

MODES 7 UND 8 IN HIGH SPEED

```

1      REM !!!!!!!!!!!!!!!  IMP INT  !!!!!!!!!!!!!!!
1000   REM KLEINES DEMO
1010   COLOR0 1 10 3 5
1020   COLORT 0 5 3 14:POKE #74,0:POKE #75,#FF
1030   MODUS%=12:REM MODE 7
1040   GOSUB 40000
1050   FILL 40,40 XMAX-40,YMAX-40 0
1060   DRAW 0,0 XMAX,YMAX 10
1070   DRAW XMAX,0 0,YMAX 10
1080   XMZ=XMAX/2:YMZ=YMAX/2:RAD%=120
1090   FOR PHI=0.0 TO 2.0*PI STEP PI/200
1100   DOT XMZ+RAD%*SIN(PHI),YMZ+RAD%*COS(PHI) 14
1110   NEXT
1120   WAIT TIME 300
1130   MODUS%=15:GOSUB 40000
1140   CURSOR 15,2:PRINT " Testen Sie selbst!   Dies ist MODE 8A."
1150   END
40000  REM SETZE MODES 7/8/7A/8A (242*512)
40010  REM AUFRUF:  MODUS%=x:GOSUB 40000
40020  REM MODUS%=12:  MODE 7      MODUS%=13:  MODE 7A
40030  REM MODUS%=14:  MODE 8      MODUS%=15:  MODE 8A
40040  REM
40050  REM NACH AUFRUF DIESER UNTERPROGRAMME SIND ALLE BILDSCHIRM-
40060  REM BEFEHLE BENUTZBAR WIE IN JEDEM ANDEREN MODUS!!
40070  REM BEI EINEM FEHLER IN EINEM NUR-GRAFIK-MODUS WIRD JEDOCH
40080  REM DAS BILD ZERSTOERT UND DIE MELDUNG IST MEISTENS NICHT
40090  REM LESBAR, DANN 'BLIND' BREAK,"MODE0" EINGEBEN!
40099  REM
40100  IF MODUS%<12 OR MODUS%>15 THEN MODE 0:PRINT :PRINT " WRONG MODUS
":STOP
40110  TEXT%=MODUS% IAND 1:COL4%=MODUS% IAND 2
40120  IF COL4%=0 THEN MODE 5:REM CURSOR LOESCHEN
40130  IF COL4%=2 THEN MODE 6
40140  HAB%=#80-(COL4% SHL 6):LDB%=#40
40150  HDB%=0:LDB%=0:IF COL4%=0 THEN HDB%=#FF:LDB%=(PEEK(#9E) IAND #F)
SHL 4
40160  ZEILENZ=#F2-TEXT%*#1E
40170  HZ=#6:IF TEXT%=0 THEN HZ=#8:REM BILD MITTIG
40175  REM COLOR-ZEILEN
40180  LINEADR%=#BFF0:COLADR%=#A1:CTR%=HAB%:GOSUB 40700
40190  IF TEXT%=0 THEN HZ=#F:COLADR%=#A1:LINEADR%=#4134:GOSUB 40700:GOT
0 40250
40195  REM DIES NUR BEI A-MODE
40200  LDB1%=LDB%:HDB1%=HDB%:HDB%=0:LDB%=0:HZ=0:COLADR%=#7F:LINEADR%=#5
0E8:CTR%=#30:GOSUB 40700
40210  HZ=#F:COLADR%=#7F:LINEADR%=#4EC0:GOSUB 40700
40220  LDB2%=LDB1%:HDB2%=HDB1%
40230  ADR%=#4EBF:ANZ%=#1E:GOSUB 40600
40240  IF COL4%=0 THEN ADR%=#3F0C:GOSUB 40500:ADR%=#4E3A:GOSUB 40500
40245  REM BILD-INIT
40250  ADR%=#BFEF:ANZ%=ZEILENZ:GOSUB 40600
40260  IF COL4%=0 THEN ADR%=#BFEF-ZEILENZ*#86+1:GOSUB 40500
40270  RESTORE:IF TEXT%=1 THEN FOR I%=1 TO 22:READ I%:NEXT
40275  REM TEST AUF RAM-PLATZ
40280  READ ADR%:IF ADR%<PEEK(#2A3)+256*PEEK(#2A4) THEN MODE 0:PRINT CH
R*(12):PRINT " OUT OF MEMORY":STOP
40285  REM PARAMETER EINPOKEN
40290  POKE #2A5,ADR% IAND #FF:POKE #2A6,ADR% SHR 8

```

```
40300 FOR ADR%=#84 TO #98:READ I%:POKE ADR%,I%:NEXT
40310 POKE #9D,MODUS%-4:REM SONST CRASH BEI MODE-WECHSEL
40315 REM SCHIRM SAEUBERN
40320 FILL 0,0 XMAX,YMAX PEEK(#9E) IAND #F:IF TEXT%=1 THEN PRINT CHR$(
12);
40330 RETURN
40340 REM
40350 REM
40498 REM IN MODE 7 EINE SCHIRMZEILE LEER SETZEN
40499 REM MUSS SEIN, SONST CRASH!!
40500 FOR LOOPADR%=-ADR%+2 TO ADR%+#80 STEP 2:POKE LOOPADR%,LDB%:POKE L
OOPADR%+1,HDB%:NEXT:RETURN
40510 REM
40599 REM CTRL-BYTES UND RAENDER SETZEN
40600 FOR I%=#0 TO ANZ%-1
40610 LOOPADR%=-ADR%-I%*#86:POKE LOOPADR%,HAB%:POKE LOOPADR%-1,LAB%:POK
E LOOPADR%-2,HDB%:POKE LOOPADR%-3,LDB%
40620 POKE LOOPADR%+#84,HDB%:POKE LOOPADR%+#85,LDB%
40630 NEXT:RETURN
40640 REM
40699 REM 4 COLOR-ZEILEN SETZEN (COLOR6, COLORT)
40700 FOR LOOPADR%=LINEADR% TO LINEADR%+15 STEP 4
40710 POKE LOOPADR%,LDB%:POKE LOOPADR%+1,HDB%:POKE LOOPADR%+2,PEEK(COL
ADR%):POKE LOOPADR%+3,CTR%+HZ
40720 COLADR%=COLADR%-1
40730 NEXT:RETURN
40740 REM
40799 REM DATEN FUER MODES 7/8
40800 DATA #4133
40810 DATA #33,#41,#3B,#80,#43,#41,#43,#41,#0B,#3F
40820 DATA #33,#41,#8F,#31,#8F,#4E,#00,#02,#F2,#1E,#86
40899 REM DATEN FUER MODES 7A/8A
40900 DATA #3F0B
40910 DATA #0B,#3F,#3B,#80,#F7,#50,#E7,#50,#CF,#4E
40920 DATA #BF,#4E,#8F,#31,#43,#41,#00,#02,#F2,#1E,#86
40999 REM GESCHAFFT!
```



```

002          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
003          x Unterprogramm zum ausfuellen eines beliebig ge- x
004          x formten Linienzuges im Graphikmode mit Farbe. x
005          x Als Basic-Programm in DALnamic-Belgien Nr.14. x
006          x Diese ML-Version ist ca. 4x schneller! x
007          x Parameteruebergabe und Unterprogrammaufruf mit: x
008          x VAR%=X+(Y SHL 16)+(C SHL 24)+(CF SHL 28) x
009          x CALLM#400,VAR% ,wobei fuer X,Y,C,CF gilt: x
010          x X,Y : eine Position innerhalb der Figur x
011          x C : Farbe der Berandungslinien der Figur x
012          x CF : Fuellfarbe, CF≠C !!! x
013          x Es ist darauf zu achten,dass die gewaehlten Far- x
014          x ben auch zulaessig sind (COLORG) und die Umrän- x
015          x dung der Figur geschlossen ist, da sonst eine x
016          x Fehlermeldung erfolgt. x
017          x Das Programm arbeitet in allen 4 Farben-modi. x
018          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
019
020          START EQU :400
021          ENDE EQU START+:117
022          ERFOR EQU :E602 ERROR HANDLING ROUTINE
023          x SCRNX(X,Y) - ROUTINE VIA RST5, DATA :27
024          x DRAW X1,Y1 X2,Y2 CF - ROUTINE VIA RST5, DATA :21
025          ADVAR EQU ENDE ADRESSE VON VAR (L,H)
026          x VERSCHIEDENE VARIABLEN:
027          X EQU ENDE+2 (L,H)
028          Y EQU ENDE+4 (L)
029          BX EQU ENDE+5 (L,H)
030          BY EQU ENDE+7 (L)
031          F EQU ENDE+8 (L)
032          K EQU ENDE+9 (L,H)
033          C EQU ENDE+11 (L)
034          CF EQU ENDE+12 (L)
035
036          ORG START
037          0400 F5 PUSH PSW PUSH ALL
038          0401 E5 PUSH H
039          0402 C5 PUSH B
040          0403 D5 PUSH D
041          x EINSPEICHERN DER WERTE IN DIE VARIABLEN
042          0404 221705 SHLD ADVAR HL UNTER ADVAR ABSP.
043          0407 7E MOV A,M FARBBYTE (C,CF) IN AKKU LAD.
044          0408 E60F ANI 15 CF-ANTEIL IN AKKU LOESCHEN
045          040A 322205 STA C FARBE C UNTER C ABSPEICHERN
046          040D 7E MOV A,M FARBBYTE (C,CF) IN AKKU LAD.
047          040E E6F0 ANI 255-15 C-ANTEIL IN AKKU LOESCHEN
048          0410 0F RRC CF SHIFTEN AUF BIT 0-3
049          0411 0F RRC
050          0412 0F RRC
051          0413 0F RRC
052          0414 322305 STA CF FARBE CF UNTER CF ABSP.
053          0417 23 INX H NAECHSTES BYTE (Y-POSITION)
054          0418 7E MOV A,M AKKU MIT Y-POSITION LADEN

```

055	0419	321805	STA	Y	Y-POSITION UNTER Y ABSP.
056	041C	23	INX	H	NAECHSTES BYTE (X-POS. HIGH)
057	041D	7E	MOV	A,M	AKKU MIT X-POS. HIGH LADEN
058	041E	321A05	STA	X+1	X-POS. HIGH UNTER X+1 ABSP.
059	0421	23	INX	H	NAECHSTES BYTE (X-POS. LOW)
060	0422	7E	MOV	A,M	AKKU MIT X-POS. LOW LADEN
061	0423	321905	STA	X	X-POS LOW-TEIL UNTER X ABSP.
062					x BEGINN DES HAUPTPROGRAMMS
063	0426	2A1905	LHLD	X	LADE HL MIT INHALT X (HL)
064	0429	23	INX	H	HL=HL+1 (X=X+1)
065	042A	221905	SHLD	X	NEUEN WERT UNTER X ABSP.
066	042D	2A1905	LOOP1	LHLD	X LADE HL MIT INHALT X (HL)
067	0430	2B	DCX	H	HL=HL-1 (X=X-1)
068	0431	221905	SHLD	X	NEUEN WERT UNTER X ABSP.
069	0434	3A1B05	LDA	Y	Y IN DEN AKKU LADEN
070	0437	4F	MOV	C,A	Y NACH C
071	0438	0D0705	CALL	SCRN	ABFRAGE PUNKT
072	043B	C22D04	JNE	LOOP1	WENN IM FELD,WEITER SUCHEN
073	043E	2A1905	LHLD	X	LADE HL MIT INHALT X (HL)
074	0441	23	INX	H	HL=HL+1 (X=X+1)
075	0442	221905	SHLD	X	NEUEN WERT UNTER X ABSP.
076	0445	221C05	SHLD	BX	BX=X UNTER BX ABSP.
077	0448	3A1B05	LDA	Y	LADE AKKU MIT INHALT Y
078	044B	321E05	STA	BY	BY=Y UNTER BY ABSP.
079	044E	3E00	MVI	A,0	
080	0450	321F05	STA	F	F ZU 0 SETZEN
081	0453	2A1905	LOOP2	LHLD	X LADE HL MIT INHLT X (HL)
082	0456	2B	DCX	H	HL=HL-1
083	0457	222005	SHLD	K	K=X-1 UNTER K ABSP.
084	045A	23	INX	H	HL=HL+1
085	045B	EB	XCHG		HL (X) NACH DE
086	045C	2A1C05	LHLD	BX	LADE HL MIT BX (HL)
087	045F	7D	MOV	A,L	VERGLEICHE BX MIT X
088	0460	8B	CMP	E	
089	0461	C28104	JNE	LBL1	WENN X≠BX SPRINGE
090	0464	7C	MOV	A,H	
091	0465	8A	CMP	D	
092	0468	C28104	JNE	LBL1	WENN X≠BX SPRINGE
093	0469	3A1E05	LDA	BY	LADE AKKU MIT INHALT BY
094	046C	5F	MOV	E,A	AKKU NACH E
095	046D	3A1B05	LDA	Y	LADE AKKU MIT INHALT Y
096	0470	B8	CMP	E	VERGLEICHE Y MIT BY
097	0471	C28104	JNE	LBL1	WENN Y≠BY SPRINGE
098	0474	3A1F05	LDA	F	LADE AKKU MIT INHALT F
099	0477	FE01	CPI	1	IST F=1?
100	0479	CA1205	JZ	RUECK	WENN F=1 BEENDEN
101	047C	3E01	MVI	A,1	LADE AKKU MIT 1
102	047E	321F05	STA	F	SETZE F ZU 1
103	0481	2A2005	LBL1	LHLD	K LADE HL MIT INHALT K
104	0484	23	INX	H	HL=HL+1 (K=K+1)
105	0485	222005	SHLD	K	NEUEN WERT FUER K ABSP.
106	0488	3A1B05	LDA	Y	LADE AKKU MIT INHALT Y
107	048B	4F	MOV	C,A	LADE Y NACH C

