden. Hier sei etwas korrigiert: Im Artikel »Am Pulsschlag der CPU« im Happy Computer Sonderheft 10/86 wurde gesagt, daß die Funktion USERF beim CPC 6128 frei sei. Daß dem nicht so ist, zeigt die offensichtliche Funktion des Floppy-Speeders auf dem CPC. Und daß beim JOYCE über diese Funktion lediglich der Bildschirmspeicher eingeblendet wird, stimmt auch nicht ganz. Davon abgesehen bildet das gesamte Sonderheft jedoch ebenfalls eine gute Einführung in das Standard-CP/M.

Das XBIOS

Bevor wir uns nun genauer dieser einmaligen Eigenschaft des Schneider-CP/M zuwenden, ist es hilfreich, einen kurzen Blick auf die Speicheraufteilung zu werfen. Beim JOYCE sieht die Memory Map in etwa so aus wie in Abbildung 3 (Memory Map JOYCE).

Die restlichen Blöcke und Bänke sind der RAM-Disk zugeteilt. Block 1 und der hier nicht abgebildete Block 2 bilden zusammen den Bildschirmbereich, über den später noch genauer berichtet werden wird. Block 7 (C000 – FFFF) ist in allen Bänken gleich, daher auch der Name COMMON. Über diesen COMMOM-Bereich erfolgt die Kommunikation der einzelnen Bänke miteinander, da sonst ein Bankswitching katastrophale Folgen hätte. Beim JOYCE wird das Bankswitching übrigens im wesentlichen durch die I/O-Adressen F0 bis F3 gesteuert. Nix genaues weiß ich aber leider auch nicht. (Ein Tip: Tracen Sie mal mit SID irgend eine BDOS-Funktion. Das Bankswitching erkennen Sie dann daran, daß die beiden letzten abgearbeiteten Befehle in etwa »MOV A,80« und »OUT F0,A« waren ... und daß Sie Ihren Rechner erst ausschalten müssen, um weiterzuarbeiten!!!



Memory Map JOYCE

Und noch ein kleiner Tip: Über die Adresse F8 läßt sich der JOYCE zu einem Reboot »überreden«: geben Sie mal unter Mallard BASIC ein: 'OUT 248,1' – das Betriebssystem wird neu geladen. Eine elegante Methode, um beispielsweise Demoprogramme zu beenden – allerdings bleibt dabei der Speicherinhalt erhalten.

Wie kommt man aber nun an die XBIOS-Funktionen heran? – Das ist ebenso kompliziert wie einfach. Kompliziert allerdings nur für den Rechner, denn der muß sich ja um solche heiklen Angelegenheiten wie Bankswitching u.a. kümmern. Der Anwender braucht also einfach nur die Funktion USERF aufzurufen...aber so einfach ist es auch wieder nicht. Immerhin muß ja geklärt werden, welche XBIOS-Funktion gewünscht wird und diese Funktion braucht ja eventuell noch einige Parameter. In der Praxis läuft es folgendermaßen ab: Zunächst muß die Einsprungadresse von USERF gefunden werden. Die erhält man durch Addition von dezimal 87 – das entspricht dem Überspringen von 29 »JMP xxxx«-Befehlen der Standard BIOS-Sprungleiste – zum Anfang derselben, und diese findet man in den Adressen 1 und 2 der TPA. Zur Kontrolle ab Adresse 0 mit SID disassemblieren: »L0« ergibt JMP xxxx. Dieses xxxx ist der Anfang der BIOS-Sprungtabelle (hier FC03), die Einsprungadresse für USERF ist also FC5A. (Ich nehme diese Adresse von nun an als Konstante, denn sowohl auf dem CPC als auch bei den CP/M-Versionen 1.2 und 1.4 des JOYCE ergibt sich diese Adresse und ich hoffe, sie wird auch noch bei Version 1.6 die gleiche sein...).

Nehmen wir jetzt nochmals das Listing zum Floppy-Speeder zur Hand: Wenn wir uns dort im Umfeld der Adresse FC5A umschauen, läßt sich schon ahnen, wie die Sache abläuft. Hinter dem Aufruf »CALL USERF« stehen zwei Bytes, die die gewünschte XBIOS-Funktion spezifizieren. In der Tat sind diese zwei Bytes die Einsprungadresse in die XBIOS-Sprungleiste, die in Block 0 angesiedelt ist. Nach dem Aufruf von USERF holt sich der Rechner also die Adresse der gewünschten XBIOS-Funktion aus den beiden dem Aufruf folgenden Bytes und führt diese dann aus (wie er dies macht, können Sie mit Trace verfolgen – allerdings nur bis zum Bankswitching...).

Und die Parameterübergabe? Die geht äußerst vielfältig vor sich. Der einfachste Fall zeigt sich bei UNSPEED: Hier werden gar keine Parameter übergeben. Andere Funktionen übernehmen ihre Parameter aus den Registern der CPU, die vor dem XBIOS-Aufruf entsprechend gefüllt werden müssen. Bei SPEEDER ist eine Parametertabelle im COMMON-Bereich nötig, es gibt sogar eine Funktion, die ein ganzes Unterprogramm im COMMON-Bereich erwartet. Die Einzelheiten werden jedoch bei der Vorstellung der einzelnen XBIOS-Funktionen erklärt werden.

Hiermit sei der grauen Theorie Genüge getan, ich hoffe, Sie haben nun das notwendige theoretische Wissen und einen ersten Einblick erhalten in das, was noch auf Sie zukommt.

- Hacker's Delight - 2. Die serielle Schnittstelle

Nun geht's los mit der genaueren Analyse der vorhandenen XBIOS-Funktionen. Dabei wird nicht in der natürlichen Reihenfolge im XBIOS vorgegangen, sondern nach der Komplexität. Dies hat zur Folge, daß die ersten Routinen, die sich mit der Floppy beschäftigen und wirklich »hard stuff« sind, erst ganz zum Schluß behandelt werden. Wundern Sie sich also nicht, wenn wir schon bei der XBIOS-Funktion Nr.18 anfangen.

Probleme, Probleme...

Obwohl in diesem Abschnitt nur drei Funktionen zur Debatte stehen, so lösen Sie doch die Probleme, die viele Benutzer der seriellen Schnittstelle bislang gehabt haben: wie stelle ich die Schnittstellenparameter ein und wie kann ich direkt auf die SIO zugreifen. Besonders DFÜ-Fans dürften bislang kein leichtes Leben auf dem JOYCE gehabt haben. Der »geschenkte Gaul«, MAIL232, bot zwar eine schöne Benutzeroberfläche in Sachen Parametereinstellung, war aber ansonsten nicht ganz »Hacker's Delight«. Alternativen wie KERMIT oder selbstgeschneiderte Terminalprogramme mochten zwar besser sein, den Komfort einer menuegesteuerten Parameterwahl konnten sie jedoch nicht bieten. Zur Neuwahl

mußte immer das Programm verlassen und mit SETSIO die neuen Werte eingestellt werden. Dieser Streß hat nun theoretisch ein Ende. Nach dieser Folge ist es möglich, DAS Terminalprogramm für den JOYCE zu schreiben; mit Parameterwahl, Protokoll, CR-LF-Umwandlung und... und...und (die Redaktion würde sich sehr freuen, demnächst das erste Top-DFÜ-Programm aus der Leserschaft veröffentlichen zu dürfen...).

Die SIO unter CP/M

Im normalen CP/M erfolgt die Ein- und Ausgabe zur seriellen Schnittstelle (bekanntlich) über die BDOS-Funktionen 3 und 4, die Statusabfrage über 7 und 8. Eine Einstellung der Schnittstellenparameter ist dagegen hardwareabhängig und kann nur über Hilfsprogramme wie SETSIO erfolgen. Mit Hilfe der XBIOS-Funktionen läßt sich dieses Problem auf dem JOYCE auch aus CP/M heraus in den Griff bekommen. Es stehen folgende drei Funktionen zur Verfügung, die das Einstellen und Auslesen der Werte ermöglichen. (Die Titelkonvention im folgenden: Funktionsnummer, Name, Einsprungadresse.)

XBIOS #18 CD_SA_INIT 00B6

Mit dieser Funktion wird die Schnittstelle initialisiert und alle Parameter außer der Baudrate eingestellt. Folgende Parameter werden über die CPU-Register übergeben:

A: enthält den Modus:
$00 = kein Handshake
$FF = Handshake

D: enthält die Anzahl der Stopbits:
$00 = 1
$01 = 1.5
$02 = 2

E: wählt die Parität:
$00 = keine
$01 = gerade
$02 = ungerade

H: Anzahl der Empfänger-Bits ($05 – $08)

L: Anzahl der Sender-Bits ($05 – $08)

Ab der CP/M-Version 1.4 stehen bei dieser Funktion noch folgende Möglichkeiten zur Verfügung:

1.) Auswahl zwischen ständiger Abfrage des SIO-Status (»polling«) oder Interrupterzeugung bei eingehenden Zeichen. Diese Optionen werden über den Accu ausgewählt:

A: Modus:
$00 = polling ohne Handshake
$FF = polling mit Handshake
$FE = Interrupt ohne Handshake
$FD = Interrupt mit Handshake

(Die übrigen Register wie oben beschrieben.)

2.) Direkte Beeinflussung der Steuerleitungen RTS und DTR. Die Auswahl erfolgt über den Accu, die übrigen Register bleiben unberücksichtigt:

A: Befehl:
$80 = DTR low
$7F = DTR high
$7E = RTS low
$7D = RTS high

Die praktische Anwendung:

```
. . .
(Register mit den Werten laden)
CALL FC5A
DW 00B6
. . .
```

XBIOS #19CD_SA_BAUD00B9

Mit dieser Funktion wird die Baudrate eingegeben. Die Parameter werden über die Register HL übergeben:

H: codierte Empfängerbaudrate ($01 – $0F)

L: codierte Senderbaudrate ($01 – $0F)

Der Code für die Baudrate ist der Tabelle zu entnehmen.

Aufruf:

```
(Werte laden)
CALL FC5A
DW 00B9
. . .
```

(Bei den Funktionen 18 und 19 wird keine Kontrolle durchgeführt. Werden unzulässige Werte eingegeben, ist das Resultat auch entsprechend...)

XBIOS #20CD_SA_PARAMS00BC

Diese Funktion ermöglicht das Lesen der aktuellen SIO-Parmeter und wird ohne Parameter aufgerufen. Nach dem Aufruf enthalten die Register die momentanen Werte (Format siehe #18 und #19):

A: Modus
B: Senderbaudrate
C: Empfängerbaudrate
D: Stopbits
E: Parität
H: Empfängerbits
L: Senderbits

Aufruf:

```
. . .
CALL FC5A
DW 00BC
```

(Werte stehen zur Verfügung)

(NB: Nach XBIOS-Aufrufen sind die Registerinhalte in der Regel zerstört, die Werte des alternate register set bleiben jedoch erhalten.)

Baudratecodes für JOYCE-SIO

Code in dezimal	Register H+L hex	Baudrate	Genauigkeit
1	01	50	0.10 %
2	02	75	0.12 %
3	03	110	0.13 %
4	04	134.5	-0.14 %
5	05	150	-0.14 %
6	06	300	0.18 %
7	07	600	-0.26 %
8	08	1200	-0.26 %
9	09	1800	-0.74 %
10	0A	2400	-0.26 %
11	0B	3600	0.18 %
12	0C	4800	-0.26 %
13	0D	7200	-2.18 %
14	0E	9600	-0.26 %
15	0F	19200	7.62 %

Baudratecodes für JOYCE-SIO

(Aufgrund der hohen Ungenauigkeit ist eine Verwendung der Baudraten 7200 und 19200 nicht zu empfehlen...)

Ein Beispiel

Das folgende Listing zeigt die Einbindung der Funktionen 18 und 19 in Turbo-Pascal. Es dient nur zur Illustrierung und kann als Anregung zum Einbau der Parameterwahl in ein Terminalprogramm verwendet werden. Der Trick ist der Notwendigkeit der INLINE-Funktion die Verwendung von absoluten Variablen für die CPU-Register, da auf diese leider nicht direkt zugegriffen werden kann. Die Anlage erfolgt im Lowbyte-Highbyte-Format, da 16-Bit-Ladebefehle verwendet werden.

(Das Programm sollte vor dem Compilieren auf CFFF gesetzt werden, um Konflikte mit den Variablen zu vermeiden.)

```
program SIO_PARA;

ä Demoprogramm: Einstellung der Schnitt-
  stellenparameter unter Turbo Pascal.
  Compiler-Endadresse auf CFFF setzen. ü

var
   A : byte absolute $D000;
   E : byte absolute $D001;
   D : byte absolute $D002;
   L : byte absolute $D003;
   H : byte absolute $D004;

ä reservierter Platz f.d. CPU-Register ü


procedure Set_Parameter;

ä Stopbits,Parity,Modus,Wortbreite ü

begin inline (
   $3A/$00/$D0/       ä LD A,(D000)  ü
   $ED/$5B/$01/$D0/   ä LD DE,(D001) ü
   $2A/$03/$D0/       ä LD HL,(D003) ü
   $CD/$5A/$FC/       ä CALL FC5A    ü
   $B6/$00);          ä DW 00B6      ü
end;



procedure Set_Baud;

ä Baudrate ü

begin inline (
   $2A/$03/$D0/       ä LD HL,(D003) ü
   $CD/$5A/$FC/       ä CALL FC5A    ü
   $B9/$00);          ä DW 00B9      ü
end;



begin

ä Hauptprogramm / Werteeingabe ü

clrscr;
writeln ('Einstellung der Schnittstellenparameter');
writeln;
write ('Modus (0/255): ');      readln (A);
write ('Stopbits (0-2): ');     readln (D);
write ('Parität (0-2): ');      readln (E);
write ('Empfangsbits (5-8): '); readln (H);
write ('Senderbits (5-8): ');   readln (L);
Set_Parameter;
write ('Empfangsbaudrate (1-15): '); readln (H);
write ('Senderbaudrate (1-15): ');   readln (L);
Set_Baud;

end.
```

SIO-Parameter unter Turbo Pascal

Zwei Problemläufe von SIO__PARA:

M>sio__para
Einstellung der Schnittstellenparameter

Modus (0/255): 255
Stopbits (0 – 2): 2
Parität (0 – 2): 0
Empfangsbits (5 – 8): 6
Sendebits (5 – 8): 8
Empfangsbaudrate (1 – 15): 1
Sendebaudrate (1 – 15): 2

M>setsio
Rx 50 Tx 75 Bits rx 6 Bits tx 8 Stop 2 Parity none Xon off Handshake on

M>sio__para
Einstellung der Schnittstellenparameter

Modus (0/255): 253
Stopbits (0 – 2): 0
Parität (0 – 2): 0
Empfangsbits (5 – 8): 8
Sendebits (5 – 8): 7
Empfangsbaudrate (1 – 15): 4
Senderbaudrate (1 – 15): 6
(Anmerkung: Die alte Version von SETSIO ignoriert den Interruptmodus...)

Hard – harder – hardest...

Soweit das Thema serielle Schnittstelle für die Software-Freaks. Aber auch für eher hardwareorientierte Programmierer hier noch einige Tips, die ihr Geld wert sind. Man kann nämlich auch direkt auf die Schnittstellenbausteine zugreifen. Dies ist besonders praktisch, wenn es auf extrem hohe Übertragungsraten ankommt, denn die Standard-CP/M Funktionen spielen da nicht mehr mit. Hier nun die Details:
Das Schnittstellenmodul CPS8256 verwendet einen Z80-DART auf Kanal A für die serielle Schnittstelle. Die Taktfrequenzen werden über einen 8253-CTC erzeugt.
Der Centronics-Ausgang besteht aus einem 8-Bit-Latch, die Steuerung (Busy, Strobe) erfolgt über die Steuerleitungen DTR und CTS des Kanal B des DART(!).

Die verwendeten I/O-Adressen sind:
$E0: DART Kanal A Daten (SIO)
$E1: DART Kanal A Kontrolle (SIO)
$E2: DART Kanal B Daten (-?-)
$E3: DART Kanal B Kontrolle (CEN)
$E4: 8253 Zähler 0 (TxC A)
$E5: 8253 Zähler 1 (RxC A)
$E6: 8253 Zähler 2 (TxRxC B)
$E7: 8253 Modus
$E8: Centronics-Latch

Mit Hilfe dieser Adressen und der intimen Kentnis der Funktionsweise der verwendeten Bausteine steht einem Direktzugriff auf die Hardware eigentlich nichts mehr im Wege.
Im Zusammenhang mit Hardware noch ein kleiner Tip: Amstrad garantiert die freie Verwendbarkeit der I/O-Adressen A0 – A7.
Wer sich Erweiterungen also selbst basteln will, hat hier schon acht ergiebige Anlaufadressen.

- Hautkontakt - 3. Die Tastatur

Weiter geht es auf unserer Forschungsreise durch die Innereien des JOYCE mit der Tastatur, wobei wir eine vollkommen neue Art der Tastaturabfrage kennenlernen werden sowie eine Alternative zum Dienstprogramm SETKEYS.

Ein wenig anders...

Unter CP/M tritt man mit der Tastatur normalerweise über die BDOS-Funktionen 1, 10 und 11, eventuell auch noch 6 in Verbindung und erhält dabei zumeist eines oder mehrere ASCII-Zeichen zurück. Gut und schön. Also schickt die Tastatur gleich komplette ASCII-Zeichen an den Rechner? Offensichtlich nicht, denn sonst müßte unsere Tastatur ja glatte 256 Tasten haben und die Tasten »SHIFT«, »ALT« und »EXTRA« wären überflüssig. Also muß die Sache mit der Tastatur wohl doch ein wenig anders funktionieren. Und in der Tat ist es so, wie die im folgenden besprochenen XBIOS-Funktionen zeigen werden.

»Tic-Tac-Tokens«

Die primären Daten, die von der Tastatur an den Rechner gesandt werden, bestehen aus zwei Teilen (praktischerweise zwei Bytes) und werden erst im Rechner dekodiert. Der Anwender bekommt mit den BDOS-Funktionen dann erst die dekodierte Version. Die Dekodierung erfolgt dabei über rechnerinterne Tabellen, die manipuliert werden können, wie SETKEYS zeigt. Doch nehmen wir uns zunächst die Primärdaten unters Mikroskop. Die »Tokens«, wie diese Daten auf englisch so schön heißen, bestehen praktisch aus zwei Bytes. Das erste Byte ist die Nummer der gedrückten Taste, die mit der Tastennummer, wie sie im JOYCE-Handbuch abgedruckt ist und wie sie auch von SETKEYS verwendet wird. Das zweite Byte zeigt durch sein Bitmuster an, welche Zusatztasten

(SHIFT, EXTRA usw.) gedrückt wurden und wird als Statusbyte bezeichnet.
Für das Statusbyte gelten folgende Zuordnungen (1 = Bedingung erfüllt):
bit 0 : (nicht definiert)
bit 1 : EXTRA gedrückt
bit 2 : CAPS LOCK (=ALT+ENTER)
bit 3 : REPEAT (durch Wiederholung erzeugt)
bit 4 : NUM LOCK (=ALT+FORM)
bit 5 : SHIFT gedrückt
bit 6 : SHIFT LOCK aktiv
bit 7 : ALT gedrückt

Im XBIOS existieren nun zwei Funktionen, die den direktesten aller Direktzugriffe auf die Tastatur erlauben, und zwar lesender- wie schreibenderweise, denn es existiert ein Tastaturpuffer, in dem noch nicht abgeholte Tokens sorgsam aufbewahrt werden:

XBIOS #30KM_KT_GET00DA

Diese Funktion liest ein Token entweder direkt von der Tastatur oder aus dem Tastaturpuffer. Wenn ein gültiges Token eingelesen wurde, wird das Carry-Flag auf high gesetzt und Register C enthält die Tastennummer, Register B den Status.

XBIOS #31KM_KT_PUT00DD

Diese Funktion schreibt ein Token in den Tastaturpuffer, wobei das Register B den Status und C die Tastennummer enthalten muß. Es kann mehr als 1 Token geschrieben werden; wenn die Pufferkapazität von ca. 10 Tokens überschritten wird, geht der Überschuß jedoch verloren. Der Puffer ist übrigens als »Last-In First-Out«-Speicher organisiert.
Das folgende Turbo Pascal-Programm zeigt die Verwendung der beiden Funktionen. Im ersten Teil wird die Tastatur abgefragt und bei Tastendruck die entsprechenden Werte angezeigt. Nach Drücken der Taste »ENTER« startet der zweite Teil, der die Meldung »ENDE!!!« in den Tastaturpuffer schreibt. Diese Meldung wird dann nach Programmende hinter den CP/M-Prompt geschrieben (aus irgendeinem Grund mußten noch zwei Dummy-Token eingefügt werden, offensichtlich kassiert CP/M irgendwo 2 Token, ohne sie anzuzeigen).

```
Listing: 'Tastatur direkt'
--------------------------


program Keyscan;

ä Demonstration des direkten Zugriffs
  auf die Tastatur unter Turbo Pascal ü

var
  i       : integer;
  B       : byte absolute $D001;
  C       : byte absolute $D000;
  Status  : arrayÄ0..8ü of stringÄ10ü;

const
  Message = #64#64#64#58#61#46#58;
    ä   LIFO: '!!!ENDE'   ü

procedure Get;
 ä liest Tastaturpuffer bis
   ein Zeichen vorhanden... ü

begin inline (
   $CD/$5A/$FC/      ä L: CALL FC5A   ü
   $DA/$00/          ä    DW 00DA     ü
   $D2/$-6/          ä    JP NC,L:    ü
   $ED/$43/$00/$D0); ä    LD(D000),BC ü
end;



procedure Put;
 ä schreibt Tastenmuster
   in Tastaturbuffer... ü

begin inline (
   $ED/$4B/$00/$D0/  ä LD BC,(D000) ü
   $CD/$5A/$FC/      ä CALL FC5A    ü
   $DD/$00);         ä DW 00DD      ü
end;


begin
 ä Hauptprogramm: Lesen+Zeigen  ü

 StatusÄ1ü := '-?-?-';
 StatusÄ2ü := 'EXTRA';
 StatusÄ3ü := 'CAPS LOCK';
 StatusÄ4ü := 'REPEAT';
 StatusÄ5ü := 'NUM LOCK';
 StatusÄ6ü := 'SHIFT';
 StatusÄ7ü := 'SHIFT LOCK';
 StatusÄ8ü := 'ALT';

 clrscr;
 writeln ('Tastatur-Demo  Keyscan:');
 writeln (' Direktzugriff auf den');
 writeln ('   Tastaturpuffer:');
 writeln;

 repeat
   Get;
 write ('Status: ');
 for i:=1 to 8 do      ä Status-Analyse ü
   begin
   if (B and 1)=1 then write ('  ',StatusÄiü);
   B:=B shr 1;
   end;
 writeln ('   Taste: ',C);
until C=78;          ä Abbruch mit ENTER ü
writeln;


ä Meldung in Tastaturpuffer schreiben ü

B:=32;                          ä <- SHIFT ü

for i:=0 to 6 do
  begin
   C:=ord(copy(Message,i+1,1));
   Put;
  end;

C:=0; B:=0; Put; Put;  ä Dummy schreiben ü

end.
```

»Tastatur direkt«

Und was macht man nun mit diesen Funktionen? Da könnte man zunächst einmal die Tastaturabfrage explizit gestalten. Denn wer garantiert mir, daß beim Empfang von »E« wirklich die Taste »E« (Nr. 58, Status SHIFT) gedrückt wurde. Dann könnte mit KM_KT_PUT ein direkter Programmstart mit Parameterübergabe (z.B. »DIRÄALLÜ«) erfolgen, was nicht mal mit Turbo-Befehl EXECUTE möglich ist. Weiterhin kann mit einer Kombination von PUT und GET die Tastatur in einen definierten Zustand gebracht werden: sollen z.B. in einem Programm nur Großbuchstaben zulässig sein, kann man sich einerseits darauf verlassen, daß der Anwender SHIFT oder CAPS LOCK verwendet oder eine Codewandlung durchführen, oder eben gleich CAPS LOCK vom Programm aus einstellen!!! Dazu lädt man mit PUT ein Token mit dem entsprechenden Status in den Puffer und liest es mit GET sofort wieder aus. Danach ist die Tastatur garantiert im gewünschten Zustand. SHIFT LOCK kann man damit jedoch nicht abschalten, da es hardwaremäßig erzeugt wird.

Tabellen, Tabellen

Und wie geht es nun weiter? – Ganz einfach... Entsprechend der Tastennummer und dem Status wird in einer Tabelle nachgeschaut, welcher Wert bei welcher Kombination erzeugt werden soll. Es existieren insgesammt fünf Tabellen, eine für den Normalzustand , und je eine für die Kombinationen SHIFT, ALT, SHIFT+ALT und EXTRA. Hier werden also die Tokens in den entsprechenden Code umgewandelt. Allerdings werden nicht alle Codes in darstellbare Zeichen umgewandelt. Die Codes $90 – $9E sind für die sogenannten »Expansions« reserviert, auf die wir später noch zurückkommen werden, Code $9F bedeutet, daß der entsprechende Tastendruck einfach ignoriert werden soll. Die Tabellen können nun entweder über SETKEYS oder direkt über eine XBIOS-Funktion manipuliert werden:

XBIOS #29KM_SET_KEY00D7

Diese Funktion ändert den Inhalt der Übersetzungstabelle für eine Taste.

B : enthält den neuen Code
C : enthält die Tastennummer
D : bezeichnet die zu ändernde Statustabelle

Bit 0 : normal
Bit 1 : SHIFT
Bit 2 : ALT
Bit 3 : SHIFT+ALT
Bit 4 : EXTRA
(übrige Bits irrelevant)

Das folgende Beispiel zeigt, wie unter Turbo Pascal »klare Verhältnisse« geschafft werden können: die Cursortasten sind Wordstar-kompatibel, STOP und EXIT funktionieren nur über SHIFT. Natürlich kann man dies auch über SETKEYS machen, aber in einem großen Programm fällt eine solche Routine nicht mehr auf und bedienungsfreundlicher ist's allemal...

Erweiterungen

Die »Expansions« sind die hilfreichen Tastataturcodes, mit denen sich »viel Text mit einem Tastendruck« erzeugen läßt. Und Sie ahnen es schon: auch dies kann man über das XBIOS erreichen. Für die Expansions ist ein spezieller Speicherbereich angelegt (Größe: unbekannt). Die folgende XBIOS-Funktion erwartet einen String, der beliebige Zeichen enthalten darf im COMMON-Bereich, sowie dessen Adresse, Länge und den Expansioncode in den Registern:

XBIOS #28KM-SET_EXPAND00D4

B : Expansion-Code ($90 – $9E)
C : Länge des Strings
HL : Adresse des Strings im COMMON-Bereich

Wurde eine Expansion erfolgreich übertragen, ist das Carry-Flag nach dem Aufruf gesetzt, ist dem nicht so, ist entweder der Expansion-Puffer voll oder ein ungültiger Expansioncode wurde gewählt.

Das folgende Assemblerlisting (für M80) belegt zu Übungszwecken die Funktionstaste »f1« mit Klartext. Wie Sie erkennen werden, ist mit der Installierung einer Expansion auch eine Tabellenänderung nötig, denn irgendwo muß ja schließlich vermerkt werden, daß Taste Nr. 2 im Normal-Status nicht mehr »↑Z« ist, sondern mit dem Expansioncode $90 belegt wurde und damit den entsprechenden Wert zu liefern hat: »Tastatur erweitert«

```
Listing: 'Tastatur anders'
--------------------------



program Keyset;
 ä Umdefinierung der Tastatur unter Turbo-Pascal:
   Cursortasten -> Wordstar, STOP & EXIT nur mit SHIFT ü


const
Keynum  = #$0E#$4F#$06#$0F#$08#$4E#$4B#$17#$4C#$42#$08#$42;
Keycod  = #$05#$18#$04#$13#$1B#$0D#$03#$2B#$2D#$03#$9F#$9F;
Keystat = #$FF#$FF#$FF#$FF#$02#$FF#$02#$FF#$FF#$02#$FD#$FD;
 ä Keynum:Tastennummer, Keycod:neuer Code, Keystat:Status ü

var
    i       : integer;
    C       : char absolute $D000;
    B       : char absolute $D001;
    D       : char absolute $D002;


procedure Set_Key;          ä aendert Wert einer Taste ü
  begin
   inline ($3A/$00/$D0/                    ä LD A,(D000) ü
           $4F/                            ä LD C,A      ü
           $3A/$01/$D0/                    ä LD A,(D001) ü
           $47/                            ä LD B,A      ü
           $3A/$02/$D0/                    ä LD A,(D002) ü
           $57/                            ä LD D,A      ü
           $CD/$5A/$FC/                    ä CALL FC5A   ü
           $D7/$00);                       ä DW 00D7     ü
end;


procedure Setup;         ä uebernimmt Werte aus Tabelle ü

begin
  for i:=1 to 12 do
  begin
    C:=copy(Keynum,i,1);
    D:=copy(Keystat,i,1);
    B:=copy(Keycod,i,1);
    Set_Key;
    end;
end;


begin

 Setup;

 ä hier geht das Hauptprogramm weiter ü

end.
```

»Tastatur anders«

```
;**********************************
;* EXPAND: legt einen String auf  *
;* Funktionstaste . 1986 by M. A. *
;**********************************
;
.Z80
ASEG
ORG 0100H
;
XBIOS   EQU 0FC5AH
SET_KEY EQU 00D7H
SET_EXP EQU 00D4H
;
KEYNUM  EQU 02H ; Taste: -f1-
KEYSTAT EQU 01H ; Status: normal
LENGTH  EQU 10H ; Stringlänge
EXPAND  EQU 90H ; Expansionnr.
;
; - String kopieren -
;
LD BC,LENGTH
LD DE,0C000H    ; nach COMMON
LD HL,STRING
LDIR
;
; - Expansion laden -
;
LD   B,EXPAND
LD   C,LENGTH
LD   HL,0C000H  ; String/COMMON
CALL XBIOS
DW   SET_EXP
;
; - Belegung ändern -
;
LD   B,EXPAND
LD   C,KEYNUM
LD   D,KEYSTAT
CALL XBIOS
DW   SET_KEY
;
; - Ende -
;
JP 0000H
;
; - String -
;
STRING:
;
DB "Funktionstaste 1"
;
;
END
```

»Tastatur erweitert

```
M>SID
CP/M 3 SID - Version 3.0
#A100
0100  MVI H,1
0102  MVI L,1
0104  CALL FC5A
0107  .
#S107
0107 00 E0
0108 00 .
#G100,109

*0109
#
```

»Tastatur schnell«

Und noch eine...

Wer kennt es nicht? Nach längeren Sitzungen vor dem heimischen EDVAltar lassen die Reflexe langsam nach und man kriegt seine Finger nicht mehr schnell genug von der Tastatur. Der Erfolg: dank der Tastaturwiederholung hat man mal wieder einige Zeichen zuviel auf dem Monitor... Und andererseits gibt es auch jene, die nicht schnell genug ihre Sternchen auf die Röhre gezaubert haben wollen. Naja, in Sachen Tastaturgeschwindigkeit sind die Geschmäcker eben verschieden. Der eine möchte seine Finger möglichst lange auf den Tasten haben können, ohne das sich der Bildschirm gleich füllt, der andere hat eine Vorliebe für horizontale Linien und ist für eine schnelle Wiederholung dankbar. Glücklicherweise sind diese Leute nicht auf die festgelegten Werte angewiesen, denn diese kann man ändern, und zwar mit folgender XBIOS-Funktion:

XBIOS #32
KM_SET_SPEED 00E0

Über die Register H und L wird die Anfangsverzögerung und die Wiederholungsrate für die Tastaturwiederholung ausgewählt. Die Einheit dafür sind 1/50-Sekunden (entspricht der Periode für die Tastaturabfrage), Werte von 0 bedeuten 256.

H : Anfangsverzögerung
L : Wiederholungsrate

Zur Illustration vollziehen Sie einfach folgende Sitzung mit SID nach, bei deren Ende Sie mit den vorgegebenen Werten schon ein gewisses »Fingerspitzengefühl« benötigen, um weiterarbeiten zu können (s. Tastatur schnell).

- Persönliche Werte - 4. DEFAULTS und anderes

Wir wiederholen zunächst, was wir in den vorangegangenen Teilen zusammengetragen haben und werden dabei auf eine sehr praktische Anwendungsmöglichkeit stoßen. Weiterhin werden vier neue XBIOS-Funktionen vorgestellt, die jedoch nicht unbedingt zu den interessantesten gehören.

Warum?
Manch einer wird sich gefragt haben, was für eine praktische Seite dieser Artikel hat. Immerhin kann das Meiste doch auch über bereits vorhandene Dienstprogramme wie SETKEYS oder SETSIO erreicht werden. Vom zeitlichen und arbeitstechnischen Aufwand her gesehen mag dies ja stimmen. Wenn man jedoch mal solche Aspekte wie Bedienungsfreundlichkeit und Speicherplatz betrachtet, zeigen sich die Vorteile der XBIOS-Routinen sofort. Zunächst zur Bedienungsfreundlichkeit: Ein Programm, welches sich seine »Umgebung« wie beispielsweise die Tastaturbelegung selbst erzeugt ist wesentlich einfacher handzuhaben als ein Programm, das zusätzlich noch auf PROFILE-Dateien und Dienstprogramme angewiesen ist. Der Anwender startet das Programm einfach und braucht sich um nichts mehr zu kümmern. Auch das Erstellen einer Startdiskette gestaltet sich wesentlich einfacher, wenn nicht mehr unzählige Dienstprogramme kopiert werden müssen. Und das bringt uns schon zum Disketten-Speicherplatz: Weniger Dienstprogramme brauchen auch weniger Platz auf der Startdiskette. Wenn beispielsweise die Funktion der Programme SETKEYS und SETSIO in einem konkreten Fall auf ein kleines COM-File verlegt wird, kann man rund 4 KByte an Platz sparen. Wenn dann auch noch die Druckerinitialisierung und ein Autostart für das Hauptprogramm installiert wird und man es sich zur Angewohnheit macht, dieses Programm nach dem Booten »von Hand« zu starten, kann man sich auch noch SETLST und SUBMIT sparen. Diese Einsparung wirkt sich sowohl auf die Diskettenkapazität als auch auf die Ladezeiten aus, denn wenn weniger Dienstprogramme zum Anlaufen nötig sind, ist auch das Hauptprogramm schneller betriebsbereit.