	CALL	SCRN	ABFRAGE PUNKT
108 048C CD0705			
109 048F C28104	JNE	LBL1	WENN NICHT IM FELD, SPRINGE
110			x RAND GEFUNDEN. ZEICHNE LINIE. AUFRUF DRAW:
111			x DE,B : X1,Y1 KOORDINATEN DER LINIE
112			x HL,C : X2,Y2 KOORDINATEN DER LINIE
113			x AKKU : FARBE DER LINIE. BEI RETURN:
114			x CY-0 : OK. CY-1: ERROR, CODE IN AKKU
115 0492 2A2005	LHLD	K	LADE HL MIT K
116 0495 2B	DCX	H	HL=K-1
117 0496 EB	XCHG		HL NACH DE
118 0497 3A1805	LDA	Y	LADE AKKU MIT INHALT Y
119 049A 47	MOV	B,A	AKKU NACH B
120 049B 4F	MOV	C,A	AKKU NACH C
121 049C 2A1905	LHLD	X	LADE HL MIT X
122 049F 3A2305	LDA	CF	LADE AKKU MIT FUELLFARBE
123 04A2 EF	RST	5	AUFRUF DRAW-ROUTINE
124 04A3 21	DATA	:21	
125 04A4 DA02E6	JC	ERROR	WENN FEHLER, ABRUCH
126 04A7 3A1805	LBL2	LDA	Y LADE AKKU MIT INHALT Y
127 04AA 3C	INR	A	ACCU=ACCU+1
128 04AB 4F	MOV	C,A	Y+1 NACH C
129 04AC 2A1905	LHLD	X	LADE HL MIT INHALT X
130 04AF CD0705	CALL	SCRN	ABFRAGE PUNKT
131 04B2 C2E904	JNE	VERZ1	WENN IM FELD, VERZWEIGEN
132 04B5 3A1805	LBL3	LDA	Y LADE AKKU MIT INHALT Y
133 04B8 4F	MOV	C,A	Y NACH C
134 04B9 2A1905	LHLD	X	LADE HL MIT INHALT X
135 04BC 23	INX	H	HL=HL+1
136 04BD CD0705	CALL	SCRN	ABFRAGE PUNKT
137 04C0 C2F304	JNE	VERZ2	WENN IM FELD, VERZWEIGEN
138 04C3 3A1805	LBL4	LDA	Y LADE AKKU MIT INHALT Y
139 04C6 3D	DCR	A	A=A-1
140 04C7 4F	MOV	C,A	Y-1 NACH C
141 04C8 2A1905	LHLD	X	LADE HL MIT INHALT X
142 04CB CD0705	CALL	SCRN	ABFRAGE PUNKT
143 04CE C2FD04	JNE	VERZ3	WENN IM FELD, VERZWEIGE
144 04D1 3A1805	LBL5	LDA	Y LADE AKKU MIT INHALT Y
145 04D4 4F	MOV	C,A	Y NACH C
146 04D5 2A1905	LHLD	X	LADE HL MIT INHALT X
147 04D8 2B	DCX	H	HL=HL-1
148 04D9 CD0705	CALL	SCRN	ABFRAGE PUNKT
149 04DC CA5304	JZ	LOOP2	WENN RAND, WEITER I. SCHL.
150 04DF 2A1905	VERZ4	LHLD	X LADE HL MIT X
151 04E2 2B	DCX	H	X=X-1
152 04E3 221905	SHLD	X	NEUEN WERT FUER X ABSP.
153 04E6 C3C304	JMP	LBL4	
154 04E9 3A1805	VERZ1	LDA	Y LADE AKKU MIT INHALT Y
155 04EC 3C	INR	A	Y=Y+1
156 04ED 321805	STA	Y	NEUEN WERT FUER Y ABSP.
157 04F0 C3D104	JMP	LBL5	
158 04F3 2A1905	VERZ2	LHLD	X LADE HL MIT INHALT X
159 04F6 23	INX	H	X=X+1
160 04F7 221905	SHLD	X	NEUEN WERT FUER X ABSP.

```

161 04FA C3A704          JMP   LBL2
162 04FD 3A1B05      VERZ3 LDA   Y          LADE AKKU MIT INHALT Y
163 0500 3D          DCR   A          Y=Y-1
164 0501 321B05          STA   Y          NEUEN WERT FUER Y ABSP.
165 0504 C3B504          JMP   LBL3
166                   x UNTERPROGRAMM SCRNR-ABFRAGE UND VERGLEICHEN.
167                   x SCRNR(X,Y)-ROUTINE AUFRUF:
168                   x HL=X, C=Y: BEI RETURN:CY=0 : OK, FARBE IN AKKU
169                   x CY=1 : ERROR, CODE IN AKKU
170 0507 EF          SCRNR RST   S          AUFRUF SCRNR(X,Y)-ROUTINE
171 0508 27          DATA :27
172 0509 DA02E6      JC    ERROR        WENN FEHLER, ABRUCH
173 050C 4F          MOV   C,A          FARBE NACH C
174 050D 3A2205      LDA   C          AKKU MIT RANDFARBE LADEN
175 0510 B9          CMP   C          HAT PUNKT RANDFARBE?
176 0511 C9          RET
177 0512 D1          RUECK POP  D          POP ALL
178 0513 C1          POP  B
179 0514 E1          POP  H
180 0515 F1          POP  PSW
181 0516 C9          RET          ENDE DES UP.
182                   x VARIABLEN-SPEICHERBEREICH
183 0517          RES   13          13 BYTE FUER VARIABLEN
184 0524          END          ENDE DES BENUTZTEN SPEICHERS

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXX
 xxxxx Symbol Tabelle xxxxx
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

ADVAR	0517	8X	051C	BY	051E	C	0522
CF	0523	ENDE	0517	ERROR	E602	F	051F
K	0520	LBL1	0481	LBL2	04A7	LBL3	04B5
LBL4	04C3	LBL5	04D1	LOOP1	042D	LOOP2	0453
RUECK	0512	SCRNR	0507	START	0400	VERZ1	04E9
VERZ2	04F3	VERZ3	04FD	VERZ4	04DF	X	0519
Y	051B						

```

0400 F5 E5 C5 D5 22 17 05 7E E6 0F 32 22 05 7E E6 F0
0410 0F 0F 0F 0F 32 23 05 23 7E 32 18 05 23 7E 32 1A
0420 05 23 7E 32 19 05 2A 19 05 23 22 19 05 2A 19 05
0430 2B 22 19 05 3A 1B 05 4F CD 07 05 C2 2D 04 2A 19
0440 05 23 22 19 05 22 1C 05 3A 1B 05 32 1E 05 3E 00
0450 32 1F 05 2A 19 05 2B 22 20 05 23 EB 2A 1C 05 7D
0460 8B C2 81 04 7C 8A C2 81 04 3A 1E 05 5F 3A 1B 05
0470 8B C2 81 04 3A 1F 05 FE 01 CA 12 05 3E 01 32 1F
0480 05 2A 20 05 23 22 20 05 3A 1B 05 4F CD 07 05 C2
0490 81 04 2A 20 05 2B EB 3A 1B 05 47 4F 2A 19 05 3A
04A0 23 05 EF 21 DA 02 E6 3A 1B 05 3C 4F 2A 19 05 CD
04B0 07 05 C2 E9 04 3A 1B 05 4F 2A 19 05 23 CD 07 05
04C0 C2 F3 04 3A 1B 05 3D 4F 2A 19 05 CD 07 05 C2 FD
04D0 04 3A 1B 05 4F 2A 19 05 2B CD 07 05 CA 53 04 2A
04E0 19 05 2B 22 19 05 C3 C3 04 3A 1B 05 3C 32 1B 05
04F0 C3 D1 04 2A 19 05 23 22 19 05 C3 A7 04 3A 1B 05
0500 3D 32 1B 05 C3 B5 04 EF 27 DA 02 E6 4F 3A 22 05
0510 B9 C9 D1 C1 E1 F1 C9 00 00 00 00 00 00 00 00
0520 00 00 00 00

```

VERSCHIEBE-PROGRAMM FÜR DOS-BENUTZER

Dieses Hilfsprogramm sorgt dafür, daß Programme, die in einen für DOS-Benutzer ungünstigen Bereich (-#19CB) umgewandelt wurden, ohne größeren Umstand auf einer Diskette gesammelt und über Directory-Anzeige ausgewählt, geladen und ausgeführt werden können.

Das Programm wird nach \$DKDIR, \$DKBOOTS, \$MSTRDOS als \$USER.BAS auf die gewünschte Diskette geschrieben.

Alle zu behandelnden Files müssen nun folgendermaßen aufbereitet werden:

1. DAI einschalten, Laufwerk aus.
2. Laden des Files von Cassette.
3. Verschieben des Files im UT an die Adresse #2006.
4. Auffüllen folgender Adressen (Hex. Low-Byte, High-Byte!)
 - a) #2000, #2001 Original-Anfang
 - b) #2002, #2003 Original-Länge (=Ende+1-Anfang)
 - c) #2004, #2005 Bei ML-Programmen: Entry-Point
Bei BASIC/ML-Programmen: 0
5. Laufwerk einschalten, DOS per RESET laden.
6. DSAVE name.BIN:0 2000 2000+länge+6 entry

BASIC/ML-Programme, die mit RUN gestartet werden, müssen dabei allerdings ab Adresse 0 bis Ende Symboltabelle verschoben und geschrieben werden.

Legt man nun die Diskette in Laufwerk 0 und drückt RESET, wird zur Auswahl das Directory gelistet. Auf den Prompt "Name?" ist nur der File-Name (nicht BIN:0) anzugeben.

Der gewünschte File wird nun geladen und an die richtige Stelle verschoben. Dabei wird das DOS zerstört. Deshalb kann ein neuer File nur mit RESET geladen werden.

ML-Programme werden sofort ab Entry-Point ausgeführt, BASIC-Programme müssen mit RUN gestartet werden.

Sollen "normale" Files geladen werden, kann das Programm nach der Directory-Anzeige abgebrochen werden.

```
100 DATA #2A,#00,#20,#EB,#2A,#02,#20
110 DATA #44,#4D,#2A,#04,#20,#22,#29
120 DATA #B3,#21,#06,#20,#7E,#12,#13
130 DATA #23,#0B,#AF,#B8,#C2,#12,#B3
140 DATA #B9,#C2,#12,#B3,#2A,#29,#B3
150 DATA #AF,#BC,#CA,#A0,#C7,#E9,#00,#00
200 FOR I=0 TO 42
210 READ A:POKE #B300+I,A
220 NEXT
300 BEF$="DIR 0":GOSUB 1000
310 PRINT :INPUT "NAME";DSN$
330 BEF$="DLOAD "+DSN$+".BIN:0":GOSUB 1000
400 BEF$="TINYDOS":GOSUB 1000
540 CALLM #B300
590 END
1000 POKE #131,3:PRINT BEF$:POKE #131,0
1090 RETURN
```


Drucken der Titel von Cassettenprogrammen

Das Programm "CHECK-LIST-DATA" von Manfred Burmeister erinnerte mich an ein unangenehmes Verhalten des DAI: Beim Aufruf der CHECK-Routine wird der Titel nur auf dem Bildschirm ausgegeben und nicht über die RS 232 auf einen Drucker. Das ist zwar verständlich, um ein ungestörtes Lesen des Bandes zu gewährleisten, aber dennoch ärgerlich. Man könnte sonst diese Routine aufrufen, gemütlich eine Tasse Kaffee trinken und in der Zwischenzeit hätte man ein gedrucktes Inhaltsverzeichnis einer Programmcassette.

Das Programm von Manfred Burmeister erschien mir für diese simple Aufgabe zu aufwendig. Ich habe deshalb das nachstehende 28 Zeilen kurze Programm entwickelt, das das Drucken der Titel übernimmt.

Zuerst wird der Cursor gelöscht, der sonst Fehler beim Lesen der Titel verursachen könnte. Sodann wird ein kurzes Maschinenprogramm abgespeichert, das dem Aufruf einer Routine zum Lesen des Programmvorspanns dient. Nach dem Aufruf dieses Maschinenprogramms wird der Cassettenrecorder wieder ausgeschaltet. Anschließend wird der Titel durch PEEK-Befehle "vom Bildschirm", also aus dem Video-RAM, gelesen. Nachfolgend wird der Titel noch einmal ge-PRINT-et. Auf dem Bildschirm sieht man nichts, da der Titel genau über den schon vorhandenen geschrieben wird. Aber diesmal erfolgt auch eine Ausgabe auf den Drucker!

Soll der nächste Titel gelesen werden, so ist die Leertaste zu drücken. Sollen alle Programmtitel der Cassette automatisch hintereinander gedruckt werden, so ist Zeile 260 zu löschen. Beim MDCR gibt es da jedoch manchmal Schwierigkeiten, da der Recorder am Ende mancher Programme zum Anfang zurückspult, obwohl noch Programme auf Band sind. Da die Laufzeit einer Minicassette ohnehin nur 1 1/2 Minuten ist, lasse man also besser die Zeile 260 drin, bleibe dabei und unterbreche gegebenenfalls durch BREAK.

Wenn man die Titel schon in eine String-Variable eingelesen hat, so bietet sich auch eine permanente Speicherung auf Band an. Das wäre ganz leicht in einem String-Array möglich. Hier wurde im Programm Nr. 2 ein anderer Weg beschritten. Ähnlich wie im Programm von Manfred Burmeister, der DATA-Zeilen verwendete, werden hier die Titel in PRINT-Zeilen geschrieben.

Die PRINT-Zeilen sollten sinnvollerweise am Ende des Programms stehen, da sonst eine zu frühe, unnütze Ausführung erfolgen kann.

Wo die speziell gekennzeichneten PRINT-Zeilen stehen, findet das Programm selbst heraus. Es prüft, ob der erste Befehl nach einer Zeilennummer ein PRINT-Befehl ist, ob der Raum zwischen den Anführungszeichen lang genug für den Titel ist, ob der erste Buchstabe ein . (Punkt) ist und ob fünf Leerzeichen folgen. Ist das der Fall, so wird der Titel in die PRINT-Zeile ge-POKE-t. Um die Prüfzeit kurz zu halten, wird

nur auf 5 Blanks geprüft. Nach Punkt und 5 Leerzeichen noch vorhandener Text kann also überschrieben werden. PRINT-Befehle, die anders aufgebaut sind, bleiben unberührt.

Der Raumersparnis halber sind nur einige PRINT-Zeilen aufgelistet. Nach meinen Erfahrungen sind 60 reservierte PRINT-Zeilen empfehlenswert, wenn auch manche Cassette noch mehr Titel enthalten kann. Dann muß allerdings auch der Start der Utility (Zeile 1020) höher gelegt werden.

Da der Titel maximal 60 Zeichen hat (sonst gibt es ohnehin schon Schwierigkeiten mit dem DAI), sollten hinter dem PRINT-Befehl, dem Anführungszeichen und dem Punkt noch 59 Leerzeichen stehen.

Mit Hilfe dieser PRINT-Zeilen kann das Inhaltsverzeichnis einer Cassette auch Nachts ganz leise auf Band gespeichert werden und der Drucker kann es am nächsten Tag herunternageln. Da die Programmlänge sich nicht ändert, bietet sich eine jeweils aktualisierte Rückspeicherung am Anfang einer Cassette an.

Noch einige Anmerkungen: Um leichte Programmänderungen zu ermöglichen, sind die logisch zusammengehörigen Programteile bis Zeile 1420 durch gleiche Endnummern (0,1,2,3,4,5) kenntlich gemacht. Das Programm wartet ebenfalls auf das Drücken der Leertaste. Soll es automatisch weitergehen, so ist Zeile 1393 zu löschen. Wenn keine PRINT-Zeile mehr frei ist, wird dies hinter dem Titel vermerkt. Die Ausgabe des Titels auf den Drucker - ohne diesen Vermerk - ist immer noch möglich. Soll das Lesen von Titeln auf die Anzahl der PRINT-Zeilen beschränkt bleiben so ist (auch) Zeile 1404 zu löschen.

Nun denn, ich hoffe, das Programm ist hilfreich.

Mit freundlichem TamderaDAI

Dr. Jürgen Palm

Drucken der Titel von Cassettenprogrammen
(Ausgabe via RS 232/V 24 mit 150 Baud)

```
100 POKE #74,0:POKE #75,0:POKE #FFF5,2
110 FOR I%=#1000 TO #1011
120 READ P%
130 POKE I%,P%
140 NEXT
150 REM Abfrage der Titel der Cassettenprogramme
160 NAME$=""
170 CALLM #1000
180 POKE #40,#30
190 X%=CURX:CURSOR 0,CURY
200 S%=PEEK(#72)+256*PEEK(#73)
210 FOR I%=1 TO X%
220 NAME$=NAME$+CHR$(PEEK(S%))
230 S%=S%-2
240 NEXT
250 PRINT NAME$
260 CALLM #D6DA
270 GOTO 150
280 DATA #F5,#E5,#D5,#C5,#21,0,0,1,#FF,0,#CD,#CE,2,#C1,#D1,#E1,#F1,#C9
*
```

```

101 REM =====
111 REM '   SUCH E NACH EINER FREIEN 'PRINT'-ZEILE
121 REM =====
130 CLEAR 256
141 POINTER%=PEEK(#29F)+256*PEEK(#2A0) \
151 FIN%=PEEK(#2A1)+256*PEEK(#2A2)-1
161 H1%=1
171 REM '   -----
181 GOTO 1000:REM Sprung zum Hauptprogramm
191 REM '   -----
201 REM ==Start der Subroutine zur 'PRINT'-Zeilensuche=
211 REM
221 IF PEEK(POINTER%+3)=#AD THEN 291
231 POINTER%=POINTER%+PEEK(POINTER%)+1
241 IF POINTER%<FIN% THEN 221
251 REM
261 REM Keine 'PRINT'-Zeile mehr frei. Zeiger H1%=0
271 REM
281 H1%=0:GOTO 341
291 H1%=POINTER%+8:H2%=0
301 IF PEEK(H1%)<>#2E THEN 231
311 H2%=H2%+1
321 IF PEEK(H1%+H2%)<>#20 THEN 231
331 IF H2%<5 THEN 311
341 RETURN
1000 REM
1010 REM
1020 MA%=#2000:REM Startadresse des Maschinenprogramms
1030 POKE #74,0:POKE #75,0:REM Kein CURSOR
1042 POKE #FFF5,2:REM 150 BAUD
1050 REM
1060 REM =====Utility laden=====
1070 REM
1080 FOR I%=MA% TO MA%+17
1090 READ P%
1100 POKE I%,P%
1110 NEXT
1120 DATA #F5,#E5,#D5,#C5,#21,0,0,1,#FF,0,#CD,#CE,2,#C1
1130 DATA #D1,#E1,#F1,#C9
1140 REM
1150 REM ===Abfrage der Titel der Cassettenprogramme===
1160 REM
1170 NAMES=""
1180 CALLM MA%
1190 POKE #40,#30:REM Cassettenrecorder ausschalten
1200 X%=CURX:CURSOR 0,CURY:REM Titellaenge und Position
1210 S%=PEEK(#72)+256*PEEK(#73)
1221 GOSUB 221
1231 IF H1%=0.0 THEN 1260
1241 IF X%<=PEEK(H1%-1) THEN 1260
1251 IF H1%<>0 THEN GOSUB 231:GOTO 1231
1260 FOR I%=0 TO X%-1
1270 CHAR%=PEEK(S%)
1281 IF H1%<>0 THEN POKE H1%+I%,CHAR%
1290 NAMES=NAMES+CHR$(CHAR%)
1300 S%=S%-2
1310 NEXT
1320 PRINT NAMES
1331 IF H1%<>0.0 THEN 1381
1341 POKE #131,1:WAIT TIME 5
1351 CURSOR X%+1,CURY+1
1361 PRINT "→ Keine 'PRINT'-Zeile mehr frei."
1371 POKE #131,0:WAIT TIME 5
1381 REM
1393 CALLM #D6DA
1404 GOTO 1150
1415 N%=N%+1:IF N%<3 THEN 1150:REM FUER 3 TITEL

```

Drucken der Titel von Cassettenprogrammen
 und Abspeichern der Titel in PRINT-Zeilen
 (Ausgabe via RS 232/V 24 mit 150 Baud)

Mit dem nachfolgend aufgelisteten Basic-Unterprogramm ist es auf einfache Weise moeglich, die im DAI-Basic nicht implementierten Anweisungen

- GOTD Label
- GOSUB Label
- RESTORE Label

auszufuehren. Dies wird durch den Aufruf eines ml-Programmes, das durch die Anweisungen in den Zeilen 50030 bis 50080 in den HEAP des Basic-Programmes geladen wird, erreicht. In Zeile 50090 werden die verschiedenen Eingangspunkte fuer das ml-Programm definiert.

Das ml-Programm ist verschieblich und kann daher an jeder Stelle im Benutzer-RAM liegen.

Die Aufrufe fuer die verschiedenen Eingangspunkte sehen wie folgt aus

- CALLM LGOTD,Label fuer GOTD Label
- CALLM LGOSUB,Label fuer GOSUB Label
- CALLM LRESTORE,Label fuer RESTORE Label

'Label' ist hierbei der Name einer beliebigen INTEGER-Variablen die, zum Zeitpunkt des Aufrufes, eine gueltige Zeilen-Nr. enthalten muss. Bei LRESTORE muss dies die Zeilen-Nr. einer DATA-Zeile sein. LGOTD, LGOSUB, LRESTORE und XML sind ebenfalls INTEGER-Variable.

```

50000 REM
50010 REM UPRO - GOTD Label, GOSUB Label, RESTORE Label
50020 REM -----
50030 DIM XML(14)
50040 XML(0)=#23234440:XML(1)=#C363DFC5:XML(2)=#23234440
50050 XML(3)=#CDEDE6C1:XML(4)=#C32DDFD5:XML(5)=#23235623
50060 XML(6)=#5EESD1CD:XML(7)=#F6CA3E04:XML(8)=#D2F5D9E5
50070 XML(9)=#2323237E:XML(10)=#E1FEA23E:XML(11)=#04C2F5D9
50080 XML(12)=#2224013E:XML(13)=#FF322301:XML(14)=#C9000000
50090 LGOTD=VARPTR(XML(0)):LGOSUB=LGOTD+7:LRESTORE=LGOTD+19
50100 RETURN

```

```
10 REM .... Demo-Programm fuer GOTO Label,
12 REM .... GOSUB Label,
15 REM .... RESTORE Label
20 REM
25 CLEAR 500
30 GOSUB 50000:REM xl-Programm erstellen
35 LGS1=400:LGT2=40:RESTORE
40 PRINT CHR$(12);" Demo fuer GOTO Label, GOSUB Label, RESTORE Labe
  1"
45 CURSOR 0,20
50 INPUT "Demo-Nr. : (1)=LGOTO, (2)=LGOSUB), (3)=LRESTORE ";LGT1
55 CURSOR 0,17
60 LGT1=LGT1*100
70 CALLM LGOTO,LGT1
80 REM -----
100 PRINT "Demo GOTO Label"
110 LGT1=500:GOTO 70
120 REM -----
200 CALLM LGOSUB,LGS1
210 GOTO 110
220 REM -----
300 PRINT "Demo RESTORE Label"
310 INPUT "DATA-Zeilen-Nr. (600, 610, 620) ";LRS1:PRINT
320 CALLM LRESTORE,LRS1
330 READ N:FOR I=1 TO N:READ D:PRINT D:NEXT
340 GOTO 110
350 REM -----
400 PRINT "Demo GOSUB Label"
410 RETURN
420 REM -----
500 PRINT "SPACE = weiter RETURN = Ende"
510 G=G+1:IF G=0 GOTO 510
520 IF G<13 THEN CALLM LGOTO,LGT2
530 END
540 REM -----
600 DATA 5,600,601,602,603,604
610 DATA 3,610,611,612
620 DATA 6,620,621,622,623,624,625
700 END
50000 REM
50010 REM UPRD - GOTO Label, GOSUB Label, RESTORE Label
50020 REM -----
50030 DIM XML(14)
50040 XML(0)=#23234440:XML(1)=#03630FC5:XML(2)=#23234440
50050 XML(3)=#0DEDE6C1:XML(4)=#0320DFD5:XML(5)=#23235623
50060 XML(6)=#5EE0D1CD:XML(7)=#F6CA3E04:XML(8)=#02F5D9E5
50070 XML(9)=#2323237E:XML(10)=#E1FEA23E:XML(11)=#04C2F5D9
50080 XML(12)=#2224013E:XML(13)=#FF322301:XML(14)=#09000000
50090 LGOTO=VARPTR(XML(0)):LGOSUB=LGOTO+7:LRESTORE=LGOTO+19
50100 RETURN
```

Harald Kogler 06/06/83

Diese Routine ersetzt "INPUT" bei der Eingabe von Zeichenketten mit folgenden Vorteilen:

1. automatisches Umschalten auf Kleinbuchstaben
2. Eingabe auch vom "Komma"
3. Vorwahl der maximalen Laenge des Strings
4. "Klingelzeichen" fuerf Buchstaben vor Rand

Aufruf: N%=60:GOSUB 61000 N%=max. moegl. Laenge (<220)

Rueckkehr: eingegebener String = X\$

gebrauchte Variablen: AAA%, LLL%, N%, X\$

```
61000 LLL%=0:X$="":POKE #2C3,#FF
61010 AAA%=GETC:IF AAA%=0 THEN 61010
61020 IF AAA%=8 AND LLL%=0 THEN 61010
61030 IF AAA%=8 THEN PRINT CHR$(8);:LLL%=LLL%-1:X%=LEFT$(X%,LLL%):GOTO61010
61040 IF AAA%=13 THEN POKE #2C3,0:PRINT:RETURN
61050 IF LLL%<N% THEN PRINT CHR$(AAA%);:X%=X%+CHR$(AAA%):LLL%=LLL%+1
61060 IF LLL%=N%-5 THEN SOUND 0 1 15 0 FREQ(1000):WAIT TIME 5:SOUND OFF
61070 GOTO 61010
```

Ins RAM laden ? J

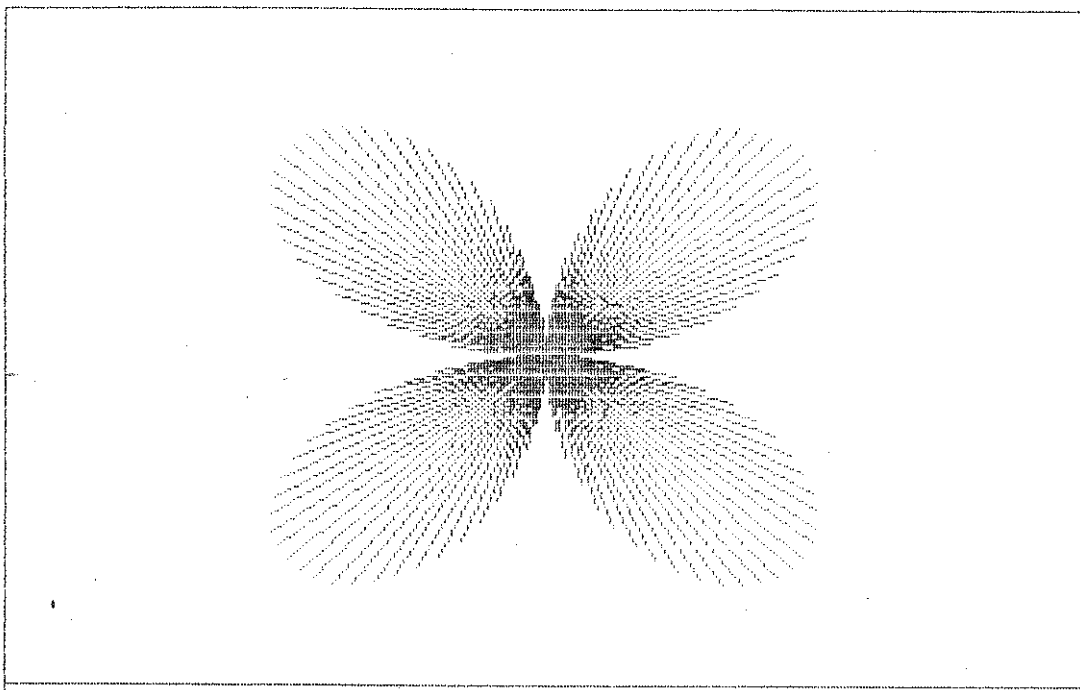
```