Zum Beispiel:
Nehmen wir mal an, Ihre Lieblings-Mailbox oder Ihr Plotter, der an der seriellen Schnittstelle angeschlossen ist, benötigen eine Einstellung, die von der Grundeinstellung abweicht. Weiterhin

```
;*************************
; D E F A U L T S . M A C
;   Einstellung häufig
;   verwendeter Werte.
;    1986 by M. Anton
;*************************
;
.Z80
ASEG      ; für M80
;
ORG 0100H
;
XBIOS     EQU 0FC5AH
CCP       EQU  0000H
DDSETUP   EQU  0083H
SETSPEED  EQU  00E0H
SIOINIT   EQU  00B6H
SIOBAUD   EQU  00B9H
KEYSET    EQU  00D7H
KEYSTAT   EQU   0FFH
UPKEY     EQU    0EH
DWNKEY    EQU    4FH
LEKEY     EQU    0FH
RIKEY     EQU    06H
;
; - VARIABLEN -
;Werte hier ändern:
;
STEPRATE EQU  08H
;
DELAY     EQU  1EH
REPEAT    EQU  02H
;
SIOMODE   EQU  00H
STOPBITS  EQU  01H
PARITY    EQU  00H
RXBITS    EQU  07H
TXBITS    EQU  07H
TXBAUD    EQU  06H
RXBAUD    EQU  06H
;
UPCODE    EQU  05H
DWNCODE   EQU  18H
LECODE    EQU  13H
RICODE    EQU  04H
;
;
;- STEPRATE -
;
LD   HL,TABLE
LD   DE,0C000H
LD   BC,0007H
LDIR
LD   HL,0C000H
CALL XBIOS
DW   DDSETUP
;
;- KEYSPEED -
;
LD   H,DELAY
LD   L,REPEAT
CALL XBIOS
DW   SETSPEED
;
;- SIOPARAM -
;
LD   A,SIOMODE
LD   D,STOPBITS
LD   E,PARITY
LD   H,RXBITS
LD   L,TXBITS
CALL XBIOS
DW   SIOINIT
;
LD   H,RXBAUD
LD   L,TXBAUD
CALL XBIOS
DW   SIOBAUD
;
;- KEYSET -
;
LD   B,UPCODE
LD   C,UPKEY
LD   D,KEYSTAT
CALL XBIOS
DW   KEYSET
LD   B,DWNCODE
LD   C,DWNKEY
LD   D,KEYSTAT
CALL XBIOS
DW   KEYSET
LD   B,RICODE
LD   C,RIKEY
LD   D,KEYSTAT
CALL XBIOS
DW   KEYSET
LD   B,LECODE
LD   C,LEKEY
LD   D,KEYSTAT
CALL XBIOS
DW   KEYSET
;
;- ENDE -
;
JP CCP
;
; - DISKTABELLE -
;
TABLE:
;
DW 320AH
DW 1EAFH
DB STEPRATE
DW 030FH
;
;
END
;
; Erweiterungen wie z. B.
; Drucker/Screeneinstellg.
; je nach Bedarf möglich...
;
; Hier abgedruckte Werte:
; Cursortasten: Wordstar
; Schnittstelle: 'Mailbox'
; Tastaturgeschw.: standard
; Steprate: 8 ms
```

wollen Sie Ihre Cursortasten immer gerne Wordstar-kompatibel haben. Also immer mit SETSIO und SETKEYS operieren? Das wären 5 KByte auf der Diskette. In einem COM-File mit XBIOS-Verwendung wäre das ganze mit einem KByte zu erledigen! Als Zugabe könnten dann noch solche Werte wie Steprate oder Tastaturrepeat verändert werden, ohne daß es platzmäßig auffällt. Hier das Beispiel, welches obengenannte Tätigkeiten ausführt sowohl als Assembler-Listing als auch als BASIC-Lader. Ändern Sie die Werte nach Ihren Bedürfnissen. Nach dem Assemblieren haben Sie ein File namens DEFAULTS.COM auf Ihrer Diskette, welches die gewählten Parameter nach dem Aufruf unter CP/(M einstellt: Dies ist natürlich nur ein Grobgerüst. Man könnte auch noch mehr Tasten definieren, Drucker und Bildschirm initialisieren (schauen Sie sich hierzu doch mal die BDOS-Funktionen 111/112 an) und...und ...und! Aber als Anregung sollte das Listing hier genügen, und wenn Sie erstmal auf den Geschmack gekommen sind, werden Sie sicherlich alle Ihre PROFILE-Dateien etwas anpassen und für jede Anwendung ein individuelles DEFAULTS-File erstellen.

Aber nun weiter mit der Vorstellung der XBIOS-Funktionen. Die folgenden vier Funktionen sind nicht unbedingt lebenswichtig, aber der Vollständigkeit halber seien sie hier genannt.

Informationen

XBIOS #33 CD_VERSION 00E3

Nach dem Aufruf dieser Funktion stehen folgende Informationen in den Registern bereit:

A : Rechner:
$00 = CPC 6128
$01 = JOYCE

BC : BIOS-Versionsnummer

HL : rechnerabhängig:
CPC 6128: ROM-Nummer
JOYCE : (nicht definiert)

XBIOS #34 CD_INFO 00E6

Nach dem Aufruf dieser Funktion stehen folgende Informationen in den Registern bereit:

A : Anzahl der Laufwerke:
$00 = 1
$FF = 2

B : Anzahl der Speicherblöcke

C : serielle Schnittstelle:
$00 = nicht vorhanden
$FF = vorhanden

HL : Adresse der Puffertabelle im COMMON Bereich

Format der Tabelle: /Eintrag1/.../EintragN/ $FF

Format der Einträge: byte 0 = CP/M Bank

byte 1,2 = Startadresse
byte 3,4 = Länge in Bytes

Die Tabelle gibt Aufschluß über die Lage der Daten- und Directory-Puffer.

Status

XBIOS #23TE_STL_ASK00C5

Mit dieser Funktion kann überprüft werden, ob die Statuszeile eingeschaltet ist. Wenn dies der Fall ist, ist nach dem Aufruf das Zero-Flag auf 0 gesetzt, ist sie ausgeschaltet, ist das Zero-Flag 1. (Bei diesem Aufruf bleiben alle Register außer A erhalten.)

XBIOS #24
TE_STL_ON_OFF 00C8

Mit dieser Funktion kann die Statuszeile ein- oder ausgeschaltet werden. Der gewünschte Zustand wird über das Register A übergeben:

A : neuer Zustand der Statuszeile:
$00 = ausgeschaltet
$FF = eingeschaltet

Wenn die Statuszeile ausgeschaltet ist, ist der Bildschirm 32 Zeilen groß und die Fehlermeldungen erscheinen nicht in der untersten Zeile, sondern mitten auf dem Schirm.

- Etwas fürs Auge - 5. Der Bildschirm

Hier werden wir uns den Bildschirm und alles, was so dazu gehört, genauer betrachten. Neben einigen für den JOYCE eher unwichtigen Routinen werden wir untersuchen, wie man direkt auf den Bildschirmspeicher und die Buchstabenmatrix zugreifen kann. Krönung dieses Teils wird ein Unterprogramm in Mallard Basic sein, das den Befehl SYMBOL der CPCRechner auch auf dem JOYCE installiert.

Zustände...

Zunächst einige weitere XBIOS-Funktionen, die den Programmierer nicht unbedingt vom Hocker reißen, in manchen Fällen jedoch hilfreich sein können.

XBIOS #21 TE_ASK 00BF

Nach dem parameterlosen Aufruf dieser Funktion stehen folgende Informationen über den Zustand des Bildschirms bereit:

B : oberste Reihe des Darstellungsfensters
C : linke Spalte des Darstellungsfensters
D : Höhe des Fensters-1
E : Breite des Fensters-1
H : Cursor-Position (Reihe)
L : Cursor-Position (Spalte)
(Die Cursor-Position bezieht sich auf das aktuelle Fenster, die Angaben der Fenster-Koordinaten auf den gesamten physikalischen Bildschirm (90 x 32), die Zählung beginnt jeweils bei 0)

XBIOS #22 TE_RESET 00C2

Diese Funktion wird ohne Parameter aufgerufen und liefert keine Parameter zurück, versetzt aber den Bildschirm in einen definierten Zustand: der Bildschirm wird gelöscht; der Cursor eingeschaltet und links oben positioniert. Dieser Befehl ist vor allem dann notwendig, wenn durch Direktzugriff auf die entsprechenden Speicherbereiche die normale Ansteuerung

Listing: Basic-Lader

```
10 ' ****************************************
20 ' *   D E F A U L T S  -  BASIC LADER    *
30 ' *    erzeugt File DEFAULTS.COM mit     *
40 ' *       den ausgewählten Werten.       *
50 ' *           1986 by M.Anton            *
60 ' ****************************************
70 '
80 ' Daten für MCode, an den angegebenen Stellen anpassen:
90 '
100 DATA &H21,&H63,&H01,&H11,&H00,&HC0,&H01,&H07,&H00,&HED
110 DATA &HB0,&H21,&H00,&HC0,&HCD,&H5A,&HFC,&H83,&H00,&H26
120 '
130 DATA &H1E : ' <- Tastatur-Delay
140 DATA &H2E
150 DATA &H02 : ' <- Tastatur-Repeat
160 '
170 DATA &HCD,&H5A,&HFC,&HE0,&H00,&H3E
180 '
190 DATA &H00 : ' <- SIO-Modus
200 DATA &H16
210 DATA &H01 : ' <- Stopbits
220 DATA &H1E
230 DATA &H00 : ' <- Parität
240 DATA &H26
250 DATA &H07 : ' <- Empfangs-Bits
260 DATA &H2E
270 DATA &H07 : ' <- Sende-Bits
280 '
290 DATA &HCD,&H5A,&HFC,&HB6,&H00,&H26
300 '
310 DATA &H06 : ' <- Emfangs-Baudrate
320 DATA &H2E
330 DATA &H06 : ' <- Sende-Baudrate
340 '
350 DATA &HCD,&H5A,&HFC,&HB9,&H00,&H06
360 '
370 DATA &H05 : ' <- Code Cursor hoch
380 '
390 DATA &H0E,&H0E,&H16,&HFF,&HCD,&H5A,&HFC,&HD7,&H00,&H06
400 '
410 DATA &H18 : ' <- Code Cursor runter
420 '
430 DATA &H0E,&H4F,&H16,&HFF,&HCD,&H5A,&HFC,&HD7,&H00,&H06
440 '
450 DATA &H04 : ' <- Code Cursor rechts
460 '
470 DATA &H0E,&H06,&H16,&HFF
480 DATA &HCD,&H5A,&HFC,&HD7,&H00,&H06
490 '
500 DATA &H13 : ' <- Code Cursor links
510 '
520 DATA &H0E,&H0F,&H16,&HFF,&HCD,&H5A,&HFC,&HD7,&H00
530 DATA &HC3,&H00,&H00,&H0A,&H32,&HAF,&H1E
540 '
550 DATA &H08 : ' <- Floppy-Steprate
560 '
570 DATA &H0F,&H03
580 '
590 '
600 ' Ladeprogramm
610 '
620 RESTORE
630 OPEN "O",1,"DEFAULTS.COM"
640   FOR i=1 TO 106
650     READ a : PRINT #1,CHR$(a);
660   NEXT i
670 CLOSE 1
680 END
```

umgangen wurde (der Funktionsaufruf zerstört alle Standardregister.)

Farben...

Von den nächsten drei Funktionen hat der JOYCE-Anwender absolut nichts, da der JOYCE ja nur zwei Farben hat. CPC-Anwender können damit wohl einige Farbzaubereien veranstalten, sind aber auf eigene Experimente angewiesen denn an dieser Stelle wird auf Beispiele verzichtet...

XBIOS #25TE_SET_INK00CB

Mit dieser Funktion werden die Farben ausgewählt, die eine »ink« darstellen (CPC-Anwender werden wissen, was gemeint ist, mir ist es mangels Erfahrungen nicht ganz so klar...). Wenn die beiden Farben gleich sind, bleibt die »ink« konstant, ansonsten wechselt sie zwischen den beiden Farben. Die zu übergebenden Parameter sind folgende:

A : »ink«-Nummer
B : erste Farbe
C : zweite Farbe

Die »ink«-Nummer wird mit $01 maskiert, um einen gültigen Wert zu erhalten. Die Farben werden mit $3F maskiert, das Ergebnis wird als 3 Zweibitzahlen behandelt, die die Intensität der Grundfarben darstellen: bit 0,1=blau / bit 2,3=rot / bit 4,5=grün. Wenn »ink 0« ausgewählt wird, ändert sich auch die Umrandung (»border«) in die entsprechende Farbe. Auf dem JOYCE ließe sich hiermit eine Umschaltung nach dem Muster des Dienstprogramms PALETTE.COM erreichen.

XBIOS #26 TE_SET_BORDER 00CE

Diese Funktion wählt die Farben für die Darstellung der Umrandung ein. Die Funktion und Parameterübergabe entsprechen #25, allerdings wird keine »ink«-Nummer übergeben. Auch hier ist auf dem JOYCE nur eine Umschaltung auf inverse Darstellung möglich, diese erfolgt durch ein gesetztes 7. bit in den Farben. Ist bit 6 gesetzt, wird der Bildschirm dunkelgetastet, d.h. der Rand bedeckt den ganzen Schirm. Dieser Effekt ließe sich eventuell einsetzen, um einen Bildaufbau zu verdecken, Experimente in diese Richtung wurden jedoch nicht unternommen.

XBIOS #27TE_SET_SPEED00D1

Mit dieser Funktion wird die Zeit gewählt, die die Farben für ink und border angezeigt werden. Zum Aufruf erhält H die Dauer für Farbe 1, L für Farbe 2. Einheit ist 1/50 Sekunde, voreingestellter Wert ist 10, ein Wert von 0 bedeutet 256.
Auf dem JOYCE blinken nicht die einzelnen Zeichen, sondern der ganze Bildschirm.
Man sieht also, auf dem JOYCE ist mit diesen Funktionen kein Blumentopf zu gewinnen. Für eigene Experimente habe ich auf der DATABOX drei Programme erstellt, die die Funktionen 26, 27 und 22 verwenden. Starten Sie die Programme wie folgt:

'BORDER' → der Bildschirm blinkt langsam / 'FLASHSP' -→ er blinkt schneller / 'NORMVID' → alles wieder klar.

Auf ein Listing an dieser Stelle möchte ich aus Platzgründen verzichten.

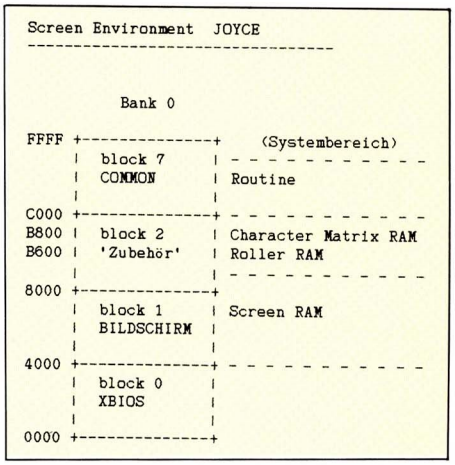
Trostpflaster!!!

Aber wie heißt es so schön: DON'T PANIC. Denn der JOYCE schlägt zurück. Die nächste XBIOS-Routine gibt es nämlich nur exklusiv auf dem JOYCE. Und was noch schöner ist, mit ihr läßt sich sogar einiges anfangen. Also: JOYCE proudly presents:

XBIOS #35 SCR_RUN_ROUTINE 00E9

Diese Routine blendet den Bildschirmspeicher »mit Zubehör« ein und führt ein im COMMON-Bereich abgelegtes Unterprogramm in diesem »Screen Environment« aus. Die Adresse des Unterprogramms wird bei Aufruf der XBIOS-Funktion über das BC-Register übergeben, die übrigen Register können Parameter für das Unterprogramm enthalten. Nach der Ausführung können alle Register außer BC Werte zurückgeben.
Um diese Routine nutzen zu können, muß man zunächst einmal wissen, wie das Screen Environment aussieht. Rein schematisch folgendermaßen:

Screen Environment JOYCE



Der Bildschirm besteht aus 720 x 256 Pixeln, die linke obere Ecke ist 0/0, rechts unten ist 255/719. Zwar ist der Bildschirmspeicher fest im Bereich ab $8000 angesiedelt, die genaue Position einer Pixelzeile ist jedoch variabel. Sie wird in einem speziellen Speicher, dem Roller-RAM gespeichert. Auf diese Weise ist zwar ein sehr schnelles Scrolling möglich, denn es brauchen ja nicht die Pixelreihen sondern nur ihre Adressen bewegt werden, für einen direkten Zugriff auf einzelne Pixel ist diese Organisation jedoch denkbar ungeeignet, da zunächst der entsprechende Bereich über das Roller-RAM gesucht werden muß. Erschwerend wirkt auch noch, daß das Roller-RAM, welches ab der Adresse $B600 beginnt, und die entsprechenden Pixelzeilen etwas komisch organisiert sind. Da sich dieser Bereich bislang meinen Experimenten entzogen hat, kann ich hier die Struktur nur so schildern, wie sie in der Originaldokumentation beschrieben wird und überlasse Experimente dem mutigen Leser:

Struktur des Roller-RAM: (Beginn: $B600)
byte 0,1: Adresse von pixel row 0
byte 2, Adresse von pixel row 1
. . .
byte 510,511: Adresse von pixel row 255.

Struktur der pixel rows:
byte 0: pixel columns 0 – 7
byte 8: pixel columns 8 – 15

. . .
byte 712: pixel columns 712 – 719

Struktur der pixel column bytes:
bit 0: pixel column 7
bit 1: pixel column 6
. . .
bit 7: pixel column 0
(Alles klar? Mir nicht unbedingt, aber vielleicht kann hier jemand anderes Klarheit schaffen...)

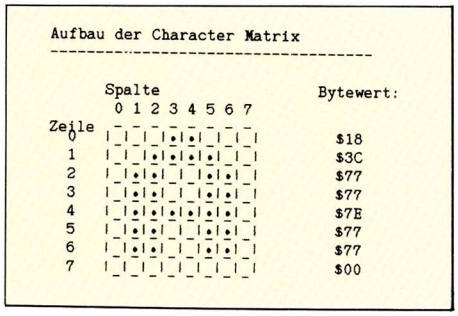
Zeichensatz

Wenden wir uns etwas übersichtlicheren Dingen zu. Zum Beispiel dem Character Matrix RAM. Hier steht nämlich die Punktmatrix jedes Bildschirmzeichens. Das Character Matrix RAM beginnt ab $B800 und besitzt folgendes Format

Byte 0 – 7 : Zeichen 0
Byte 8 – 15 : Zeichen 1
Byte 2040 – 2047: Zeichen 255

Der Eintrag für jedes Zeichen besteht aus 8 Bytes, die mit ihrem Bitmuster die Pixelzeilen darstellen. Das folgende Bild zeigt die Zeichenmatrix für »A« und die entsprechenden Bytewerte, wie sie ab $BA08, dem Beginn des Buchstabeneintrags stehen:

Aufbau der Character Matrix



Und da diese Matrixtabelle im RAM steht, ist sie folglich auch veränderbar. Mit der XBIOS-Funktion #35 steht Ihrem persönlichen Zeichensatz nichts mehr im Wege.

Symbole...

Unsere ersten Gehversuche im Screen Environment widmen wir also der Zeichenmatrix. Das folgende BASIC-Programm ermöglicht die Manipulation aller 256 Zeichen des JOYCE über einfache DATA-Zeilen. In diesen Zeilen wird zunächst der ASCII-Code des zu verändernden Zeichens abgelegt, danach die 8 Bytes, die die einzelnen Zeilen darstellen. Bewirkt wird die Umwandlung durch ein kleines Maschinenprogramm, welches die 8 Bytes über die XBIOS-Funktion #35 an die entsprechende Stelle im Matrix-RAM verschiebt. Beendet wird die Umwandlung, sobald ein ungültiger ASCII-Code angetroffen wird. Danach geht es mit dem Hauptprogramm und dem geänderten Zeichensatz weiter. Hier das Listing:

SYMBOLS

```
SYMBOLS
-------
10 '************************************************
20 '*     SYMBOLS: Veränderung der Zeichensatz-     *
30 '*     matrix des JOYCE unter Mallard Basic.     *
40 '*                                               *
50 '*                                               *
60 '*               1986 by M.Anton                 *
70 '************************************************
80 '
90 '
100 '
110 GOSUB 58000 : ' MCode initialisieren
120 '
130 GOSUB 50000 : ' Zeichensatz ändern
140 '
150 ' - Hier beginnt das Hauptprogramm -
160 '
200 PRINT CHR$(27)+"E"+CHR$(27)+"H"+CHR$(27)+"f"
210 PRINT : PRINT : PRINT
220 PRINT CHR$(253);
222 FOR i=1 TO 79 : PRINT "-"; : NEXT i
223 PRINT CHR$(13);
225 GOSUB 380
230 FOR i= 1 TO 80
240 PRINT CHR$(8)+CHR$(252);
245 GOSUB 380
250 PRINT CHR$(8)+CHR$(32)+CHR$(253);
255 GOSUB 380
260 NEXT i
270 PRINT : PRINT : PRINT
280 PRINT CHR$(255);
290 GOSUB 380
300 FOR i=1 TO 80
310 PRINT CHR$(8)+CHR$(254);
315 GOSUB 380
320 PRINT CHR$(8)+CHR$(32)+CHR$(255);
325 GOSUB 380
330 NEXT i
365 PRINT CHR$(27)+"e"
370 END
380 FOR j=1 TO 300 : NEXT j : RETURN
47998 '
47999 ' - Endgültiges Ende des Hauptprogramms -
48000 END
49000 '
49100 '
49200 '------------------------------------------------
49300 '
49400 ' - SYMBOLS -
49500 '
50000 RESTORE 50200
```

```
50010 READ Zeichencode
50020 IF Zeichencode<0 OR Zeichencode>255 THEN RETURN
50030 FOR i=0 TO 7
50040   READ x : POKE Tabstart+i,x
50050 NEXT i
50060 Charstart=Zeichencode*8+&HB800+2^16
50070 Highaddr=INT(Charstart/256)
50080 Lowaddr=Charstart-Highaddr*256
50090 POKE &HF510,Lowaddr
50100 POKE &HF511,Highaddr
50110 CALL Codestart
50120 GOTO 50010
50130 '
50140 ' - Daten für Zeichen im Format: -
50150 ' - ASCII-Code, 8 Reihen-Bytes   -
50160 ' - Abbruch wenn Zeichencode     -
50170 ' - ungleich 0..255              -
50180 '
50200 DATA 255,0,0,0,0,0,0,255,0
50210 DATA 254,0,0,0,8,20,20,55,0
50220 DATA 253,&H1C,&H3E,&H7B,&H7F,&H7F,&H7F,&H3E,&H1C
50230 DATA 252,&H1C,&H3E,&H76,&H7C,&H78,&H7C,&H3E,&H1C
50240 DATA -1
57000 '
57100 '------------------------------------------------
57200 '
57300 ' - INIT -
57400 '
58000 MEMORY &HF4FF
58010 RESTORE 58110
58020 Codestart=&HF500
58030 Tabstart=&HF515
58040 FOR i=0 TO 20
58050   READ x : POKE Codestart+i,x
58060 NEXT i
58070 RETURN
58080 '
58090 ' Data für MCode
58100 '                    ORG F500
58110 DATA &H01,&H09,&HF5 : '         LD BC,Routine
58120 DATA &HCD,&H5A,&HFC : '         CALL XBIOS
58130 DATA &HE9,&H00      : '         DW 00E9
58140 DATA &HC9           : '         RET
58150 DATA &H21,&H15,&HF5 : 'Routine: LD HL,Tabstart
58160 DATA &H01,&H08,&H00 : '         LD BC,0008
58170 DATA &H11,&H00,&H00 : '         LD DE,Charstart
58180 DATA &HED,&HB0      : '         LDIR
58190 DATA &HC9           : '         RET
58200 '                    Tabstart: /8 Rowbytes/
58210 ' XBIOS 00E9 ruft "Routine" auf, "Routine"
58220 ' kopiert 8 Bytes ab Tabstart in das
58230 ' Character-Matrix-RAM
```

Das integrierte Demoprogramm definiert vier Zeichen um und bewegt sie dann über den Bildschirm.

Mit Hilfe der beiden Unterprogramme können Sie von jetzt an Ihren Programmen die ganz persönliche Note geben, allerdings nur auf dem Bildschirm, die Druckermatrix ändert sich durch diese Manipulationen nicht. Die geänderten Zeichen bleiben solange erhalten, bis Sie den Rechner ausschalten. Mit dem Programm ORIGINAL.COM auf der DATABOX können Sie jedoch jederzeit unter CP/M den Originalzeichensatz wiederherstellen.

- Weiche Scheiben - 6. Die Floppy (1)

Die nächsten beiden Teile werden der JOYCE-Floppy genauer auf den Zahn fühlen – oder besser den Schreib-/Lesekopf, denn Zähne hat sie ja keine (was auch sicherlich den Disketten sehr schlecht bekommen würde).

Zustände

Die ersten beiden Funktionen sind jene, die für den Floppy-Speeder verwendet wurden; ein Beispiel gibt es also schon. Sie ermöglichen die Manipulation des Floppy-Timings, welches für alle Laufwerke gültig ist.

XBIOS #0 DD_INIT 0080

Diese Funktion initialisiert den Disk-Treiber und stellt alle Werte auf die Standardwerte. Es werden keine Parameter übergeben, die Standardregister sind nach dem Aufruf zerstört. Die Standardwerte sind:

Motor ein-Verzögerung: 1 Sekunde
Motor aus-Verzögerung: 5 Sekunden
'write current off': 1.75 ms
'head settle time': 30 ms
Steprate: 12 ms
'head load time': 4 ms
'head unload time: 480 ms
non-DMA Modus

XBIOS #1 DD_SETUP 0083

Mit dieser Funktion können obige Werte verändert werden. Zum Aufruf muß eine Tabelle mit den neuen Parametern im COMMON-Bereich bereitstehen, deren Adresse über das Register HL übergeben wird. Nach dem Aufruf sind alle Standardregister zerstört.
Format der Tabelle:
byte 1: Motor ein-Verzögerung, Einheit 100 ms
byte 2: Motor aus-Verzögerung, Einheit 100 ms
byte 3: write current off-time, Einheit 10 ms
byte 4: head settle time, Einheit 1 ms
byte 5: Steprate, Einheit 1 ms
byte 6: head unload delay, Einheit 32 ms
byte 7: bits 7-1 head load delay, bit 0: non-DMA

Mit der folgenden Funktion kann der Zustand eines Laufwerks abgefragt werden:

XBIOS #8
DD_DRIVE_STATUS 0098

Vor dem Aufruf müssen im Register C die Bits für Laufwerksnummer und Kopfnummer gesetzt werden:

bit 0 = Laufwerk A:
bit 1 = Laufwerk B:
bit 2 = Kopf (für zweiseitige Laufwerke)

Nach dem Aufruf enthält A den Status:
bit 7 : (nicht definiert)
bit 6 : schreibgeschützt
bit 5 : ready
bit 4 : Track 0
bit 3 : (nicht definiert)
bit 2 : Kopf
bit 1,0 : Laufwerk
Wenn für Laufwerk B: bit 6 und bit 5 »0« sind, so ist das Laufwerk nicht vorhanden.
Die nächsten drei Funktionen ermöglichen eine direkte Steuerung des Floppy-Motors:

XBIOS #12
DD_L_ON_MOTOR 00A4

Diese Funktion schaltet den Floppy-Motor nach Ablauf der Verzögerungszeit ein. Parameter werden keine übergeben, alle Register bleiben erhalten.

XBIOS #13
DD_L_T_OFF_MOTOR 00A7

Diese Funktion startet die Abschaltverzögerung. Nach dem Ablauf der eingestellten Zeit schaltet der Motor aus. (Die Routine kehrt jedoch schon vor Ablauf dieser Zeit wieder zurück. Wenn inzwischen der zuständige Programmteil überschrieben wird, bleibt der Motor ständig an – siehe FAIRLIGHT!!!)
Es sind keine Parameter nötig, alle Register bleiben erhalten.

XBIOS #14
DD_L_OFF_MOTOR 00AA

Mit dieser Funktion wird der Motor sofort abgeschaltet. Die Standardregister werden zerstört.

Direkt auf die Scheibe...

Die nächsten drei XBIOS-Funktionen ermöglichen den direkten Zugriff auf Diskettensektoren:

XBIOS #2
DD_READ_SECTOR 0086

Diese Funktion liest einen Sektor in den Speicher. Vor dem Aufruf müssen folgende Parameter in den Registern stehen:

B : CP/M Bank
C : Laufwerk
D : logischer Track
E : logischer Sektor
HL : Adresse des Puffers in Bank (B)
IX : Adresse des XDPB im COMMON-Bereich
(XDPB = eXtended Disk Parameter Block, wird im nächsten Abschnitt behandelt.)

Wenn die Operation erfolgreich verläuft, wird das CARRY-Flag gesetzt. Bei Fehlern ist das CARRY-Flag nicht gesetzt und A enthält den Fehlergrund. Vor der Ausgabe einer Fehlermeldung wird zunächst ein neuer Versuch gestartet.

XBIOS #3
DD_WRITE_SECTOR 0089

Diese Funktion schreibt den Pufferinhalt (Adresse in HL, Bank in B) in einen Diskettensektor. Die Parameter und Ergebnisse entsprechen denen in #2.

XBIOS #4
DD_CHECK_SECTOR 008C

Diese Funktion vergleicht den Inhalt eines Puffers mit einem Sektor auf

Diskette. Die zu übergebenden Parameter entsprechen den Funktionen #2 und #3.

Das Ergebnis wird folgendermaßen dargestellt: Trat kein Fehler auf und die Bereiche sind identisch, sind CARRY und ZERO-Flag gesetzt. Sind die beiden Bereiche nicht identisch, ist nur das CARRY-Flag gesetzt. Trat ein Diskettenfehler auf, sind weder CARRY- noch ZERO-Flag gesetzt und A enthält den Fehlergrund.

Alle drei Funktionen zerstören die Standardregister, die Indexregister bleiben jedoch erhalten.

Statusbyte bei Floppyfehlern

```
Statusbyte bei Floppy-Fehlern
-----------------------------


Fehler -> CARRY-Flag nicht gesetzt.
Grund:

Wert in A   |  Bedeutung
-------------------------------------------
$00| Laufwerk nicht bereit
$02| Suche erfolglos
$03| Datenfehler
$04| keine Daten
$05| fehlende 'data adress mark'
$08| Diskette gewechselt (= Sektoren-
|  nummern stimmen nicht überein.)
```

Verhalten bei Fehlern

```
Verhalten bei Fehlern
---------------------

Bei auftretenden Fehlern versucht die Floppy, nach
folgendem Muster zu einem Erfolg zu kommen:

- 3 Versuche
- Neupositionierung
- 4 Versuche
- innerste Spur anfahren
- 4 Versuche
- Neupositionierung
- 4 Versuche

Erst danach wird eine Fehlermeldung gegeben.
```

Direkte Kontrolle...

Mit den folgenden drei Funktionen ist ein direkter Zugriff auf den Floppycontroller FDC 765 möglich. (Genaue Kenntnis des IC vorausgesetzt):

XBIOS #15
DD__L__READ 00AD

Mit dieser Routine können Befehle wie »READ DATA«, »READ DELETED DATA« und »READ A TRACK« direkt an den Floppycontroller gesendet werden. Vor dem Aufruf muß der Motor laufen (→ #12) und HL die Anfangsadresse einer im COMMON-Bereich gelegenen Parametertabelle enthalten.

Die Parametertabelle hat folgendes Format:

byte 0 : CP/M Bank
byte 1,2 : Pufferadresse
byte 3,4 : Anzahl zu übertragender Bytes
byte 5 : Anzahl der FDC 765-Kommandobytes
byte 6...: Kommandobytes

Nach dem Aufruf enthält HL die Adresse des Ergebnis-Puffers im COMMON-Bereich. Der Ergebnispuffer hat folgendes Format:

byte 0 : Anzahl der empfangenen Ergebnisbytes
byte 1...: Ergebnisbytes
(Register AF, BC, DE zerstört.)

XBIOS #16
DD__L__WRITE 00B0

Diese Routine ermöglicht das direkte Senden der Befehle »WRITE DATA«, »WRITE DELETED DATA«, »FORMAT A TRACK«, »SCAN...« an den Floppycontroller. Die übergebenen und empfangenen Parameter entsprechen denen der Funktion #15.

XBIOS #17
DD__L__SEEK 00B3

Diese Funktion fährt eine bestimmte Spur auf der Diskette an. Vor dem Aufruf muß der Motor angeschaltet werden und folgende Werte bereitstehen:

C : bits 0,1 = Laufwerk
bit 2 = Kopfnummer
D : physikalische Spur
IX : Adresse des extended DPB im COMMON-Bereich

Verlief die Operation erfolgreich, so wird das CARRY-Flag gesetzt. Ansonsten ist CARRY nicht gesetzt und A enthält den Fehlergrund (0 = Laufwerk nicht bereit, 2 = nicht gefunden). Vor der Ausgabe eines Fehlers wird nach Bild 2 verfahren. Das ZERO-Flag wird auf alle Fälle gesetzt, die Register B und IY werden zerstört.

Für die XBIOS-Funktionen #2–5 (#5 siehe nächster Abschnitt) ist diese Funktion nicht notwendig, da sie die entsprechende Spur eigenständig anfahren.

- Enorm in Form - 7. Die Floppy (2)

Das Thema dieses Abschnitts sind die unterschiedlichen Diskettenformate, allerdings wiederum nur in der Theorie. Für die Praxis könnten (theoretisch) folgende Anwendungen dabei herauskommen: Nutzbarmachung der Spuren 41 – 43 auch für den JOYCE sowie Kopierschutz und Kopierprogramme. In Verbindung mit einer 5,25"-Zoll Floppy ergäbe sich auch die interessante Möglichkeit eines Programms zur Verarbeitung verschiedenster Diskformate, was aber alles nicht an dieser Stelle geschehen soll (...und kann).

Formate

Mancher JOYCE-Benutzer wird sich schon gewundert haben, daß einige (kommerzielle) Programme wie z.B. DR DRAW u.a. in einem Format geliefert werden, das der JOYCE zwar lesen und schreiben kann und welches 178 KByte aufnehmen kann, vom Dienstprogramm DISCKIT jedoch freundlich zurückgewiesen wird. Dann sind ihm vielleicht auch schon Disketten untergekommen, auf die beim besten Willen nur 169 KByte passen wollten. Die Erklärung des Phänomens: Es handelt sich dabei um Disketten, die mit den CPC-Formaten »DATA ONLY« (178 KB) und »SYSTEM« (169 KB) formatiert wurden. Lesen und schreiben kann der JOYCE diese zwar, selber formatieren kann er sie jedoch nicht – zumindest nicht mit DISCKIT. Schade, da vor allem 178 KByte auf einer Diskette, von der nicht gebootet werden soll, schon sehr praktisch wären... Aber woran erkennt der JOYCE nun, welche Diskette ihm gerade untergejubelt wurde? – Ganz einfach: Auf Spur 1, Sektor 0 jeder Diskette stehen Informationen über die Struktur der Diskette, die einfach eingelesen werden:

byte 0 : Diskettentyp:
$00 = Standard-JOYCE
$01 = DATA ONLY
$02 = SYSTEM
(andere Werte reserviert)
byte 1 : Diskettenart:
(vgl. auch Tabelle 2)

$00 = einseitig
$01 = zweiseitig=flip sides
$02 = zweiseitig=up and over
byte 2 : Tracks pro Seite
byte 3 : Sektoren pro Track
byte 4 : log2 (Sektorengröße)-7
byte 5 : Anzahl reservierter Tracks
byte 6 : log2 (Blockgröße)-7
byte 7 : Anzahl Directoryblocks
byte 8 : gap length (read/write)
byte 9 : gap length (format)

Diese Informationen werden beim Formatieren mit DISCKIT automatisch dorthin geschrieben, im JOYCE-Format folgt ab Byte 16 das Ladeprogramm für das (erstbeste) EMS-Systemfile. Steht an dieser Stelle jedoch nichts (oder »nichts rechtes«), wird die Diskette garnicht erst angenommen.

Double Density-Formate

```
Tabelle: Double Density-Formate
-------------------------------


a: 'Flip Sides':

   Seite 0  Track 0 2 4 6 8 ... 158
            ------------------------
   Seite 1  Track 1 3 5 7 9 ... 159


b: 'Up and Over':

   Seite 0  Track 0 1 2 3 4 5 ... 79
            ------------------------
   Seite 1  Track 159 158 157 ... 80

(Das JOYCE Zweitlaufwerk verwendet Muster a.)
```

Parameterblöcke

Aus diesen eingelesenen Informationen wird dann ein Disk Parameter Block (DPB) initialisiert, der aus dem Standard-CP/M bekannt sein sollte.

Aber wenn es beim JOYCE schon ein XBIOS gibt, warum sollte da nicht auch noch ein erweiterter Parameterblock existieren? Und in der Tat, es gibt ihn, diese XDPB! Er besteht aus dem Standard-DPB und einigen zusätzlichen Informationen und besitzt folgendes Format (die Werte für die »kleinen« Disketten sind in der Tabelle bereits enthalten).

Durch ein entsprechendes Patchen des XDPB sollte es nun möglich sein, (fast) beliebige Diskettenformate bearbeiten zu können, was jedoch nur in Verbindung mit 5,25-Zoll Floppys so richtig interessant wird.

Funktionen...

XBIOS-Funktionen, die diesen XDPB benötigen, hatten wir bereits in der letzten Folge kennengelernt (#2–4). Und wie kommt man nun an den XDPB ran? Unter CP/M erhält man seine Adresse über die BDOS-Funktion 31, man kann ihn jedoch auch mittels einiger XBIOS-Funktionen erreichen und anlegen.
Doch zuvor noch zwei andere Funktionen, von denen besonders die erstere im Zusammenhang mit verschiedenen Formaten besonders interessant sein dürfte.

XBIOS #5
DD_FORMAT 008F

Mit dieser Funktion ist es möglich, einen Track zu formatieren. Ein XDPB für das entsprechende Laufwerk muß bereits im COMMON-Bereich zur Verfügung stehen, weiterhin muß eine Tabelle mit den Informationen über die Sektoren bereitstehen. Die zu übergebenden Parameter sind folgende:

B : CP/M Bank der Tabelle
C : Laufwerk
D : logische Track-Nr.
E : Füllbyte, normalerweise $E5
HL : Adresse der Tabelle
IX : Adresse des XDPB im COMMON-Bereich

Die Tabelle hat das Format (1. Sektoreintrag... letzter Sektoreintrag). Die Einträge müssen folgendes Format aufweisen:

byte 0 : Track Nr.
byte 1 : Kopf Nr.
byte 2 : Sektor Nr.
byte 3 : log2(Sektorgröße)-7

Die Sektornr. kann zwar eine beliebige Zahl sein, als Standard wird vom BIOS jedoch nur 1 – 9, ggf. mit einem Offset von $40 (SYSTEM) oder $C0 (DATA) akzeptiert. Andere Werte können nur in Verbindung mit eigenen Abfragen (Kopierschutz) verwendet werden. Für einen optimalen Zugriff empfiehlt es sich, im Standard-Formaten (512 Byte-Sektoren) die Sektoren nach folgendem Muster anzuordnen (Hard-skew): 1-6-2-7-3-8-4-9-5
Eine Markierung des Tabellenendes ist nicht nötig, da die Anzahl der Sektoren aus dem XDPB übernommen

```
Tabelle: XDPB-Daten für Single Density Disketten
------------------------------------------------


byte  | Inhalt                      | JOYCE  | SYSTEM | DATA
------------------------------------------------------------
 0,1  | Records pro Spur            |   36   |   36   |   36
  2   | Blockverschiebefaktor       |    3   |    3   |    3
  3   | Blockmaske                  |    7   |    7   |    7
  4   | Extentmaske                 |    0   |    0   |    0
 5,6  | Blockanzahl-1               |  174   |  170   |  179
 7,8  | Directoryeinträge-1         |   63   |   63   |   63
9,10  | Directoryblocks/allocation  | $C000  | $C000  | $C000
11,12 | Größe der Prüfsumme         |   16   |   16   |   16
13,14 | Reservierte Spuren          |    1   |    2   |    0
 15   | Record-Verschiebefaktor     |    2   |    2   |    2
 16   | Record-Verschiebemaske      |    3   |    3   |    3
------------------------------------------------------------
 17   | 'Seitenstruktur' (0-2)      |    0   |    0   |    0
 18   | Spuren pro Seite            |   40   |   40   |   40
 19   | Sektoren pro Spur           |    9   |    9   |    9
 20   | Nummer des ersten Sektors   |    1   |   $41  |   $C1
21,22 | Sektorengröße               |  512   |  512   |  512
 23   | gap length (read/write)     |   42   |   42   |   42
 24   | gap length (format)         |   82   |   82   |   82
 25   | 'Modus'                     |  $60   |  $60   |  $60
 26   | 'freeze flag'               |    0   |    0   |    0


   byte 17:  0 = einseitig
             1 = zweiseitig/flip sides
             2 = zweiseitig/up and over

   byte 25:  bit 7 = multi-track operation:
                     1 - multi-track
                     0 - single track
             bit 6 = Modulationsart:
                     1 - MFM
                     0 - FM
             bit 5 = deleted data adress mark:
                     1 - überspringen
                     0 - nicht überspringen
             bit 4-0 = 0

   byte 26:  freeze flag:
               $00 = automatische Formaterkennung ein
               $FF = automatische Formaterkennung aus
```

wird. Wurde die Operation erfolgreich ausgeführt, wird das CARRY-Flag gesetzt, ansonsten kann der Fehlergrund nach Bild 1 aus A entnommen werden, das Verhalten bei Fehlern Bild 2. Außer IX werden alle Register zerstört.

XBIOS #9 DD_READ_ID 009B

Diese Funktion liest die Identifikation des ersten gefundenen Sektors. Vor dem Aufruf sind folgende Parameter zu übergeben:

C : Laufwerk
D : Track Nr.
IX : Adresse des XDPB im COMMON-Bereich

Verlief die Operation erfolgreich, so enthält A die Sektornummer und das CARRY-Flag ist gesetzt. Ansonsten erfolgt eine Fehlerbehandlung und -darstellung wie in Bild 1 und 2. HL enthält nach dem Aufruf auf alle Fälle die Adresse einer Ergebnistabelle, deren erstes Byte ihre Länge enthält.

XDPBs nach Wahl...

Mit den folgenden Funktionen kann ein (X)DPB initialisiert werden.

XBIOS #6 DD_LOGIN 0092

Mit dieser Funktion wird versucht, das Format einer Diskette zu erkennen und bei Erfolg ein XDPB initialisiert. Dazu sind das Laufwerk über Register C und die (gewünschte) Adresse des XDPB im COMMON-Bereich über IX zu übergeben. Verlief die Operation erfolgreich (CARRY-Flag gesetzt), so wird ein XDPB initialisiert und folgende Informationen werden zurückgegeben:

A : Diskettentyp
  $00 = Standard-JOYCE
  $01 = SYSTEM
  $02 = DATA ONLY
  (andere Werte wie von Diskette gelesen)
DE : Allocation vector
HL : Größe der hash table

Ansonsten erfolgt eine Fehlerbehandlung und -meldung wie beschrieben. Die Register B und C sind auf alle Fälle zerstört.

Übersicht über die XBIOS-Funktionen

Nr.	Name	Funktion	Adresse	Abschn.	Beispiel
0	DD_INIT	Disk-Treiber initialisieren	0080	0,6	UNSPEED
1	DD_SETUP	Disk-Parameter setzen	0083	0,6	SPEEDER
2	DD_READ_SECTOR	Sektor lesen	0086	6	-
3	DD_WRITE_SECTOR	Sektor schreiben	0089	6	-
4	DD_CHECK_SECTOR	Sektor überprüfen	008C	6	-
5	DD_FORMAT	Spur formatieren	008F	7	-
6	DD_LOGIN	Diskette anmelden	0092	7	-
7	DD_SEL_FORMAT	Standardformat wählen	0095	7	-
8	DD_DRIVE_STATUS	Laufwerkszustand abfragen	0098	6	-
9	DD_READ_ID	Sektoren-ID lesen	009B	7	-
10	DD_L_DPB	Standard-DPB initialisieren	009E	7	-
11	DD_L_XDPD	eXtended DPB initialisieren	00A1	7	-
12	DD_L_ON_MOTOR	Floppymotor einschalten	00A4	6	-
13	DD_L_T_OFF_MOTOR	Motor mit Verzögerung aus	00A7	6	-
14	DD_L_OFF_MOTOR	Motor sofort aus	00AA	6	-
15	DD_L_READ	FDC 765-READ Kommandos	00AD	6	-
16	DD_L_WRITE	FDC 765-WRITE Kommandos	00B0	6	-
17	DD_L_SEEK	Spur anfahren	00B3	6	-
18	CD_SA_INIT	SIO-Parameter setzen	00B6	2	SIO_PARA
19	CD_SA_BAUD	Baudrate setzen	00B9	2	SIO_PARA
20	CD_SA_PARAMS	Parameter abfragen	00BC	2	-
21	TE_ASK	Cursorposition abfragen	00BF	4	-
22	TE_RESET	Bildschirm zurücksetzen	00C2	4	(NORMVID)
23	TE_STL_ASK	Statuszeile an?	00C5	3	-
24	TE_STL_ON_OFF	Statuszeile schalten	00C8	3	-
25	TE_SET_INK	ink-Farbe setzen	00CB	4	-
26	TE_SET_BORDER	border-Farbe setzen	00CE	4	(BORDER)
27	TE_SET_SPEED	Blinkgeschwindigkeit	00D1	4	(FLASHSP)
28	KM_SET_EXPAND	Expansion-Text bestimmen	00D4	2	EXPAND
29	KM_SET_KEY	Tastaturbelegung ändern	00D7	2	KEYSET
30	KM_KT_GET	Tastaturpuffer lesen	00DA	2	KEYSCAN
31	KM_KT_PUT	Tastaturpuffer schreiben	00DD	2	KEYSCAN
32	KM_SET_SPEED	Tastaturgeschwindigkeit	00E0	2	SID-Bsp.
33	CD_VERSION	Versionsnr. abfragen	00E3	3	-
34	CD_INFO	Systeminformationen	00E6	3	-
35	SCR_RUN_ROUTINE	Bildschirmspeicher einbelenden (*)	00E9	4	SYMBOLS

(beispiel): nur auf DATABOX (*): nur auf dem JOYCE

XBIOS #7 DD_SEL_FORMAT 0095

Diese Routine initialisiert einen XDPB für Standard-Formate ohne Rücksicht auf das tatsächliche DiskFormat (verwendbar z.B. zur Voreinstellung bei Formatierprogrammen). Der gewünschte Diskettentyp (0 – 2, siehe #6) ist über A, die Adresse des XDPB über IX zu übergeben. Bei Erfolg werden die Informationen wie bei #6 zurückgegeben, ansonsten erfolgt eine Fehlermeldung über das nicht ge-

Floppyparameter im Systemfile

Version	Filename	Start-adr.	Steprate-adresse	Speichern von $0100 bis
CP/M v 1.2	J12DCPM3.EMS	$0C50	$0C54	$A0FF
CP/M v 1.4	J14GCPM3.EMS	$0D40	$0D44	$A0FF
LocoScript 2.1	J21GLOCO.EMS	$5220	$5224	$B0FF

setzte CARRY-Flag und A, wobei hier nur die Meldung #06 (falsches Format) auftritt, da nicht auf die Diskette zugegriffen wird.

XBIOS #10 DD_L_DPB 009E

Diese Routine initialisiert einen Standard-DPB aus einer Spezifikationstabelle. Dazu müssen über HL die Adresse der Tabelle und über IX die Adresse des DPB (beide im COMMON-Bereich) übergeben werden.

Die Tabelle hat folgendes Format:

byte 0 : Diskettentyp
byte 1 : Diskettenart
byte 2 : Tracks/Seite
byte 3 : Sektoren/Track
byte 4 : log2(Sektorengröße)-7
byte 5 : Anzahl reservierter Spuren
byte 6 : log2(Blockgröße)-7
byte 7 : Anzahl Directoryblocks
byte 8 : gap length (read/write)
byte 9 : gap length (format)

Bei Erfolg wird das CARRY-Flag gesetzt und A gibt den Diskettentyp zurück, ansonsten wird die Meldung #06 (falsches Format) zurückgegeben.

Die Standard-Register außer IX werden zerstört.

XBIOS #11 DD_L_XDPB 00A1

Diese Routine entspricht #10, allerdings wird mit ihr ein XDPB initialisiert.

Hiermit wäre das Kapitel Floppy abgehakt. Ich hoffe, die Floppy-Freaks können mit diesen Hinweisen etwas sinnvolles anfangen (Erfolgsmeldungen in Form von Programmen sind erwünscht...), ich konnte es leider bislang noch nicht.

- Steprate dauerhaft - 8. Das dicke Ende

Es ist vollbracht...wir sind am Ende unserer XBIOS-Reise angelangt. Jetzt fehlt eigentlich nur noch eine Karte unserer Reiseroute – und die Rückkehr an den Ausgangspunkt! Die folgende Tabelle zeigt nochmals alle besprochenen Routinen mit Name, Funktion und Adresse sowie die Folge, in der die Routine besprochen wurde und welche Beispiele gegeben wurden:

Kreisbewegung

Erinnern Sie sich eigentlich noch, wie der Artikel angefangen hat? Da war doch irgendwas mit der FloppySteprate oder so... Richtig. Und um das Maß vollzumachen – bzw. den Kreis rund – verändern wir nochmals die Steprate. Diesmal allerdings etwas anders – und dauerhaft!

Folgendes Gedankenexperiment: Wenn mit der XBIOS-Funktion #0 Standardwerte für das Floppy-Timing eingestellt werden können, so müssen diese ja irgendwo zu finden sein. Und was liegt näher, als im CP/M-Systemfile danach zu suchen... Da sich die Tabelle nicht freiwillig meldet, muß halt mit einem Debugger gesucht werden. Erinnern wir uns: die Standardtabelle besteht aus der Bytefolge '0A 32 AF 1E 0C 0F 03'. Wenn uns nun diese Folge irgendwo begenet, kann es sich ja fast nur um diese Tabelle handeln, deren 5. Byte ($0C) die Steprate ist. Wenn wir dieses Byte im EMS-File ändern und die geänderte Version abspeichern, dann wird bei jedem Booten die neue Steprate eingestellt.

Und zur allgemeinen Freude steht die Tabelle tatsächlich in den EMS-Files, wie das folgende Bild zeigt:

Ändern Sie also mittels SID oder DDT die jeweilige Stelle in Ihrem EMSFile (möglichst nicht auf der Originaldiskette) von $0C in Ihre gewünschte Steprate und speichern Sie das geänderte File auf Ihre neue Boot-Diskette. Beim Booten von dieser Diskette haben Sie dann immer die angepaßte Steprate automatisch verfügbar.

Vorsicht Falle...

Das LocoScript EMS-File muß mit 'SET J21*.EMSÄSYSÜ' erst noch unsichtbar gemacht werden, wenn es nicht in der Diskettenverwaltung von LocoScript auftauchen soll. Unter LocoScript läßt sich die Steprate nicht mehr ändern, es sei denn eine ungepatchte Version wird gebootet. Unter CP/M läßt sich die Steprate jedoch beliebig zurücksetzen; allerdings nicht mehr mit dem »alten« UNSPEED. Die in UNSPEED verwendete XBIOS-Funktion würde zur Einstellung der Standardwerte genau jene Tabelle verwenden, die wir gepatcht haben, und das bringt ja nichts. Stattdessen muß mit Funktion #1 (vgl. SPEEDER) mit dem Normalwert $0C als Steprate verwendet werden!

Damit aber endgültig genug in Sachen Steprate und XBIOS – zumindest von meiner Seite aus. Von Ihnen sollten nun Anwendungen und Tips kommen...

Zugrundeliegende Literatur:

CP/M on the PCW8265, issue 1 draft 2 / 14-6-85
Copyrighted 1985 by Locomotive Software Ltd.
Mit freundlicher Genehmigung der Schneider Rundfunkwerke AG, Türkheim.

Ausblick

Im Anschluß an diesen wahrhaft beinharten Artikel finden Sie zwei Anwendungen, die Ihnen auf der Basis des XBIOS die Tür zum Inneren des Joyce öffnen helfen sollen.

- Klare Verhältnisse - Wiederherstellung der Standard-Tastatur

Wer kennt nicht das Problem: Nach dem Arbeiten mit Programmen, die die Tastaturbelegung ändern, wie beispielsweise dBase, läuft eigentlich so gut wie nichts mehr, wenn man beispielsweise mit dem BASIC-Editor RPED weiterarbeiten möchte. Hier schafft das Programm RESET wirkungsvolle Abhilfe.

Nach einer Änderung der Tastaturbelegung mittels SETKEYS laufen Programme, die die Standardtastatur voraussetzen, nur noch eingeschränkt oder überhaupt nicht. Kein Wunder, denn mit den Cursorkommandos von dBase oder Wordstar kann das Mallard-BASIC relativ wenig anfangen.

Es gibt zwei Möglichkeiten zur Abhilfe: Man könnte natürlich CP/M neu

booten. Aber das dauert erstens seine Zeit und zweitens geht dabei der eventuell noch benötigte Inhalt der RAM-Disk verloren. Dann könnte man sich auch noch mit einem erneuten SETKEYS die Ursprungswerte wiederherstellen. Aber das ist auch noch nicht ganz optimal, denn es ist im Vergleich zu der im folgenden vorgestellten dritten Möglichkeit immernoch relativ platz- und zeitaufwendig. Immerhin müssen Sie für die dazu notwendigen Dateien 3 KByte auf der Diskette opfern.

Das folgende Assemblerlisting für M80 von Microsoft erledigt diese Sache mit einem nur 1 KByte langen COM-File. Und als Zugabe wird gleich noch der Bildschirm aufgeräumt. Durch die Ausnutzung der XBIOS-Funktionen läßt sich somit das Zurücksetzen in den Ausgangszustand sehr schnell und ökonomisch erreichen. Statt SETKEYS (dateiname) einfach RESET – und schon ist alles wie frisch nach dem Einschalten. Sämtliche Editierfunktionen sind wieder standardmäßig zu erreichen und der Bildschirm präsentiert sich auch wie gewohnt. Und so sieht Ihr neuer Helfer aus:

RESET.MAC (für M80)

```
Listing: RESET.MAC (für M80)
----------------------------

;*********************************
;* R E S E T -Wiederherstellung *
;* der Standardtastaturbelegung *
;*      1987 by M.Anton         *
;*********************************
;
.Z80
ASEG
;
ORG 0100H
;
XBIOS   EQU 0FC5AH
EXPAND  EQU 00D4H
SET_KEY EQU 00D7H
RESET   EQU 00C2H
BDOS    EQU 0005H
P_BLOCK EQU 111D
;
ANZAHL  EQU 25D
;
;
;- Belegung ändern -
;
LD   HL,TABSTART
PUSH HL
LD   B,ANZAHL
;
LOOP:
POP  HL
LD   A,B
LD   (TEMP),A
LD   C,(HL)
INC  HL
LD   B,(HL)
INC  HL
LD   D,(HL)
INC  HL
PUSH HL
CALL XBIOS
DW   SET_KEY
LD   A,(TEMP)
LD   B,A
DJNZ LOOP
;
;- Expansions ändern -
;
LD   BC,0005H
LD   HL,STRINGS
LD   DE,0C000H
LDIR
;
LD   B,8FH
LD   C,03H
LD   HL,0C000H
CALL XBIOS
DW   EXPAND
LD   B,90H
LD   C,02H
LD   HL,0C003H
CALL XBIOS
DW   EXPAND
;
;- Bildschirmreset -
;
LD   C,P_BLOCK
LD   DE,CCB
CALL BDOS
;
;- Programmende -
;
JP   0000H
;
;
TEMP: DS 1
;
;
;- Tastaturtabelle -
;
TABSTART:
DB 66D,03H,03H
DB 02D,1AH,03H
DB 00D,11H,03H
DB 73D,13H,03H
DB 77D,10H,03H
DB 16D,07H,01H
DB 75D,08H,03H
DB 10D,15H,03H
DB 03D,17H,03H
DB 11D,17H,03H
DB 20D,1DH,03H
DB 14D,1FH,03H
DB 23D,16H,03H
DB 15D,01H,03H
DB 06D,06H,03H
DB 05D,06H,03H
DB 01D,12H,03H
DB 79D,1EH,03H
DB 79D,05H,04H
DB 16D,0BH,04H
DB 76D,1CH,03H
DB 72D,18H,04H
DB 08D,1BH,03H
DB 13D,8FH,01H
DB 13D,90H,02H
;
;- Expansions -
;
STRINGS:
DB 06H,02H,02H
DB 06H,02H
;
;- Resetstring -
;
CCB:
DW START
DW 0011H
START:
DB 1BH,71H,1BH,75H,1BH,65H
DB 1BH,62H,3FH,1BH,79H
DB 1BH,31H,1BH,45H,1BH,48H
;
END
```

Zunächst wird über die entsprechenden XBIOS-Funktionen die Tastatur nach Standard belegt. Abschließend wird ein Resetstring an den Bildschirm gesendet, der die Modi revers, 24 x 80, unterstreichen, auschaltet und den Cursor und die Statuszeile an sowie »Grün auf Schwarz« einstellt und den Bildschirm löscht (das etwas merkwürdige Operieren mit dem Stack und dem Zwischenspeicher TEMP ist deshalb nötig, da einerseits die XBIOS-Funktionen relativ wenig Rücksicht auf die Register nehmen, andererseits wird Register B sowohl zur Parameterübergabe als auch für die DJNZ-Schleife gebraucht).

RESET-Basiclader

```
10 ' ************************
20 ' * RESET-Basiclader     *
30 ' * erzeugt RESET.COM    *
40 ' * auf Bezugslaufwerk   *
50 ' *                      *
60 ' *  1987 by M.Anton     *
70 ' ************************
80 '
90 ' - Daten f. MCode -
95 '
100 DATA &H21,&H4C,&H01,&HE5,&H06,&H19,&HE1,&H78,&H32,&H4B,&H01,&H4E,&H23,&H46,&H23,&H56
110 DATA &H23,&HE5,&HCD,&H5A,&HFC,&HD7,&H00,&H3A,&H4B,&H01,&H47,&H10,&HE9,&H01,&H05,&H00
120 DATA &H21,&H97,&H01,&H11,&H00,&HC0,&HED,&HB0,&H06,&H8F,&H0E,&H03,&H21,&H00,&HC0,&HCD
130 DATA &H5A,&HFC,&HD4,&H00,&H06,&H90,&H0E,&H02,&H21,&H03,&HC0,&HCD,&H5A,&HFC,&HD4,&H00
140 DATA &H0E,&H6F,&H11,&H9C,&H01,&HCD,&H05,&H00,&HC3,&H00,&H00,&H00,&H42,&H03,&H03,&H02
150 DATA &H1A,&H03,&H00,&H11,&H03,&H49,&H13,&H03,&H4D,&H10,&H03,&H10,&H07,&H01,&H4B,&H08
160 DATA &H03,&H0A,&H15,&H03,&H03,&H17,&H03,&H0B,&H17,&H03,&H14,&H1D,&H03,&H0E,&H1F,&H03
170 DATA &H17,&H16,&H03,&H0F,&H01,&H03,&H06,&H06,&H03,&H05,&H06,&H03,&H01,&H12,&H03,&H4F
180 DATA &H1E,&H03,&H4F,&H05,&H04,&H10,&H0B,&H04,&H4C,&H1C,&H03,&H48,&H18,&H04,&H08,&H1B
190 DATA &H03,&H0D,&H8F,&H01,&H0D,&H90,&H02,&H06,&H02,&H02,&H06,&H02,&HA0,&H01,&H11,&H00
200 DATA &H1B,&H71,&H1B,&H75,&H1B,&H65,&H1B,&H62,&H3F,&H1B,&H79,&H1B,&H31,&H1B,&H45,&H1B
210 DATA &H48
230 '
234 ' - Checksum -
236 '
240 RESTORE
250 FOR i=1 TO 177
260   READ a : s=s+a
270 NEXT i
280 IF s<>9194 THEN PRINT "Fehler in DATA-Zeilen (oder Pr)fsumme falsch,,,)" : END
290 '
294 ' - Ladeprogramm -
296 '
300 RESTORE
310 OPEN "O",1,"RESET.COM"
320 FOR i=1 TO 177
330   READ a : PRINT #1,CHR$(a);
340 NEXT i
350 CLOSE 1
360 END
```

An fremden Ufern... - Fremdformate auf dem JOYCE -

Und noch eine nachträgliche Anwendung zum XBIOS-Artikel...wenn man schon einen erweiterten Disk Parameterblock hat, so sollte man ihn wenigstens etwas ausnützen.

»Etwas« insofern, als hier nur eine praktische Anwendung gezeigt wird, die auch noch etwas höheren Hardwareaufwand, nämlich ein 5,25-Zoll Floppy voraussetzt. Aber zusammen mit den übrigen Beispielen der Serie ergibt sich ein weites Experimentierfeld für Floppy-Manipulationen.

Große Scheiben...

Am Anfang war die 3-Zoll Diskette. Und die war teuer. Ebenso war es das Zweitlaufwerk. Technisch Talentierte klemmten also nach Umschiffung aller technischen Probleme ein 5,25-Zoll Zweit- oder Drittlaufwerk an den JOYCE. Das war zwar billiger, der eigentliche Kontakt zur CP/M-Welt, die sich überwiegend auf großen Disketten abspielt, war aber noch nicht vorhanden, da der JOYCE sich beharrlich weigerte, nicht von ihm erstellte Sachen zu akzeptieren. Doch damit ist jetzt Schluß, denn mit LOGIN muß der JOYCE nun »essen, was vorgesetzt wird«.

Das Prinzip

Allerdings wird auch nur so heiß gegessen, wie gekocht wird. Es kann zwar relativ viel verändert werden, in der Praxis wurde jedoch bisher nur das große VORTEX-Format nutzbar gemacht (was immerhin auch schon etwas ist, wenn man beispielsweise an den Public Domain-Sektor denkt). Das Zauberwort heißt eXtended Disk Parameter Block (XDPB), in dem so ziemlich alles über den Aufbau einer Diskette steht. Und wenn man den nun ändert, kann man schon einiges servieren... von wegen, denn normalerweise versucht der JOYCE beim Einloggen einer Diskette deren Format zu bestimmen. Und dabei werden halt leider nur die vier bekannten Formate (JOYCE-Standard, JOYCE-DD, DATA ONLY und SYSTEM) akzeptiert. Man kann diese Formaterkennung ausschalten, und zwar mit dem letzten Byte des XDPB, dem Freeze-Flag (vgl. Abschnitt 7 XBIOS). Ist dieses nämlich $FF, nimmt der JOYCE an, daß die Diskette im Laufwerk ist, die durch den XDPB spezifiziert wird. Das Prinzip ist also einfach: einen dem gewünschten Format entsprechenden XDPB initialisieren und einfrieren. Damit hat sich's. Genau dieses macht LOGIN – und noch etwas mehr. Denn wenn man z.B. wieder mit den Standardwerten arbeiten will, sollte das Freeze-Flag zurückgesetzt werden und...und...und!

Bedienung

Tatsächlich ist der größte Teil von LOGIN Bedienungskomfort, hat also mit dem eigentlichen Anliegen kaum etwas zu tun, zumindest nicht direkt. Bevor Sie nun auf Formatjagd gehen können, müssen Sie natürlich noch etwas arbeiten, denn 7 KByte Turbo Pascal geben sich nicht von allein ein. Aber dann kann es losgehen.

LOGIN wird entweder mit oder ohne Parameter aufgerufen. Ein Aufruf ohne Parameter führt in den Verwaltungsteil. Dort werden die Parameterfiles erstellt oder die aktuellen Belegungen ausgegeben. Dieser Teil ist die erste Anlaufstelle für die Erstellung eines neuen Formats. Geben Sie unter Punkt 3 den neuen XDPB (in Abschnitt 7 beschrieben) ein, der auf Diskette abgespeichert wird. Dieses File wird dann später zum Login eines neuen Formates verwendet.

Beim Aufruf mit Parametern gibt es folgende Optionen:

1 Parameter: entweder Ändern des Freeze-Flag oder Anmeldung eines neuen Formats für Drive B:

'LOGIN FREEZE'
'LOGIN NOFREEZE'

oder

'LOGIN VORTEX'

2 Parameter: Formatanmeldung für ein bestimmtes Laufwerk (A/B):

'LOGIN A SYSTEM'

4 Parameter: Anmeldung zweier neuer Formate:

'LOGIN B VORTEX A DATA'

Dabei ist zu beachten, daß die einzelnen Parameter durch Leerzeichen getrennt werden und daß die angegebenen Parameterfiles wirklich vorhanden sind. Für diese Files wird die Extension ».FRM« vorausgesetzt. Bei den Angaben für das Laufwerk (1./3. Parameter) darf nur »A« oder »B« (ohne Doppelpunkt) verwendet werden, bei den Filenamen (ohne Extension!) kann auch ein Laufwerk angegeben werden:

'LOGIN B M:VORTEX A DATA'

Ungültige Werte werden soweit wie möglich abgefangen (doppelseitige Formate auf A sind unmöglich), für die Korrektheit des XDPB besteht jedoch Eigenverantwortung...

Praxis

Es empfiehlt sich, LOGIN auf dem RAM-Laufwerk zu haben, damit ein Zurücksetzen in den Normalzustand ganz einfach mit »M: LOGIN NOFREEZE« möglich ist (wenn beispielsweise beide Laufwerke geändert sind und auf keiner Diskette dieses Formates »LOGIN« ist, hilft nur noch ein Reboot mit allen Konsequenzen).

Das bisher einzige zugängliche Fremdformat ist das »große VORTEX-Format« (Bild 2), der Rest ist noch zu erforschendes Gebiet. Die Arbeitsstufen zum Sichern einer VORTEX-Diskette wären also folgende (LOGIN und ein Kopierprogramm, z. B. SWEEP) auf M: oder A: vorausgesetzt):

- LOGIN VORTEX
- SW /Alle Files auf Standarddisks in A kopieren/
- LOGIN NOFREEZE
- SW /Alle Files A nach B kopieren/

Forschung

Erfahrungen mit anderen Formaten oder auch Schreibvorgängen liegen bislang noch nicht vor. Hier sind die Anwender um Meldungen gebeten. Für Fremdformate auf A: hat sich (außer der Festlegung auf bestimmte Formate) noch keine Anwendung ergeben. Zu erwägen wäre hier jedoch die Formatierung mit XFORMAT bis Spur 43, die mit einem entsprechenden XDPB theoretisch »legale« 190 KByte auf A: ermöglichen würde. Also bitte: Forscher an die Front!!!

PS: Es erübrigt sich wohl, darauf hinzuweisen, daß der Autor für Hard oder Softwareschäden im Zusammenhang mit Floppy-Experimenten nicht haftet!!!

Der XDPB für VORTEX

Records pro Track: 36
Blockverschiebefaktor: 5
Blockmaske: 31
Extent-Maske: 3
Blockanzahl-1: 176
Directoryeinträge-1: 127
Allocation 0: 128
Allocation 1: 0
Prüfsummenvektor: 32
reservierte Spuren: 2
Phys. Record-Versch.-Faktor: 2
Phys. Record-Versch.-Maske: 3
Seitenverteilung: 129
Tracks pro Seite: 80
Sektoren pro Track: 9
Erste Sektornummer: 1
Sektorengröße: 512
Gaplength=read-write: 30
Gaplength=format: 80
Modus: 96
Freeze: 255

(M.Anton)

JOYCE MULTI-DATABASE & TOOLKIT

(Umfangreiches Universal Dateisystem & Dienstprogramme)

- **Universell einsetzbar** – Verwaltet von der Adress-Personaldatei bis zur Münzsammlung **alles**
- Arbeitet auf **Joyce PCW 8256 und 8512**
- Wurde nicht von anderen Computern übernommen, sondern auf dem Schneider Joyce entwickelt, z. B. erfolgt die Bildschirmausgabe auf 90 x 30 Zeichen (nicht mit den üblichen 80 x 24 Zeichen)
- **Direktzugriff** (Random Access)
- Die **einzigartige SPEED DISC Funktion** ermöglicht ein **5-8fach** schnelleres Arbeiten mit der **MULTI DATABASE**, durch konsequente Nutzung der RAM-Floppy
- Zehn frei definierbare Datensätze pro Datenfeld
- Suchen nach **beliebigen Kriterien**
- Suchen nach **ganzen oder teilweisen Begriffen**
- Suchverknüpfung **und/oder** möglich
- Leichtes Erstellen einer **Selektionsdatei.** Es können problemlos neue Dateien eingerichtet werden, die nur Elemente enthalten, die vorher von DATEN SUCHEN gefunden wurden
- Etiketten-**Sofortdruck** (ohne Übernahme der Daten in eine Datei)
- Adressetikettendruck auf beliebige Formate. Frei wähl- und definierbare Überschriften
- Komfortable Benutzerführung durch **Pull Down Menue** ähnlich dem Locoscript
- Bedienungsfehler werden weitgehendst vom Programm abgefangen und angezeigt
- Eingebauter **Taschenrechner** mit Grundrechnungsarten und Memoryfunktion
- **Ausdruck in NEUN verschiedenen Schriftarten** möglich
- Auf einer Diskette können mehrere Dateien angelegt werden
- **Optimale Übersicht am Bildschirm**
- Kein ständiges Wechseln der Programm- und Datendiskette nötig
- **u. v. a. m.**

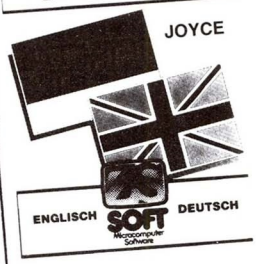
INTERLOGIC TOOLKIT

Dienstprogramm zur **MULTI-DATABASE** – Ermöglicht das leichte **sortieren, kopieren, mischen, löschen, umbenennen** von Dateien. – Der Taschenrechner ist auch hier verfügbar! – **Jetzt mit elektronischem Notizblock!** – Sehr komfortabel – **u. v. a. m.**

Multi-Database & Toolkit für alle JOYCE PCW nur DM 49.—

Bei diesem Preis fällt das Umsteigen auf die MULTI-DATABASE leicht!!!

CPC·COMPUTER DICTIONARY



Ganz Eilige können unsere HOTLINE 08652/3545 benutzen!

Wir würden uns über ein Gespräch mit Ihnen sehr freuen

Lieferzeit nur ca. 3-4 Werktage nach Eingang der Bestellung

Alle Preise verstehen sich in D-Mark und sind inkl. MwSt. – Lieferungen erfolgen per Nachnahme oder V-Scheck. Porto/Versandkosten DM 4.— im Inland – DM 12.— im Ausland.