0 / *****
1 / * Graphikcopy V2 30.4.83 Helge Rebhan *
2 / *
3 / * Bewirkt Hardcopyausdruck im allen Graphikmodi *
4 / * Geschrieben fuer Epson MX80 F/T II. Kann aber *
5 / * durch aendern der Escape-Sequenzen fuer alle *
6 / * graphikfaehigen Drucker benutzt werden. *
7 / *
8 / *****
9 /
10 DW PRINT: DB44
11 DW SPACE: 411B / Escape-Sequenzen
12 DW CARR: OD08
13 DW ESCAPE: 4B1B
14 /
15 ORG 1000
1000 C5 16 PUSHB
1001 3E 03 17 MVIA 3 / Drucker einschalten
1003 32 31 01 18 STA 0131
1006 2A 94 00 19 LHLD 0094
1009 22 B1 10 20 SHLD XMAX
100C 3A 98 00 21 LDA 0098
100F 6F 22 MOVLA
1010 26 00 23 MVIH 00
1012 22 B3 10 24 SHLD XBYTE / Anzahl der Bytes/Zeile
1015 11 EA BF 25 LXID BFEA
1018 19 26 DADD / Topadresse
1019 EB 27 XCHG
101A 2A B3 10 28 NEXTLN: LHLD XBYTE
101D 29 29 DADH
101E 29 30 DADH / Anzahl der Bytes fuer
101F 29 31 DADH / 8 Zeilen subtr.
1020 EB 32 XCHG
1021 CD 1A DE 33 CALL DE1A
1024 E5 34 PUSHH
1025 EB 35 XCHG
1026 2A 88 00 36 LHLD 0088
1029 CD 14 DE 37 CALL DE14 / Ende des Videograms erreicht ?
102C D2 97 10 38 JNC ENDE / ja
102F 3E 08 39 MVIA 8
1031 21 AB 10 40 LXIH TEXT / Escape K + (XMAX) ausgeben
1034 CD 44 DB 41 CALL PRINT / Drucker damit auf Graphik
1037 E1 42 POPH / gesetzt
1038 E5 43 PUSHH
1039 E5 44 PUSHH
103A 2A B3 10 45 LHLD XBYTE
103D 7D 46 MOVAL
103E D6 06 47 SUI 6
1040 4F 48 MOVCA
1041 44 49 MOVBH
1042 E1 50 POPH
1043 C5 51 ZEILE: PUSHB
1044 1E 08 52 MVIE 8 / Position der Bits
1046 06 80 53 MVIB 80 / Maske zum Herausfiltern der
1048 E5 54 BLOCK: PUSHH / Bits
1049 0E 00 55 MVIC 00 / Zaehler
104B 51 56 MOVDC
104C 0C 57 BYTE: INRC / naechstes Byte
104D C5 58 PUSHB
104E 7E 59 MOVAM
104F A0 60 ANA B
1050 F5 61 PUSHPSW
1051 79 62 MOVAC
1052 93 63 SUB E

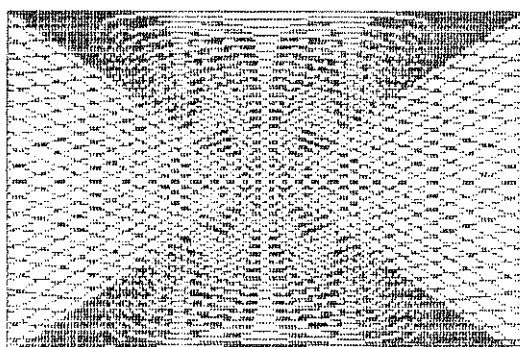
```


1053	4F		64	MOVCA	/ in C jetzt Anzahl der noetigen
1054	CA	6C	10	JZ CONT	/ RLC oder RRC Befehle
1057	FA	63	10	JM MINUS	
105A	F1		67	POPFSW	
105B	07		68	LOOP1: RLC	
105C	0D		69	DCR C	
105D	C2	5B	10	JNZ LOOP1	
1060	C3	6D	10	JMP LABEL	
1063	F1		72	MINUS: POPFSW	
1064	0F		73	LOOP2: RRC	
1065	0C		74	INR C	
1066	C2	64	10	JNZ LOOP2	
1069	C3	6D	10	JMP LABEL	
106C	F1		77	CONT: POPFSW	/ Bit an richtiger Position
106D	B2		78	LABEL: ORA D	/ nach D abspeichern
106E	57		79	MOVDA	
106F	D5		80	PUSHD	
1070	EB		81	XCHG	
1071	2A	B3	10	LHLD XBYTE	/ Adr. fuer naechstes Byte
1074	19		83	DADD	
1075	D1		84	POPD	
1076	C1		85	POPB	
1077	79		86	MOVAC	
1078	FE	08	87	CPI B	/ Byte komplett ?
107A	C2	4C	10	JNZ BYTE	
107D	7A		89	MOVAD	
107E	CD	60	DD	CALL DD60	/ ja, ausdrucken
1081	1D		91	DCR E	
1082	78		92	MOVAB	
1083	0F		93	RRC	/ Maske aendern
1084	47		94	MOVBA	
1085	E1		95	POPH	
1086	D2	48	10	JNC BLOCK	
1089	C1		97	POPB	/ 8*8 Bits sind ausgedruckt
108A	2B		98	DCXH	
108B	2B		99	DCXH	/ Videoadr -2
108C	0B		100	DCXB	/ Bytes/Zeile -2
108D	0B		101	DCXB	
108E	78		102	MOVAB	
108F	B1		103	ORA C	/ = 0 ?
1090	C2	43	10	JNZ ZEILE	
1093	D1		105	POPD	/ ja, naechste Zeile
1094	C3	1A	10	JMP NEXTLN	
1097	CD	5E	DD	107 ENDE: CALL DD5E	/ Print CR
109A	3E	1B		108 MVIA 1B	/ Print Escape 2
109C	CD	60	DD	109 CALL DD60	/ setzt Drucker in den
109F	3E	32		110 MVIA 32	/ urspruenglichen Zustand
10A1	CD	60	DD	111 CALL DD60	/ zurueck
10A4	AF		112	XRAA	
10A5	32	31	01	113 STA 0131	/ Drucker ausschalten
10A8	E1		114	POPH	
10A9	C1		115	POPB	
10AA	C9		116	RET	
10AB	1B	41		117 TEXT: SPACE	
10AD	08	0D		118 CARR	
10AF	1B	4B		119 ESCAPE	
10B1	00		120	XMAX: NOP	/ Buffer fuer XMAX
10B2	00		121	NOP	
10B3	00		122	XBYTE: NOP	/ und Bytes/Zeile
10B4	00		123	NOP	

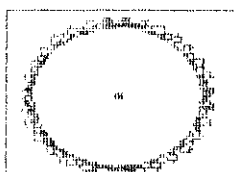
Ein Beispiel in Mode 6 ...



und eins in Mode 4 ...



am schoensten ist Mode 2 ...



P A I N T

PAINT ist eine 'Spende' der Firma INDATA. Der Autor ist mir leider nicht bekannt.

Dies ist eine Maschinenroutine ähnlich der in NP29 beschriebenen. Sie erlaubt das Füllen geschlossener Bereiche beliebiger Form und 'kriecht' um alle Ecken.

Kurzbeschreibung:

1. rasend schnell!!!
2. läuft nur in MODE 6 (bei Y-Anpassung auch in MODE 6A, d.h. im A-Modus Y%+#2C. Oben ist immer 255)
3. füllt bis zu 64 Bereiche gleichzeitig
4. Füllfarbe kann gleich Randfarbe sein
5. der Rand muß geschlossen sein! (Bildschirmrand ist keine Grenze!)

Zur Durchführung muß ein beliebiger Punkt innerhalb der Fläche, die Füll- und die Randfarbe angegeben werden:

POKE in #200C und #200D	X% (low,high)
POKE in #200E	Y% (oben ist immer 255!!)
POKE in #200F	Füllfarbe (0..3 oder 20..23)
POKE in #2010	Randfarbe (0..3 oder 20..23)

CALLM #2000 um eine Fläche zu füllen.

Um mehrere Felder gleichzeitig zu füllen:

1. CALLM #2003 um die FIFO-Pointer zurückzusetzen
 2. Parameter einpoken (alle Werte können unterschiedlich sein)
 3. CALLM #2006 Parameter übergeben
- Schritte 2 und 3 bis zu 64 mal wiederholen
4. CALLM #2009 Start von PAINT

Kleines Demo:

```
10 REM MULTIPAIN
20 COLORG 8 0 5 14
50 MODE 0:MODE 6
100 GOSUB 1000
200 CALLM #2003
210 POKE #200C,160:POKE #200D,0
220 POKE #200E,5
230 POKE #200F,2:POKE #2010,0
240 CALLM #2006
250 POKE #200E,240:POKE #200F,2:CALLM #2006
260 POKE #200E,100:POKE #200F,0:CALLM #2006
270 CALLM #2009
500 WAIT TIME 200:GOTO 50
1000 FILL 20,4 XMAX-20,YMAX-4 21
1010 FOR I=1 TO 100
1020 H=RND(15)+2:V=RND(25)+2:X=RND(270)+30:Y=RND(180)+40
1030 DRAW X-H,Y-V X+H,Y-V 20:DRAW X+H,Y-V X+H,Y+V 20
1040 DRAW X+H,Y+V X-H,Y+V 20:DRAW X-H,Y+V X-H,Y-V 20
1050 NEXT
1060 RETURN
```

Übersetzt und bearbeitet: Bernd Preusing

Hex-Dump des PAINT-Programmes:

(Unterstrichene Adressen müssen beim Verschieben angepaßt werden!)

```

2000  C3 2B 21 C3 35 21 C3 41 21 C3 56 21 00 00 00 00
2010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2020  00 00 00 00 00 2B 20 2B 20 00 00
    
```

Bereich von #202B bis #212A unwichtig (Pufferspeicher).

```

212B  CD 35 21 CD 41
2130  21 CD 56 21 C9 E5 21 2B 20 22 25 20 22 27 20 E1
2140  C9 E5 D5 C5 F5 CD FB 23 2A 11 20 22 13 20 CD 82
2150  22 F1 C1 D1 E1 C9 E5 D5 C5 F5 2A 25 20 EB 2A 27
2160  20 7B AD CA A3 21 11 11 20 7E 12 23 13 7E 12 23
2170  7E 32 15 20 23 7E 0F 0F E6 03 32 0F 20 7E E6 03
2180  32 10 20 CD 43 24 AF 2F 32 22 20 32 24 20 CD AB
2190  21 2A 27 20 22 29 20 CD 0D 23 2A 29 20 22 27 20
21A0  C3 5A 21 F1 C1 D1 E1 C9 E5 D5 C5 F5 CD 8E 23 CD
21B0  44 22 CD 33 23 3A 1C 20 32 21 20 47 3A 22 20 AB
21C0  CA CB 21 7B B7 CC 6B 22 3A 1E 20 32 23 20 47 3A
21D0  24 20 AB CA DB 21 7B B7 CC 7B 22 3A 15 20 E6 01
21E0  C2 00 22 3A 1D 20 B7 C2 3F 22 3A 15 20 0F 32 15
21F0  20 3A 21 20 32 22 20 3A 23 20 32 24 20 C3 AF 21
2200  CD 85 23 3A 1A 20 E6 80 C2 3F 22 2A 11 20 2B 2B
2210  22 11 20 3A 1A 20 B7 C2 EA 21 3A 19 20 47 3A 21
2220  20 AB C2 EA 21 3A 1B 20 47 3A 23 20 AB C2 EA 21
2230  2A 11 20 3A 17 20 77 2B 3A 18 20 77 C3 00 22 F1
2240  C1 D1 E1 C9 2A 11 20 46 2B 4E 3A 15 20 2F 5F A0
2250  47 7B A1 4F 3A 17 20 57 3A 15 20 A2 B0 47 3A 1B
2260  20 5F 3A 15 20 A3 B1 77 23 70 C9 2A 11 20 11 5A
2270  00 19 22 13 20 C3 82 22 2A 11 20 11 A6 FF 19 22
2280  13 20 3A 15 20 32 16 20 CD B9 22 2A 25 20 EB 21
2290  13 20 7E 12 23 13 7E 12 13 3A 16 20 12 13 3A 0F
22A0  20 07 07 6F 3A 10 20 B5 12 2A 25 20 22 29 20 CD
22B0  0D 23 2A 29 20 22 25 20 C9 E5 D5 C5 F5 3A 1A 20
22C0  F5 3A 1D 20 F5 CD 95 23 3A 16 20 E6 80 C2 E4 22
22D0  CD 69 23 3A 1D 20 B7 C2 00 23 3A 16 20 07 32 16
22E0  20 C3 CB 22 CD 9C 23 3A 1A 20 4F E6 01 C2 00 23
22F0  2A 13 20 23 23 22 13 20 79 B7 CA E4 22 C3 DA 22
2300  F1 32 1D 20 F1 32 1A 20 F1 C1 D1 E1 C9 E5 D5 2A
2310  29 20 23 23 23 23 22 29 20 EB 21 2B 20 7D 2F 6F
2320  7C 2F 67 23 19 7C B7 CA 30 23 21 2B 20 22 29 20
2330  D1 E1 C9 E5 D5 C5 F5 3A 15 20 47 3A 19 20 A0 3E
2340  00 CA 45 23 2F 32 1C 20 3A 1B 20 A0 3E 00 CA 52
2350  23 2F 32 1E 20 AF 7B 1F 47 3A 1A 20 A0 3E 00 CA
2360  63 23 2F 32 1D 20 C3 80 23 E5 D5 C5 F5 AF 3A 16
2370  20 17 47 3A 1A 20 A0 3E 00 CA 7D 23 2F 32 1D 20
2380  F1 C1 D1 E1 C9 E5 2A 11 20 2B 2B C3 AB 23 E5 2A
2390  11 20 C3 AB 23 E5 2A 13 20 C3 A2 23 E5 2A 13 20
23A0  23 23 D5 C5 F5 C3 C3 23 D5 C5 F5 11 5A 00 19 CD
23B0  CE 23 32 19 20 11 4C FF 19 CD CE 23 32 1B 20 11
23C0  5A 00 19 CD CE 23 32 1A 20 F1 C1 D1 E1 C9 E5 C5
23D0  56 2B 5E 3A 10 20 CD E4 23 6F 3A 0F 20 CD E4 23
23E0  B5 C1 E1 C9 D5 47 E6 02 C2 EE 23 7A 2F 57 7B E6
23F0  01 C2 F7 23 7B 2F 5F 7A A3 D1 C9 E5 D5 C5 F5 3A
2400  0E 20 6F 26 FF 23 29 4D 44 29 29 09 4D 44 29 29
2410  29 09 01 EB BF 09 EB 2A 0C 20 7C 0F 7D 1F 1F 1F
2420  E6 3F 87 2F 4F 06 FF 03 EB 09 22 11 20 2A 0C 20
2430  7D E6 07 47 3E 01 0F 05 F2 36 24 32 15 20 F1 C1
2440  D1 E1 C9 3A 0F 20 E6 02 3E 00 CA 4E 24 2F 32 17
2450  20 3A 0F 20 E6 01 3E 00 CA 5C 24 2F 32 1B 20 C9
    