INTERLOGIC – Peter Herzog
Grabenweg 30, 8240 Schönau a. Königssee

Joyce Dictionary Set Leistungsübersicht:

- **Elektronisches Wörterbuch & Vokabeltrainer**
- ca. **40.000** fest gespeicherte Wörter
- ca. **20.000** fest gespeicherte Stichwörter
- Durchschnittliche Zugriffszeit im Wörterbuch auf einen Begriff **nur ca. 6 sek.**
- **individuell erweiterbar**
- **Lernerfolg** durch den Vokabeltrainer
- Komplettset **Deutsch/Englisch & Englisch/Deutsch**

Bitte beachten Sie den Testbericht in PC-INTERNATIONAL 3/87, Seite 80

JOYCE DICTIONARY SET jetzt nur **DM 99.—**

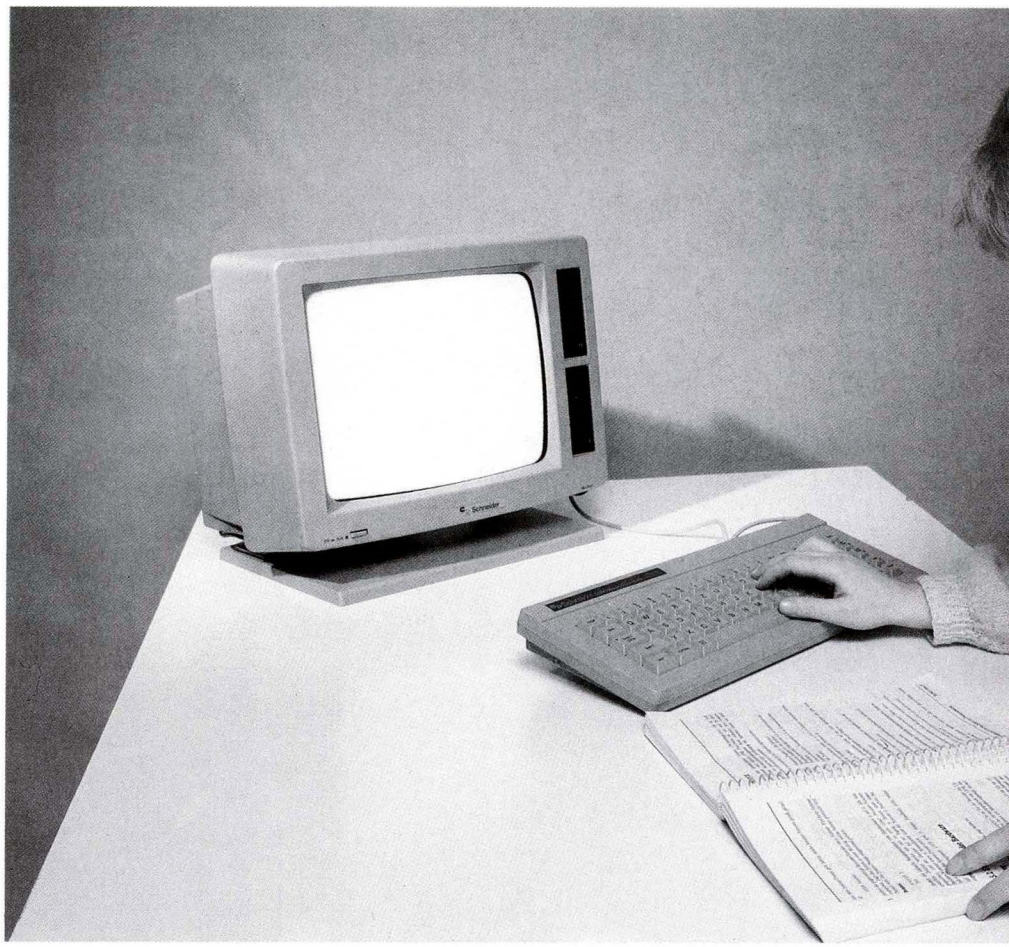
INTERLOGIC · Postfach 2210
8240 Schönau am Königssee · Telefon 08652-3545

Listing LOGIN

```
program Login;

ä Lesen von Fremdformaten ü

type XDPB = record
            SPT    : integer;
            BSH,
            BLM,
            EXM    : byte;
            DSM,
            DRM    : integer;
            AL0,
            AL1    : byte;
            CKS,
            OFF    : integer;
            PSH,
            PHM,
            Side,
            TPSi,
            SePT,
            First  : byte;
            Size   : integer;
            Gap1,
            Gap2,
            Mode,
            Freeze : byte;
            end;

var   Ein     : XDPB;
      Patch_A : XDPB absolute $FF4A;
      Patch_B : XDPB absolute $FF65;
      Datei   : file of XDPB;
      Name    : stringä10ü;
      Wahl,
      Drive,
      Drive_1,
      Drive_2 : char;
      Ok      : boolean;


procedure Scan_Key;
begin
  while not keypressed do;
  read (kbd,Wahl);
  Wahl:=upcase(Wahl);
end;


procedure Intro;
begin
  clrscr;
  writeln ('-----------------------------------------------');
  lowvideo;
  writeln ('*** Disk-Login für JOYCE    1987 by M. Anton ***');
  normvideo;
  writeln ('-----------------------------------------------');
  writeln; writeln;
end;


procedure Frage;
begin
  Wahl:='x';
  writeln;
  write ('Abspeichern (J/N)? ');
  repeat
    Scan_Key;
  until Wahl in ä'J','N'ü;
  writeln;
end;




procedure Einlesen;
begin
  assign (Datei,Name+'.FRM');
  reset (Datei);
  read (Datei,Ein);
  close (Datei);
end;


procedure Abspeichern;
begin
  writeln; writeln;
  write ('Format abspeichern als: ');
  readln (Name);
  assign (Datei,Name+'.FRM');
  rewrite (Datei);
  write (Datei,Ein);
  close (Datei);
end;


procedure Eingeben;
begin
  with Ein do begin
    write('Records pro Track: ');
    readln (SPT);
    write('Blockverschiebefaktor: ');
    readln (BSH);
    write('Blockmaske: ');
    readln (BLM);
    write('Extent-Maske: ');
    readln (EXM);
    write('Blockanzahl-1: ');
    readln (DSM);
    write('Directoryeinträge-1: ');
    readln (DRM);
    write('Allocation 0: ');
    readln (AL0);
    write('Allocation 1: ');
    readln (AL1);
    write('Prüfsummenvektor: ');
    readln (CKS);
    write('reservierte Spuren: ');
    readln (OFF);
    write('Phys. Record-Versch.-Faktor: ');
    readln (PSH);
    write('Phys. Record-Versch.-Maske: ');
    readln (PHM);
    write('Seitenverteilung: ');
    readln (Side);
    write('Tracks pro Seite: ');
    readln (TPSi);
    write('Sektoren pro Track: ');
    readln (SePT);
    write('Erste Sektornummer: ');
    readln (First);
    write('SektorengröBe: ');
    readln (Size);
    write('Gaplength/read-write: ');
    readln (Gap1);
    write('Gaplength/format: ');
    readln (Gap2);
    write('Modus: ');
    readln (Mode);
    write ('Freeze: ');
    readln (Freeze);
  end;
end;


procedure Ausgeben;
begin
  with Ein do begin
    writeln('Records pro Track: ',SPT);
    writeln('Blockverschiebefaktor: ',BSH);
    writeln('Blockmaske: ',BLM);
    writeln('Extent-Maske: ',EXM);
    writeln('Blockanzahl-1: ',DSM);
    writeln('Directoryeinträge-1: ',DRM);
    writeln('Allocation 0: ',AL0);
    writeln('Allocation 1: ',AL1);
    writeln('Prüfsummenvektor: ',CKS);
    writeln('reservierte Spuren: ',OFF);
    writeln('Phys. Record-Versch.-Faktor: ',PSH);
    writeln('Phys. Record-Versch.-Maske: ',PHM);
    writeln('Seitenverteilung: ',Side);
    writeln('Tracks pro Seite: ',TPSi);
    writeln('Sektoren pro Track: ',SePT);
    writeln('Erste Sektornummer: ',First);
    writeln('SektorengröBe: ',Size);
    writeln('Gaplength/read-write: ',Gap1);
    writeln('Gaplength/format: ',Gap2);
```

```
    writeln('Modus: ',Mode);
    writeln('Freeze: ',Freeze);
  end;
end;


procedure Change_A;
begin
  Einlesen;
  if Ein.Side<>0 then
    begin
      writeln ('DS-Formate auf A: nicht möglich...');
      Ok:=false;
    end
  else Patch_A:=Ein;
end;

procedure Change_B;
begin
  Einlesen;
procedure B_oder_F;
begin
  Name:=paramstr(1);
  if (Name<>'FREEZE') and (Name<>'NOFREEZE') then Change_B
  else if Name='FREEZE'  then begin
                                Patch_A.Freeze:=$FF;
                                Patch_B.Freeze:=$FF;
                              end
       else  begin
               Patch_A.Freeze:=0;
               Patch_B.Freeze:=0;
             end;
end;

procedure A_oder_B;
begin
  Drive:=paramstr(1);
  Name:=paramstr(2);
  if Drive in ['A','B'] then
    begin
      case Drive of
        'A' : Change_A;
        'B' : Change_B;
      end;
    end
  else
   begin
     writeln (^G,'Falsche Parameter...');
     Ok:=false;
   end;
end;

procedure A_und_B;
begin
  Drive_1:=paramstr(1);
  Drive_2:=paramstr(3);
  if (Drive_1=Drive_2) or
     (not (Drive_1 in ['A','B'])) or
     (not (Drive_2 in ['A','B']))
       then
         begin
           writeln (^G,'Falsche Parameter...');
           Ok:=false;
         end
       else
         begin
           Name:=paramstr(2);
           case Drive_1 of
             'A' : Change_A;
             'B' : Change_B;
           end;
           Name:=paramstr(4);
           case Drive_2 of
             'A' : Change_A;
             'B' : Change_B;
           end;
        end;
end;

procedure Aktuell_Anzeigen;
begin
  Intro;
  writeln ('Parameter für Drive A:');
  writeln;
  Ein:=Patch_A;
  Ausgeben;
  Frage;
  if Wahl='J' then Abspeichern;
  Intro;
  writeln ('Parameter für Drive B:');
  writeln;
  Ein:=Patch_B;
  Ausgeben;
  Frage;
  if Wahl='J' then Abspeichern;
end;


procedure File_Anzeigen;
begin
  Intro;
  writeln;
  write ('Name des Files: ');
  readln (Name);
  Einlesen;
  Intro;
  writeln ('Parameter von ',Name,'.FRM');
  writeln;
  Ausgeben;
  while not keypressed do;
end;


procedure Erstellen;
begin
  Intro;
  Eingeben;
  Abspeichern;
end;



procedure Menu;
begin
  Wahl:='X';
  repeat
  Intro;
  writeln;
  writeln ('1 = Aktuelle Parameter anzeigen');
  writeln;
  writeln ('2 = Parameter-File anzeigen');
  writeln;
  writeln ('3 = Parameter-File erstellen');
  writeln; writeln;
  writeln ('0 = E N D E');
  writeln; write ('Wahl: ');
  repeat
    Scan_Key;
  until Wahl in ['0'..'3'];
  writeln (Wahl);
  case Wahl of
    '1' : Aktuell_Anzeigen;
    '2' : File_Anzeigen;
    '3' : Erstellen;
  end;
  until Wahl='0';
end;


procedure Modify;
begin
  case paramcount of
    1         : B_oder_F;
    2,3       : A_oder_B;
    4..maxint : A_und_B;
  end;
if not Ok then writeln ('Operation nicht ganz erfolgreich...');
end;


{ Hauptprogramm (!!!) }

begin
  intro;
  Ok:=true;
  if paramcount=0 then Menu else Modify;
end.
```

Achtung BASIC-Programmierer!



- Stört es Sie auch, die BASIC-Befehle Zeichen für Zeichen eintippen zu müssen?
- Stört es Sie auch, bei Programmänderungen jeweils EDIT Zeilennummer eingeben zu müssen, d.h. keinen Bildschirm-Editor zu haben?
- Würden Sie Ihr Programmlisting auch einmal gerne in einer beliebigen Schriftart ausdrucken (ohne großen Aufwand)?
- Haben Sie schon einmal in einem fertigen Programm Variablen umbenannt und wenn ja, wie denken Sie daran zurück?

Nun sind Sie sicherlich gespannt auf einen Tip, der nichts kostet, wenn Sie die LocoScript-Version 1.21 (oder größer) besitzen:

Schreiben Sie Ihre BASIC-Programme in Zukunft mit LocoScript! Davon haben Sie folgendes:

Die oben genannten Nachteile kehren sich in Vorteile um, denn

- Sie können die BASIC-Befehle als Sätze auf Tasten legen (siehe Artikel »Vom Umgang mit Sätzen«).
- Sie können mit dem Cursor kreuz und quer über den Bildschirm fahren.
- Sie wählen die Schriftart wie gewohnt über Menues aus.
- Sie benennen Variablen um mit dem »Ersetzen-Menue«.

Darüber hinaus können Sie Ihre Programme komprimieren, indem Sie mit dem »Ersetzen-Menue« den eckigen Pfeil für »Neue Zeile« (RETURN) durch einen Doppelpunkt ersetzen. In dem komprimierten Text müssen Sie aber spätestens alle drei Zeilen ein RETURN setzen, sonst wird die zulässige Zeilenlänge in BASIC überschritten.

Natürlich hat auch dieses Verfahren eine Kehrseite:

- Sie müssen die Zeilennummern von Hand einfügen, was besonders bei Sprunganweisungen Aufmerksamkeit erfordert.

Ein Programm läßt sich mit LocoScript schreiben und in ein BASIC-Programm umwandeln, ein BASIC-Programm kann man jedoch nicht mehr in einen von LocoScript lesbaren Text zurückverwandeln. Einzige Möglichkeit: mit 'f7', sub-Modus Text einfügen läßt sich ein Basic-Programm in den gerade bearbeiteten Text einfügen. Dies ist langwierig, aber nur ein Nachteil, wenn Sie in BASIC Änderungen durchführen (z.B. RENUM) und sich danach überlegen, daß Sie lieber mit dem komfortablen Bildschirmeditor arbeiten.

Doch nun zur praktischen Seite. Laden Sie LocoScript und erstellen Sie einen Text auf Laufwerk M mit dem Namen ZAEHLER.TXT.

Schreiben Sie:

```
PRINT »Beispiel eines unter LocoScript erstellten Programms«
PRINT:PRINT
PRINT »Zaehlprogramm«
PRINT:PRINT
INPUT »Untere Grenze = «,unten
INPUT »Obere Grenze = «,oben
'
For i = unten TO oben
PRINT i
NEXT i
'
PRINT:PRINT
PRINT »Programmende«
END
```

Nun fügen Sie die Zeilennummern ein:

```
10 PRINT »Beispiel eines unter Loscript erstellten Programms«
20 PRINT:PRINT
30 PRINT »Zaehlprogramm«
usw.
```

Benutzen Sie dazu die »Zeile«-Taste, um an den Zeilenanfang zu gelangen.

Beenden Sie die Bearbeitung der Datei ZAEHLER.TXT (EXIT) und verschieben (f4) Sie sie auf Ihre LocoScript-Diskette (Laufwerk A). Der Cursor muß sich nun auf ZAEHLER.TXT befinden. Sie drücken die Taste f7 und wählen die Option »ASCII-Datei erstellen« aus. Sie werden aufgefordert, die Zielgruppe für die ASCII-Datei anzugeben; wählen Sie Gruppe 0 auf Laufwerk M aus und drücken ENTER. Bevor Sie zum zweiten Mal ENTER drücken, benennen Sie den Text um in ZAEH-



LER.BAS. Dieses Menue können Sie mit ENTER bestätigen, da das benötigte »unformatiert« bereits ein Häkchen hat.

Nun ist eine ASCII-Datei Ihres LocoScript-Textes erstellt. Dies können Sie überprüfen, indem Sie versuchen, diese Datei zu bearbeiten. Es wird die Fehlermeldung »Kein LocoScript-Text, Operation abbrechen« erscheinen. Ihr Programm muß jetzt noch auf Ihre BASIC-Diskette kopiert werden. Dafür drücken Sie (f1) für Diskwechsel, entnehmen die LocoScript-Diskette und schieben die BASIC-Diskette in das Laufwerk. Sie drücken wieder (f1) und sehen nun den Inhalt Ihrer BASIC-Diskette. Mit Taste (f3) kopieren Sie die Datei ZAEHLER.BAS von Laufwerk M in Gruppe 0 auf Laufwerk A.

(Die Diskettenoperationen unter Locoscript sind übrigens in Heft 10/86 auf Seite 39 abgedruckt.)

Nun drücken Sie gleichzeitig die Tasten SHIFT, EXTRA, EXIT. Falls Sie das Betriebssystem CP/M nicht auf Ihrer BASIC-Diskette haben, wird Sie Ihr Computer durch ein dezentes Piepen darauf aufmerksam machen.

Nachdem Sie Mallard-BASIC geladen haben, geben Sie wie gewohnt ein: run »zaehler« ENTER und das Programm wird seinen Weg gehen. Programmänderungen können Sie wie gewohnt in BASIC durchführen oder aber Ihre LocoScript-Diskette einschieben, die Änderung durchführen und nochmal eine neue ASCII-Datei erstellen.

Komprimieren

Das komprimierte Programm sieht nach dem Ersetzen des eckigen Pfeils durch einen Doppelpunkt so aus:

```
PRINT »Beispiel eines unter LocoScript erstellten Programms«:PRINT:PRINT:PRINT »Zaehlprogramm«:PRINT:PRINT:INPUT »Untere Grenze = «,unten:INPUT »Obere Grenze = «,oben:':For i = unten to oben:PRINT i:NEXT i:':PRINT:PRINT:PRINT »Programmende«:END
```

Nach Einfügen der Zeilennummern und einzelner Leerzeichen läßt sich das Programm in zwei Programmzeilen schreiben:

```
10 PRINT »Beispiel eines unter LocoScript erstellten Programms«:PRINT: PRINT:PRINT »Zaehlprogramm«:PRINT:PRINT:INPUT »Untere Grenze = «,unten :INPUT »Obere Grenze = «,oben:'
20 For i = unten to oben:PRINT i:NEXT i:':PRINT:PRINT:PRINT »Programmende«:END
```

Dies ist zwar nicht übersichtlich, aber dafür haben Sie das Programm als Text auch noch gespeichert. In diesem Beispiel spart man durch Komprimieren ca. 12% Speicherplatz.

Falls Sie einen Joyce der ersten Stunde besitzen, und beim Versuch, dieser Anleitung zu folgen, die Option »ASCII-Datei erstellen« im besagten Menue vermissen, dann arbeiten Sie noch mit der alten LocoScript-Version 1.1. Falls sich dieser Verdacht beim Betrachten des Ladebildes bestätigt, senden Sie Ihre Originaldiskette mit der Bitte um Umtausch an Firma Schneider/Türkheim zurück.

(K.-P.Roethke)

Vom Umgang mit »Sätzen«

Für Vielschreiber oder schlicht zur Vereinfachung kann das Erstellen von eigenen Sätzen eine große Hilfe sein. Und selbst, wer nicht viel schreibt, wird Floskeln wie »Sehr geehrte Damen und Herren,« bzw. »Mit freundlichen Grüßen« immer wieder benötigen.

Beim Abfassen längerer Berichte wie z.B. Examensarbeiten kommen die Vorteile jedoch voll zur Geltung. Daß man Worte bzw. Sätze abspeichern kann, ist zwar schön, aber eigentlich noch nichts besonderes.

Die eleganten Möglichkeiten beginnen nämlich dann, wenn man auch die Codes wie z.B. (+Us)(-Us) mit abspeichert. Man erspart sich damit je nach Anwendung das Einblenden der verschiedenen Menues.

Ein Beispiel:

Die Überschrift eines Kapitels soll in einer anderen Zeichendichte, kursiv und fett gedruckt, außerdem soll die besondere Schrift (die nur auf dem Ausdruck zu sehen ist) am Bildschirm durch inverse Darstellung hervorgehoben werden.

Für diese Aktion müssen die Menues »Hervorheben« bzw. »Schrift« am Anfang und Ende der Überschrift eingeblendet werden. Bedeutend einfacher und schneller geht dies, wenn man einmal die Überschrift in die gewünschte Form bringt und sich dann einen Überschriften-Satz anlegt, der wie folgt aussehen könnte:

(+Kursiv)(+ZeichZ10)(+Fett)(+Inv)ü (-Fett)(-Inv)(-Kursiv)(-ZeichZ)

Diesen Satz inklusive zwei mal RETURN für eine Zwischenzeile legt man dann auf Buchstaben U oder Ü wie Überschrift ab. Das ü wird dann einfach mit der gewünschten Überschrift überschrieben. Falls die Überschrift nur 30 Zeichen lang ist, kann dies auch elegant mit dem »Ersetzen«-Menue erledigt werden.

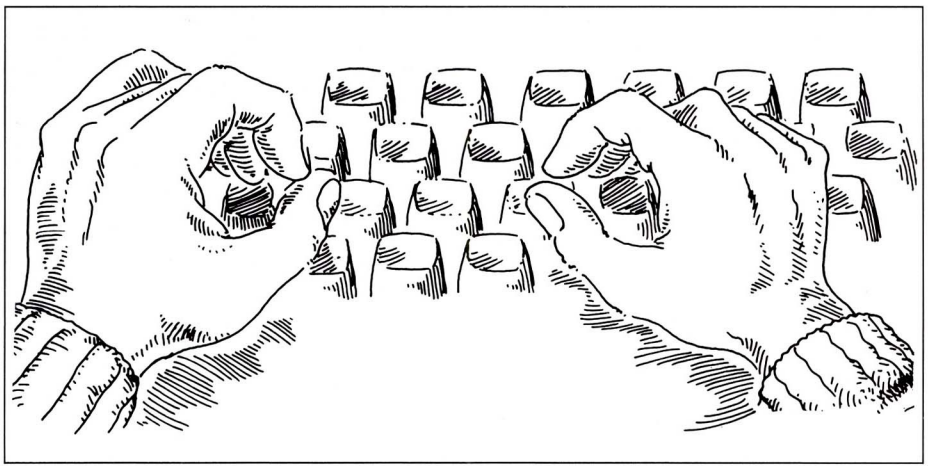
Eine weitere Anwendung der Code-Speicherung ist die mathematische Schreibweise, die im Anschluß an die Vorgehensweise für das Erstellen eigener Sätze beschrieben wird.

Vorgehensweise für das Erstellen eigener Sätze

1. Neustart des Programms LocoScript.
   In der Diskverwaltung existiert die Datei SAETZE.STD nur in Gruppe 0 (BRIEFE) = Laufwerk A.
2. In Gruppe 0 = Laufwerk M dann den Text »bearbeiten«, aus dem die neuen Sätze abgespeichert werden sollen. Dazu muß der betreffende Text von Laufwerk A auf Laufwerk M kopiert werden. Man kann auch einen neuen Text erstellen, der nur aus den gewünschten Redewendungen besteht. Dies hat den Vorteil, daß die Sätze in einer zusätzlichen Datei gespeichert sind, falls man beim Abspeichern etwas vergißt.
3. Löschen der bisherigen (alten) Sätze. Die bisherige Belegung der Tasten mit Sätzen muß während der »Bearbeitung« (d.h. nicht in der Diskverwaltung) mit der Funktionstaste f8 angezeigt werden. Bei Neuanlegen einer Sätzedatei empfiehlt es sich, die alten Sätze zu löschen. So bringt man alte und neue Sätze nicht durcheinander.

   Löschen geschieht durch die Tastenfolge COPY-COPY-Buchstabe.

   Nach Betätigung der f8-Taste wird die Datei SAETZE.STD automatisch von Laufwerk A auf M kopiert. Die alten Sätze sind nun zunächst nur in der Datei SAETZE.STD auf Laufwerk M gelöscht.
4. Neue Sätze den gewünschten Buchstaben zuweisen durch COPY-Bereich (=Satz)-COPY-Buchstabe.
5. Wichtig: Nun Taste f8 drücken und Option »Alle Sätze speichern« auswählen und mit ENTER bestätigen. Nun sind die neuen Sätze in der Datei SAETZE.STD in Gruppe 0 = Laufwerk M gespeichert.
6. Aus laufender Textbearbeitung mit EXIT aussteigen.



7. Datei SAETZE.STD in Gruppe 0 = Laufwerk A löschen (f6). (Dies waren die alten Sätze)
8. Datei SAETZE.STD von Gruppe 0 = Laufwerk M in Gruppe 0 (BRIEFE) = Laufwerk A verschieben (f4). Dies sind die neuen Sätze. Sie werden beim nächsten Start des gesamten Programms den betreffenden Tasten zugewiesen.
   Die meiste Arbeit ist nun erledigt.
9. Kontrolle: Warmstart durchführen (SHIFT + EXTRA + EXIT).
   Die neuen Sätze müssen sich nun bei Textbearbeitung in jeder beliebigen Gruppe einblenden lassen.
10. Empfehlung: Die Datei SAETZE.STD in diejenige Gruppe kopieren, in der die erstellten Sätze am meisten verwendet werden. Um einen Bezug zur Satzspeicherdatei herzustellen, sollte dann die Extension STD umbenannt werden, z.B. BRF für Brief- Redewendungen oder MAT für mathematische Formelschreibweise.

Will man wissen, welche Sätze in der Datei SAETZE.LET bzw. SAETZE.NUL (zu finden in Gruppe 3 = SCHABLON) gespeichert sind, geht man die obige Beschreibung ab Punkt 8 durch, wobei z.B. die Datei SAETZE.STD durch die Datei SAETZE.LET ersetzt werden muß.

Anwendung der Codes-Speicherung für die mathematische Schreibweise

Wie im vorigen Überschriftenbeispiel gezeigt wurde, können die Codes zur Textveränderung mit abgespeichert werden.

Bei mathematischen Formeln können hoch-/tiefgestellte Zeichen auf dieselbe Weise rationell verwirklicht werden.

Der Nachteil, daß hoch-/tiefgestellte Zeichen nur in Entwurfsqualität gedruckt werden, kann beinahe ausgeglichen werden, wenn man diese Zeichen doppelt (f3) hervorhebt. Der Druckkopf fährt dann zweimal (wie bei NLQ-Schrift) leicht versetzt über die Stelle, man hat dann »Quasi-NLQ-Schrift«.

Eine weitere Anwendung ist die Formelschreibweise:

Eine Formel mit Bruchstrichen sieht am besten aus, wenn sie mit Zeilenabstand 1/2 gedruckt wird.

Beim Schreiben einer Gleichung schreibt man zuerst die Zeichen in Höhe des Bruchstrichs. Der Bruchstrich selbst wird aus Bindestrichen zusammengesetzt, wobei jeweils eine kleine Lücke entsteht. Die Länge des Bruchstrichs richtet sich nach der maximalen Anzahl Zeichen im Zähler bzw. Nenner des Bruchs.

Ist die »Mittelzeile« geschrieben, werden Zähler und Nenner plaziert.
Bei dieser Vorgehensweise sollte der Bruch nicht in Proportionalschrift geschrieben werden, da man sonst Schwierigkeiten beim genauen Plazieren der Zeichen bekommen wird.

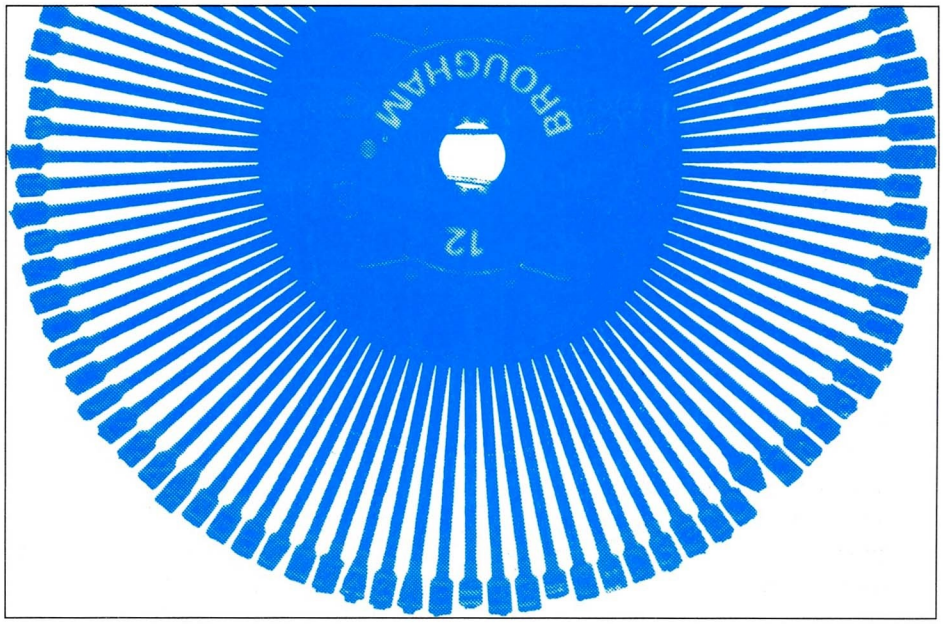
Man kann die Codes für die Änderung des Zeilenabstandes speichern oder ohne Verwendung von Sätzen ein anderes Layout einblenden (f2), indem man den Zeilenabstand 1/2 und z.B. Zeichendichte 10 vereinbart hat.

Der Phantasie sind keine Grenzen gesetzt.....

(K.-P.Roethke)

Umwandlungsprogramm

für Centronics-Typenraddrucker



Wer mit dem Schriftbild seines serienmäßigen Joyce-Druckers nicht zufrieden ist, der kann über die Centronics-Schnittstelle der CPS 8256 seine Texte an einen Typenraddrucker ausgeben. Eine ganz neue Spielart wurde in CPC International 10/86 mit dem SD 15 von SchneiderData vorgestellt.

Zu diesem Zweck werden LocoScript-User in vorbildlicher Weise mit einer Option bedient, die die Erstellung formatierter oder unformatierter ASCII-Files erlaubt.

Leider enthält die deutsche Sprache allerlei Sonderzeichen (ä,ö,ü,ß), die im ASCII-Standard nicht vorgesehen sind; selbst bei deutscher LANGUAGE-Einstellung vermag die über LocoScript erstellte ASCII-Datei keine Umlaute darzustellen.

Das hier vorgestellte Programm UMWAND.BAS bietet nun eine einfache Möglichkeit, unter Basic eine Umformung der durch LocoScript erstellten ASCII-Dateien vorzunehmen.

Dazu bringt man das Programm und die zu transformierende ASCII-Datei auf eine Diskette, lädt Basic und startet mit 'RUN "UMWAND" <RETURN>. Das Programm erfragt den Namen der zu transformierenden Datei (mit Angabe der Extension) und speichert den korrigierten Text in einer neuen Datei namens "TEXTNEU.ASC". TEXTNEU kann nun direkt an den Centronics-Drucker gegeben werden.

Hinter der Programmzeile 210 lassen sich je nach Bedarf, Sprache und Drucker weitere Umdefinitionen einfügen, wobei n der »neue« ASCII-Code und a der »alte« (eingestellte) ASCII-Code sind. Der »Überlaufschutz« verhindert die bei längeren Texten nach zwei bis drei Seiten auftretende Fehlermeldung "String space full in 100". Durch die Bearbeitung mit LINE INPUT # bleibt das Seitenlayout erhalten. Parallel zur Umwandlung wird die bearbeitete Zeile in der ursprünglichen Form auf dem Bildschirm dargestellt, um den Stand des Vorganges anzuzeigen. Nach Beendigung der Umformung meldet sich der Rechner mit Klingelzeichen und TYPEd die neue Version aus; die dient zur schnellen Kontrolle, ob auch alle Textteile bearbeitet wurden.

Das Verfahren benötigt auf der Diskette einen freien Speicherplatz vom Umfang der zu transformierenden Datei. Daher sollte man bei langen Texten die Disk möglichst freihalten. Bei Benutzung des Standardlaufwerks begrenzt sich die Textlänge durch die Diskettenkapazität von 173 kB und den Programmumfang von 2 kB auf maximal 85 kB. Dabei ist darauf zu achten, daß die ASCII-Kopie oft einige kB mehr beansprucht als die LocoScript-Fassung. (F.Jochem)

```
LISTING >UMWAN<, REMARK = >REM<.
<13> 10 ' TEXTUMWANDLUNG FuR CENTRONICS-DRUCKER
<88> 20 DIM text$(250)
<57> 30 INPUT "ZU TRANSFORMIERENDE ASCII-DATEI. $$$.$$$
     :";datei$
<74> 40 OPEN "I",1,datei$
<21> 50 OPEN "O",2,"TEXTNEU.ASC"
< 9> 60 eintraege = 0
<67> 70 umlauf% = 1
<53> 80 WHILE NOT (EOF(1))
<70> 90 eintraege = eintraege + 1
<54> 100        LINE INPUT #1,text$(eintraege)
<41> 105        PRINT text$(eintraege)
<17> 110        IF text$(eintraege) = "" GOTO 600
<89> 120        ' Definition der zu tranformierenden Zeichen
<49> 130        ON umlauf% GOTO 150,160,170,180,190,200,210
< 4> 150        a = 240 : n = 123 : GOTO 500 : REM kleines ä
<81> 160        a = 208 : n =. 91 : GOTO 500 : REM großes. Ä
<62> 170        a = 243 : n = 124 : GOTO 500 : REM kleines ö
<39> 180        a = 211 : n =. 92 : GOTO 500 : REM großes. Ö
<98> 190        a = 244 : n = 125 : GOTO 500 : REM kleines ü
<56> 200        a = 212 : n =. 93 : GOTO 500 : REM großes. Ü
<84> 210        a = 186 : n = 126 : GOTO 500 : REM..... ß
<39> 500        ' Umwandlungen
<44> 510        stelle% = INSTR(1,text$(eintraege),CHR$(a))
<52> 520        IF stelle% = 0 GOTO 550
<65> 530        MID$(text$(eintraege),stelle%,1) = CHR$(n)
<62> 540        GOTO 510
<54> 550        umlauf% = umlauf% + 1
<15> 560        IF umlauf% <= 7 THEN GOTO 130 ELSE GOTO 600
<78> 600        ' Einlesen in neue Datei "TEXTNEU.ASC"
< 6> 610        PRINT #2,text$(eintraege)
<27> 630        umlauf% = 1
<36> 700                'Überlaufschutz
<97> 710                IF eintraege > 100 GOTO 2000
<11> 1000 WEND
<83> 1010 CLOSE
<54> 1020 bell$ = CHR$(7) : PRINT bell$ : PRINT bell$ :
     PRINT
<56> 1050 TYPE textneu.asc
<68> 1100 END
<49> 2000 ' Überlaufschutz
<52> 2010 COMMON umlauf%,datei$
<61> 2020 COMMON RESET
<63> 2030 DIM text$(250)
<76> 2050 GOTO 80
```

Listing 1: Das Umwandlungsprogramm

Tastendefinition + Startdiskette

Tips zur Erstellung einer Tastendefinitionsdatei und einer Startdiskette für Basic-Programmierer.



Laut Handbuch des JOYCE ist das Umdefinieren der Tastatur relativ einfach und problemlos: Zunächst erstellt man eine Tastendefinitionsdatei (am besten mit dem RPED.BAS Editor), speichert diese (z.B. mit TASTEN.001 bezeichnete) Datei zusammen mit dem SETKEYS.COM Dienstprogramm auf eine Startdiskette und definiert die Tastatur durch den Befehl A>SETKEYS TASTEN.001 unter CP/M um. Zuvor muß man lediglich noch herausgefunden haben, was ein »Zeichen«, was ein »Steuer-Code« und was ein »Erweiterungsstring« ist (mit denen kann man nämlich laut Handbuch die Tasten neu belegen).

Doch ganz so einfach ist das leider nicht. Dies merken Sie spätestens dann, wenn Sie sich den Start von MALLARD-BASIC, wie im Handbuch 1/II (Anhang 1, Seite 9 unten) angegeben, durch Umdefinieren der Taste f1/2 (Tasten-Nr. 02) erleichtern möchten, indem Sie genau nach Anweisung in einer Tastendefinitionsdatei dem Erweiterungszeichen E #80 den Erweiterungsstring "BASIC↑M" und der Taste 02 dieses Erweiterungszeichen zuordnen. Nach dem Aufruf von SETKEYS TASTEN.001 (falls Sie die Datei so genannt haben) läßt sich MALLARD-BASIC auch tatsächlich durch Druck auf die Taste f1/2 starten, allerdings werden Sie verblüfft feststellen, daß ein Druck auf die STOP-Taste den gleichen (hier unerwünschten) Effekt erziehlt. Sollten Sie den Erweiterungsstring ("BASIC ↑M") nun dem Erweiterungszeichen E #81 zuordnen, erleben Sie die Überraschung spätestens beim Aufruf des RPED.BASEditors: Wenn Sie dann nämlich laut Menueanweisung die Taste f1/2 betätigen, um eine vorhandene Datei aufzurufen, tut sich nichts! Jetzt wartet der EDITOR nämlich darauf, daß ihm über die Taste f1/2 vom Erweiterungszeichen E ↑81 ein '↑Z' (Control-Z) »geliefert« wird. Stattdessen erhält er aber "BASIC)M"und das sagt ihm hier herzlich wenig!

Dies Problem entsteht dadurch, daß ganze Befehlswörter ("LIST", "EDIT" usw.) nicht direkt einer bestimmten Taste (z.B. der f1-Taste) zugeordnet werden können, sondern nur auf dem Umweg über die dafür vorgesehenen 31 Erweiterungszeichen (#80-#9F). Diese Erweiterungszeichen werden aber beim Laden des Betriebssystems mit CTRL-Codes belegt und bestimmten Tasten zugeordnet (vgl. Handbuch 1.II, Anhang 1, Seite 11). Auf diese Tastenbelegung sind

```
ALT U=einf Zeile | RUBOUT=DEL Zeile | ↑ ↓ → ← DEL CLR HOME ↵ RETURN | EXIT STOP

E #80 "PRINT CHR$(27);↑"E↑";CHR$(27);↑"H↑"↑M"
E #81 "↑aE↑aH↑M"
E #82 "BASIC↑M"
E #83 "SYSTEM↑M"
E #84 "RUN↑M"
E #85 "SAVE ↑""
E #86 "LOAD ↑""
E #87 "LIST "
E #88 "EDIT "
E #8B "OPTION FILES ↑""   TASTE:          BELEGUNG/FUNKTION:
E #8D "TYPE M:H↑M"
66 N S A E SA "↑C"          -STOP          =STOP
02 N S        "↑Z"          -F1/2          =↑Z
02              SA "↑'#8B'"                =WAHL DES LAUFWERKS UNTER BASIC
03 N            "↑'#8D'"    -[EINBL]       =TASTENBELEGUNG AUF DEM BILDSCHIRM ANZEIGEN
00 N S          "↑Q"        -F3/4          =↑Q
00      A       "↑'#86'"                   =LOAD
00        E SA  "↑'#85'"                   =SAVE
73 N S          "↑S"        -F5/6          =↑S
73      A       "↑'#87'"                   =LIST
73        E SA  "↑'#88'"                   =EDIT
77 N S          "↑P"        -F7/8          =↑P
07 N            "↑'#82'"    -[##]          =BASIC LADEN
07              SA "↑'#83'"                =RUCKSPRUNG INS CP/M-SYSTEM
78        E     "↑'#84'"    -ENTER         =START VOM BASICPROGRAMMEN
75 N S          "↑H"        -CAN           =↑H
75      A       "↑'#80'"                   =BILDSCH. LOSCHEN/BASIC
75              SA "↑'#81'"                =BILDSCH. LOSCHEN/SYSTEM
                                                          Laufwerk ist A:
```

Bild 1: Inhalt der TASTEN, STD-Datei

```
ALT U=einf Zeile | RUBOUT=DEL Zeile | ↑ ↓ → ← DEL CLR HOME ↵ RETURN | EXIT STOP

Die Startdiskette stellt die Tastatur des JOYCE so ein, daß die Arbeit mit
MALLARD-Basic erleichtert wird. Standardbefehle, die besonders beim Programmieren
häufig gebraucht werden, sind jetzt durch Tastendruck aufrufbar (Außerdem werden
einige nützliche CP/M Dienstprogramme in das 'Laufwerk' M kopiert):

                [##]  :                 (Laden von MALLARD-Basic)
[ALT]+[SHIFT]+[##]    :                 (Rückkehr zu CP/M)
[EXTRA]+[ENTER]       :                 (Start von Basic-Programmen)

[ALT]+[SHIFT]+[f1/2]: OPTION FILES "?   (? = Laufwerksnummer M oder A)
[ALT]+      [f3/4]: LOAD "?             (? = Name des zu ladenden Programms)
[ALT]+[SHIFT]+[f3/4]: SAVE "?           (? = Name, unter dem das Progr. gespeichert wird)
[ALT]+      [f5/6]: LIST ?-?            (? = Nr. der zu listenden Zeilen von-bis)
[ALT]+[SHIFT]+[f5/6]: EDIT ?            (? = Nr. der zu editierenden Zeile)

[ALT]+      [CAN] :                     (Bildschirmlöschung unter BASIC)
[ALT]+[SHIFT]+[CAN] :                   (Bildschirmloschung unter CP/M)

Basic-Programme, die nicht länger als 200 Zeilen werden, können mit dem speziellen
Basic-Editor (RPED.BAS) erstellt werden.

Unter CP/M wird der EDITOR mit          [e] + [ENTER]    aufgerufen.
Unter MALLARD-Basic wird er mit   run "rped" + [ENTER]   aufgerufen (Vorher muß
  eventuell das Laufwerk M (mit OPTION FILES "M") angewählt werden).

ACHTUNG: Diese Hinweise können jederzeit durch [EINBL] auf den Bildschirm geholt werden

                                                          Laufwerk ist A:
```

Bild 2: Inhalt der Datei "H"

nun die mitgelieferten Dienstprogramme abgestimmt (z.B. BASIC.COM, RPED.BAS, DISCKIT.COM usw.).

Wenn Sie also die Belegung der Erweiterungszeichen ändern, funktioniert bei diesen Programmen die Steuerung nicht mehr so, wie vorgesehen. Demnach können Sie für die Steuerung eigener Programme die Tastaturbelegung (fast) beliebig ändern, sofern Sie mit derselben Tastaturbelegung (fast) beliebig ändern, sofern Sie mit derselben Tastaturbelegung keine anderen Programme steuern wollen.

Falls Sie aber vorhaben, die Tastatur Ihres JOYCE so zu ändern, daß sie Ihnen einerseits das Programmieren in BASIC erleichtert, andererseits z.B. die Nutzung von RPED.BAS nicht wesentlich erschwert, müssen Sie etwas vorsichtiger zu Werke gehen: Wollen Sie z.B. das Erweiterungszeichen E #80 mit einem neuen Erweiterungsstring belegen, müssen Sie der STOP-Taste (Tasten-Nr. 66) in der neuen Tastendefinitions-Datei direkt das '↑C' (Control-C) zuweisen. Die entsprechende Zeile in der Datei müßte lauten: 66 N S A E SA "↑C" (vgl. Abb. 1). Damit behält die STOP-Taste auch weiterhin ihre vorgesehene Funktion. Um die Funktionstasten f1, f2 usw. für die Belegung mit häufig gebrauchten BASIC-Befehlen freizubekommen, kann man ihre vordefinierte Funktion ('Lieferung' von ↑Z, ↑Q, ↑S usw.) auf eine andere »Tastenebene« (z. B. die ALT-»Ebene«) verlegen. Dadurch brauchen Sie dann bei der Steuerung von RPED.BAS lediglich die Tasten ALT + f1 zu drücken, wenn laut Menueanweisung die Taste f1 zu betätigen wäre.

Ein Beispiel dafür, wie die Tastatur für die Programmierung in BASIC umdefiniert werden könnte, ohne die voreingestellten Funktionen völlig zu verlieren, zeigt Abb. 1. Da die Tasten f1, f2, f3, Ausbl, EZ, ZEILE (bzw. die durch sie ausgelösten CTRL-Codes) zur Steuerung des RPED.BAS-Editors benötigt werden, wurden ihnen die entsprechenden CTRL-Codes direkt zugewiesen bzw. auf eine andere Tastenebene verlegt. Wenn Sie die Tastatur mit dieser Datei umdefinieren, haben Sie einige der beim BASIC-Programmieren besonders häufig erforderlichen Befehle auf Tastendruck parat: LOAD", SAVE", LIST, EDIT usw. Außerdem können Sie den Bildschirm sowohl unter CP/M als auch unter MALLARD-BASIC durch Tastendruck löschen. Die geänderte Tastenbelegung geht auch aus Abb. 2 hervor.

Die Tasten-Definitionsdatei können Sie folgendermaßen anlegen: Nach dem Laden des Betriebssystems starten Sie mit dem Befehl A>rped.sub (oder A>basic rped) den auf derselben Diskettenseite vorhandenen Basic-Editor (RPED.BAS) und folgen dann den Anweisungen auf dem Bildschirm. Nachdem Sie einen Namen für die neu zu erstellende Datei eingegeben haben (z.B. TASTEN.001), erscheint der eigentliche Bildschirm-Editor und Sie brauchen nur noch die Datei-Zeilen aus Abb. 1 fehlerfrei abzutippen. (Fehlerkorrektur, Zeileneinschübe usw. sind bei diesem Editor kein Problem und in der Kopfleiste ausreichend erklärt). Falls Sie eine so erstellte Datei auf Ihre Startdiskette (vgl. Schneider International 6/86, S. 23ff.) kopieren und in die PROFILE.SUB-Datei die Befehlszeile SETKEYS TASTEN.001 (ebenfalls mit dem RPED.BAS-Editor!) einfügen, wird die Tastatur Ihres JOYCE bereits beim AUTO-START (s.u.) umdefiniert.

Übrigens: Durch Betätigen von ALT + ENTER können Sie die Tastatur auf Großschreibung umstellen: alle eingegebenen Buchstaben werden dann in der Großschreibung wiedergegeben, die anderen Zeichen werden »normal« ausgegeben. Nach erneuter Betätigung der gleichen Tastenkombination funktioniert die gesamte Tastatur wieder »normal«. (So läßt sich die Tastatur auch bei LocoScript umstellen!).

Und so werden selbstdefinierte Tastaturdateien in eine CP/M-Startdiskette integriert: Abb.3 zeigt als Directory-Listing eine Zusammenstellung von CP/M-Dienstprogrammen, die für Sie ganz nützlich sein könnten, wenn Sie BASIC-Programme erstellen oder bearbeiten wollen. Neben den Dienstprogrammen, die Sie von der mitgelieferten Systemdiskette auf Ihre Startdiskette kopieren können, müssen Sie noch die Dateien E.SUB, H1, PROFILE.SUB und TASTEN.001 selbst anlegen und auf dieser Diskette speichern:

Die Datei E.SUB enthält lediglich die Befehlszeile BASIC RPED und ermöglicht es, unter CP/M den RPED.BAS-Editor durch Betätigen von [e]+[RETURN] zu starten (vgl. Abb. 2). Die Datei H1 (ohne weitere Angaben!) enthält den Text der Abb. 2, der zur Information über die Tastenbelegung mit <type h1> auf den Bildschirm geholt werden kann bzw. dort nach dem AUTO-START erscheint. Abb. 4 zeigt Ihnen, wie die PROFILE.SUB-Datei aussehen muß, damit Ihr JOYCE beim Start einigermaßen »benutzerfreundlich« eingestellt wird. Wie man die Startdiskette am einfachsten zusammenstellt, können Sie im Heft 6 von Schneider International oder im JOYCE-Handbuch nachlesen.

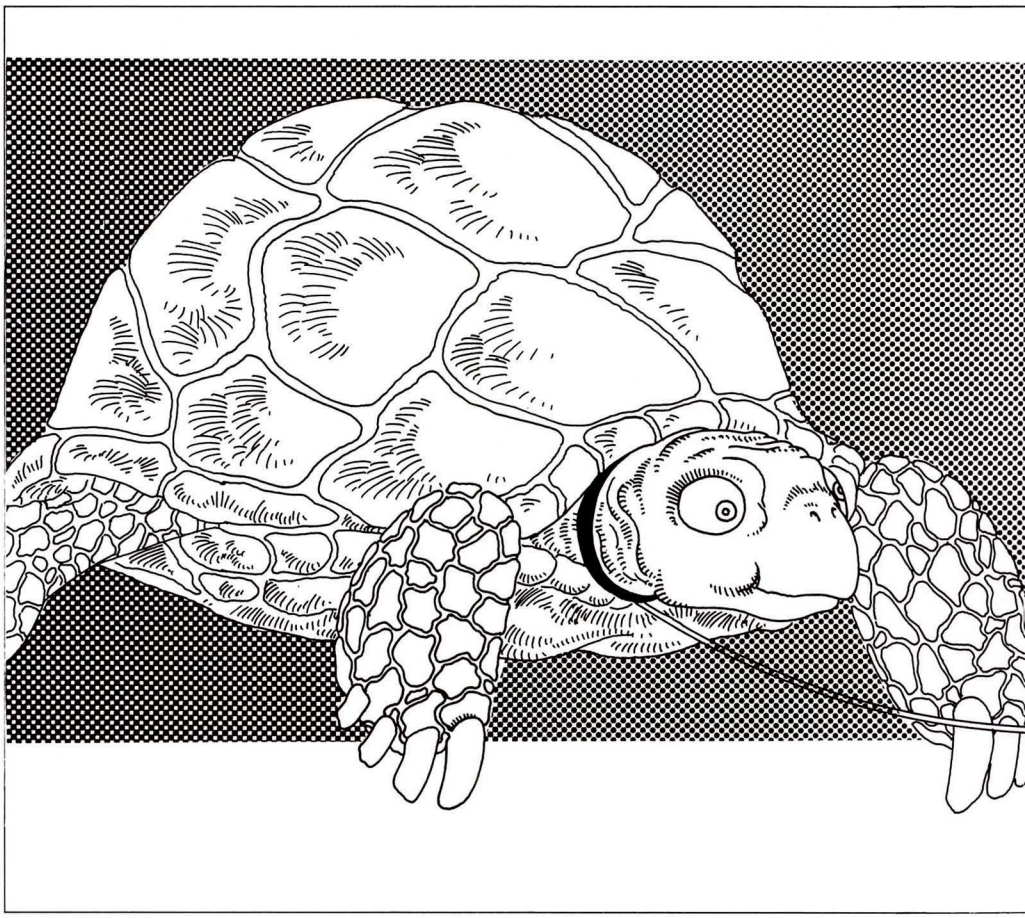
(Klaus Wiltsch)

```
ESC U=einf Zeile | AUSBL=DEL Zeile | F1 F2 F3 ESC DEL ZEILE ↵ RETURN | EXIT STOP
PALETTE 1 0
SETKEYS TASTEN.STD
SETDEF M:,A:&ORDER=(SUB,COM) TEMPORARY=M:U
PIP
<M:=basic.com
<M:=dir.com
<M:=e.sub
<M:=pip.com
<M:=rename.com
<M:=rped.bas
<M:=submit.com
<M:=H
<
TYPE H
```

Bild 3: Beispiel einer PROFILE.SUB-Datei für Ihre Startdiskette.

```
Directory For Drive A:  User 0

    Name     Bytes  Recs  Attributes      Name     Bytes  Recs  Attributes
------------ ------ ----- ----------  ------------ ------ ----- ----------
BASIC    COM   28k   224  Dir RO      DIR      COM   15k   114  Dir RW
E        SUB    1k     1  Dir RW      ERASE    COM    4k    29  Dir RW
H              2k    12  Dir RW      J14GCPM3 EMS   40k   320  Dir RO
PALETTE  COM    1k     8  Dir RW      PIP      COM    9k    68  Dir RW
PROFILE  SUB    1k     2  Dir RW      RENAME   COM    3k    23  Dir RW
RPED     BAS    7k    56  Dir RW      SET      COM   11k    81  Dir RW
SET24X80 COM    1k     8  Dir RW      SETDEF   COM    4k    32  Dir RW
SETKEYS  COM    2k    16  Dir RW      SETLST   COM    2k    16  Dir RW
SUBMIT   COM    6k    42  Dir RW
TASTEN   STD    2k    10  Dir RW      TYPE     COM    3k    24  Dir RW

Total Bytes     =   143k  Total Records =   1094  Files Found =   20
Total 1k Blocks =   143   Used/Max Dir Entries For Drive A:   23/  64
```

Bild 4: Ihre Startdiskette sollte diese Datei enthalten.



LOGO Grafik auf

Wenn mit JOYCE Balken- und Blockdiagramme oder geometrische Figuren gezeichnet werden sollen, muß man nicht unbedingt ein teures Grafik-Programm kaufen.

Füttert man LOGO mit den folgenden abgedruckten Prozedur-Listings, so lassen sich alle Zeichen-Wünsche, die auf einer Eck-Figur basieren, präzise und komfortabel realisieren.

Fehlende Farbigkeit wird bei allen Figuren durch stufenlos einstellbare Schraffuren ersetzt. Alle Grundfiguren mit Eckformen haben ihre eigene Prozedur, alle Elemente lassen sich bequem zu einem Gesamtbild ergänzen.

Gezeichnet wird immer senkrecht zur x-Achse des Zeichenstiftes (Schildkröte). Der Zeichenstift kehrt an den Ausgangspunkt zurück bzw. wandert an einen zum einfachen Weiterzeichnen geeigneten Punkt.

Mit HIDE TURTLE kann die Zeichengeschwindigkeit noch einmal erhöht werden. Mit SETSPLIT 2 erhält man einen optimalen Grafikbildschirm (genügend Bildfläche mit ca. 600 * 400 Einheiten und ausreichend Text-Platz zur Beobachtung von Eingaben und Fehlermeldungen).

Alle Graphiken lassen sich durch Drücken der EXTRA- und PTR-Taste oder mit Hilfe des Programms COPYPIC (siehe Schneider CPC International

```
Anhang: Sammlung der LOGO-Prozeduren

*******************************************************************************

to diagramm :beg :b1 :a :b :c :d :e :f :g
pu ht setx -300 sety :beg pd
rt 45 rechteck 30 ((((:b1 - 1) * 90) + 50) 0 45 135 lt 45
if :a < 0 [pu fd :a pd block (:a * (-1)) 2 pu bk :a pd]
if :a > -1.e-06 [lösch :a block :a 2]
if :b < 0 [pu fd :b pd block (:b * (-1)) 3 pu bk :b pd]
if :b > -1.e-06 [lösch :b block :b 3]
if :c = 1 [stop] []
if :c < 0 [pu fd :c pd block (:c * (-1)) 4 pu bk :c pd]
if :c > -1.e-06 [lösch :c block :c 4]
if :d = 1 [stop] []
if :d < 0 [pu fd :d pd block (:d * (-1)) 5 pu bk :d pd]
if :d > -1.e-06 [lösch :d block :d 5]
if :e = 1 [stop] []
if :e < 0 [pu fd :e pd block (:e * (-1)) 6 pu bk :e pd]
if :e > -1.e-06 [lösch :e block :e 6]
if :f = 1 [stop] []
if :f < 0 [pu fd :f pd block (:f * (-1)) 7 pu bk :f pd]
if :f > -1.e-06 [lösch :f block :f 7]
if :g = 1 [stop] []
if :g < 0 [pu fd :g pd block (:g * (-1)) 8 pu bk :g pd]
if :g > -1.e-06 [lösch :g block :g 8]
pu home
end

*******************************************************************************

to lösch :v
if :v = 0 [pe rt 45 fd 30 rt 45 fd 50 rt 135 fd 30 rt 45 fd 50 rt 90 pd]
if :v > 0 [pe rt 45 fd 30 rt 45 bk 22 fd 74 bk 2 rt 135 fd 30 rt 45 fd 50
rt 90 pd]
end

*******************************************************************************

to block :höhe :schraffur
if :höhe > 0 [rechteck :höhe 50 :schraffur 90 90]
fd :höhe rt 45
rechteck 30 50 (:schraffur + 2) 45 135
lt 45 bk :höhe rt 90 fd 50 lt 90
if :höhe > 0 [rechteck :höhe 30 (:schraffur + 2) 45 135]
if :schraffur = 0 [rt 90 pu fd 40 pd lt 90]
if :schraffur = 0 [stop] []
if :schraffur < 5 [if :höhe > 0 [pe fd :höhe rt 45 fd 30 bk 30 lt 135 fd 50
bk 50 rt 90 bk :höhe pd]]
if :schraffur > 4 [if :höhe > 0 [lt 90 fd 1 rt 90 fd :höhe bk :höhe rt 90
fd 1 lt 90]]
rt 90 pu fd 40 pd lt 90
end
```

```
*******************************************************************************

to rechteck :höhe :breite :schraffur :winkela :winkelb
make "x item 1 tf
make "y item 2 tf
make "z item 3 tf
make "breit (:breite / 2)
make "schraf (:schraffur / 2)
fd :höhe rt :winkela fd :breite rt :winkelb fd :hihe rt :winkela fd :breite
if :schraf > 0 [repeat ((((:breite - :schraf) / :schraf) + 1) / 2) [bk
:schraf rt :winkelb fd :höhe rt :winkela fd :schraf lt :winkela bk :hihe lt
:winkelb]]
if :schraf = 0 [rt :winkelb]
if :schraf = 0 [stop] []
pu setx :x sety :y seth :z pd
end
make "z 0
make "y 0
make "x 290
make "schraf 5
make "q 6
make "breit 15

*******************************************************************************

to ringe :radius :segment :abstand :ende
make "x item 1 tf
make "y item 2 tf
make "z item 3 tf
make "seg (360 / :segment)
make "umfang (:radius * 2)
if :radius < :ende [stop] []
if :umfang < 2 [stop] []
pu fd :radius rt 90 pd
repeat (:umfang / :seg) [fd 3.14159 rt (360 / :umfang)]
pu setx (:x) sety (:y) seth (:z) pd
ringe (:radius - :abstand) :segment :abstand :ende
end
make "umfang 70
make "z 0
make "y 0
make "x 0
make "seg 8

*******************************************************************************

*******************************************************************************

to säule :höhe :schraffur
make "q item 1 tf
make "r item 2 tf
pu ht fd (:höhe + 25) pd
zirkel 25 360
fill
```



Joyce

– August '86) zu Papier bringen. Bei COPYPIC muß allerdings zuerst die erstellte Grafik mit SAVEPIC »Dateiname« gespeichert werden.

Achten Sie aber darauf, daß eine solche Bilddatei unabhängig von der Bildgröße 23 kByte auf Ihrer Diskette verschlingt.

Zum Abtippen der Prozeduren verwenden Sie am besten den Befehl ED "Prozedur-Name". Vergessen Sie nicht, nach dem Probedurchlauf die Prozedur abzuspeichern mit SAVE "Prozedur-Name". Welche Figuren stehen nun zur Verfügung ?

Das Quadrat:
Um ein Quadrat zu zeichnen, wird die RECHTECK-Prozedur aufgerufen.

Für die Variablen HÖHE und BREITE wählt man einen beliebigen Wert ungleich Null mit HÖHE gleich BREITE. Das Quadrat kann man mit SCHRAFFUR gleich Null unschraffiert lassen, oder mit einem beliebigen Wert für diese Variable schraffieren. Die Eingabe für alle WINKEL-Variablen muß 90 Grad sein.

Das Rechteck:
Die RECHTECK-Prozedur aufrufen, beliebige Werte für HÖHE und BREITE ungleich Null eingeben, die gewünschte SCHRAFFUR anwählen und für die WINKEL-Variablen jeweils 90 Grad eingeben.

Das Parallelogramm:
Wiederum die RECHTECK-Prozedur laden und die gleichen Werte wie für ein Rechteck eingeben. Nur kann man nun die WINKEL-Variablen ungleich 90 Grad einstellen. Addiert müssen diese Werte aber 180 bei positiven Gradzahlen, oder -180 bei negativen Gradzahlen, ergeben.

Das Trapez:
Die TRAPEZ-Prozedur anwählen, sich für beliebige Werte für die BODEN- und DACH-Breite ungleich Null entscheiden, einen STEIGungswert ungleich Null bestimmen und schließlich schraffieren.

Beachten Sie: Die Gesamtbreite des Trapezes ist (2 * BODEN) + DACH. Die Höhe des Trapezes ergibt sich aus der Multiplikation von STEIGung und BODEN-Breite.

Das gleichschenklige Dreieck:
Erneut ruft man hierzu die TRAPEZ-Prozedur auf. Einziger Unterschied zum Trapez bei der Bestimmung der Variablen ist, das die Breite des DACHes gleich Null gesetzt wird. Die Gesamtbreite des Dreiecks ergibt sich aus der Verdoppelung des Wertes für die BODEN-Breite. Alle anderen Werte entsprechen denen des Trapezes.

```
pu bk :schraffur seth 180
lt 90 fd 2 rt 90
repeat (:höhe / :schraffur) [schraf (:höhe / 2) :schraffur 180]
pu setx :q sety :r seth 0 seth 90 fd 80 lt 90 pd
end

*******************************************************************************

to schraf :höhe :schraffur :segment
seth 90
zirkel 25 :segment
seth 180 pu fd :schraffur pd
end

*******************************************************************************

to zirkel :radius :segment
make "x item 1 tf
make "y item 2 tf
make "z item 3 tf
make "seg (360 / :segment)
make "umfang (:radius * 2)
pu fd :radius rt 90 pd
repeat (:umfang / :seg) [fd 3.14159 rt (360 / :umfang)]
pu setx (:x) sety (:y) seth (:z + :segment) pd
end
make "umfang 50
make "z 90
make "y 25
make "x 262
make "r 0
make "q 260
make "seg 2
make "radius 180

*******************************************************************************

to zirkel :radius :segment
make "x item 1 tf
make "y item 2 tf
make "z item 3 tf
make "seg (360 / :segment)
make "umfang (:radius * 2)
pu fd :radius rt 90 pd
repeat (:umfang / :seg) [fd 3.14159 rt (360 / :umfang)]
pu setx (:x) sety (:y) seth (:z + :segment) pd
end

*******************************************************************************

to ringe :radius :segment :abstand :ende
make "x item 1 tf
make "y item 2 tf
make "z item 3 tf
```

```
make "seg (360 / :segment)
make "umfang (:radius * 2)
if :radius < :ende [stop] []
if :umfang < 2 [stop] []
pu fd :radius rt 90 pd
repeat (:umfang / :seg) [fd 3.14159 rt (360 / :umfang)]
pu setx (:x) sety (:y) seth (:z) pd
ringe (:radius - :abstand) :segment :abstand :ende
end

*******************************************************************************


to torte :a :b :c :d :e :f :g :h :i :j
zirkel 100 360
fd 100 bk 100 rt :a fd 101 bk 101 lt (:a / 2) pu fd 50 pd fill pu bk 50 rt
(:a / 2)
if :b > 0 [ringe 97 :b 3 0 rt :b fd 100 bk 100]
if :c > 0 [ringe 94 :c 6 0 rt :c fd 100 bk 100]
if :d > 0 [ringe 91 :d 9 0 rt :d fd 100 bk 100]
if :e > 0 [ringe 88 :e 12 0 rt :e fd 100 bk 100]
if :f > 0 [ringe 85 :f 15 0 rt :f fd 100 bk 100]
if :g > 0 [ringe 82 :g 18 0 rt :g fd 100 bk 100]
if :h > 0 [ringe 79 :h 21 0 rt :h fd 100 bk 100]
if :i > 0 [ringe 76 :i 24 0 rt :i fd 100 bk 100]
if :j > 0 [ringe 73 :j 27 0 rt :j fd 100 bk 100]
seth 0
end
make "umfang 104
make "z 45
make "y 0
make "x 0
make "seg 1.14285714285714

*******************************************************************************
*******************************************************************************

to zirkel :radius :segment
make "x item 1 tf
make "y item 2 tf
make "z item 3 tf
make "seg (360 / :segment)
make "umfang (:radius * 2)
pu fd :radius rt 90 pd
repeat (:umfang / :seg) [fd 3.14159 rt (360 / :umfang)]
pu setx (:x) sety (:y) seth (:z + :segment) pd
end
make "umfang 100
make "z 0
make "y 50
make "x 100
make "seg 1

*******************************************************************************
```

Die Raute:
Tippen Sie load "raute", geben Sie dann eine beliebige BREITE größer Null ein, anschließend einen Wert für die STEIGung. Die Variable SENKung belegen Sie mit dem Wert der STEIGung. SCHRAFFUR gleich Null läßt die Raute unschraffiert. Schraffur größer Null schraffiert die Figur. Die Gesamtbreite der Raute ergibt sich aus der Verdoppelung des BREITE-Wertes, die Gesamthöhe aus der Multiplikation von BREITE und STEIGung mal 2.

Der Drachen:
Einzige Änderung zur Raute: Die SENKung wird hier ungleich zur STEIGung gewählt. Daraus ergibt sich auch eine andere Berechnung der Gesamthöhe. Diese errechnet sich beim Drachen aus der Formel (BREITE * STEIGung) + (BREITE * SENKung).

Das Vieleck:
Vielecke lassen sich mit der VIELECK-Prozedur zeichnen. Die Seitenlänge kann beliebig größer Null gewählt werden.

Für die Anzahl der Ecken muß zunächst eine kleine Berechnung angestellt werden. Es ist 360 durch die Anzahl der gewünschten Ecken zu teilen und das Ergebnis für die zweite Variable zu verwenden. Die dritte Variable ist immer 360, es sei denn, die Figur soll geöffnet sein.

Auch eine TORTEngraphik ist nun kein großes Problem mehr.

Koppelt man die ZIRKEL-Prozedur mit der Prozedur für RINGE, so erreicht man leicht das gesteckte Ziel.

Die Prozedur TORTE erstellt ein Diagramm mit bis zu elf Tortenstücken. Zehn Stücke werden als Variablen A bis J eingegeben und erhalten eine automatisch wechselnde Schraffur. Teilstück 11 ergibt sich aus der Differenz von 360 minus der Summe von A bis J. Es wird nicht schraffiert. Benötigt man aber z.B. nur fünf Stücke, so werden die verbleibenden Variablen einfach mit der Ziffer Null belegt.

Beachten Sie aber dabei Folgendes: Ist die Summe der angebenen Variablen ungleich 360, so ergibt sich ein nicht schraffiertes Feld in Größe der Differenz. Die für die Variablen eingegebenen Zahlenwerte entsprechen der Feldgröße in Grad des Kreissegmentes. Will man also Prozentwerte grafisch umsetzen, so sind die Prozentzahlen mit 3.6 zu multiplizieren, um auf die von der TORTEN-Prozedur verlangten Gradangaben zu kommen. Natürlich ist der Einbau einer automatischen Umrechnung in Prozent möglich, indem man in der Prozedur alle Eingaben mit 3.6 multipliziert. Die Prozedur TORTE wird dann aber auf die Umsetzung von Prozentanteilen beschränkt.

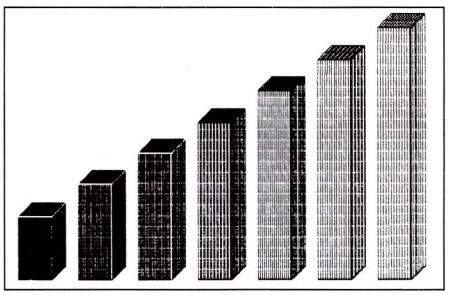


Bild 1: Blockdiagramm mit LOGO

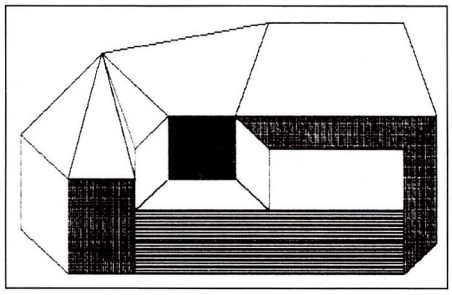


Bild 2: Eckfiguren zusammengesetzt

Die vielbenötigten Prozeduren ZIRKEL und RINGE sind auch allein zu benutzen. Mit ZIRKEL läßt sich in Abhängigkeit von den Variablen RADIUS und SEGMENT ein Kreis zeichnen. RINGE bringt eine Ringfigur auf den Bildschirm. Dabei kann durch RADIUS der Radius des äußersten Ringes und durch ENDE der Radius des innersten Ringes bestimmt werden. ABSTAND gibt die Radiusabnahme von Ring zu Ring an. SEGMENT schließlich ermöglicht die Auswahl des zu zeichnenden Ausschnitts.

Noch ein Tip zum Schluß: Beim Speichern der Prozeduren empfiehlt es sich, einige Dinge zu beachten, die später ein schnelleres Arbeiten ermöglichen. Wenn mit save "Prozedur-Name" eine Prozedur gespeichert werden soll, so werden von JOYCE nicht nur die angesprochenen Prozedurdefinitionen auf Diskette geschrieben, sondern auch alle anderen im Arbeitsspeicher befindlichen Daten unter dem angegebenen Namen mit abgelegt. So wird z.B. durch save "rechteck" die Prozedur RECHTECK und DIAGRAMM geladen, wenn sich beide Formeln im Arbeitsspeicher befinden. Dieses Verhalten von JOYCE hat Vor- und Nachteile.

Eck-Figuren mit LOGO

Figur	Eingaben	Beispiel
Quadrat	rechteck a,a,0 bis b,90,90	100 100 4 90 90
Rechteck	rechteck a,b,0 bis c,90,90	100 150 5 90 90
Parallelogramm	rechteck a,b,0 bis c,d,e	100 150 3 45 135
Trapez	trapez a,b,0.1 bis c,0 bis d	50 100 1 3
Dreieck(glschkl)	trapez a,0,0.1 bis b,0 bis c	50 0 1 3
Dreieck(links)	dreieck 0,0.1 bis a,b,c	0 2 3 150
Dreieck(rechts)	dreieck a,-0.1 bis -a,b,c	150 -2 3 0
Raute	raute a,0.1 bis b,0.1 bis b,0 bis c]	50 1 1 3
Drachen	raute a,0.1 bis b,0.1 bis c,0 bis d]	50 1 3 3
Fünfeck	vieleck a,72,360	
Sechseck	vieleck a,60,360	
Siebeneck	vieleck a,51.42,360	
Achteck	vieleck a,45,360	
Neuneck	vieleck a,40,360	
Zehneck	vieleck a,36,360	

Rechtwinklige Dreiecke lassen sich mit der DREIECK-Prozedur zeichnen, indem man nur eine rechte oder linke Hälfte eines beliebigen Dreiecks zeichnet.

Tabelle 1: Eckfiguren mit Logo

Arbeitet eine Formel selbständig, d.h. ohne den Gebrauch von Sub-Prozeduren, so ist es unsinnig, mehr als nötig auf Diskette zu speichern. Denn das kostet Speicherplatz und führt zu langen Wartezeiten beim Programmaufruf mit LOAD.

Ist aber die Prozedur, wie z.B. DIAGRAMM auf mehrere Sub-Prozeduren angewiesen, so empfiehlt es sich, diese mit abzulegen.