```

```

1      REM ----- IMP INT -----
100  T1$=" * * * S C H A L T A L G E B R A * * *"
195  PRINT CHR$(12);PRINT :PRINT T1$;PRINT :PRINT
200  PRINT "   E R L A E U T E R U N G";PRINT
205  PRINT "   Mit diesem Programm koennen die FUNKTIONSTABELLEN von"
210  PRINT "   SCHWALTBRA-FUNKTIONEN mit den 5 Eingangen"
215  PRINT "   (VARIABLEN) A,B,C,D und E dargestellt werden."
220  PRINT "   Bis zu 10 Funktionsgleichungen  $Z = f(A,B,C,D,E)$ "
225  PRINT "   koennen gleichzeitig mit den in DAI-BASIC vorhandenen"
230  PRINT "   logischen Operatoren unter den STATEMENT-Nr."
235  PRINT "   1000-10000 formuliert und vorbereitet werden.";PRINT
240  PRINT "   EINGABE DER FUNKTIONSGLEICHUNG   BREAK-TASTE - IMP INT"
245  PRINT "   DARST. DER FUNKTIONSTABELLE     SPACE-TASTE"
280  Q=BETC:IF Q=0 GOTO 200
300  PRINT CHR$(12);PRINT T1$;PRINT
310  INPUT "   FUNKTIONS-NR. ";Q:PRINT
320  PRINT
400  I=0:K=0:GOSUB 900
410  FOR A=0 TO 1:FOR B=0 TO 1:FOR C=0 TO 1:FOR D=0 TO 1:FOR E=0 TO 1
420  ON Q GOSUB 1000,2000,3000,4000,5000,6000,7000,8000,9000,10000
425  Z=ABS(Z)
430  CURSOR K+4,CURY:PRINT TAB(5+K);I;TAB(9+K);"I";TAB(11+K);A;B;C;D;E;TAB(22+K);"I ";Z:CURSOR K+3,CURY
440  I=I+1
450  IF I=16 THEN K=27:CURSOR K+3,18:GOSUB 900
460  NEXT:NEXT:NEXT:NEXT:NEXT
463  PRINT :CURSOR 0,CURY
465  ON Q GOTO 801,802,803,804,805,806,807,808,809,810
470  W=BETC:IF W=0 GOTO 470
475  GOTO 300
801  LIST 1000-1995:GOTO 470
802  LIST 2000-2995:GOTO 470
803  LIST 3000-3995:GOTO 470
804  LIST 4000-4995:GOTO 470
805  LIST 5000-5995:GOTO 470
806  LIST 6000-6995:GOTO 470
807  LIST 7000-7995:GOTO 470
808  LIST 8000-8995:GOTO 470
809  LIST 9000-9995:GOTO 470
810  LIST 10000-10995:GOTO 470
900  CURSOR 6+K,CURY:PRINT "0Z I A B C D E I Z"
905  CURSOR 5+K,CURY:PRINT "-----+-----+-----"
910  RETURN
1000  Z=A IAND ( INOT B) IAND C IAND D
1999  RETURN
2000  Z=( INOT A) IAND B IAND ( INOT C) IOR (( INOT A) IAND B IAND D) IOR (A IAND ( INOT B) IAND C IAND ( INOT D))
2999  RETURN
3000  Z= INOT A IAND B IAND (C IOR D) IOR (A IAND ( INOT B) IAND (( INOT C) IOR ( INOT D)))
3999  RETURN
4000  Z=( INOT A IAND B IAND ( INOT C)) IOR ( INOT A IAND B IAND D) IOR (A IAND ( INOT B) IAND C IAND ( INOT D))
4999  RETURN
5000  Z=( INOT A IAND B IAND ( INOT D)) IOR (A IAND ( INOT B) IAND ( INOT C) IAND ( INOT D)) IOR (A IAND ( INOT B) I
AND C IAND D)
5999  RETURN
6000  Z1=( INOT A IAND B IAND ( INOT C) IAND D) IOR ( INOT A IAND B IAND C IAND ( INOT D))
6010  Z2=(A IAND ( INOT B) IAND ( INOT C) IAND D) IOR (A IAND B IAND ( INOT C) IAND ( INOT D))
6020  Z=Z1 IOR Z2
6999  RETURN
7000  Z=( INOT A IAND C IAND D) IOR (A IAND ( INOT B) IAND ( INOT C))
7999  RETURN
8000  Z=( INOT B IAND C IAND D) IOR (A IAND ( INOT B) IAND C) IOR (A IAND B IAND ( INOT C) IAND ( INOT D))
8999  RETURN
9000  REM
9999  RETURN
10000 REM
10999 RETURN

```

DISPLAY STOP

Das folgende kleine Maschinenprogramm bietet die Möglichkeit, den Ablauf D'ISPLAYs in der UT'ILITY zu unterbrechen und wieder fortzusetzen.

Gestartet wird mit "V6" - es erscheint die Adresse des Vektors 6 (D578)
Eingabe von "300" (= Ändern der Adresse des Vektors 6).

Gestoppt wird nun mit der "eckigen Klammer" (neben CHAR DEL), fortgesetzt wird mit SPACE.

Bevor man ins BASIC zurückkehrt, sollte der Interruptvektor mit Z3 wieder initialisiert werden (= beenden des Programms)

DISPLAY STOP SPRUNGADRESSEN (300-31D :G V6 - 300)

+J 311:317 +C 314:D6DA +J 31C:D578

DIESER PROGRAMMBLOCK BESITZT : 3 SPRUENGE

DIE LAENGE DES PROGRAMMBLOCKS : 29 BYTES

DISPLAY STOP DISASSEMBLER AUSDRUCK

DISPLAY STOP

SEITE 1

300	F3	DI			
301	F5	PUSH	PSW		
302	C5	PUSH	B		U
303	D5	PUSH	D		U
304	E5	PUSH	H		
305	1101FF	LXI	D:FF01		
308	2107FF	LXI	H:FF07		I
30B	3608	MVI	M:8		6
30D	1A	LDAX	D		
30E	17	RAL			
30F	FE40	CPI	:40		S
311	C21703	JNZ	:317		R
314	000A06	CALL	:D6DA		J z v
317	E1	POP	H		
318	D1	POP	D		9
319	C1	POP	B		Q
31A	F1	POP	PSW		
31B	FB	EI			
31C	C378D5	JMP	:D578		S x u

ENDE DES AUSDRUCKS
END PROGRAM

HEX DUMP

0300 31D

0300 F3 F5 C5 D5 E5 11 01 FF 21 07 FF 36 08 1A 17 FE
0310 40 C2 17 03 0D 0A 06 E1 D1 C1 F1 FB 03 78 D5

NP 36

```

10 MODE 0:PRINT CHR$(12):PRINT
20 PRINT " I N T E G R A T I O N (SIMPSON'SCHE NAEHERUNG)":PRINT
30 PRINT " Mit diesem PROGRAMM koennen Sie jedes BESTIMMTE INTEGRAL"
32 PRINT " annaehern genau berechnen."
34 PRINT " Nach Drucken der BREAK-Taste und EINGABE der FUNKTION"
36 PRINT " (in BASIC) unter der STATEMENT-Nr. 11000 sowie EINTASTEN"
38 PRINT " des BEFEHLS 'RUN' werden Sie aufgefordert weiter DATEN"
40 PRINT " einzugeben."
42 PRINT " Das sind ANFANGS- und ENDWERT von X sowie die Anzahl der"
44 PRINT " SUBINTERVALLE, die fuer die Genauigkeit des ERGEBNISSES"
46 PRINT " verantwortlich ist."
48 PRINT " Nach Ausgabe des ERGEBNISSES koennen SIE durch Druucken"
50 PRINT " der SPACE-Taste die GRAFISCHE DARSTELLUNG der FUNKTIONS-"
52 PRINT " KURVE und der FLAECHE darunter (INTEGRAL) erhalten."
54 PRINT " NEGATIVE FLAECHEEN weden POSITIV aufgefasst, sodass alle"
56 PRINT " FLAECHEENTEILE addiert werden.":PRINT
60 PRINT " WEITER GEHT'S mit SPACE-Taste, wenn Sie die FUNKTION"
62 PRINT " bereits eingegeben haben."
70 Q=GETC:IF Q=0.0 GOTO 70
500 MODE 0:PRINT CHR$(12):PRINT
510 PRINT " I N T E G R A T I O N (SIMPSON'SCHE NAEHERUNG)":PRINT
520 PRINT " FUNKTION":LIST 11000:CURSOR 0,CURY+1:PRINT " ":PRINT
530 PRINT " EINGABE":INPUT " XA = ";XA:INPUT " XE = ";XE:PRINT
540 PRINT " ANZAHL DER SUBINTERVALLE N = ";:INPUT N:PRINT
550 H=(XE-XA)/N:S=0.0
560 FOR I=1.0 TO N-1.0 STEP 2.0:X=I*H+XA:GOSUB 11000:S=S+4.0*ABS(Y):NEXT
570 FOR I=2.0 TO N-2.0 STEP 2.0:X=I*H+XA:GOSUB 11000:S=S+2.0*ABS(Y):NEXT
580 X=XA:GOSUB 11000:S=S+ABS(Y)
590 X=XE:GOSUB 11000:S=S+ABS(Y)
600 S=INT(S*H/3.0*100.0+0.5)/100.0
605 PRINT
610 PRINT " INTEGRAL ZWISCHEN XA = ";XA;" UND XE = ";XE
620 PRINT " BETRAEGT S = ";S
630 Q=GETC:IF Q=0.0 GOTO 630
640 GOTO 60000
11000 Y=SQR(25.0-X*X)
11005 RETURN
60000 REN
60130 MODE 0:PRINT CHR$(12)
60150 COLOR 12 0 10 0:MODE 5A:XD=XE-XA:PX=XMAX/(XD*1.2):Y0=1E-10:YU=1E10
60152 IF XA>0.0 THEN PX=XMAX/(XE*1.2)
60155 FOR X=XA TO XE STEP 0.1
60160 GOSUB 11000
60163 IF Y>100 OR Y<(-100) THEN Y=0.0
60165 IF Y>Y0 THEN Y0=Y
60167 IF Y<YU THEN YU=Y
60168 NEXT
60170 IF Y0>0.0 AND YU<0.0 THEN Y0=Y0-YU:PY=INT(215.0/(Y0*1.2)):Y0=10.0+ABS(YU)*PY:GOTO 60200
60180 IF Y0<0.0 AND YU<0.0 THEN Y0=ABS(YU):PY=INT(215.0/(Y0*1.2)):Y0=190.0:GOTO 60200
60190 IF Y0>0.0 AND YU>0.0 THEN Y0=ABS(Y0):PY=INT(215.0/(Y0*1.2)):Y0=10.0
60200 DRAW 5,Y0 XMAX-5,Y0 0
60205 X0=ABS(XA)*PX+20.0:IF XA>0.0 THEN X0=20.0
60210 DRAW X0,5 X0,210 10
60450 FOR X=XA TO XE STEP 1.0/PX
60455 GOSUB 11000
60460 XX=X*PX+X0
60465 IF XX<0.0 OR XX>XMAX THEN XX=X0
60470 YY=Y*PY+Y0
60475 IF YY<0.0 OR YY>YMAX THEN YY=Y0
60477 DRAW XX,Y0 XX,YY 10
60480 DOT XX,YY 0:NEXT
60490 PRINT " I N T E G R A T I O N DER FUNKTION"
60500 LIST 11000:CURSOR 0,CURY+1:PRINT " "
60502 PRINT " MIT ";N;" SUBINTERVALLE"
60505 PRINT " VON XA =";XA;" BIS XE =";XE;" IST S =";S;
60510 Q=GETC:IF Q=0.0 GOTO 60510
60520 GOTO 500

```

```

100 CLEAR 1000:PRINT CHR$(12):PRINT
110 PRINT " KUBISCHE GLEICHUNG - GLEICHUNG 3.GRADES":PRINT
120 PRINT " Y = A*X^3 + B*X^2 + C*X + D":PRINT
130 INPUT " EINGABE A = ";A:PRINT
140 INPUT " EINGABE B = ";B:PRINT
150 INPUT " EINGABE C = ";C:PRINT
160 INPUT " EINGABE D = ";D:PRINT :PRINT
170 PRINT " LOESUNGEN FUER DIE GLEICHUNG":PRINT
180 PRINT " Y = ";A;"*X^3";:IF SGN(B)<0.0 THEN PRINT " -";:GOTO 184
4
182 PRINT " +";
184 PRINT ABS(B);"*X^2";:IF SGN(C)<0.0 THEN PRINT " -";:GOTO 188
186 PRINT " +";
188 PRINT ABS(C);"*X";:IF SGN(D)<0.0 THEN PRINT " -";:GOTO 192
190 PRINT " +";
192 PRINT ABS(D);" = 0"
200 R=B/A:S=C/A:T=D/A:P=S-R*R/3.0:Q=2.0*R*R*R/27.0-R*S/3.0+T
210 RAD=Q*Q/4.0+P*P*P/27.0
220 IF RAD<0.0 GOTO 500
300 H=SQR(RAD)
310 U1=Q/(-2.0)+H:V1=SGN(U1):U1=ABS(U1)^(1.0/3.0)*V1
320 V1=Q/(-2.0)-H:V1=SGN(V1):V1=ABS(V1)^(1.0/3.0)*V1
330 X1=U1+V1-R/3.0
340 XR=(U1+V1)/(-2.0)-R/3.0:XI=(U1-V1)/2.0*SQR(3.0)
350 IF RAD=0.0 GOTO 450
360 PRINT :PRINT " X1 = ";X1
370 PRINT " X2 = ";XR;" + j";XI
380 PRINT " X3 = ";XR;" - j";XI
390 GOTO 600
450 PRINT :PRINT " X1 = ";X1
460 PRINT " X2 = ";X1
470 GOTO 600
500 AR=SQR(P*P*P/(-27.0))
510 PH=ACOS(Q/(-2.0)/AR)
515 VR=SGN(AR):RR=ABS(AR)^(1.0/3.0)*VR*2.0
520 X1=RR*COS(PH/3.0)-R/3.0
530 X2=RR*COS(PH/3.0+2.0*PI/3.0)-R/3.0
540 X3=RR*COS(PH/3.0+4.0*PI/3.0)-R/3.0
550 PRINT :PRINT " X1 = ";X1
560 PRINT " X2 = ";X2
570 PRINT " X3 = ";X3
600 PRINT :PRINT :PRINT :PRINT " WEITER GEHT'S MIT TASTE 'SPACE'"
610 W=GETC:IF W=0.0 GOTO 610
620 GOTO 100

```