Dadurch spart man erheblich Zeit, denn alle für die Ausführung einer Prozedur benötigten Sub-Prozeduren werden vom Rechner gleich mit geladen. Schließlich sind sie auf diese Weise Bestandteil der Hauptprozedur. Das Opfern von Speicherplatz für diese Verfahrensweise lohnt sich, denn es ist wesentlich bequemer und schneller durch load "diagramm" die Prozedur einsatzbereit zu haben, als jede Sub-Prozedur und DIAGRAMM selbst einzeln aufzurufen.

Umrechnungstabelle von Steigungen in Winkel

Steigung	Winkel
0	0 Grad
0.25	75.96 Grad
0.5	63,44 Grad
0.75	53,13 Grad
1	45 Grad
2	26,57 Grad
3	18,44 Grad
4	14,04 Grad
5	11,31 Grad

Diese Umrechnungstabelle erleichert das Anlegen von Parallelogrammen an Trapeze. Dies ist z.B. für das Zeichnen dreidimensionalen Häusern nötig.

Tabelle 2: Umrechnung von Steigungen in Winkel

Um den Arbeitsspeicher nun vor dem Speichern einer Routine – ob mit oder ohne Sub-Routine – wie gewünscht zu präparieren, stehen folgende LOGO-Befehle zur Verfügung: ER-ALL für das Löschen aller Definitionen im Arbeitsspeicher. ER gefolgt vom Prozedur-Namen für das Löschen einzelner Prozeduren und LOAD zum Dazuladen.

Die Befehle POPS, POTS und PO-ALL helfen dabei, sich einen schnellen Überblick über die Inhalte des Arbeitsspeichers zu verschaffen. Mit ED "Prozedur-Name" wird eine Prozedur zum Bearbeiten vorbereitet. Nach soviel Formeln nun aber viel Erfolg mit Ihren LOGO-Graphiken.

(M.Böhmer)

Submit-Ausführung mit Basic

Mit Hilfe dieser Routine kann man jetzt vom BASIC in das Betriebssystem springen, dort beliebige CP/M-Befehle ausführen, und dann wieder ins BASIC zurückspringen. Zu beachten ist, daß die CP/M-Befehle in umgekehrter Reihenfolge der Abarbeitung in der Direktzugriffsdatei $$$.SUB stehen. Das Vorhandensein der Datei SUBMIT.COM auf der entsprechenden Diskette ist nicht erforderlich. Nach Abarbeiten der Datei $$$.SUB wird diese wieder auf der Diskette gelöscht.

E.O. Dumke

```
Beispiellisting von SUB.BAS                                    *****
===========================                                    *****
                                                               *****
10 REM *****  SUB.BAS                                          *****
15 REM *****  (c)   Ernst-Otto Dumke                           *****
20 REM *****
25 REM *****        D-6087 Büttelborn 2
30 REM *****
35 DIM a$(3)
40 a$(3)="pip m:=a:sub.bas":a$(2)="dir m:":a$(1)="basic"
45 OPEN "R",1,"a:$$$.sub",128:FIELD #1,128 AS b$
50 FOR i=1 TO 3:a$(i)=CHR$(LEN(a$(i)))+a$(i)+CHR$(0)
55 LSET b$=a$(i):PUT #1,i:NEXT i:CLOSE:SYSTEM
```

Für unsere ständige Joyce-Rubrik suchen wir noch

Programme
Tips + Tricks

zur Veröffentlichung. Honorar nach Vereinbarung.

Einsendungen an:
DMV Daten & Medien Verlagsges. mbH,
Fuldaer Str. 6, 3440 Eschwege

Computerstore GmbH
Hochstraße 11
8500 Nürnberg 80
Tel.: 09 11 / 28 90 28

PUBLIC-DOMAIN-Software:	
alle Programme mit dt. Anleitungen	
XLISP & E-PROLOG	30,–
JRT-PASCAL	30,–
ASSEMBLER-Paket	30,–
CP/M-Hilfsprogramme	30,–
SMALL-C	30,–
FORTH 83	30,–
ALLE 6 DISKETTEN mit Anl.	180,–
3"-Markendisketten 10 ST	75,–
RAM-Erweiterung512K	80,–
Zweitlaufwerk FD-2	598,–
NACHRÜSTSATZ FD-2 & RAM	668,–
Typenraddrucker SD 15	698,–
Programm LOCO 15	59,–
MOUSE-PACK	248,–
Schnittstelle CPS 8256	148,–
JOYSTICK-Controller	98,–
dto. mit SOUND & Lautspr.	139,–



INFOCOM-Software:	
Wishbringer	69,–
Ballyhoo	79,–
Zork I, II, III	je 79,–
STARGLIDER	74,90
CYRUS II Chess	49,–
SUPERCOPY	85,–
VOKABELTRAINER	59,–
DR DRAW (dt.)	199,–
DR GRAPH (dt.)	199,–
TURBO-PASCAL	199,–
WORDSTAR, dBase, MULTIP.	je 199,–
NEVADA-FORTRAN (dt.)	109,–
NEVADA-COBOL (dt.)	109,–
Z-ASS (Macro-Assembler)	109,–
CRACKER (Kalkulation)	174,–
DATAMAT	298,–
FINANZBUCHHALTUNG	194,–
C-BASIC-Compiler 80	174,–
TASWORD 8000 (dt.)	148,–

Wir führen zu den Original-SCHNEIDER-Produkten Artikel verschiedener Firmen wie DATA BECKER, VORTEX, STAR-DIVISION, CUMANA, GERDES, PROFISOFT, ARIOLA, RUSH-WARE, MARKT & TECHNIK, SYBEX, STAR, SCHNEIDER-DATA usw.!

JOYCE-Tips

Ich programmiere nun schon längere Zeit und Mallard Basic auf meinem Joyce. Durch die hohe Arbeitsgeschwindigkeit und hervorragende Datei-Verwaltung mit Jetsam bin ich damit sehr zufrieden.

Einige kleine Probleme gab es aber doch:

1) Der Input Befehl ist sehr unkomfortabel. Mit »Bildschirm-Input« aus Heft 10/86 kann man das Problem sehr komfortabel lösen; diese Routine belegt mir jedoch zu viel Speicherplatz.

2) Eine Uhr läßt sich nicht realisieren, obwohl die Routinen dafür im BDOS des CP/M Plus vorhanden sind.

3) Ich betreibe einen DMP-2000 zusätzlich am Joyce, um Formulare zu bedrucken. Am Joyce-Drucker ist mir die Möglichkeit des automatischen Papiereinzugs und der vielfältigen Schriftgestaltungsmöglichkeiten (z.B. Elite in NLQ) wichtig, die ich auch benutzen möchte. Ein Umschalten zwischen den beiden Druckern ist in Basic jedoch nicht möglich.

Diese Probleme beseitigt das untenstehende Maschinensprach-Programm:

Nach Aufruf des Basic-Laders mit »GOSUB 65520« werden die Probleme werden damit wie folgt gelöst:

1) Bei der Input-Routine wird wie folgt verfahren:

Zuerst wird in die Speicherstelle &H80 die maximale Eingabelänge und in Speicherstelle &H82 der Wert 0 geschrieben. Hier liegt der Edit-Puffer für CP/M. Da Basic diesen jedoch verschiebt, muß der Puffer durch Aufruf von &HF5CD wieder an diese Stelle geschoben werden. Dann ruft man die Routine &HF5C4 auf und die Eingabe kann erfolgen. Es stehen alle Editiermöglichkeiten mit <−, −>, ZEILE,EZ, DEL−>, <−DEL und AUSBL zur Verfügung. Beim Drücken von RETURN steht die Eingabe ab der Speicherstelle &H82. Die wirkliche Länge der Eingabe steht in der Speicherstelle &H81. Für mich erledigt folgendes kleines Unterprogramm diese Aufgabe. Vor dem Aufruf muß maxlen die maximale Länge der Eingabe enthalten. Hinterher enthält x$die Eingabe.

```
10000 x$=""
10010 POKE &H80,maxlen:POKE
      &H82,0
10020 CALL dskres%:CALL inp%
10030 FOR yyy=&H82 TO &H81
      +PEEK(&H81):x$=x$+
      CHR$(PEEK(yyy)): NEXT
10040 RETURN
```

Das neue Input bietet aber noch mehr! Man kann auch schon vorhandene Zeichenketten verbessern. Dazu muß ähnlich verfahren werden. Allerdings entfällt der POKE &H82,0 und in die Speicherstelle ab &H82 wird der zu verbessernde Inhalt des Strings geschrieben. Darauf muß eine 0 folgen! Hier wieder eine kleine Basic Routine. maxlen enthält wieder die maximale Länge, und x$ enthält den alten String.

```
10500 CALL dskres%:POKE &H80,
      maxlen:maxlen:x$=x$+
      CHR$(0)
10510 FOR yyy=1 TO LEN (x$):
      POKE &H81+yyy,ASC(MID
      $(x$,yyy)):NEXT
10520 CALL inp%
10530 x$=" ":FOR yyy=&H82 TO
      &H81+PEEK(&H81):x$=x$
      +CHR$(PEEK(yyy)):NEXT
10540 RETURN
```

Falls CTRL-S und CTRL-C wirkungslos bleiben sollen, ist am Anfang des Programms einzugeben CALL disable%.

2) Die Benutzung des Datums im internen Timer ist meiner Meinung nach nicht sehr sinnvoll. Wer läßt seinen JOYCE schon tagelang angeschaltet und benötigt eine automatische Datumsumschaltung! Falls jemand diese Möglichkeit trotzdem nutzen möchte: Das Datum besteht aus einer 16-Bit-Zahl, die die Anzahl Tage, die seit dem 1.1.78 vergangen sind, angibt. Diese sinnvoll zu verwerten, ist ein Problem für sich.

Die eingebaute Uhr zu benutzen, ist hingegen kein Problem: Um den Timer zu stellen, schreibt man in die Speicherstelle &HF5A4 die Stunden, in &HF5A5 die Minuten und ruft danach &HF5A6 auf. Das Datum, falls es jemand benutzt, steht in der o.a. Form in den Speicherstellen &HF5A2 und &HF5A3 in der Form Low-Byte − High-Byte.

Allerdings sind die Minuten u. Stunden BCD-codiert, was aber kein großes Problem darstellt. Die folgende Basic-Zeile stellt den Timer:

```
POKE &HF5A4, VAL("&H"+
stunde$):POKE &HF5A5,VAL
("&H"+minute$):CALL dset%
```

Falls Std. und Min. nicht als Zeichenkette vorliegen, müssen sie vorher mit STR$ umgewandelt werden.

Der Timer wird ähnlich gelesen: Nach Aufruf von &HF5B3 enthält &HF5B1 die Stunden und &HF5B2 die Minuten. In Basic:

```
CALL dget%:stunde$=HEX$
(PEEK(&HF5B1)):minute$=HEX$-
(PEEK(&HF5B2))
```

3) Um den Drucker auf DMP umzuschalten, muß erst der AUXOUT:-Kanal für das Gerät LST: zur Verfügung gestellt werden. Dies geschieht mit Hilfe des DEVICE-Programmes auf Systemdiskette 3. Eingabe unter CP/M:

```
A>DEVICE AUXOUT:=CEN
```

Danach kann in Basic der DMP 2000 durch OPTION LPRINT=dmp% angesprochen werden. Durch OPTION LPRINT kann wieder auf den JOYCE-Drucker zurückgeschaltet werden.

Dies alles funktioniert natürlich nur nach Ablaufen des Basic-Laders durch Aufruf mit »GOSUB 65520«.

(Patrick Bosch)

```
65520 REM --- DATA ---
65521 DATA 5C,0C,00,00,0E,68,11,A2,F5,CD,05,00,C9   'SET DATE AND TIME
65522 DATA 00,00,00,00,0E,69,11,AF,F5,CD,05,00,C9   'GET DATE AND TIME
65523 DATA 59,0E,04,CD,05,00,C9,00                  'PRINT TO AUX
65524 DATA 11,00,00,0E,0A,CD,05,00,C9               'INPUT
65525 DATA 0E,0D,CD,05,00,C9                        'DMA-BUFFER TO &H80
65526 DATA 11,0A,00,0E,6D,CD,05,00,C9               'CTRL-C,-S DISABLED
65527 MEMORY &HF5A1:RESTORE 65521:FOR I=&HF5A2 TO &HF5DB:READ A$:POKE
I,VAL("&H"+A$):NEXT
65528 dset%=&HF5A6:dget%=&HF5B3:dmp%=&HF5BC:inp%=&HF5C4:dskres%=&HF5CD
65529 disable%=&HF5D3
65530 RETURN
```

Noch'n Thema: FUSSNOTEN

In Heft 5'86 hat ein Herr Straub aus Berlin geklagt, daß das Textprogramm Locoscript so unkomfortabel sei und keine automatische Fußnotenverwaltung enthalte.

Da ich mein Maschinchen zum gleichen Zweck brauche (hauptsächlich wissenschaftliche Texte), kann ich in dieses Klagegeheul nur aus tiefstem Herzen einstimmen. Aber wenn schon, dann bitte auch: automatischen Index, automatisches Inhaltsverzeichnis, und Silbentrennung/Trennhilfe, Desktop, Mehrspaltenbearbeitung und Rechtschreibprüfung; und alles blitzschnell auf dem Schirm. Ich hör' die Programmierer stöhnen. Ist ja recht, CP/M auf »Z80 A« hat seine Grenzen, aber Fußnoten und Trennhilfe wären wirklich nicht nur Luxus, sondern sehr hilfreich (liest irgendwer von »Locomotive« eigentlich die entsprechenden Leserbriefe und hört sich das bei den Händlern anfallende Gejammer an?).

Zurück zu den Fußnoten

Man schreibt seinen Text so vor sich hin, und wenn eine Fußnote auftaucht, tippt man, wie mit der Schreibmaschine auch, an der entsprechenden Stelle eine HOCHGESTELLTE Fußnotennummer. Damit diese später im gedruckten Text gut zu sehen ist, kann/sollte man sie gleichzeitig FETT hervorheben. Mit nur »fett« allerdings ist sie auf dem Bildschirm schlecht zu sehen, weswegen man sie (einschließlich des nachfolgenden Leerzeichens) für die Arbeit am Schirm auch gleich noch INVERS hervorheben sollte. Also: HOCH, FETT und INVERS.

Als nächstes tippt man in aller Seelenruhe den Absatz zuende, ohne sich weiter um die Fußnote zu kümmern. Ist das geschehen, fügt man direkt unter dem Absatz, in dem die Fußnotennummer steht, die eigentliche Fußnote (Fußnotentext) an. Das heißt, daß die Fußnote (zunächst!) INNERHALB des Textes steht und nicht am Ende der entsprechenden Seite. Damit man sie auf dem Bildschirm später gut vom Text unterscheiden kann, markiert man die GESAMTE Fußnote, wie die Fußnotennummer vorher auch, INVERS.

Als nächstes tippt man in aller Seelenruhe den Absatz zuende, ohne sich weiter um die Fußnote zu kümmern. Ist das geschehen, fügt man direkt unter dem Absatz, in dem die Fußnotennummer steht, die eigentliche Fußnote (Fußnotentext) an. Auf diese Weise bleiben die Fußnoten bei den Textstellen, zu denen sie gehören.

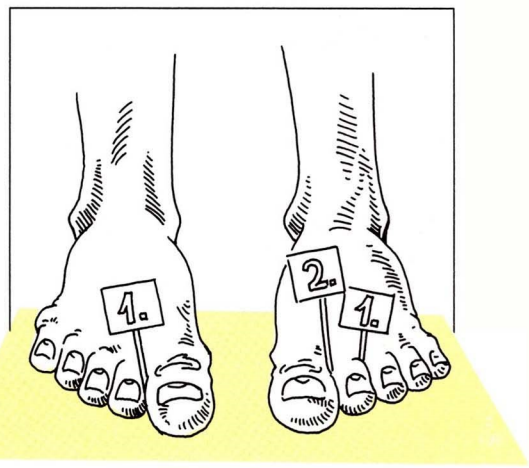
Ist der Text abgeschlossen und korrigiert, werden die Fußnoten mit

COPY / AUSBL / EINBL

an das Ende der jeweiligen Textseite verschoben. Dadurch, daß die Fußnoten zuvor im Text gestanden haben, ist gewährleistet, daß sie beim Verschieben exakt den Fuß der Seite füllen. Jetzt noch einige wichtige Feinheiten:

Fußnotenlayout

Prinzipiell kann man natürlich für die Fußnoten jede beliebige Schrift wählen. Allerdings sollte man sich darüber



im klaren sein, wie diese später im Schriftbild aussieht. Am besten experimentiert man ein wenig. Mit ein paar Textproben mit verschiedenen Schrifttypen, Zeilenabständen und Zeilendichten hat man ganz schnell eine zufriedenstellende Kombination gefunden.

Ich habe die besten Erfahrungen mit der 17-er Schrift gemacht. Diese kann hochgestellt sein, was ich empfehle, da die nicht hochgestellte Version dieser Schrift gegenüber dem Haupttext zu »laut« wirkt und etwas in den Vordergrund tritt. Auch kann sie mit einer Zeilendichte von 8 Zeilen/Zoll und einem Zeilenabstand von 1 eingege-

So sieht der Text **VOR** dem Einfügen der Fußnote aus:

Man schreibt seinen Text so vor sich hin, und wenn eine Fußnote auftaucht, tippt man, wie mit der Schreibmaschine auch, an der entsprechenden Stelle eine **HOCHGESTELLTE** Fußnotennummer. Etwa so wie hier:[1] Damit diese später im gedruckten Text gut zu sehen ist, kann/sollte man sie gleichzeitig **FETT** hervorheben.[2] Mit nur 'fett' allerdings ist sie auf dem Bildschirm schlecht zu sehen, weswegen man sie für die Arbeit am Schirm auch noch gleich **INVERS** hervorheben sollte. Also: **HOCH, FETT und INVERS.**

So **NACH** dem Einfügen der Fußnote:

Man schreibt seinen Text so vor sich hin, und wenn eine Fußnote auftaucht, tippt man, wie mit der Schreibmaschine auch, an der entsprechenden Stelle eine **HOCHGESTELLTE** Fußnotennummer. Etwa so wie hier:[1] Damit diese später im gedruckten Text gut zu sehen ist, kann/sollte man sie gleichzeitig **FETT** hervorheben.[2] Mit nur 'fett' allerdings ist sie auf dem Bildschirm schlecht zu sehen, weswegen man sie für die Arbeit am Schirm auch noch gleich **INVERS** hervorheben sollte. Also: **HOCH, FETT und INVERS.**

1 Diese Fußnote ist mit einer Zeichendichte von 17 Zeichen/Zoll und einer Zeilendichte von 8 Zeilen/Zoll geschrieben, Zeilenabstand 1, Schrift hochgestellt. Zu beachten wäre noch die Layoutgestaltung und das Setzen der Ränder. Das kommt später.

ben werden. So sind die Fußnoten sehr platzsparend gestaltet und trotzdem noch gut lesbar.

Diese Werte (Schriftbreite, Zeilendichte und Zeilenabstand) legt man im Kopfinfo fest:

f2 – LAYOUT EINFÜGEN/
LAYOUT ÄNDERN

und gibt dem Fußnotenlayout eine Nummer (bequemlichkeitshalber die 1).

Bei jedem Einfügen einer Fußnote wird dann nur noch die entsprechende Layoutnummer aufgerufen,

f2, LAYOUT EINFÜGEN

der Fußnotentext eingegeben und mit der Funktion

(-LayouT) und RETURN

abgeschlossen.

Falls man zur Übersicht unter

f1 = ANZEIGE

die Layoutlineale anzeigen läßt, darf man sich nicht wundern, wenn das Lineal erst UNTER der ersten Fußnotenzeile steht (auch aus diesem Grund ist das Markieren der ganzen Fußnote invers sinnvoll). Die erste Zeile wird trotzdem korrekt gedruckt.

Da im normalen Textlayout (Basislayout) mit einer kleineren Zeichendichte (z.B. Proportionalschrift) geschrieben wird als im Fußnotenlayout, die Anzahl der möglichen Zeichen/Zeile in der Fußnote bei gleicher Randeinstellung aber leicht über den rechten Bildschirmrand reichen kann, ist es sinnvoll, ZUERST die Randeinstellung der Layouts so zu wählen, daß im Fußnotenlayout MAXIMAL 90 Zeichen zwischen die Ränder passen (weniger dürfen es sein, aber mehr wären ungünstig für die Arbeit am Bildschirm).

Bei Verwendung von z.B. Proportionalschrift im Basislayout und 17-er Schrift in den Fußnoten wäre also eine Textbreite von 62 Zeichen (im Basislayout) angebracht.

Zur Berechnung

- In der Fußnote maximale Zeichenzahl = 90.
- 90 / Zeichendichte der gewählten Fußnotentype (in diesem Fall 17) = Breite des Textes in Zoll (hier 5,2).
- Diese Zollzahl x Zeichendichte Haupttext (hier Proportionalschrift = ca 12) ergibt die Textbreite (hier 5,2 x 12 = 62,4) und damit die Randeinstellung für die Layouts.
- Dabei ist es noch sinnvoll, in den Layouts (die ja die gleiche Randeinstellung haben) den LINKEN RAND auf 0 (!) zu setzen, den rechten Rand der ermittelten Textbreite entsprechend. Das bewirkt, daß der Fußnotentext mit bis zu 90 Zeichen/Zeile KOMPLETT auf dem Bildschirm angezeigt wird und nicht über den rechten Rand des Schirmes schreibt.

Damit der Text dann beim Ausdrucken ordentlich in der Mitte der Seite steht, und nicht direkt am linken Rand gedruckt wird wegen der Randeinstellung im Layout, gibt man vor dem Ausdrucken des Textes in der Druckerkontrollstellung PTR unter f3 = AKTIONEN einen entsprechenden linken Abstand für den Ausdruck ein.

Um sicherzustellen, daß eine Fußnote, auch wenn sie auf einer Textseite ziemlich weit unten auftaucht, noch auf diese Seite paßt oder aber komplett mit dem zugehörigen Absatz/Teil des Absatzes auf die nächste Seite kommt, kann man am Ende der Fußnote noch den Befehl HALTEN (siehe Handbuch) und die entsprechende Zeilenzahl (Anzahl der Zeilen der Fußnote + Zeilananzahl des zugehörigen Absatzes) eingeben.

Um die Fußnoten optisch vom Text zu trennen, kann man zu guter Letzt beim Verschieben der Fußnoten aus dem Text an das Ende der Seite noch einen trennenden Strich UNTERSTRICHENE LEERZEILE = +U / HARD-SPACES / -U zwischen Text und Fußnoten einfügen.

Wichtig dabei ist, daß diese Trennlinie UNBEDINGT mit HARD-SPACES (ca 10-12 Stck.) erstellt werden muß. Ist das nicht der Fall und besteht sie nur aus normalen Leerzeichen, wird sie vom Programm nicht als eigenständige Zeile eingefügt, sondern an das Ende der vorhergehenden Zeile angehängt. Außerdem wird sonst auch ein am Ende der vorhergehenden Zeile stehendes Soft-Trenn nicht berücksichtigt beim Ausdrucken.

Damit das nicht zur endlos wiederholten Fummelei ausartet, macht man diese unterstrichene Leerzeile einmal und legt sie als SATZ dauerhaft ab (Leerzeile herstellen; COPY / AUSBL / Satzbuchstabe / f8 = BLOCK / ALLE SÄTZE SPEICHERN / ENTER).

Beim Verschieben der Fußnoten an das Ende der Seite blendet man dann nur noch diese so abgelegte Zeile mit EINBL / Satzbuchstabe ein.

Dabei ist aber zu berücksichtigen, daß diese Zeile Platz braucht, die Zeile und die Fußnoten also ZWISCHEN der vorletzten und der letzten Zeile der Seite eingefügt werden müssen (die Zeile war ja zuvor nicht im Text vorhanden).

Zum Schluß noch einmal alles in der richtigen Reihenfolge und in Kurzfassung:

1. Festlegen des Basislayouts und des Fußnotenlayouts

Berechnen, wieviel Zoll von 90 Zeichen der gewählten Fußnotenschrift eingenommen werden: 90 / Zeichendichte Fußnotenschrift = Anzahl Zoll.

Berechnen, wieviel Zeichen der gewählten Textschrift auf diese Anzahl Zoll passen: Anzahl Zoll x Zeichendichte Textschrift = maximale Textbreite Basislayout. Ist die so ermittelte Textbreite zu groß (zu kleine Ränder rechts und links auf dem Ausdruck), wählt man sie einfach entsprechend kleiner.

Festlegen der Ränder im Basislayout (die brauchen dann im Fußnotenlayout nicht mehr geändert zu werden!):

f7 / Kopfinfo ändern / f7 / f1.

Linken Rand auf 0, rechten Rand entsprechend der ermittelten Textbreite setzen.

Sonstige Werte der Layouts in der Kopfinfo festlegen (Schriftdichte, Zeilendichte, Zeilenabstand, Kursiv an/aus. Für das Fußnotenlayout geht das nicht über f7, sondern über f2 / Layout ändern / Layoutnummer).

BÜNDIG (Blocksatz) sollte vorerst NICHT eingegeben werden. Das sollte erst ganz zum Schluß geschehen, wenn der Text korrigiert und druckfertig ist und alle Silbentrennungen erledigt sind. Mit der Bildschirmanzeige von Blocksatz ist es sehr schwer abzuschätzen, wieviel Silben eines Wortes noch in die vorige Zeile passen.

2.Schreiben des Textes

Normales Schreiben des Textes. Im Text auftauchende Fußnotennummern HOCH, FETT und INVERS eingeben. Fußnoten fortlaufend beziffern. NACH Beendigung des Absatzes, in dem die Fußnotennummer steht (nach Lust und Laune auch später), die zugehörige Fußnote eingeben.

3. Fußnote eingeben

Den Cursor direkt unter den Absatz stellen, in dem die Fußnotennummer steht. Dann die zu dieser Nummer gehörende Fußnote eingeben. Dazu:

- Fußnotenlayout einfügen f2=LAYOUT / LAYOUT EINFÜGEN / LAYOUTNUMMER (Fußnotenlayout) / ENTER oder +LT-Nr. / ENTER
- Fußnote INVERS markieren: +I
- Eventuell Schrift hochstellen: +HH
- Fußnotentext eingeben.
- Unter Umständen Fußnote und zugehörige Textzeile als Zeilenpaket halten (Siehe Handbuch). Bewirkt, daß Fußnote und zugehöriger Text auf jeden Fall auf der gleichen Seite stehen und nicht über zwei Seiten umbrochen werden können.
- am Ende des Fußnotentextes mit: -I Invers wieder abschalten, ebenso das Hochstellen mit: -HH und das Fußnotenlayout mit: -LT (Angabe der Layoutnummer ist nicht nötig); zurück zum Basislayout.
- DANN ERST den Fußnotentext mit RETURN (Zeilenschaltung) beenden. Andernfalls kann es passieren, daß bei einem eventuellen Verschieben der Fußnote an eine andere Textstelle das -LT nicht mit verschoben wird.

4. Text weiter bearbeiten

Text weiter schreiben, korrigieren, ändern, ergänzen, Silbentrennungen usw. Erst dann, wenn der Text mit Ausnahme der Fußnotenanordnung druckfertig ist,

5. Fußnoten an das Ende der jeweiligen Seite verschieben

- Eventuell Anfertigen einer Trennlinie zwischen Fußnoten und Text. Diese als Satz abspeichern: f8=BLOCK
- wenn Trennlinie, dann diese zwischen vorletzter und letzter Zeile der Seite einfügen.
- Fußnoten aus dem Text an das Ende der Seite verschieben mit: COPY / AUSBL / EINBL

Wenn Trennlinie, dann direkt hinter dieser und VOR der ehemals letzten Seitenzeile.

Wenn keine Trennlinie verwendet wird, kann/sollte vor dem Verschieben der Fußnote das Seitenende (letztes Wort) invers markiert werden, da beim Ausblenden der Fußnote der nachfolgende Text natürlich um eine entsprechende Anzahl von Zeilen nach oben rückt. Damit steht die Textzeile, unter der die Fußnote eingeblendet werden muß, nicht mehr am Seitenende; markiert ist sie aber sehr leicht wiederzufinden.

Die Fußnoten werden nun direkt hinter der letzten Seitenzeile eingefügt (da die Fußnoten invers markiert wurden, ist immer klar ersichtlich, wo die nächste hin muß).

Eventuell kann es vorkommen, daß beim Verschieben der Fußnote an das Ende der Seite die am Anfang der Fußnote stehende Fußnotennummer vom Programm noch in die vorhergehende Textzeile gestellt wird (genügend Platz am Ende der Zeile). In einem solchen Fall müssen die ersten Zeichen/Wörter der Fußnote mit HARDSPACES verbunden werden (das verhindert einen Umbruch der verbundenen Teile über zwei Zeilen). Dieser Effekt tritt bei der Verwendung einer zwischen Text und Fußnote eingeschobenen Trennlinie (unterstrichene Leerzeile) nicht auf, da diese mit RETURN abschließt.

6. Eventuell auf Blocksatz umschalten

Das muß in ALLEN im Text verwendeten Layouts geschehen, also auch im Fußnotenlayout.

7. Fertigen Text drucken

Da wegen der Randeinstellung der Layouts (linker Rand = 0) der Text direkt am linken Rand beginnend ausgedruckt würde, muß vor dem Ausdrucken in der Druckerkontrollstellung PTR unter f3=AKTIONEN der für den Ausdruck gewünschte linke Rand/Abstand eingegeben werden. Beim Ausdrucken von Korrekturabzügen kann man diese Funktion dazu nutzen, den linken Rand entsprechend schmal vorzugeben und dadurch einen entsprechend breiteren rechten Rand für Korrekturen zu erhalten.

Ein Letztes

Da beim Eingeben der Fußnotentext erst NACH dem Verlassen des Fußnotenlayouts mit RETURN abgeschlossen wurde, hat dieser Zeilenvorschub natürlich den im Basislayout eingestellten Wert. Beim Verschieben mehrerer Fußnoten an ein Seitenende bleibt dieser Wert erhalten, was zur Folge hat, daß der Abstand zwischen den einzelnen Fußnoten der Zeilenschaltung des Basislayouts entspricht.

Wenn dieser Abstand zu groß ist, löscht man nach dem Verschieben der Fußnoten die Funktion -LayouT am Ende der Fußnoten. Dadurch wird der folgende Zeilenvorschub zwischen den Fußnoten mit dem im Fußnotenlayout angegebenen Wert durchgeführt. Bei der LETZTEN Fußnote am Ende der Seite allerdings muß das -LayouT stehenbleiben!

ALLES KLAR??? (R.Sprunkel)



dBase II auf dem JOYCE PLUS

Nachdem Schneider den JOYCE PLUS mit zweitem Diskettenlaufwerk und einer 368 KByte RAM-Disk auf den Markt gebracht hat, eröffnen sich für den Benutzer von dBase II neue Möglichkeiten. Zum Einen ist das zweite Laufwerk mit 720 KByte dazu prädestiniert eine Menge Daten abzuspeichern, zum anderen erlaubt die RAM-Disk einen schnellen Zugriff auf die Programmteile von dBase II.

Jeder dBase II Benutzer wird wohl über kurz oder lang sein erstes dBase-Programm schreiben, um die Benutzerfreundlichkeit zu erhöhen und an seine speziellen Bedürfnisse anzupassen. Diese Programme ermöglichen eine Menuesteuerung, die selbst absolute Computerneulinge sicher durch das Programm führen.

War es bislang so, und auf dem »kleinen« JOYCE auch weiterhin, daß mit vielen Diskettenwechsel und langen Zugriffszeiten gearbeitet werden mußte, ermöglicht nun das zweite Laufwerk und die erweiterte RAM-Disk ein schnelles und bequemes Arbeiten mit dBase II.

Um einen schnellen Zugriff auf Programmteile zu erhalten, laden wir die dBase-Programme auf die RAM-Disk (Laufwerkskennung M:). dBase kann nun von der RAM-Disk aus gestartet werden und mit »SET DEFAULT TO B« wird die Diskette im zweiten Laufwerk zum Datenträger definiert.

Gehen wir nun einmal davon aus, daß das dBase-Programm mit dem Namen »BUECHER.CMD«, welches als Beispiel auf Seite B der Original dBase-Diskette zum Lieferumfang gehört, beim Starten automatisch aufgerufen werden soll. Folgende Schritte sind dazu notwendig: Zunächst erstellen wir uns in Laufwerk A: eine Startdiskette von dBase mit folgenden Files:

DBASE.COM
DBASEOVR.COM
DB.SUB
DB.KEY

Die HELP-Funktion (File DBASEOVR.TXT) benötigen wir nicht, da wir ein dBase-Programm mit Menuesteuerung aufrufen wollen.
Dann kopieren mit PIP noch folgende Files mit auf die Diskette:

PIP.COM
TYPE.COM
SETKEYS.COM
SUBMIT.COM
DIR.COM
J14GCPM3.EMS

Nun erstellen wir mit dem Editor RPED folgende Befehlsdatei mit dem Namen PROFILE.SUB: (diese datei wird bei jedem Start von CP/M ausgeführt)

```
SETKEYS DB.KEY
PIP
<M:=DO.COM
<M:=DBASEOVR.COM
<M:=DBASEMSG.TXT
<M:=START.CMD
<
M:
DO START
```

Dann kopieren wir mit PIP folgende dBase-Dateien von Seite B der Sicherheitskopie von dBASE auf eine leere CF 2 Diskette in Laufwerk B:

```
PIP
* B:=A: *.CMD
```

(mit diesem Befehl werden alle »Bücher«-Programm-Dateien kopiert)

```
* B:=A: *.NDX
```

(mit diesem Befehl werden alle Indexdateien kopiert)

```
* B:=A: *.FRM
```

(mit diesem Befehl werden alle Reportdateien kopiert)

```
* B:=A: BUECHER.DBF
```

(mit diesem Befehl wird die eigentliche Bücherdatei kopiert)

```
CP/M Plus  Amstrad Consumer Electronics plc
v 1.4, 61K TPA, 2 Laufwerke, 112K Laufwerk M:
A)dir A: ÄALLÜ
Scanning Directory...
Sorting  Directory...
Directory For Drive A:  User  0

    Name     Bytes   Recs   Attributes      Name     Bytes   Recs   Attributes
------------ ------ ------ ------------ ------------ ------ ------ ------------
DB       KEY    2k     16 Dir RW       DB       SUB    1k      1 Dir RW
DBASEOVR COM   41k    328 Dir RW       DIR      COM   15k    114 Dir RW
DO       COM   20k    158 Dir RW       J14GCPM3 EMS   40k    320 Dir RW
PIP      COM    9k     68 Dir RW       PROFILE  SUB    1k      1 Dir RW
SETKEYS  COM    2k     16 Dir RW       START    CMD    1k      4 Dir RW
SUBMIT   COM    6k     42 Dir RW       TYPE     COM    3k     24 Dir RW

Total Bytes     =    141k  Total Records =    1092  Files Found =   12
Total 1k Blocks =    141   Used/Max Dir Entries For Drive A:   17/  64

A)
```

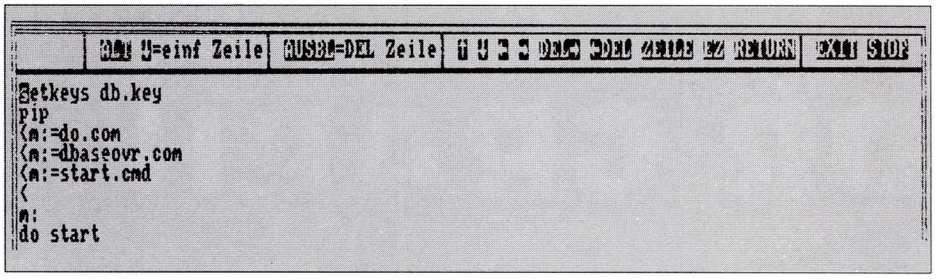
```
CP/M Plus  Amstrad Consumer Electronics plc
v 1.4, 61K TPA, 2 Laufwerke, 112K Laufwerk M:
A)dir B: ÄALLÜ
Scanning Directory...
Sorting  Directory...
Directory For Drive B:  User  0

    Name     Bytes   Recs   Attributes      Name     Bytes   Recs   Attributes
------------ ------ ------ ------------ ------------ ------ ------ ------------
AUTOREN  NDX    4k     20 Dir RW       BUECHER  CMD    2k     10 Dir RW
BUECHER  DBF    6k     44 Dir RW       BUECHER  FRM    2k      2 Dir RW
BUECHER1 CMD    2k     16 Dir RW       BUECHER2 CMD    4k     31 Dir RW
BUECHER3 CMD    4k     20 Dir RW       BUECHER4 CMD    2k     15 Dir RW
BUECHER5 CMD    2k      9 Dir RW       BUECHER6 CMD    2k      9 Dir RW
BUECHER7 CMD    2k     12 Dir RW       BUECHER8 CMD    2k     12 Dir RW
B_TITEL  NDX    6k     48 Dir RW       RAEUME   FRM    2k      2 Dir RW
RAEUME   NDX    2k      8 Dir RW       RAUEME   NDX    2k      8 Dir RW
S_GEBIET NDX    2k     16 Dir RW

Total Bytes     =     48k  Total Records =     282  Files Found =   17
Total 1k Blocks =     42   Used/Max Dir Entries For Drive B:   17/ 256

A)
```

Wir können hierzu nach schriftlicher Auskunft der Firma SCHNEIDER die normalen CF2 Diskette benutzen. Die Produktion der CF 2 DD Disketten wurde aufgegeben, da die CF 2 Disketten ohne Qualitätseinbuße benutzt werden können.

Eine Programm-Datei wird unter dBase naormalerweise mit dem Befehl »DO«, in unserem Beispiel also mit »DO BUECHER« gestartet. dBase bietet aber die Möglichkeit eine Programm-Datei direkt vom Betriebssystem aus zu starten. Dazu müssen wir die Programm-Datei »DBASE.COM« umbenennen in »DO.COM«. Um dies unter CP/M zu erreichen, schreiben wir (nach Verlassen vom dBase II-System natürlich):
a>REN DO.COM=DBASE.COM
Das DBASE.COM File auf unserer Stratdiskette heißt nun also DO.COM. Das Programm »Bücher« kann jetzt mit dem Befehl »DO BUECHER« direkt vom Betriebssystem aus aufgerufen werden.

Damit dies nun auch automatisch beim Starten geschieht, erstellen wir unter dBase mit dem MODIFY COMMAND-Befehl folgende Programmdatei mit dem Namen START.CMD:

SET DEFAULT TO B
DO BUECHER



Diese dBase-Programmdatei ist Bestandteil unserer oben erstellten Datei PROFILE.SUB, wird beim Starten also mit aufgerufen und bringt uns in das »Bücher«-Programm. Da sich dieses Programm auf der Diskette im Laufwerk B: befindet, wird zuvor mit »SET DEFAULT TO B« dieses Laufwerk für dBase zum Bezugslaufwerk definiert.

Wir brauchen jetzt nur noch den Joyce einzuschalten, beide Disketten einzulegen und können am Bildschirm verfolgen, wie die einzelnen Programmschritte aufgerufen werden.

Zunächst erscheint die gewohnte CP/M Meldung. Danach wird die PROFILE.SUB datei aufgeführt. Die Tastatur wird mit dem Befehl SETKEYS DB.KEY definiert. Jetzt werden mit PIP alle benötigten Programme auf die RAM-Disk kopiert. Die RAM-Disk wird mit A>M: zum Erstlaufwerk erklärt. Von dort wird mit dem Befehl »DO START« gleichzeitig dBase gestartet und die dBase-Befehlsdatei »START« aufgerufen. Diese »START«-Datei schaltet wie oben beschrieben auf das Laufwerk B um und startet in unserem Beispiel das Programm »Bücher«.

Das Laufwerk B bleibt solange unser Bezugslaufwerk, bis wir mit QUIT den dBase-Lauf beenden. Wir kehren dann in das Betriebssystem zur RAM-Disk mit der Laufwerkskennung M> zurück.

Wenn nun dBase vom Laufwerk M> gestartet werden soll, muß diese natürlich mit »DO« geschehen und nicht wie gewohnt mit »DBASE«. Wollen wir wieder ins »Bücher«-Programm, schreiben wir »DO BUECHER«.

Schauen wir uns zum Schluß mit »DIR ÄALLÜ« den Inhalt unserer beiden Disketten an. Sie müßten folgende Files aufweisen: (H. Höster)

Alles für den Joyce

Farbband Drucker	29.90
Parallel-Seriell-Schnittstelle	148,--
1 MB-Laufwerk (FD 2)	598,--
Bildschirmfilter	89,--
Disketten 3" CF 1DD	98,--
Diskettenbox 3"/40	39.80
Diskettenbox 3"/80	49.80
Joystick Compet. Pro	49,--
Endlospapier 1000 Blatt	29.80
200 Etiketten 70x70	16,--
Joyce Papierführung	37,--
Schaltplan Joyce	15,--

Hardware Preise

Joyce PCW 8256	1.578,--
Joyce Plus 8512	2.198,--

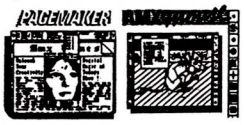
Public Domain Service

Ca.1000 Programme auf 256 Disketten warten auf Ihren Einsatz! Unter anderem sind dabei:
Pascal Compiler • C-Compiler • Forth Lisp • dBase Programme • Assembler etc.
Jede Diskette 20,--

Joyce Software

Business-Star	298,--
Fibu-Star	298,--
dBase II, Wordstar je	199,--
Multiplan	199,--
Prompt (Datei)	69,--
Prompt Druck	39,--
Mica (CAD)	198,--
Star Mail	98,--
Star Base	198,--
Dr. Graph	199,--
Dr. Draw	199,--
Turbo-Pascal	225,--
Profirem (Address, Lager, Fakturierung)	178,--
Fakturem	98,--
Addressdatei	58,--
Turbo Address	169,--
Small C	99,--
C-Basic-Compiler	169,--
RH-Dat	89,--
Vereinsverwaltung	199,--
Fibuking	136,--
Joyce Mailing-Syst.	189,--
Dateistar	98,--
Faktum 10	148,--
Headline	198,--
Thingi (Konzepthalter)	**24.80**

AMX-Mouse + Joystick-Controler + Adapter
(mit deutschem Handbuch) 328,--



Pagemaker 178,--
(mit deutschem Handbuch)

Grafpad III (neu) 548,--
(mit deutschem Handbuch)

Sensationell!!
Für alle die umsteigen wollen auf einen PC!
Multitronic XT
• AT-Gehäuse • Netzteil 150 Watt • Prozessor 8088 • 4.77 MHZ • DIN Tastatur (deutsch) • RAMSet 640 KB • Color Grafikkarte • Multi/O (FD, RS 232, P, U, G) • Laufwerk 360 KB • Daten : 640 KB, Centronics-Ausgang, V24/RS232-Ausgang, RGB/Chinch-Ausgang •
1.598,-- DM

Elektric Studio Produkte für Joyce
Lightpen *: 278,-- DM
- Freihandzeichnen auf dem Monitor
- Abspeichern möglich
- Hardcopy der bilder auf Joyce Drucker in 3 Größen möglich
- gängige Funktionen wie Bleistift, Sprühdose, Radiergummi, Invers

Maus *: 448,-- DM
- hochwertig mit Interface
- Zeichensoftware wie beim Lightpen
- Joystickanschluß am Interface

Adapter (*) 29,-- DM

Supercopy - Joyce
Das Diskettenkopierprogramm der Superlative (auch für CPC 464, 664 und 6128 erhältlich).
SUPERCOPY kopiert 99,9% der auf dem Markt befindlichen Software!
(Sollte SUPERCOPY einmal etwas nicht schaffen, senden Sie die Orginaldiskette an den Hersteller, dann erhalten Sie kostenlos eine neue Version incl. der Erkennung des neuen Kopierschutzes). 89,-- DM

Vokabeltrainer 59,.. DM

Verbentrainer 49,-- DM

Spiele für den Joyce

1. Aftershock	59,--	DM
2. Batman	55,--	DM
3. Bridge Players	65,--	DM
4. Cyrus Chess	59,--	DM
5. 3D Clock Chess	59.90	DM
6. Fairlight	59,--	DM
7. Heroes of Karn	59,--	DM
8. Pawn	79.90	DM
9. S.A.S Raid	49.90	DM
10. Tomahawk	69,--	DM
11. Jewels of Darkness	78,--	DM

dk` tronics-Produkte für Joyce

256 k Erweiterung auf 512 kRam	109,--	DM
Joystick-Controler (Interface) *	69,--	DM
Joystick-Controler + Musik/Sound *	158,--	DM
Echtzeituhrenmodul *	138,--	DM
Adapter (alle Module mit *)	29,--	DM

Joyce Festplatte (Hard Disk)

10 Megabyte Hard Disk	1.698,-- DM
20 Megabyte Hard Disk	1.998,-- DM

Karl-Heinz • Potsdamer Ring 10 • 7150 Backnang Tel. 07191/1528
weeske
Computer - Elektronic

Achtung! Wir liefern auch gern in die Schweiz und nach Ostereich!
Zahlung per Scheck oder Überweisung (Vorkasse) wegen komplizierter Nachnahmeabwicklung erwünscht.

Tips zu DR GRAPH

Hiermit möchte ich einen, wie ich finde, wertvollen (weil zeitsparenden) Tip an alle weitergeben, die mit DR Graph auf einem JOYCE PLUS oder einem vollerweiterten PCW 8256 (d.h., 2. Laufwerk und Speichererweiterung auf 512K) arbeiten.

Die zugebendermaßen recht langwierige Erstellung von Grafik (besonders »Multiple Graphs«) im Display oder Printoutput läßt sich nämlich durch einen einfachen Trick verkürzen:

Auf eine fertige **WORKDISK** für das 2. Laufwerk (nach Anweisung im Handbuch erstellt) bringt man mit einem geeigneten Editor die **SUB-Datei <grafik.sub>** wie **in der Anlage dokumentiert** und legt gleichzeitig die Suchordnung mit **<setdef>** wie angegeben fest (wichtig!, damit die benötigten Files immer auf **M:** gesucht werden). **<setdef>** und **<pip>** muß man also vorher auf die **WORKDISK** kopieren.

Die Datei **<assign.sys>** auf der **WORKDISK** ist so eingerichtet, daß **<ddfxhr8.prl>** (die Druckersteuerung!) immer auf dem Laufwer B: gesucht wird. Auch läßt sich mit einem Editor (z.B.: **<basic rped>**) leicht ändern, so daß die neue Datei den Befehl enthält, die Druckersteuerung jetzt auf **M:** zu suchen.

Nach wie vor muß man **CP/M®** in Laufwerk **A**: mit einer geeigneten Startdiskette starten. Die nun folgende Eingabe **A:<submit B:grafik>** startet die von uns auf **B:** eingerichtete **SUB-Datei** und man muß nun zunächst einmal zuschauen, wie **JOYCE** mit **PIP** arbeitet. Das dauert ein bißchen, doch die »verlorene« Zeit holt man leicht wieder heraus, denn jetzt kann man mit **DR.GRAPH®** wie gewohnt arbeiten – nur schneller!

Die Zugriffszeiten auf das Laufwerk entfallen und man kann in den Menues nur so Herumfahren. Auch das Erstellen von **MULTIPLE GRAPHS**, ihr *Display* und der *Printeroutput* gehen nun viel viel schneller.

Extra-Tip

Man kann die Bearbeitungsgeschwindigkeit noch mal erhöhen, wenn man die Graphen zunächst auf **M:** abspeichert und den MULTIGRAPH entsprechen erstellt mit:

< File name a: **M:**Beispiel 1 >
< File name b: **M:**Beispiel 2 >
< File name c: **M:**Beispiel 3 >
< File name d: **M:**Beispiel 4 >

Soll ein solcher Graph ausgedruckt werden, so bleibt der Bildschirm schwarz, die Laufwerke werden nicht mehr abgefragt und die Aufforderung *»Papier anlegen«* erscheint viel schneller als bisher.

Wer es ausprobiert hat ist sofort überzeugt!
!!! Nicht vergessen, vor dem Ausschalten die Graphen von **M:** auf eine Datendiskette zu kopieren und so zu sichern!!!

Übrigens: Man kann jetzt nach dem kompletten Startprogramm auch die **WORKDISK** aus dem Laufwerk **B:** nehmen und dafür eine andere Diskette einlegen. Zum Beispiel eine, auf der man nun seine schnell erstellten Graphen alle speichern möchte (das geht jetzt mit mit <OUTPUT file name: **B:**Beispiel 1>) – Darauf ist ja genug Platz.

Und hier die Version für PCW 8256 mit 512 KB RAM:

Statt sich als Diskjockey betätigen zu müssen, wenn nach Diskette A oder B gefragt wird, kann man durch ein gezieltes Erstellen einer WORKDISK das gesamte Programm auf dem Laufwerk M: laufen lassen. Das spart nicht nur das ständige Wechseln der Disketten, sondern auch jede Menge Zeit.

Zuerst erstellt man aus Sicherheitsgründen eine Backup-Version der Original Programmdiskette. Von dieser Schutzkopie ausgehend wird nun die eigentliche WORKDISK erstellt:

Kopieren Sie folgende Programme von der Bachup-Diskette des Originals auf Ihre leere (formatierte!) **WORKDISK Seite A**

⇒ ASSIGN.SYS (geänderte Version!)
⇒ DDFXHR8.PRL (oder andere Druckertreiber)
⇒ DDSCREEN.PRL
⇒ GRAPH.COM ⇒ GRAPHY.MNU ⇒ GRAPHZ.TBL
⇒ GSX.SYS

zusätzlich die CP/M Hilfsprogramme

⇒ J14GCPM3.EMS (CP/M Startdatei)
⇒ PIP.COM
⇒ SUBMIT.COM

sowie eine mit BASIC RPED erstellte Datei **PROFILE.SUB**, die wie folgt aussieht:

```
pip
<m:=a:graph?.*
<m:=a:*.prl
<m:=a:*.sys
<m:=b:*.*
m:
graph.com
```

Auf Seite B: der WORKDISK kopiert man dann die restlichen Dateien für DR GRAPH®

⇒ GRAPH 1.OVL bis GRAPH9.OVL
⇒ GRAPH A.OVL bis GRAPHD.OVL

Ebenfalls mit dem Editor ändert man die ASSIGN.SYS Datei so, daß die Gerätesteuerung ebenfalls auf M: gesucht werden. Danach muß die Datei so aussehen:

```
21 m:ddfxhr8
01 m:ddscreen
```

Hat man alle diese Arbeiten erledigt (die »verlorene« Zeit holt man beim schnelleren Arbeiten mit dem Grafikprogramm leicht wieder heraus), sollte

beim Einschalten von JOYCE und Einlegen der **WORKDISK Seite A** zunächst CP/M selbständig geladen werden. Die Datei PROFILE.SUB gelangt automatisch zur Ausführung. JOYCE arbeitet mit PIP und es erscheint dann die Aufforderung: *»Bitte Diskette für **B:** einlegen«.*

Wenden Sie also Ihre WORKDISK und legen Sie **Seite B** ein. Dann arbeitet JOYCE wieder mit PIP bis schließlich die Meldung, *»Bitte Diskette für **A:** einlegen«,* erscheint.

Hier wird nun einfach die Leertaste gedrückt oder eine DATENDISKETTE in **A:** eingelegt und dann die Leertaste gedrückt. Daraufhin startet JOYCE das Programm DR GRAPH® vom Laufwerk **M:** aus.

Die Zugriffszeiten auf das Laufwerk entfallen und man kann in den Menues nur so herumfahren. Auch das Erstellen von »MULTIPLE GRAPHS«, ihr *Display* und der *Printeroutput* gehen nun viel viel schneller.

Extra-Tip

Man kann die Bearbeitungsgeschwindigkeit hierbei noch mal erhöhen, wenn man die Graphen zunächst auf **M:** abspeichert und den MULTIGRAPH entsprechend erstellt mit:

< File name a: **M:**Beispiel 1 >
< File name b: **M:**Beispiel 2 >
< File name c: **M:**Beispiel 3 >
< File name d: **M:**Beispiel 4 >

Soll ein solcher Graph ausgedruckt werden, so bleibt der Bildschirm schwarz, das Laufwerk wird nicht mehr abgefragt und die Aufforderung *»Papier anlegen«* erscheint viel schneller als bisher.

Wer es ausprobiert hat, ist sofort überzeugt.

!!! Nicht vergessen vor dem Ausschalten die Graphen von **M:** auf eine Datendiskette zu kopieren und so zu sichern!!!

Übrigens: Hat man nach dem kompletten Startprogramm die **WORKDISK** aus dem Laufwerk **A:** genommen und dafür eine DATENDISKETTE eingelegt, kann man nun seine schnell erstellten Graphen darauf speichern.

Das geht jetzt mit <OUTPUT file name **A:**Beispiel 1> direkt aus dem Menue der DR GRAPH®.

(U. Schmidt)

So sollte die SUB-Datei aussehen!

```
B>type grafik.sub
pip
<m:=B:graph?.*Är vü
<m:=B:*.prlÄr vü
<m:=B:assign.sysÄr vü
<m:=B:gsx.sysÄr vü
<
setdef M:,A:,B: ÄORDER=(SUB,COM)ü
m:
graph
```

So sieht die geänderte Datei aus!

```
B>
B>type assign.sys
21 m:ddfxhr8
02 m:ddscreen
01 m:ddscreen
```


Inserenten

ABD Computer 7
Büro für Software Entwicklung 35
DMV 21, 29, 47, 69, 123
Ferdi's Computer Software 71
G Computerstore 107
Gerdes 63
Hashagen 73
Henschke 65
Interlogic 95
Keil 38
Kempenich 53
Kotulla 115
Micro Market Worms 33
Naujoks 33
Sunshine Software 116
Schneider Data 12
Star Division 2
TG-Soft 73
Van der Zalm 55
Vortex 124
Wade 53
Waldeck Software 71
Weeske 113
Werder 38

Software ist viel zu teuer!

Haben Sie bisher viel Geld für Programme ausgegeben, die dann doch nicht Ihren Vorstellungen entsprachen? Da haben wir ein tolles Angebot für Sie: Public Domain-Software für CP/M aus den USA! Nur bei uns bekommen Sie diese Programme mit deutschen Systemmeldungen und einem gedruckten Handbuch in deutscher Sprache – garantiert lauffähig auf Ihrem **Schneider-CPC** und **Joyce**:

Diskette 1: Pascal Compiler **JRT-Pascal** – vollständiges Pascal mit Strings bis 64 KByte, EXTERN-Overlays und vielen weiteren Vorteilen *

Diskette 2: **Assembler -Paket** mit Z80-Assembler, Linker, Debugger und intelligentem Disassembler

Diskette 3: Künstliche Intelligenz – Interpreter für die KI-Sprachen **XLISP** und **E-PROLOG**

Diskette 4: C-Compiler **Small-C** – verarbeitet sogar Fließkommazahlen und besitzt umfangreiche Bibliotheken *

Diskette 5: **Forth-83** – komplette Implementation von Forth mit Assembler, Decompiler, Screen-Editor, Kommentar-Screens ...

Diskette 6: CP/M-Utilities wie **SQ** (komprimiert Dateien bis 60%), **Diskettenmonitor, UNERA, Mehrspaltendruck, Super-Directory** u.v.a.

* Die Disketten 1, 3 und 4 benötigen CP/M 2.2 mit Speichererweiterung (64K genügen) oder CP/M Plus. Alle anderen auch ohne Erweiterung. Disketten 1–4 mit WordStar-kompatiblen Editor.

Jede Diskette kostet sage und schreibe nur 30,– Mark. Bitte Diskettenformat (3-Zoll oder Vortex 5,25-Zoll) angeben! Lieferung per Nachnahme oder Vorauskasse. Keine Versand- oder Verpackungsgebühren!

MARTIN KOTULLA

Grabbestraße 9, 8500 Nürnberg 90, Telefon: 09 11/30 33 33

Weitere Bezugsquellen (bitte Versandbedingungen erfragen):

Techn. Büro Ingeborg Hochholzer, Erhard-Prunner-Str. 1, 8062 Markt Indersdorf, Telefon 08136/1625 (auch 3,25 Zoll/Vortex)

TESCO GmbH, Rüdenhausenerstraße, 8714 Wiesentheid, Telefon 09383/1237 (auch andere CP/M-Formate)

Weeske Computer-Elektronik, Potsdamer Ring 10, 7150 Backnang, Telefon 07191/1528-29

Computerstore, Hochstraße 11, 8500 Nürnberg 80, telefon 0911/28 90 28

Computer Solutions, Belgradstraße 9, 8000 München 40, Telefon 089/3088048 (auch andere CP/M-Formate, Apple)

Weitere Händleranfragen sind stets willkommen!

Schriftartanwahl in Basic

Wer hat sich nicht schon bei der Suche nach den Steuercodes des Joyce-Druckers geärgert? Das Schriftbild bei LLIST und LPRINT Anweisungen ist durch die Grundeinstellung vorgegeben und nicht so einfach zu ändern.

Das vorliegende Programmchen soll hier Abhilfe schaffen.

Vorgehensweise:

1. A> Basic >RETURN<
2. RUN "DRUCKER.BAS" >RETURN<

Das Programm meldet sich mit einer Erklärung und fragt, ob die Grundeinstellung beibehalten oder geändert werden soll.

Wollen Sie eine Änderung vornehmen, so haben Sie die Wahl zwischen vier Möglichkeiten:

1. PICA-Schriftbild
2. ELITE-Schriftbild
3. KOPAKT-Schrift
4. PROPORTIONAL-Schrift

Anschließend können Sie zu jeder der vier Schriftarten noch folgende Optionen anwählen (1 = ja; 0 = nein)

1. + Kursive Darstellung
2. + Doppelanschlag
3. + Fettdruck
4. + doppelte Breite
5. + Korrespondenz-Qualität

Die somit gewählte Einstellung bleibt in jedem Falle bis zu einem Warmstart erhalten.

(A.Wilke)

```
LISTING >DRUCKER<, REMARK = >REM<.
<31> 10 'Arnulf Wilke..7520 Bruchsal
<48> 20 esc$=CHR$(27)..... 'Steuercode
<42> 30 an$=esc$+"p"..... 'Inverse Darstellung an
<79> 40 aus$=esc$+"q"..... 'Inverse Darstellung aus
<21> 50 cls$=esc$+"E"+esc$+"H" 'Bildschirm löschen
<49> 60 a$=esc$+CHR$(64)..... 'Grundeinstellung des Dr
     uckers
<65> 70 X$=STRING$(77,"="):Y$=STRING$(37,"-")
<84> 80 :
<39> 90 '.....Beschreibung des Programms
< 6> 100 LPRINT a$:PRINT cls$:PRINT an$:PRINT X$
<77> 110 PRINT" Die Grundeinstellung des Druckers ist:
     PICA-Schrift an (10 Zeichen pro Zoll)"
<59> 120 PRINT Y$;
<13> 130 PRINT".. Kursiv aus, Doppelanschlag aus,.....
     ":PRINT Y$;
<16> 140 PRINT".. Fettdruck aus, doppelte Breite aus,.
     ":PRINT Y$;
<28> 150 PRINT".. Korrespondenz-Qualität aus.....
     ":PRINT X$:PRINT aus$
<11> 160 PRINT X$:PRINT aus$
<28> 170 :
<93> 180 '.....Wahl der Grundeinstellung
<87> 190 PRINT" Die Grundeinstellung soll erhalten blei
     ben..... = 0 /RETURN/"
<13> 200 PRINT" Die Grundeinstellung soll geändert werd
     en..... = 1 /RETURN/"
< 2> 210 INPUT" Bitte treffen Sie Ihre Wahl.....
     ..... = ",a:IF a=0 GOTO 490
<19> 220 :
<13> 230 '.....Wahl der Schriftart
< 3> 240 PRINT an$:PRINT X$
< 3> 250 PRINT" Bitte wählen Sie eine der 4 Schriftarte
     n aus:..... "
<82> 260 PRINT X$
<52> 270 PRINT" PICA .....-Schrift. ( .. 10 Zeichen p
     ro Zoll) = 1 /RETURN/..... "
<34> 280 PRINT" ELITE .....-Schrift. ( .. 12 Zeichen p
     ro Zoll) = 2 /RETURN/..... "
< 5> 290 PRINT" KOMPAKT ....-Schrift. ( .. 17 Zeichen p
     ro Zoll) = 3 /RETURN/..... "
<97> 300 PRINT" PROPORTIONAL-Schrift. (ca. 12 Zeichen p
     ro Zoll) = 4 /RETURN/..... "
< 3> 310 PRINT X$:PRINT aus$
<84> 320 INPUT" Bitte treffen Sie Ihre Wahl.....
     ..... ",a:IF a<1 OR a>4 GOTO 320
<22> 330 :
<20> 340 '.....Einstellen der Schriftart
<93> 350 IF a=1 THEN a$=a$
<49> 360 IF a=2 THEN a$=esc$+"M"
<32> 370 IF a=3 THEN a$=esc$+CHR$(15)
<63> 380 IF a=4 THEN a$=esc$+"p1"
<34> 390 :
<88> 400 '.....Wahl und Einstellen des Zusatzes
<98> 410 PRINT an$:PRINT X$
<31> 420 PRINT" Bitte wählen Sie -zusätzlich- aus:.....
     ... ja = 1 / nein = 0. /RETURN/..... "
<78> 430 PRINT X$
<91> 440 PRINT" + Kursive Darstellung..... ";:
     GOSUB 510:IF b=1 THEN a$=a$+esc$+"4"
<20> 450 PRINT" + Doppelanschlag..... ";:
     GOSUB 510:IF b=1 THEN a$=a$+esc$+"G"
<53> 460 PRINT" + Fettdruck..... ";:
     GOSUB 510:IF b=1 THEN a$=a$+esc$+"E"
< 9> 470 PRINT" + doppelte Breite..... ";:
     GOSUB 510:IF b=1 THEN a$=a$+esc$+"W1"
<84> 480 PRINT" + Korrespondenz-Qualität..... ";:
     GOSUB 510:IF b=1 THEN a$=a$+esc$+"m1"
<50> 490 LPRINT a$:PRINT:PRINT:PRINT:PRINT Y$:PRINT". D
     er Drucker ist neu eingestellt !. "
< 7> 500 PRINT Y$:PRINT aus$:PRINT:PRINT:END
<96> 510 PRINT aus$;:INPUT"..... .. Ihre Wahl = ",b:P
     RINT an$;:RETURN
```


Wetten, der JOYCE kann mehr als Sie glauben?!

Z.B. Spielen:

Batman (3-D-Action-Adventure)		47.90	Bounder (schnelles Arcadegame)		39.90
Classic Collection		45.90	Tomahawk (3-D Hubschrauber-Simul.)		59.90
(PacMan, Frogger, Invaders)			Frank Bruno's Boxing (Boxkampf)		56.90
Cyrus II 3-D Chess (Schach)		47.90	Football Fortunes (Fußball-Strategie)		59.90
Strike Force Harrier (Flugsimul.)		59.90	The Pawn (engl. Super-Adventure)		69.90
Fairlight (3-D-Action-Adventure)		47.90	Southern Belle + Air Traffic Control		49.90
INFOCOM-Adventures	je	79.90	(Dampfloksimul. + Flughafentowersimulation)		

Z.B. Arbeiten:

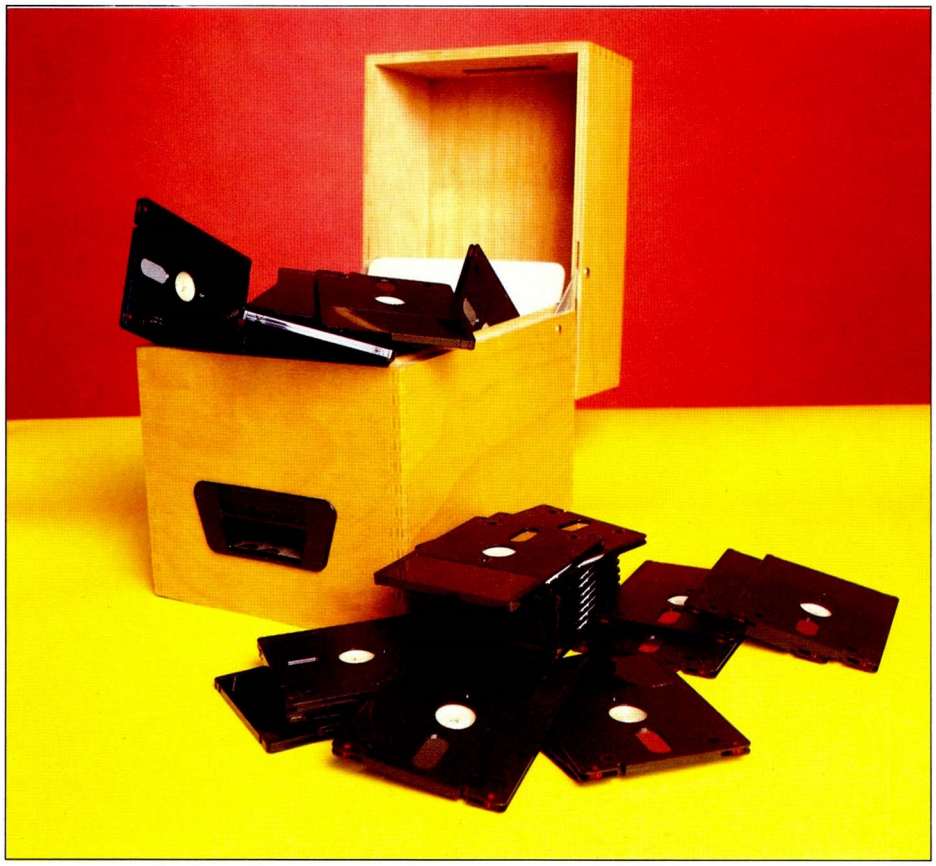
DR Draw / DR Graph	je	199.00	dBASE II / Multiplan	je	199.00
Finanzbuchhaltung		194.00	Business-Star (Auftragsbearb.)		298.00
Public Domain Software	je Disk	29.90	Statistic-Star (Business-Graphik)		98.00

Und wenn Sie für Ihr spezielles Problem noch keine Lösung gefunden haben, rufen Sie uns doch mal an! Oder lassen Sie Sich unseren kompletten JOYCE-Katalog mit noch mehr tollen Angeboten schicken.



SUNSHINE-Software, Auf dem Schilde 14,
5270 Gummersbach, Tel. (0 22 61) 7 57 52
*** In Luxemburg unt. Tel. (0 03 52) 81 82 12 ***

Zahlungsweise: Vorkasse (Scheck) oder Nachnahme (zzgl. Nachnahmegebühren). Bei einem Auftragswert über DM 100.– liefern wir bis auf NN-Gebühren frei Haus, darunter berechnen wir eine Versandkostenpauschale von DM 5.–.



Komfortable Diskverwaltung

1. Problemstellung Warum DISK.BAS ?

Wenn der Anwender CP/M-Plus mit seiner Kopie der System-Diskette initialisiert, endet das Booten auf der CP/M-Betriebsebene. Ist die PROFILE.SUBDatei aktiv, werden noch einige Funktionen ausgeführt, eventuell wird BASIC geladen. Spätestens jetzt muß nach dem Diskettenwechsel umständlich mit »LOAD« oder »RUN« ein Programm geladen bzw. gestartet werden. Ist das gewünschte Programm nicht auf der Disk, gibt es eine Fehlermeldung. Um das Inhaltsverzeichnis zu erhalten muß »DIR« eingegeben werden.

Fazit: Die Handhabung auf der Systemebene ist umständlich und langwierig. Daher muß Abhilfe geschaffen werden.

Hier setzt DISK.BAS Version 2.0 an und bietet dem Anwender mit seinen an LOCOSCRIPT angelehnten Datenbild ein übersichtliches Mittel, oft anfallende Funktionen programmunterstützt auszuführen.

2. Zielgruppe

Für wen ist DISK.BAS gedacht? DISK.BAS ist für alle Anwender die im privaten Bereich mit dem Schneider JOYCE arbeiten und überwiegend BASIC-Programme verwenden.

3. Installation

Wo sollte DISK.BAS installiert werden? Es sollte auf der Kopie der System-Diskette mit der CP/M-Plus gebootet wird, installiert werden. Da das Programm ohne Kommentare nur ca. 7300 Byte lang ist, kann es durch das Umsetzen von einigen COM-Files auf die System-Diskette 2 erreicht werden, daß bei aktiver PROFILE.SUB-Datei das Booten mit DISK.BAS endet. Dem Anwender steht dann sofort ein komfortables Hilfsmittel zur Verfügung. Es ist zu empfehlen, DISK.BAS auch auf alle anderen Disketten mit BASIC-Programmen zu kopieren. Wenn diese Programme mit einer Option zum Aufruf von DISK.BAS versehen sind, kann man bequem von einem Programm ins andere gelangen, ohne auf der BASIC-Systemebene arbeiten zu müssen.

4.Was kann DISK.BAS?

DISK.BAS enthält die nachfolgend aufgeführten Funktionen:

a) DIRectory der Laufwerke A: und M:
   Diskwechsel in A: möglich
b) Programmende mit Verbleib in BASIC
c) Programmende mit Rückkehr zur CP/M-Systemebene
d) Laden eines BASIC-Programms
e) Starten eines BASIC-Programms
f) Umbenennen von Files
g) Löschen von Files
h) Überprüfen von Dateien im ASCII-Format mittels der TYPE-Funktion

Die Funktionen d-h gelten für beide Laufwerke (A: + M:).

5.Was kann DISK.BAS nicht?

a) Kopieren oder Versetzen von Dateien.
b) Aufrufen und Ausführen von COM- oder SUB-Files
c) Abfragen u. Anzeigen eines Laufwerk B: (Programm kann ggfs. erweitert werden.)
d) Anzeigen von mehr als 21 Files je Spalte.

Die entsprechenden Erweiterungen wurden entweder nicht vorgenommen, weil sie nicht möglich sind, oder aber, um nicht zuviel Speicherplatz auf Disk zu belegen.

6.Wie wird DISK.BAS installiert?