```

=====
100 MODE 0:PRINT CHR$(12):PRINT TAB(15);"H E R Z K U R V E":PRINT :PRINT
:WAIT TIME 50:PRINT :PRINT
110 PRINT " MATHEM. FORMEL : Y = X^(2/3)+(A^2-X^2)^(1/2)":PRINT TAB(13)
);"UND Y = X^(2/3)-(A^2-X^2)^(1/2)":PRINT :PRINT
120 INPUT " EINGABE DER GROESSE 'A': ";A
130 MODE 6:IF A>YMAX/2.4 THEN 200
140 MODE 6:COLORG 10 0 15 5:MX=XMAX/2.0:MY=YMAX/2.0
150 FOR X=-A TO A:IF X>A THEN X=A
160 D=SQR(A*A-X*X):W=ABS(X)^(9.0/10.0):DOT X+MX,W+D+MY 0:DOT X+MX,W-D+
MY 0
170 NEXT
180 Q=GETC:IF Q=0.0 GOTO 180
190 GOTO 100
200 MODE 0:PRINT " 'A' WURDE ZU GROSS GEWAHLT.BITTE GEBEN SIE 'A' NE
U EIN":GOTO 120

```



```

10 CLEAR 5000:GOSUB 5000
20 A1$=" 1 - GAUSS'SCHER ALGORITHMUS - DETERMINANTE"
22 A2$=" 2 - MATRIZEN-ADDITION"
24 A3$=" 3 - MATRIZEN-SUBTRAKTION"
26 A4$=" 4 - MATRIZEN-MULTIPLIKATION"
28 A5$=" 5 - INVERSE MATRIX"
30 B$=" BITTE GEBEN SIE EIN"
32 C$=" EINGABE MATRIX A"
34 D$=" EINGABE MATRIX B"
36 E$=" AUSGABE"
38 F$=" KOEFFIZIENTEN-MATRIX"
70 PRINT :PRINT " MATRIZEN-OPERATIONEN":PRINT
80 PRINT A1$:PRINT A2$:PRINT A3$:PRINT A4$:PRINT A5$:PRINT
90 PRINT " BITTE WAELHEN SIE"
110 Q%=GETC:IF Q%=0 GOTO 110
115 Q%=Q%-48
120 IF Q%>5 THEN PRINT " FALSCH EINGABE, BITTE WIEDERHOLEN.":GOTO
110
130 ON Q% GOSUB 200,740,1000,1270,1600
200 REM GAUSS'SCHER ALGORITHMUS
230 GOSUB 5000:PRINT :PRINT A1$:PRINT
250 INPUT " MATRIX-ORDNUNG N = ";N%:PRINT
260 PRINT :N%=N%-1:DIM A(N%,N%+1)
310 FOR I%=0 TO N%:FOR J%=0 TO N%+1
330 IF J%=N%+1 THEN PRINT :PRINT " A(";I%+1;") = ";:INPUT A(I%,
J%):PRINT :GOTO 360
340 PRINT " A(";I%+1;",";J%+1;") = ";:INPUT A(I%,J%):PRINT
360 NEXT:PRINT :NEXT
380 PRINT :GOSUB 6120
600 PRINT " LOESUNGEN":PRINT
620 FOR I%=0 TO N%
630 PRINT " ERGEBNIS X";I%+1;" = ";A(I%,N%+1)/A(I%,I%)
640 NEXT
690 GOSUB 5010:GOTO 10
720 REM MATRIZEN-ADDITION
740 GOSUB 5000:PRINT :PRINT A2$:PRINT :PRINT B$:GOSUB 6100:GOSUB 6110
790 GOSUB 6400
800 PRINT :PRINT C$:PRINT :GOSUB 6000:GOSUB 6050
910 FOR I%=1 TO N%:FOR J%=1 TO M%
930 C(I%,J%)=A(I%,J%)+B(I%,J%)
940 PRINT " C(";I%;",";J%;") = ";C(I%,J%):NEXT:NEXT
970 GOSUB 5010:GOTO 10
1000 REM MATRIZEN-SUBTRAKTION
1020 GOSUB 5000:PRINT :PRINT A3$:PRINT
1030 PRINT B$:GOSUB 6100:GOSUB 6110:GOSUB 6400
1060 PRINT :PRINT C$:PRINT :GOSUB 6000:GOSUB 6050
1080 FOR I%=1 TO N%:FOR J%=1 TO M%:C(I%,J%)=A(I%,J%)-B(I%,J%)
1210 PRINT " C(";I%;",";J%;") = ";C(I%,J%):NEXT:NEXT
1240 GOSUB 5010:GOTO 10
1270 REM MATRIZEN-MULTIPLIKATION
1290 GOSUB 5000:PRINT A4$:PRINT :PRINT B$
1310 PRINT :INPUT " ZEILENZahl MATRIX A = ";L%
1320 PRINT :INPUT " SPALTENZahl MATRIX A = ";M%
1330 PRINT :INPUT " ZEILENZahl MATRIX B = ";M1%
1335 IF M1%<>M% GOTO 4900
1340 PRINT :INPUT " SPALTENZahl MATRIX B = ";N%
1350 DIM A(L%,M%),B(M%,N%),C(L%,N%):PRINT :PRINT :PRINT C$:PRINT
1370 FOR I%=1 TO L%:FOR J%=1 TO M%
1390 PRINT " A(";I%;",";J%;") = ";:INPUT A(I%,J%):PRINT
1400 NEXT:PRINT :NEXT
1410 PRINT :PRINT :PRINT D$:PRINT
1420 FOR I%=1 TO M%:FOR J%=1 TO N%
1430 PRINT " B(";I%;",";J%;") = ";:INPUT B(I%,J%):PRINT
1440 NEXT:PRINT :NEXT
1450 PRINT :PRINT " LOESUNG: MATRIX C":PRINT
1490 FOR I%=1 TO L%:FOR J%=1 TO N%:FOR K%=1 TO M%

```

```

1520 C(I%,J%)=C(I%,J%)+A(I%,K%)*B(K%,J%)
1530 NEXT:PRINT "      C(";I%,";",";J%;" ) = ";C(I%,J%):NEXT:PRINT :NEXT
1550 GOSUB 5010:GOTO 10
1580 REM INVERSE MATRIX
1600 GOSUB 5000:PRINT :PRINT A$;PRINT :PRINT B$;PRINT :INPUT "      N =
      ";N%:N%=N%-1
1660 DIM A(N%,N%+1),B(N%,N%+1):PRINT :PRINT F$:PRINT
1690 FOR I%=0 TO N%:FOR J%=0 TO N%
1710 PRINT "      B(";I%+1,";",";J%+1;" ) = ";:INPUT B(I%,J%):PRINT :NEXT:P
PRINT :PRINT :NEXT:PRINT
1740 FOR I%=0 TO N%:FOR J%=0 TO N%:A(I%,J%)=B(I%,J%):NEXT:NEXT
1790 J%=N%+1
1810 FOR I%=0 TO N%:A(I%,J%)=0:NEXT
1850 A(R%,J%)=1:R%=R%+1:GOSUB 6120
1900 FOR I%=0 TO N%
1910 PRINT "      A(";I%+1,";",";R%;" ) = ";A(I%,N%+1)/A(I%,I%)
1920 NEXT:PRINT
1940 IF R%<=N% THEN GOSUB 5010:GOTO 1740
1950 END
4900 PRINT :PRINT "      F E H L E R ! - FORDERUNG:"
4910 PRINT "      SPALTENZAHL MATRIX A = ZEILENZAHL MATRIX B"
4920 GOSUB 5010:GOTO 10
5000 PRINT CHR$(12):RETURN
5010 Q%=GETC:IF Q%=0 GOTO 5010
5015 RETURN
5020 PRINT "      KEINE EINDEUTIGE LOESUNG MOEGlich":RETURN
6000 FOR I%=1 TO N%:FOR J%=1 TO M%
6020 PRINT "      A(";I%,";",";J%;" ) = ";:INPUT A(I%,J%):PRINT :NEXT:NEXT:
PRINT
6040 PRINT :PRINT D$:PRINT :RETURN
6050 FOR I%=1 TO N%:FOR J%=1 TO M%
6070 PRINT "      B(";I%,";",";J%;" ) = ";:INPUT B(I%,J%):PRINT :NEXT:NEXT:
PRINT
6090 PRINT :PRINT E$:PRINT :RETURN
6100 PRINT :INPUT "      ZEILENZAHL " ;N%:RETURN
6110 PRINT :INPUT "      SPALTENZAHL " ;M%:RETURN
6120 D=1:FOR K%=0 TO N%
6130 IF K%=N% GOTO 6250
6140 U=ABS(A(K%,K%))
6160 FOR I%=K%+1 TO N%
6170 IF ABS(A(I%,K%))<U GOTO 6230
6180 U=ABS(A(I%,K%)):D=-D
6200 FOR J%=K% TO N%+1
6210 Z=A(K%,J%):A(K%,J%)=A(I%,J%):A(I%,J%)=Z
6220 NEXT J%
6230 NEXT I%
6250 IF ABS(A(K%,K%))<=1E-7 GOTO 5020
6270 FOR I%=0 TO N%
6280 IF I%=K% GOTO 6340
6290 T=A(I%,K%)/A(K%,K%)
6310 FOR J%=K%+1 TO N%+1:A(I%,J%)=A(I%,J%)-A(K%,J%)*T
6330 NEXT J%
6340 NEXT I%
6350 NEXT K%
6370 IF R%<>0 THEN PRINT "      TASTE 'SPACE' DRUECKEN !"
6380 FOR I%=0 TO N%:D=D*A(I%,I%):NEXT
6390 PRINT :PRINT "      DETERMINANTE A = ";D:PRINT
6395 RETURN
6400 DIM A(N%,M%),B(N%,M%),C(N%,M%):PRINT :RETURN

```

Das nachstehende Programm ermöglicht eine, rein softwaremäßige Sprachausgabe, die jedoch nur Versuchsqualität hat. Der Speicherbedarf beträgt etwa 1,8 KByte/s.

Immer wenn die Wechselspannung am Eingang des Kassettenrecorders vom positiven in den negativen Bereich wandert (oder umgekehrt), ändert eine Flipflopschaltung ihren Zustand. Über ein Maschinenprogramm wird dieses Flipflop (Bit 7 Adresse :FD00) abgefragt. Trotz einer Abfragengeschwindigkeit von ca. 1500 Bit/s läßt die Sprachqualität der Ausgabe mit dem Rauschgenerator noch viele Wünsche offen. Ich habe aber mit den Soundgeneratoren keine besseren Erfahrungen gemacht. Die Geschwindigkeit beeinträchtigt die Sprachausgabe wenig, so daß Speicher eingespart werden kann. Z.B wird die Sprache verständlicher wenn nur die Aufnahme verringert wird (vor Zeile 26 zehn NOPs).

Zum Speichern von Sprache wird eine besprochene Kassette abgespielt und dabei das Maschinenprogramm mit CALLM :300 aufgerufen. Das Sprechen erfolgt dann auf CALLM :400. Die Ausgabe kann natürlich auch auf den Kassettenrecorder gespeichert werden. Dazu wird statt dem Rauschgenerator der Kassetten Datenausgang (Bit 0 Adresse :FD06) angesprochen. Um Zeit zu sparen schreibt das Programm immer erst Bit 0 des ganzen Speicherbereiches voll, und benutzt dann Bit 2, Bit 3 usw.. Die Sprechdaten werden in dieser Programmfassung zwischen :500 und :A000 abgespeichert.

Falls ihr Verbesserungsvorschläge habt, schreibt sie mir oder veröffentlicht sie doch in der nächsten Ausgabe!

Thorsten Laux
Kurt-Schumacher-Str. 31
2900 Oldenburg
Tel.: 0441/53858

```

0001      xx0300      ORG      :300
0002 0300 F5      PUSH   PSW
0003 0301 C5      PUSH   B
0004 0302 D5      PUSH   D
0005 0303 E5      PUSH   H
0006 0304      ;SPRACHSYNTHESE
0007 0304      ;
0008 0304      ;
0009 0304      ;
0010 0304 0E0A     MVI    C,10
0011 0306      ;AUF SIGNAL WARTEN
0012 0306 3A00FD   SIGNAL LDA    :FD00
0013 0309 E680     ANI    128
0014 030B 0680     MVI    B,128
0015 030D 90      SUB    B
0016 030E C20603   JNZ    SIGNAL
0017 0311 0D      DCR    C
0018 0312 C20603   JNZ    SIGNAL
0019 0315      ;
0020 0315 0E08     MVI    C,8
0021 0317 1601     MVI    D,1
0022 0319 1EFE     MVI    E,254
0023 031B 2605     IN      MVI    H,:5
0024 031D 2E00     MVI    L,0
0025 031F 3A00FD   AUFNAHME LDA   :FD00
0026 0322 90      SUB    B
0027 0323 FA2C03   JM     NEG
0028 0326 7E      MOV    A,M
0029 0327 B2      ORA    D
0030 0328 77      MOV    M,A
0031 0329 C32F03   JMP    LOOP
0032 032C 7E      NEG    MOV    A,M
0033 032D A3      ANA    E
0034 032E 77      MOV    M,A
0035 032F 23      LOOP   INX    H
0036 0330 7C      MOV    A,H
0037 0331 FEAO     CPI    :A0
0038 0333 C21F03   JNZ    AUFNAHME
0039 0336 7B      MOV    A,E
0040 0337 07      RLC
0041 0338 5F      MOV    E,A
0042 0339 7A      MOV    A,D
0043 033A 07      RLC
0044 033B 57      MOV    D,A
0045 033C 0D      DCR    C
0046 033D C21B03   JNZ    IN
0047 0340 E1      POP    H
0048 0341 D1      POP    D
0049 0342 C1      POP    B
0050 0343 F1      POP    PSW
0051 0344 C9      RET
0052 0345      ;

```