Vorschlag zur Änderung der CP/M-Systemdiskette:
Die Files: SETSIO .COM
LANGUAGE.COM
werden umgesetzt auf Diskette 2 (CP/M-Dienstprogramme), Seite 2. Das ist die Seite, auf der sich HELP.COM befindet. Die Datei PROFILE.GER bzw. PROFILE.ENG (je nach Diskette) wird in PROFILE.SUB umbenannt. Mit dem Editor RPED.BAS wird diese Datei geändert. Mit SHOW.COM ist zu prüfen ob der freie Speicherplatz auf der

der CP/M-Systemdiskette jetzt ausreichend ist. (Platzbedarf 8 KByte).
Ggf. sollte geprüft werden, ob auf die Datei RPED.SUB verzichtet werden kann. Zuletzt ist mit PIP.COM das Programm DISK.BAS auf die Systemdiskette zu kopieren.
Mit einem Neustart des Systems ist zu prüfen, ob jetzt alles funktioniert. Wenn nicht, ist nochmals mit SHOW.COM der Speicherplatz auf der Disk zu überprüfen und wie bereits beschrieben, weiter zu verfahren.

7. Sonstiges

Nähere Erläuterungen insbesondere eine Bedienungsbeschreibung ist nicht erforderlich, da das Programm sich während der Bedienung selbst erklärt. Falsche Eingaben werden abgefangen.

Beim Abtippen sollten sämtliche Erläuterungen im Listing weggelassen werden, um Speicherplatz zu sparen. Entsprechende Zeilen werden durch GOSUB- bzw. GOTO-Befehle nicht angesprochen. Bei der Erstellung wurde auch darauf geachtet, daß die besonders zeitaufwendigen Abschnitte des Programms so gestaltet sind, das unnötige Wartezeiten möglichst nicht entstehen.

Aus diesem Grund wurden die Routinen zum Einlesen der Dateinamen der Laufwerke A: und M: getrennt geschrieben, da so der Programmablauf schneller ist.

(T. Westermann)

```
LISTING >DISK    <, REMARK = >'<.

<86> 10 'DISK.BAS -. 1987 by Thomas Westermann
<29> 20 OPTION RUN
<85> 30 DIM lw$(128),disk$(5,22),feld$(11)
<49> 40 es$=CHR$(27)
<42> 50 cls$=es$+"H"+es$+"E"
<67> 60 on$=es$+"p":of$=es$+"q"
<60> 70 con$=es$+"e":cof$=es$+"f"
<25> 80 bell$=CHR$(7)
<63> 90 n$="-Files"
<50> 100 n$(1)="Stück"
<84> 110 n$(2)=SPACE$(8)
<55> 120 feld$(1)=es$+"X"+CHR$(36)+CHR$(50)+CHR$(35)+CH
     R$(36)
<25> 130 feld$(2)=es$+"X"+CHR$(36)+CHR$(110)+CHR$(35)+C
     HR$(36)
<62> 140 FOR a=3 TO 8
<53> 150 READ b
<45> 160 feld$(a)=es$+"X"+CHR$(41)+CHR$(b)+CHR$(53)+CHR
     $(44)
<48> 170 NEXT
<53> 180 DATA 33,48,63,78,93,108
<74> 190 feld$(9)=es$+"X"+CHR$(36)+CHR$(33)+CHR$(36)+CH
     R$(60)
<31> 200 feld$(10)=es$+"X"+CHR$(36)+CHR$(63)+CHR$(35)+C
     HR$(60)
<58> 210 feld$(11)=es$+"X"+CHR$(36)+CHR$(93)+CHR$(36)+C
     HR$(59)
<82> 220 voll$=es$+"X"+CHR$(32)+CHR$(32)+CHR$(63)+CHR$(
     121)
<48> 230 DEF FNpkt$(zei,sp,zchn$)=es$+"Y"+CHR$(32+zei)+
     CHR$(32+sp)+zchn$
< 4> 240 PRINT cls$;cof$;FNpkt$(0,0,on$+"DISK... .BAS"+
     SPACE$(22)+"Diskverwaltung"+SPACE$(17)+"V2.0 by Th
     omas Westermann")
<45> 250 PRINT FNpkt$(1,0,". L. = Laden *.BAS"+SPACE$(5
     )+"R. = Run *.BAS"+SPACE$(6)+"U. = Umbenennen"+SPA
     CE$(3)+"E.. = Löschen"+SPACE$(2)+"T = Type Datei")
     ;
<25> 260 PRINT". f1 = Diskwechsel"SPC(5)"f3 = Ende-BASI
     C"SPC(5)"f7 = Ende-CP/M"SPC(4)"CAN = Abruch"SPC(17
     )of$+FNpkt$(3,0,STRING$(90,154))
<81> 270 FOR a=0 TO 90 STEP 30
<81> 280 PRINT FNpkt$(3,a,CHR$(158))
<53> 290 NEXT
<51> 300 FOR a=4 TO 7
<79> 310 FOR b=0 TO 90 STEP 30
<28> 320 PRINT FNpkt$(a,b,CHR$(149))
<42> 330 NEXT
<44> 340 NEXT
< 7> 350 PRINT FNpkt$(8,0,CHR$(151)+STRING$(29,154)+CHR
     $(155)+STRING$(29,154)+CHR$(155)+STRING$(28,154)+C
     HR$(157))
<10> 360 FOR a=15 TO 75 STEP 30
<71> 370 PRINT FNpkt$(8,a,CHR$(158))
<52> 380 NEXT
<16> 390 PRINT FNpkt$(8,30,CHR$(159));FNpkt$(8,60,CHR$(
     159))
<47> 400 FOR a=0 TO 90 STEP 15
< 4> 410 IF a=45 THEN c=133 ELSE c=149
<19> 420 FOR b=9 TO 29
<74> 430 PRINT FNpkt$(b,a,CHR$(c));
<45> 440 NEXT
<47> 450 NEXT
<62> 460 FOR a=9 TO 11 STEP 2:PRINT feld$(a);"Laufwerk
     ";
<24> 470 IF a=9 THEN PRINT FNpkt$(0,22,n$(1));
<74> 480 IF a=11 THEN PRINT"M:"+".. "+n$(2)+n$(1)
<27> 490 PRINT "COM+SUB "+n$+n$(2)+n$(1):PRINT"BASIC..
     "+n$+n$(2)+n$(1)
<61> 500 PRINT "Sonstige"+n$+n$(2)+n$(1);
<45> 510 NEXT:PRINT voll$
<71> 520 OPTION RUN
<47> 530 OPTION FILES"A"
<15> 540 IF r=2 THEN r=0:GOTO 820
< 7> 550 FOR a=1 TO 64
<14> 560 lw$(a)=FIND$("*.*",a)
<42> 570 IF lw$(a)="" THEN a=64
<54> 580 NEXT
<77> 590 OPTION FILES"M"
<84> 600 FOR a=65 TO 128
<61> 610 lw$(a)=FIND$("*.*",a-64)
<77> 620 IF lw$(a)=""THEN a=128
<45> 630 NEXT
<50> 640 OPTION FILES"A"
<33> 650 FOR a=0 TO 5:z(a)=0:NEXT
<10> 660 FOR a=1 TO 64
<65> 670 IF lw$(a)=""THEN a=64:GOTO 730
< 6> 680 pr$=RIGHT$(lw$(a),3)
<81> 690 IF pr$="COM"THEN disk$(0,z(0))=lw$(a):z(0)=z(0
     )+1:GOTO 730
<92> 700 IF pr$="SUB"THEN disk$(0,z(0))=lw$(a):z(0)=z(0
     )+1:GOTO 730
< 3> 710 IF pr$="BAS"THEN disk$(1,z(1))=lw$(a):z(1)=z(1
     )+1:GOTO 730
<61> 720 disk$(2,z(2))=lw$(a):z(2)=z(2)+1
<46> 730 NEXT
<93> 740 FOR a=65 TO 128
<88> 750 IF lw$(a)=""THEN a=128:GOTO 810
< 3> 760 pr$=RIGHT$(lw$(a),3)
<85> 770 IF pr$="COM"THEN disk$(3,z(3))=lw$(a):z(3)=z(3
     )+1:GOTO 810
<16> 780 IF pr$="SUB"THEN disk$(3,z(3))=lw$(a):z(3)=z(3
     )+1:GOTO 810
<26> 790 IF pr$="BAS"THEN disk$(4,z(4))=lw$(a):z(4)=z(4
     )+1:GOTO 810
<22> 800 disk$(5,z(5))=lw$(a):z(5)=z(5)+1
<43> 810 NEXT
<49> 820 PRINT feld$(1);cls$;z(0)+z(1)+z(2):PRINT z(0):
     PRINT z(1):PRINT z(2);
<95> 830 PRINT feld$(2);cls$;z(3)+z(4)+z(5):PRINT z(3):
     PRINT z(4):PRINT z(5);
<71> 840 PRINT voll$;
<89> 850 FOR a=0 TO 5
<62> 860 PRINT feld$(a+3);
<78> 870 FOR b=0 TO 20
<52> 880 PRINT disk$(a,b)
<59> 890 NEXT
```

```
<11> 900 PRINT voll$
<44> 910 NEXT
< 3> 920 IF LEN(disk$(0,0))<12 THEN disk$(0,0)=SPACE$(1
     2)
<52> 930 prog$=FNpkt$(9,1,disk$(0,0)):hoehe=0:seite=0:P
     RINT on$+prog$+of$
< 5> 940 OPTION STOP
<55> 950 z$=INKEY$:IF z$=""THEN 940
<84> 960 z$=UPPER$(z$):z=ASC(z$)
<33> 970 IF z=26 THEN 1170
<94> 980 IF z=17 THEN 1240
<86> 990 IF z=16 THEN 1290
<10> 1000 IF z=76 THEN 1340
<73> 1010 IF z=82 THEN 1520
<97> 1020 IF z=85 THEN 1560
<61> 1030 IF z=69 THEN 1710
<94> 1040 IF z=84 THEN 1880
<31> 1050 IF z=31 OR z=1 OR z=6 OR z=30 THEN 1080
< 6> 1060 PRINT bell$;
<71> 1070 GOTO 940
<38> 1080 IF z=30 AND hoehe<20 THEN hoehe=hoehe+1
<46> 1090 IF z=31 AND hoehe>0 THEN hoehe=hoehe-1
<91> 1100 IF z=6 AND seite< 5 THEN seite=seite+1
<41> 1110 IF z=1 AND seite>0 THEN seite=seite-1
<89> 1120 PRINT prog$
<88> 1130 IF disk$(seite,hoehe)=""THEN disk$(seite,hoeh
     e)=STRING$(12,32)
<89> 1140 prog$=FNpkt$(9+hoehe,1+seite*15,disk$(seite,h
     oehe))
<93> 1150 PRINT on$+prog$+of$
<70> 1160 GOTO 940
<81> 1170 menu$(0)="DISKWECHSEL:"
<68> 1180 menu$(1)=""
<54> 1190 GOSUB 2420
<33> 1200 GOSUB 2060
<92> 1210 GOSUB 2120
<87> 1220 IF z=8 THEN GOSUB 2100:GOTO 940
<94> 1230 GOTO 1780
<82> 1240 menu$(0)="PROGRAMMENDE. in. BASIC"
<61> 1250 menu$(1)=""
<19> 1260 GOSUB 2400
<59> 1270 IF z=8 THEN 1220
<85> 1280 PRINT cls$;con$;menu$(0):END
<22> 1290 menu$(0)="PROGRAMMENDE.. zu.. CP/M".
<48> 1300 menu$(1)=""
< 6> 1310 GOSUB 2400
<57> 1320 IF z=8 THEN 1220
<61> 1330 PRINT cls$;menu$(0);con$:SYSTEM:END
<10> 1340 menu$(0)="LADEN:
< 2> 1350 menu$(1)=of$+RIGHT$(prog$,12)+on$
<21> 1360 GOSUB 2400
<61> 1370 IF z=8 THEN 1220
<90> 1380 pruef$=RIGHT$(prog$,3)
<95> 1390 IF pruef$<>"BAS"THEN 1450
<60> 1400 IF RIGHT$(prog$,12)=STRING$(12,32) THEN 1450
<78> 1410 GOSUB 2450
<72> 1420 PRINT cls$;
<28> 1430 PRINT menu$(0);" ";menu$(1);of$;con$
<63> 1440 GOTO 2470
<94> 1460 menu$(1)="Datei kein BASIC-Programm"
<21> 1470 menu$(2)="bzw.unzulässiger Dateiname!"
<70> 1480 menu$(3)="Abbrechen!... >CAN< = Ja"
<64> 1490 GOSUB 2060
<95> 1500 GOSUB 2120
<55> 1510 GOTO 1220
<52> 1520 menu$(0)="RUN"
<61> 1530 menu$(1)=""
<87> 1540 r=1
<19> 1550 GOTO 1350
<49> 1560 menu$(0)="UMBENENNEN"
< 1> 1570 menu$(1)="Alter Name: "+RIGHT$(prog$,12)
<64> 1580 menu$(2)="Neuer Name: "+of$+STRING$(8,32)+".
     . "+on$
<76> 1590 GOSUB 2430
<41> 1600 GOSUB 2060
< 1> 1610 GOSUB 2120
<17> 1620 name$=""
<91> 1630 GOSUB 2200
<51> 1640 IF RIGHT$(prog$,12)=STRING$(12,32)THEN z$="":
     GOTO 1450
<50> 1650 IF LEN(name$)=0 THEN PRINT of$;:GOSUB 2100:GO
     TO 940
<97> 1660 GOSUB 2450
<31> 1670 PRINT of$;:NAME RIGHT$(prog$,12) AS name$
<19> 1680 OPTION FILES"A"
<38> 1690 name$=""
<95> 1700 GOTO 1780
<21> 1710 menu$(0)="L ö S C H E N"+bell$+bell$
<11> 1720 menu$(1)="Programm: "+of$+RIGHT$(prog$,12)+on
     $
<20> 1730 GOSUB 2400
<60> 1740 IF z=8 THEN 1220
<69> 1750 IF disk$(seite,hoehe)=STRING$(12,32)THEN 1450
< 0> 1760 GOSUB 2450
<73> 1770 KILL disk$(seite,hoehe)
<21> 1780 OPTION FILES"A"
<24> 1790 FOR a=1 TO 128
<16> 1800 lw$(a)=""
< 2> 1810 NEXT
<77> 1820 FOR a=0 TO 5
<48> 1830 FOR b=0 TO 22
<82> 1840 disk$(a,b)=""
<14> 1850 NEXT
<17> 1860 NEXT
<42> 1870 GOTO 2350
<96> 1880 menu$(0)="T Y P E :"
<19> 1890 menu$(1)="Datei: "+of$+RIGHT$(prog$,12)+on$
<15> 1900 GOSUB 2400
<55> 1910 IF z=8 THEN 1220
<87> 1920 menu$(0)="..... A C H T U N G :"+bell$
<51> 1930 menu$(1)="Die Dateiausgabe kann mit"
< 8> 1940 menu$(2)="...... >ALT< + >S<"
<11> 1950 menu$(3)="... angehalten werden."+bell$
<65> 1960 GOSUB 2060
<25> 1970 GOSUB 2120
<76> 1980 IF z=8 THEN 1220
< 4> 1990 PRINT cls$;
<68> 2000 GOSUB 2450
<43> 2010 DISPLAY RIGHT$(prog$,12)
<89> 2020 OPTION FILES"A"
<76> 2030 PRINT FNpkt$(29,34,on$+"Bitte >RETURN< drücke
     n!"+of$);
<55> 2040 GOSUB 2160
<42> 2050 r=2:GOTO 240
<69> 2060 FOR a=1 TO 4
<66> 2070 PRINT FNpkt$(3+a,31,on$+STRING$(29,32)+of$)
< 8> 2080 NEXT
< 7> 2090 RETURN
<67> 2100 PRINT feld$(10);cls$;voll$;
<84> 2110 RETURN
<47> 2120 FOR a=4 TO 7
<55> 2130 PRINT FNpkt$(a,33,on$+menu$(a-4)+of$)
<97> 2140 NEXT
< 2> 2150 IF z$="U" THEN RETURN
<95> 2160 z$=INKEY$:IF z$=""THEN 2160
<74> 2170 z=ASC(z$)
<92> 2180 IF z=13 OR z=8 THEN RETURN
<65> 2190 PRINT bell$:GOTO 2160
<90> 2200 a$=INKEY$:IF a$=""THEN 2200
< 4> 2210 a=ASC(a$)
<84> 2220 IF a=13 THEN RETURN
<76> 2230 IF a=127 AND LEN(name$)>0 THEN name$=LEFT$(na
     me$,LEN(name$)-1):GOTO 2320
<87> 2240 IF a=8 THEN name$="":RETURN
<47> 2250 IF a=39 OR a=32 THEN 2300
<36> 2260 IF a=46 AND LEN(name$)<9 THEN name$=name$+STR
     ING$(8-LEN(name$),32)+".":GOTO 2320
<64> 2270 IF a>96 AND a<123 OR a>47 AND a<58 THEN 2300
<15> 2280 IF a>65 AND a<90 THEN 2300
<40> 2290 PRINT bell$:GOTO 2200
<42> 2300 a$=UPPER$(a$):name$=name$+a$
<88> 2310 IF LEN(name$)>8 THEN MID$(name$,9,1)="."
<34> 2320 IF LEN(name$)>12 THEN name$=LEFT$(name$,12):P
     RINT bell$;
<81> 2330 PRINT FNpkt$(6,45,of$+STRING$(12,32)+on$);FNp
     kt$(6,45,of$+name$+on$);
<46> 2340 GOTO 2200
<79> 2350 GOSUB 2100
<44> 2360 FOR a=1 TO 8
< 4> 2370 PRINT feld$(a);cls$
<14> 2380 NEXT
<97> 2390 PRINT voll$:GOTO 550
<40> 2400 GOSUB 2420:GOSUB 2060:GOSUB 2120
<90> 2410 RETURN
<69> 2420 menu$(2)=">ENTER<.. =.. Ja"
<18> 2430 menu$(3)=">CAN<.... =.. Nein"
< 0> 2440 RETURN
<80> 2450 IF seite>2 THEN OPTION FILES"M"
< 6> 2460 RETURN
<29> 2470 IF r=0 THEN LOAD RIGHT$(prog$,12)
<26> 2480 IF r=1 THEN RUN RIGHT$(prog$,12)
< 3> 2490 END
```

Tips zum Softwarekauf

Nach Lesen der Marktübersicht wissen Sie nun so halbwegs, welche Programme für welche Anwendungen auf dem Markt sind. Natürlich hilft Ihnen eine Marktübersicht beim Kauf von Software nur wenig, wenn es darum geht, aus der Vielfalt der vorhandenen Software die für Sie richtige herauszusuchen. Aus diesem Grunde haben wir hier einige Tips zusammengestellt, die Ihnen helfen werden, die für Sie richtige Anwendersoftware zu finden.

Wenn Sie sich entschlossen haben, bestimmte Aufgaben, die in Ihrem Unternehmen anfallen, per Computer und einem entsprechenden Programm zu erledigen, sollten Sie sich erst einmal ein sogenanntes Pflichtenheft für das zu kaufende Programm anlegen. Hier tragen Sie genau ein, was dieses Programm für Sie erledigen soll.

Ein Pflichtenheft für Ihre Lagerverwaltung würde ich z.B. anhand der folgenden Fragen erstellen:

Wieviele Artikel sollen verwaltet werden? Hierbei würde ich einen genügend großen Aufschlag für spätere Erweiterungen des eigenen Sortiments einplanen.

Ist das Programm in der Lage, bei Unterschreiten eines eingegebenen Mindestbestand eine Meldung zu machen? Selbstverständlich muß es möglich sein, den Mindestbestand zu jedem Artikel extra festzulegen.

Kann das Programm einzelne Artikel zu Artikelgruppen zusammenfassen.

Kann das Programm auch Lieferantenkonten verwalten?

Unterstützt mich das Programm bei meinen Bestellungen?

Kann das Programm mir Artikelumsätze und Artikelgruppenumsätze sowohl mengen- als auch wertmäßig anzeigen?

Kann es Etiketten ausdrucken?

Druckt es Lager-, Bestell- und Bewegungslisten usw.?

Sie werden sich zur Erstellung eines Pflichtenheftes für eine Lagerverwaltung natürlich wesentlich mehr Fragen stellen müssen. Die hier exemplarisch aufgelisteten Fragen geben Ihnen jedoch hoffentlich genug Anhaltspunkte, um sich selbst ein Pflichtenheft zu erstellen.

Wenn Sie Ihr Pflichtenheft (meist ist es nur ein Zettel) soweit fertiggestellt haben, sollten Sie nicht gleich zu Ihrem Händler stürzen, um das erste Programm zu kaufen, das den Kriterien des Pflichtenheftes genügt.

Der nächste Schritt zu »Ihrem« Programm sollte zunächst darin bestehen, einen kleinen Testlauf für die Programme zu entwerfen, die Sie sich beim Händler anschauen möchten. Keine Angst, dies wird keine Doktorarbeit, bewahrt Sie jedoch später vor Enttäuschungen.

Für ein Finanzbuchhaltungsprogramm zum Beispiel denken Sie sich ungefähr dreißig Buchungen aus, die möglichst viele verschiedene Konten ansprechen. Dazu gehören mindestens eine Sammelbuchung im Soll sowie eine Sammelbuchung im Haben. Ich kenne Programme, die eine Sammelbuchung zum Abenteuer werden lassen.

Weiterhin sollten Sie mindestens jeweils zwei Warenein- und Ausgänge mit Skonto in Ihren Testlauf einbauen. Nicht jedes Programm ist in der Lage, solche Buchungen ordnungsgemäß unter Berücksichtigung der veränderten Vor- oder Umsatzsteuer durchzuführen.

Da die freie Gestaltungsmöglichkeit neuer Konten bei einigen Finanzbuchhaltungsprogrammen dort ihre Grenzen findet, wo die Salden dieser neu definierten Konten zur Ermittlung der Gewinn- und Verlustrechnung herangezogen werden sollen, ist es ratsam, sich auch mehrere Buchungen auszudenken, die ein neu zu definierendes Kosten- oder Ertragskonto betreffen.

Nachdem Sie Ihre Testbuchungen vorbereitet und natürlich auch einen Abschluß dieser Buchungen erstellt haben, sind Sie in der Lage, jedes Finanzbuchhaltungsprogramm zumindest soweit zu testen, wie es einem Laien zumutbar ist. Unter der Voraussetzung, daß Ihr »zu Fuß« gemachter Abschluß richtig ist, muß ein Programm, das den Anforderungen einer ordnungsmäßigen Buchführung genügen will, den gleichen Abschluß wie Ihre Testbuchungen produzieren.

Wenn Ihr Händler Ihnen später Programme vorführt, geben Sie Ihren Testlauf ein und können so innerhalb kürzester Zeit feststellen, ob es lohnenswert ist, sich mit dem Programm weiter zu beschäftigen.

Zur Durchführung des Testlaufes noch ein Tip: machen Sie soviel und vor allem so dumme Fehler, wie Sie eigentlich nicht unterlaufen dürften. Verzeiht Ihnen das Programm diese Fehler ohne Fehlermeldung und Abbruch des falschen Befehls (nicht des Programms), lassen Sie die Finger davon.

Ein derartiger Testlauf läßt sich, analog den zur Finanzbuchhaltung gemachten Angaben, auch mit anderen Programmen durchführen.

Nachdem Sie nun so halbwegs wissen, was ihr zukünftiges Programm leisten soll und wie Sie angebotene Programme ein wenig auf Ihre Leistungsfähigkeit hin abklopfen können, sollten Sie sich langsam einmal einen Überblick über das Angebot an entsprechenden Programmen verschaffen.

Studieren und sammeln Sie Anzeigen und Software-Reviews aus Computer-

zeitschriften und versuchen Sie, möglichst viele zusätzliche Informationen über Programme einzuholen, die Ihnen interessant erscheinen. Hören Sie sich nach Benutzern dieser Programme um. Der kompetenteste Sachverständige für ein Anwenderprogramm ist immer noch der, der schon einige Zeit damit gearbeitet hat.

So mit Vorinformationen versorgt, können Sie sich nun auf den Weg zu Ihrem Händler machen. Da viele Händler keine große Auswahl an Anwenderprogrammen bieten, kann es passieren, daß Sie verschiedene Händler aufsuchen müssen, um die Programme zu sehen und testen zu können, die Sie interessieren. Dies ist zwar etwas mühselig, erspart Ihnen jedoch manch überflüssigen Griff in die Brieftasche.

Nehmen Sie Ihr Pflichtenheft zur Hand und machen Sie Notizen zu jedem Programm, das Ihnen angeboten wird. Wenn Sie den Eindruck haben, daß ein bestimmtes Programm Ihren Vorstellungen schon ziemlich nahe kommt, dann geben Sie Ihren Testlauf ein. Notieren Sie sich auch hier wieder alles, was Ihnen zur Beurteilung des Programmes wichtig erscheint.

Wenn das Programm nach dem Testlauf die richtigen Ergebnisse produziert hat, kann es in die engere Wahl genommen werden. Wenn Sie so mehrere Programme gefunden haben, die sowohl den Testlauf bestanden haben, als auch die Kriterien Ihres Pflichtenheftes erfüllen, sind Sie Ihrem Ziel schon ziemlich nah.

Sie müssen nur noch unter einer kleinen Auswahl von guten und selbst getesteten Programmen auswählen.

Um Ihnen die Wahl zwischen gleichwertigen Programmen zu erleichtern, hier noch einige Entscheidungshilfen.

Ein nicht kopiergeschütztes Programm ist allein schon deshalb benutzerfreundlicher, weil Sie sich Sicherheitskopien erstellen können. Ich würde auf alle Fälle ein nicht kopiergeschütztes Programm kaufen.

Es gibt Programme, die so umfangreich sind, daß sie nicht komplett in den Arbeitsspeicher eingelesen werden können. Das bedeutet, daß während der Arbeit mit diesen Programmen permanent auf die Diskette zugegriffen werden muß und Sie selbst bei Ihrer Arbeit immer wieder warten müssen, bis der Diskettenzugriff beendet ist. Ein Programm, das einerseits die Speicherkapazität des Joyce überfordert, andererseits jedoch die Möglichkeit, Programmteile auf der RAM-Disc abzulegen, nicht ausschöpft, würde ich nicht kaufen.

Großen Wert lege ich bei einem Programm darauf, daß eingegebene Daten nicht bei jedem Arbeitsschritt auf die Diskette geschrieben werden müssen. Entweder muß hierzu genug Platz im Speicher sein, oder die Daten werden erst einmal auf die RAM-Disc geschrieben. Weiterhin sollte ein gutes Programm anzeigen, wenn die Kapazität der Diskette oder der RAM-Disc nicht mehr ausreicht, die eingegebenen Daten aufzunehmen. Ein Programm, welches lautstark eine Fehlermeldung ausgibt und dann mit den entsprechenden Datenverlusten abstürzt, ist natürlich nicht alltagstauglich.

Ein verständliches und gut gegliedertes Handbuch mit einem umfangreichen Index sollte heute zum Standard eines guten Programmes gehören. Leider hat sich dies noch nicht bis zu allen SoftwareHerstellern herumgesprochen.

Manche Programme werden nicht über Menues, sondern ausschließlich über Control-Codes gesteuert. Dadurch steigt die Häufigkeit von Fehlbedienungen in einem derartig hohen Maß, daß ich vom Kauf eines solchen Programmes leider abraten muß.

Ein wichtiges Kriterium bei der endgültigen Auswahl des Programms ist natürlich auch der Preis. Trotzdem würde ich nicht immer das kostengünstigste Programm auswählen. Ich würde mich bei meiner Entscheidung auch davon leiten lassen, welches Programm mir »sympathisch« erscheint.

Manchmal begegnen mir Programme, die – nach allen möglichen »objektiven« Bewertungsmaßstäben gemessen – gleich gut sind. Trotzdem habe ich das Gefühl, mit einem dieser Programme besser arbeiten zu können, als mit dem anderen. Dieses Programm ist mir einfach »sympathischer« als die anderen. Auch wenn es teurer sein sollte – ich würde ein mir sympathisches Programm kaufen und kann Ihnen nur empfehlen, ähnlich zu verfahren.

Denn für die Effizienz Ihrer Arbeit mit dem Programm ist es auch wichtig, daß Sie gern damit arbeiten. Also scheuen Sie sich nicht, unter Umständen hundert Mark mehr für ein Programm auszugeben, was Ihnen »sympathisch« ist – das Geld wird sich schnell in Form einer höheren Arbeitsproduktivität amortisiert haben.

Sparen Sie also hier nicht an der falschen Stelle. Die falsche Stelle zum Sparen ist für viele auch ein Versandhaus. Wenn Sie ganz sicher sind, daß Sie nie mehr Fragen zu einem Programm haben werden, ist ein Versandhaus vielleicht auch die richtige Adresse für Ihren Einkauf. Ansonsten bin ich jedoch der Meinung, daß im Preis für Anwenderprogramme auch ein Teil für die entsprechende Beratung kalkuliert ist. Wenn Sie diese Beratung benötigen, sollten Sie dort einkaufen, wo Sie sicher sein können, sie auch nach dem Kauf noch zu erhalten.

Einiges, was Sie jetzt gelesen haben, wird Ihnen schon bekannt gewesen sein. Anderes werden Sie vielleicht für überflüssig oder zu aufwendig erachten. Ich bin mir jedoch sicher, daß Sie die für Sie richtige Software kaufen werden, wenn Sie so vorgehen, wie es hier beschrieben wurde.

(W. Huske)

Meine Meinung zu diesem Sonderheft:

Hier haben Sie Gelegenheit, uns Ihre Meinung zu diesem Sonderheft mitzuteilen. Wir hoffen auf eine rege Teilnahme, nur so können wir unsere Fehler erkennen, Ihre Anregungen verarbeiten und es das nächste Mal besser machen, denn – Nobody's perfect !

Füllen Sie die freigelassenen Felder für Ihre Bemerkungen aus und senden Sie das Blatt (oder eine Fotokopie, falls Sie Ihr Sonderheft nicht zerpflücken möchten) an o.g. Anschrift. Wir danken schon jetzt für Ihre aktive Unterstützung!

An den
DMV Verlag
Stichwort: Joyce-Sonderheft Nr. 1
Postfach 250
3440 Eschwege

Das finde ich gut: _____

_____

_____

Das finde ich nicht gut: _____

_____

_____

Ich wünsche mir: _____

_____

_____

_____

Name und Anschrift: _____

_____

_____

Computerkonfiguration: _____

Ihre persönlichen Daten werden nur zu internen Verwaltungszwecken elektronisch gespeichert.

Von Kennern und Könnern geschätzt ...
... ein ausgezeichneter Jahrgang

Experten halten die 86er Ausgaben des führenden Schneider Magazins für einen ausgezeichneten Jahrgang.

Experten sagen voraus, daß die 87er Ausgaben noch besser und informativer werden.

Sichern Sie sich diese Qualität durch ein Abonnement.

Vom Preisvorteil gegenüber dem Einzelbezug wollen wir gar nicht weiter reden.



Schneider International im Abonnement:

Bundesrepublik Deutschland
12 Ausgaben 60,– DM
6 Ausgaben 30,– DM

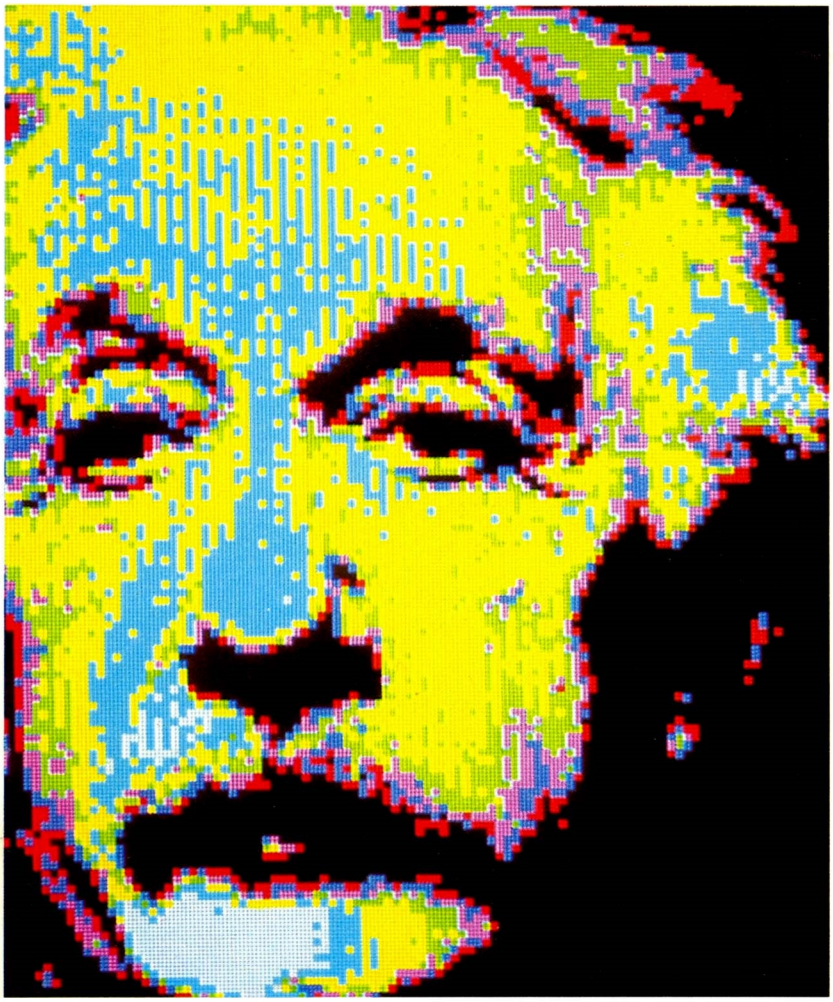
Auslandspreise:

Europa
12 Ausgaben 90,– DM
6 Ausgaben 45,– DM

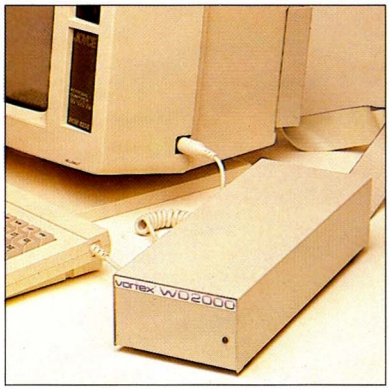
Außereuropäisches Ausland
12 Ausgaben 120,– DM
6 Ausgaben 60,– DM

Die Versandkosten sowie Porto sind in diesen Preisen bereits enthalten.

DMV Verlagsgesellschaft mbH · PC Schneider International
Postfach 250 · Fuldaer Straße 6 · 3440 Eschwege



...UND PLÖTZLICH WIRD DER SCHNEIDER JOYCE ZUM ECHTEN TEXTSYSTEM.



Um es ganz offen zu sagen:

Was fangen Sie mit einem Textverarbeitungssystem an, das weniger als 10.000 DIN A4 Seiten Speicherkapazität hat?

Deshalb unsere Empfehlung:
Rüsten Sie Ihren Schneider Joyce mit unserem Festplattensystem **WD 2000** auf, damit Sie ihn tatsächlich professionell nutzen können.

Übrigens: Die vortex-Festplatte WD 2000 mit 20 MB Speicherkapazität paßt auch zu allen anderen Schneider CPC-Computern.

Der Infoscheck bringt weitere Informationen!

J/4/87

I·N·F·O-S·C·H·E·C·K

Bitte schicken Sie mir umgehend Unterlagen für die vortex-Festplatte WD 2000 für den Schneider Joyce.

_____

_____

_____

vortex Computersysteme GmbH
Falterstraße 51-53 · 7101 Flein · Telefon (07131) 52061

vortex
COMPUTERSYSTEME

...UND PLÖTZLICH HABEN SIE EINEN PROFI-COMPUTER