```

0053 0345 ;
0054 0345 ;AUSGABE
0055 xx0400 ORG :400
0056 0400 F5 PUSH PSW
0057 0401 C5 PUSH B
0058 0402 D5 PUSH D
0059 0403 E5 PUSH H
0060 0404 0E08 MVI C,8
0061 0406 1601 MVI D,1
0062 0408 2605 INÄ MVI H,:5
0063 040A 2E00 MVI L,0
0064 040C 7E SPRECH MOV A,M
0065 040D A2 ANA D
0066 040E CA1904 JZ NOTSPR
0067 0411 3EFO MVI A,240
0068 0413 3205FD STA :FD05
0069 0416 C32104 JMP NEUIN
0070 0419 3E00 NOTSPR MVI A,0
0071 041B 3205FD STA :FD05
0072 041E C32104 JMP NEUIN
0073 0421 23 NEUIN INX H
0074 0422 7C MOV A,H
0075 0423 FEAO CPI :A0
0076 0425 C20C04 JNZ SPRECH
0077 0428 7A MOV A,D
0078 0429 07 RLC
0079 042A 57 MOV D,A
0080 042B 0D DCR C
0081 042C C20804 JNZ INÄ
0082 042F E1 POP H
0083 0430 D1 POP D
0084 0431 C1 POP B
0085 0432 F1 POP PSW
0086 0433 C9 RET

```

Programm im Hex-Code

```

0002 0300 F5 C5 D5 E5 0E 0A 3A 00 FD E6 80 06 80 90 C2 06
0016 0310 03 0D C2 06 03 0E 08 16 01 1E FE 26 05 2E 00 3A
0025 0320 00 FD 90 FA 2C 03 7E B2 77 C3 2F 03 7E A3 77 23
0036 0330 7C FE A0 C2 1F 03 7B 07 5F 7A 07 57 0D C2 1B 03
0047 0340 E1 D1 C1 F1 C9
0056 0400 F5 C5 D5 E5 0E 08 16 01 26 05 2E 00 7E A2 CA 19
0066 0410 04 3E F0 32 05 FD C3 21 04 3E 00 32 05 FD C3 21
0072 0420 04 23 7C FE A0 C2 0C 04 7A 07 57 0D C2 08 04 E1
0083 0430 D1 C1 F1 C9

```

```

1      REM ..... IMP INT .....
10     REM PROGRAMM ZUM ERSTELLEN VON 128 ZEICHEN FUER FX-80
20     REM AB Y=170 FREI FUER TEXT-TEST-ZEILE PROPORT. !!!!
100    CLEAR 15000
105    MAXL=10:REM 11 BYTE = 1 ZEICHEN NORMAL
106    NR=0:REM INIT F. CHAR-NR
107    X0=10:Y0=10
120    DIM CHARS(127,MAXL+1):REM 128 ZEICHEN a 1+11 BYTE
125    DIM EC(MAXL,7)
130    COLOR 8 0 1 9
135    MODE 6A
150    GRL=10:REM ABSTAND LINIEN MUSS GERADE SEIN!!
160    GR=7:REM GROESSE PUNKTE
180    GOSUB 3600
500    REM MENUE
502    GOSUB 1800:REM LEERES GITTER
505    PRINT CHR$(12);
506    PRINT " 1 - ZEICHEN UEBERNEHMEN          5 - STRING ZEIGEN"
510    PRINT " 2 - NACHSTES Z. (LEER)"
520    PRINT " 3 - ARRAY LADEN"
530    PRINT " 4 - ARRAY ABSPEICHERN";
535    CURSOR 30,0
540    G=GETC:IF G<#31 OR G>#35 THEN 540
550    ON G-#30 GOTO 600,1000,700,800,900
555    STOP
600    REM UEBERNEHMEN
610    PRINT CHR$(12);" U E B E R N E H M E N : "
620    INPUT "VON WELCHER NUMMER (0..127) ";NR:PRINT
630    IF NR<0 OR NR>127 THEN 600
640    GOSUB 4000:REM IN EC() UND ZEICHNEN
650    INPUT "ALS WELCHE NACHHER ABLEGEN (0..127) ";NR:PRINT
660    GOTO 1020
700    REM ARRAY LADEN
710    PRINT CHR$(12);" L A D E N : "
720    INPUT "FILENAME";FN$:PRINT
730    LOADA CHARS FN$
740    FOR NR=0 TO 127:GOSUB 3000:NEXT
750    NR=0
760    GOTO 500
800    REM ABSPEICHERN
810    PRINT CHR$(12);" A B S P E I C H E R N : "
820    INPUT "FILENAME";FN$:PRINT :IF FN$="" THEN 810
830    SAVEA CHARS FN$
840    NR=0
850    GOTO 500
900    REM STRING ZEIGEN
910    PRINT CHR$(12);" S T R I N G   Z E I G E N "
920    INPUT "STRING";S$:PRINT :IF S$="" THEN 900
930    Y1=175:GOSUB 5000
940    GOTO 500
1000   REM ZEICHEN INPUT MIT CURS+RET
1005   DIM EC(MAXL,7):REM 11 BYTE 0,0=LINKS-UNTEN
1010   GOSUB 1800:GOSUB 3000
1020   PRINT CHR$(12);NR
1021   PRINT "  CURSOR          CHDEL = LOESCHEN      SPACE = SETZEN"
1022   PRINT "  RETURN = ABLEGEN          N = NOTAUSGANG"
1025   X=0:Y=0:CC=23:GOSUB 1700:REM CURSOR SETZEN
1030   G=GETC:IF G<>78 AND G<>13 AND G<>32 AND G<>8 AND (G<16 OR G>19) T
HEN 1030
1035   IF G=13 THEN CC=20:GOSUB 1700:GOSUB 2000:NR=(NR+1) MOD 128:GOTO 5
00
1040   IF G=32 THEN WERT=1:GOSUB 1600:GOTO 1120
1050   IF G=8 THEN WERT=0:GOSUB 1600:GOTO 1120

```

```

1055 IF G=78 THEN 500
1060 CC=20:GOSUB 1700:ON G-15 GOTO 1070,1080,1090,1100
1070 IF Y=7 THEN 1110:Y=Y+1:GOTO 1110
1080 IF Y=0 THEN 1110:Y=Y-1:GOTO 1110
1090 IF X=0 THEN 1110:X=X-1:GOTO 1110
1100 IF X=MAXL THEN 1110:X=X+1:GOTO 1110
1110 CC=23:GOSUB 1700
1120 GOTO 1030
1600 REM TEST , OB SETZEN OD. LOESCHEN MOEGLICH
1610 OK=1:IF X>0 THEN GOSUB 1650
1620 IF X<MAXL THEN GOSUB 1660
1630 IF OK=1 THEN EC(X,Y)=WERT:COL=20+WERT:GOSUB 1900
1640 RETURN
1650 IF EC(X-1,Y)=1 THEN OK=0
1655 RETURN
1660 IF EC(X+1,Y)=1 THEN OK=0
1670 RETURN
1700 REM CURSOR SETZEN
1705 YY=Y0+GRL*Y+GRL/2:XX=X0+(GRL*X+GRL)/2
1710 DRAW X0-8,YY X0-2,YY CC
1720 DRAW X0+6*GRL+2,YY X0+6*GRL+8,YY CC
1730 DRAW XX,Y0-8 XX,Y0-2 CC
1740 DRAW XX,Y0+8*GRL+2 XX,Y0+8*GRL+8 CC
1750 RETURN
1800 REM ZEICHEN LEER:-
1810 FILL 0,0 X0+6*GRL+10,Y0+8*GRL+10 20
1820 FOR I=0 TO 6:DRAW X0+I*GRL,Y0 X0+I*GRL,Y0+8*GRL 23:NEXT
1850 FOR I=0 TO 8:DRAW X0,Y0+I*GRL X0+6*GRL,Y0+I*GRL 23:NEXT
1880 RETURN
1899 REM
1900 REM EINEN PUNKT ZEICHNEN OD LOESCHEN
1910 REM EING: X,Y, COL (20,21)
1920 FILL X0+GRL/2*X+2,Y0+GRL*Y+2 X0+GRL/2*X+1+GR,Y0+GRL*Y+1+GR COL
1930 IF X MOD 2=1 AND COL=20 THEN DRAW X0+(X+1)/2*GRL,Y0+Y*GRL X0+(X+1)
) /2*GRL,Y0+(Y+1)*GRL 23
1990 RETURN
1999 REM
2000 REM ZEICHEN IN CHARS PACKEN UND KLEIN ZEICHNEN
2010 REM EINGANG: NR, EC()
2030 INPUT "UNTERLÄNGE <J/N>";UL$:PRINT
2040 IF UL$="J" THEN UL=0:GOTO 2070
2050 IF UL$="N" THEN UL=1:GOTO 2070
2060 GOTO 2030
2070 INPUT "PROPORTIONAL LINKS,RECHTS (0..11) ";LR,RR:PRINT
2080 IF LR>7 OR RR>11 OR RR-LR<4 THEN 2070
2090 FBYTE=(UL SHL 7)+(LR SHL 4)+RR
2100 CHARS(NR,0)=FBYTE:PRINT "FBYTE=";FBYTE
2110 FOR HOR=0 TO 10
2120 BYTE=0
2130 FOR I=0 TO 7
2140 BYTE=BYTE IOR (EC(HOR,I) SHL I)
2150 NEXT I
2160 CHARS(NR,HOR+1)=BYTE
2170 NEXT HOR
2180 FOR HOR=1 TO 11:PRINT " ";CHARS(NR,HOR);:NEXT
2190 GOSUB 3000:REM KLEIN ZEICHNEN
2200 PRINT :PRINT " LEERTASTE!!!":CALLM #D6DA:PRINT CHR$(12);
2210 RETURN
3000 REM CHAR NR AUS CHARS ZEICHNEN
3010 REM BERECHNET POS AUS NR
3020 REM 128 ZEICHEN STARTX=104,STARTY=4
3030 REM EIN ZEICHEN X=12+ 3 FREI:Y=18+ 3 FREI
3040 X1=88+(NR MOD 16)*15:Y1=4+(7-NR/16)*21

```

```

3090 REM LAEUFT IN 3500 REIN!!!
3500 REM CHAR NR AN POS X1,Y1 ZEICHNEN AUS CHARS:COL=22
3505 Y2=Y1+((CHARS(NR,0) SHR 7) IAND 1)*2
3510 FILL X1,Y1 X1+MAXL+1,Y1+2*8+1 20
3520 FOR VER=0 TO 7
3530 FOR HOR=0 TO MAXL
3540 C2=(CHARS(NR,HOR+1) SHR VER) IAND 1:REM (X,0)=FBYTE
3560 IF C2=1 THEN FILL X1+HOR,Y2+2*VER X1+HOR+1,Y2+2*VER+1 22
3570 NEXT
3580 NEXT
3590 RETURN
3600 REM GITTER FUER KLEINE
3610 FOR I=0 TO 16:DRAW 86+I*15,2 86+I*15,2+8*21 23:NEXT
3620 FOR I=0 TO 8:DRAW 86,2+I*21 86+16*15,2+I*21 23:NEXT
3630 RETURN
4000 REM CHAR AUS CHARS() IN EC() UND ZEICHNEN
4005 REM EING: NR
4010 COL=21:FOR HOR=0 TO MAXL
4020 FOR VER=0 TO 7
4030 POINT=(CHARS(NR,HOR+1) SHR VER) IAND 1
4040 EC(HOR,VER)=POINT
4050 IF POINT=1 THEN X=HOR:Y=VER:GOSUB 1900
4060 NEXT
4070 NEXT
4080 RETURN
5000 REM ZEICHNE STRING S$ OBEN PROPORT.
5010 REM EING : S$, Y1
5020 X1=20:FLAG=0
5030 FILL X1,Y1 XMAX,Y1+20 20
5040 FOR I=0 TO LEN(S$)-1
5050 HNR=ASC(MID$(S$,I,1)):FBYTE=CHARS(HNR,0)
5060 Y2=Y1+((FBYTE SHR 7) IAND 1)*2
5070 LR=(FBYTE SHR 4) IAND 7:RR=FBYTE IAND 15
5075 IF RR=11 THEN RR=10:FLAG=1
5080 FOR J=LR TO RR
5090 FOR VER=0 TO 7
5100 C2=(CHARS(HNR,J+1) SHR VER) IAND 1
5110 IF C2=1 THEN FILL X1,Y2+2*VER X1+1,Y2+2*VER+1 22
5120 NEXT
5130 X1=X1+1:IF X1>334 THEN 5170
5140 NEXT
5150 IF FLAG=1 THEN X1=X1+1:REM 12.PUNKT IMMER 0
5160 NEXT
5170 RETURN

```

Dies ist z.B. die Schriftart "SARMODERN", die einigen Leuten überhaupt nicht gefällt, mir jedoch sehr! 

ABCDEFGHIJKLMN OPQRSTUVWXYZäöüabcde fghijklmnopqrstuvwxyzaöü1234567890
 ♥ ° Ω ← ↑ → ↓ ⌈ ⌋ ⌌ ⌍ ⌎ ⌏ ⌐ ⌑ ⌒ ⌓ ⌔ ⌕ ⌖ ⌗ ⌘ ⌙ ⌚ ⌛ ⌜ ⌝ ⌞ ⌟ ⌠ ⌡ ⌢ ⌣ ⌤ ⌥ ⌦ ⌧ ⌨ 〈 〉 ⌫ ⌬ ⌭ ⌮ ⌯ ⌰ ⌱ ⌲ ⌳ ⌴ ⌵ ⌶ ⌷ ⌸ ⌹ ⌺ ⌻ ⌼ ⌽ ⌾ ⌿ Ⓚ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ ⓐ ⓑ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ ⓚ ⓛ ⓜ ⓝ ⓞ ⓟ ⓠ ⓡ ⓢ ⓣ ⓤ ⓥ ⓦ ⓧ ⓨ ⓩ ⓪ ⓫ ⓬ ⓭ ⓮ ⓯ ⓰ ⓱ ⓲ ⓳ ⓴ ⓵ ⓶ ⓷ ⓸ ⓹ ⓺ ⓻ ⓼ ⓽ ⓾ ⓿ Ⓚ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ ⓐ ⓑ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ ⓚ ⓛ ⓜ ⓝ ⓞ ⓟ ⓠ ⓡ ⓢ ⓣ ⓤ ⓥ ⓦ ⓧ ⓨ ⓩ ⓪ ⓫ ⓬ ⓭ ⓮ ⓯ ⓰ ⓱ ⓲ ⓳ ⓴ ⓵ ⓶ ⓷ ⓸ ⓹ ⓺ ⓻ ⓼ ⓽ ⓾ ⓿

Um die entworfenen Zeichen aus dem Array in den Drucker zu laden:

```

FOR I=32 TO 127
PRINT CHR$(27); "&"; CHR$(0); CHR$(1); CHR$(1);
FOR J=0 TO 11:PRINT CHR$(CHARS(I,J)); :NEXT
NEXT

```

Bernd Preusing

Schreibschrift mit dem EPSON FX 80

Das Programm habe ich auf meinem DAI Personal Computer entwickelt, es dürfte aber ohne große Änderungen auf vielen Personal- bzw. Home-Computern laufen. Durch POKE#131,3 (Zeile 245) wird der Drucker angesprochen, so daß auch normale PRINT-Befehle auf den Drucker ausgegeben werden. Üblicherweise würde man sonst statt PRINT immer LPRINT schreiben, um die darauf folgenden Zeichen auf den Drucker auszugeben.

Das Programm nützt die Möglichkeit aus, daß der FX 80 in einem RAM vom Anwender definierte Zeichen abspeichern kann. Diese Zeichen können auf passende Tasten des Rechners gelegt werden. Jedes dieser Zeichen wird durch 11 Bytes definiert, wobei ein Byte eine Spalte der Matrix des Zeichens festlegt. In einer Spalte wird also ein Punkt gedruckt, wenn im zugehörigen Byte ein Bit gesetzt (d.h.=1) ist.

Das Programm nützt für alle Zeichen die 8 Bit pro Spalte aus. Trotzdem steht für Unterlängen noch eine Punktzeile zur Verfügung: mit Hilfe eines "Attributs" (=Kontrollbyte) kann das Zeichen um eine Punktzeile tiefer gedruckt werden. (Siehe Programmzeile 1465; "DESCX=11". 11 ist Attribut für tiefergestellte Zeichen (DESCENDER), 139 ist Attribut für normalstehende Zeichen.

Die Daten (Zeile 2001 bis 2168) enthalten als erste Zahl den ASCII-Code des Zeichens, das durch das selbstdefinierte Zeichen ersetzt werden soll. Dann kommen die 11 Bytes für die Zeichendefinition.

Zu den einzelnen Abschnitten:

10-260: wird erklärt in den Kommentaren

400-720: die Schriftart (Pica oder Schreibschrift) wird durch die Variable ART% gewählt, ART%=1 ist Pica, ART%=2 ist Schreibschrift. Die Druckart wird durch das Senden von Steuerzeichen gewählt; PRINT EMP\$; bedeutet, daß der Druck verstärkt wird (emphasized). Diese Steuerzeichen werden ab Zeile 3000 definiert. Double strike bedeutet, daß die Zeichen doppelt gedruckt werden, wobei beim zweiten Druck ein Papiervorschub von einem 216tel Zoll erfolgt. Dadurch werden vertikale Abstände zwischen den Punkten gefüllt. Das Steuerzeichen dafür im Programm ist DOU\$;, Aufhebung durch DDOU\$;

Zu beachten ist noch, daß beim Ausdrucken des Programms keine Zeichen in den Drucker eingelesen waren, und die Umlaute des Druckers kann ich über die Tastatur des Rechners nicht alle erreichen.

1325: Das Zeichen-ROM des Druckers wird in das RAM geladen.

1327: Es werden Zeichen aus dem RAM gedruckt

1403-1490: Lädt die Umlaute für Pica bzw. die Schreibschrift

1800: Liest die 12 Daten für jedes Zeichen

1820-1835: Sendet diese Daten mit Angabe des ASCII-Codes zum Drucker. Anschließend stehen diese Zeichen zur Verfügung, solange der Drucker eingeschaltet ist

(Konrad Plachetta)

```

1      REM *****DRUCKDEMO SCHREIBSCHRIFT FX 80 *****
2      REM ***          C      KONRAD PLACHETTA 6/83          ***
3      REM *****
10     CLEAR '150
15     DIM D(11)
245    POKE #131,3
250    GOSUB 1150:REM Einstellen linker Rand
260    GOSUB 3000:REM Einlesen Steuerzeichen-Variable
400    ART=1:GOSUB 1325:REM Drucken in Pica-Schrift
405    REM -----Druckprogramm-----
407    PRINT EMP$;
410    PRINT "Dies ist die n^chterne, normale Schriftart Pica des FX 80.
      "
420    PRINT :PRINT :PRINT "Und so sieht die selbstdefinierte Schrift au
      s: "
500    PRINT :ART=2:GOSUB 1325:REM DRUCKEN IN 'SCHREIBSCHRIFT'
505    PRINT PROP$;:REM Die Buchstaben werden direkt aneinander gedruckt

510    GOSUB 600:PRINT
515    PRINT DPROP$;:PRINT DOU$;:PRINT "Double strike : "
517    GOSUB 620
520    PRINT EN$;DOU$;:PRINT :PRINT "ENLARGED :":PRINT EN$;:GOSUB 700
522    PRINT EN$;DOU$;:GOSUB 710
525    PRINT EN$;DOU$;:GOSUB 720
529    PRINT DPROP$;:PRINT DEMP$;
540    PRINT COND$;DOU$;:PRINT :PRINT "CONDENSED :":GOSUB 620
542    PRINT DCOND$;DDOU$;
550    PRINT DEMP$;:PRINT EL$;:PRINT :PRINT "ELITE :":GOSUB 620
555    PRINT DOU$;:PRINT "Elite double strike :":GOSUB 620
560    PRINT DEL$;
570    GOTO 570
600    PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ 'B"
610    PRINT "abcdefghijklmnopqrstuvwxyz /A^#"
620    PRINT "Die n^chterne Computerschrift kann ersetzt werden "
640    PRINT "durch schreibschriftähnliche Buchstaben !":RETURN
700    PRINT "Die n^chterne Computer-":RETURN
710    PRINT "schrift kann ersetzt werden durch":RETURN
720    PRINT "schreibschriftähnliche Buchstaben !":RETURN
1149   REM -----EINSTELLEN LINKER RAND-----
1150   PRINT CHR$(27);"1";CHR$(5);
1160   RETURN
1300   REM -----DEFINITION DES ZEICHENSATZES-----
1325   POKE #131,3:PRINT CHR$(27);": ";CHR$(0);CHR$(0);CHR$(0);
1327   PRINT CHR$(27);"%";CHR$(1);CHR$(0);
1340   ON ART GOSUB 1400,1421
1345   RETURN
1399   REM ----PROGR. UMLAUTE PICA-----
1400   REM
1403   FOR ZEICHEN=1 TO 8
1405   GOSUB 1800
1407   DESC=139
1410   GOSUB 1820
1411   NEXT
1412   RESTORE
1415   RETURN
1420   REM -----PROGR. SCHREIBSCHRIFT-----
1421   REM
1425   FOR ZEICHEN=1 TO 8
1430   GOSUB 1800
1435   NEXT
1440   DESC=139
1445   FOR ZEICHEN=1 TO 50
1450   GOSUB 1800
1455   GOSUB 1820
1460   NEXT
1465   DESC=11

```

```

1470 FOR ZEICHEN=1 TO 9
1475 GOSUB 1800
1480 GOSUB 1820
1485 NEXT
1487 RESTORE
1490 RETURN
1799 REM -----READ Routine-----
1800 READ AS,D(1),D(2),D(3),D(4),D(5),D(6),D(7),D(8),D(9),D(10),D(11)
1805 RETURN
1819 REM -----ZEICHEN EINPROGRAMMIEREN-----
1820 PRINT CHR$(27); "&"; CHR$(0); CHR$(AS); CHR$(AS);
1825 PRINT CHR$(DESC);
1830 FOR I=1 TO 11:PRINT CHR$(D(I));:NEXT
1840 RETURN

```

```

1999 REM -----DATEN FUER ZEICHEN-----
2001 DATA 91,4,10,160,10,32,10,160,28,2,0,0
2003 DATA 94,0,60,128,2,0,2,128,60,2,0,0
2005 DATA 126,60,130,0,2,0,2,0,130,60,0,0
2007 DATA 47,0,28,162,0,34,0,162,28,0,0,0
2009 DATA 39,28,162,0,34,0,34,0,162,28,0,0
2011 DATA 35,126,128,0,128,18,128,18,108,0,0,0
2013 DATA 93,6,136,20,32,68,32,20,136,6,0,0
2015 DATA 42,0,0,0,48,0,48,0,0,0,0,0

```

```

2018 REM -----Schreibschrift-Zeichen-----
2020 DATA 65,1,2,4,8,16,40,64,136,64,62,1
2022 DATA 66,1,129,126,1,128,17,128,17,136,102,0
2024 DATA 67,16,0,28,34,81,128,17,128,97,0,2
2026 DATA 68,129,0,255,0,129,0,129,0,66,36,24
2028 DATA 69,0,0,108,146,1,144,1,128,65,0,2
2030 DATA 70,0,64,128,1,144,1,254,0,145,0,130
2032 DATA 71,28,34,65,0,129,0,137,0,137,66,12
2034 DATA 72,1,254,0,8,0,8,0,8,0,254,1
2036 DATA 73,0,64,130,1,128,1,128,1,254,0,2
2040 DATA 75,0,129,0,255,0,24,32,70,129,0,129
2042 DATA 76,0,8,1,126,129,1,161,64,1,0,2
2044 DATA 77,1,254,0,64,32,16,32,64,0,254,1
2046 DATA 78,1,254,0,64,32,16,8,4,0,127,128
2048 DATA 79,28,34,65,128,1,128,1,128,66,36,152
2050 DATA 80,1,128,127,0,146,0,146,0,146,98,2
2052 DATA 81,28,34,65,128,1,128,9,132,66,38,154
2054 DATA 82,1,128,127,0,152,4,146,1,144,97,2
2056 DATA 83,0,2,1,96,145,0,145,0,145,74,4
2058 DATA 84,0,64,129,0,129,126,128,2,128,2,128
2060 DATA 85,128,0,254,1,0,1,0,1,254,1,2
2062 DATA 86,128,28,226,1,0,1,2,4,248,0,128
2064 DATA 87,128,30,225,0,1,14,1,0,129,126,128
2066 DATA 88,129,64,34,20,8,16,40,68,2,129,0
2070 DATA 90,64,1,130,21,136,17,160,81,128,1,2
2072 DATA 93,1,2,4,8,148,32,68,32,28,130,1
2074 DATA 126,64,62,129,0,1,0,129,0,127,0,2
2076 DATA 39,12,146,33,64,1,64,1,64,34,156,0

```

```

2080 REM -----KLEINBUCHSTABEN-----
2082 DATA 97,2,0,6,9,16,1,16,1,14,17,2
2084 DATA 98,4,0,126,129,16,129,96,1,14,0,4
2086 DATA 99,2,0,6,9,16,1,16,1,8,1,2
2088 DATA 100,2,12,2,17,0,17,8,1,254,1,2
2090 DATA 101,2,0,6,9,20,1,20,9,0,1,2
2094 DATA 104,2,4,0,127,128,16,168,64,15,0,1
2096 DATA 105,2,4,8,0,92,2,1,0,1,0,2
2100 DATA 107,2,4,0,127,128,20,162,65,8,1,2
2102 DATA 108,2,4,0,126,129,16,161,64,1,0,2
2104 DATA 109,8,16,15,0,16,0,15,0,16,0,15
2112 DATA 110,4,8,0,31,0,16,0,16,15,0,2
2114 DATA 111,4,0,14,1,16,1,16,1,18,12,2
2116 DATA 114,4,8,16,15,0,8,0,16,8,0,4
2118 DATA 115,4,8,16,33,16,1,8,5,2,0,2

```

	norm. ASCII	Taste
"	123	[
ä	125	↑
ü	93	~
ö	124	/
ß	92	/
Å	126	#
Ä	91]
•	42	(Mal-Zeichen)

Ä	91
ü	93
ö	92

```

2120 DATA 116,2,36,0,254,1,32,1,0,1,0,2
2122 DATA 117,4,8,0,30,1,0,1,0,31,0,2
2124 DATA 118,4,8,0,30,1,0,1,0,30,0,16
2126 DATA 119,2,0,30,1,0,1,30,1,0,1,30
2128 DATA 120,1,16,10,0,4,2,8,17,8,0,4
2130 DATA 122,4,8,16,0,21,2,21,8,21,0,2
2132 DATA 91,2,0,6,9,80,1,16,1,94,1,2
2134 DATA 94,4,8,0,30,65,0,1,0,95,0,2
2136 DATA 47,2,4,10,81,0,17,0,81,10,4,2
2145 REM -----DESCENDER-KLEINBUCHSTABEN-----
2147 DATA 112,4,8,16,32,31,0,36,0,36,0,24
2149 DATA 113,4,8,16,40,4,32,4,32,31,0,4
2151 DATA 121,4,8,48,8,4,0,5,0,63,0,4
2153 DATA 103,4,8,16,8,37,0,37,0,63,0,4
2155 DATA 102,4,0,10,0,127,128,18,160,66,0,4
2157 DATA 106,4,8,1,16,161,30,0,2,0,2,4
2159 DATA 35,4,8,0,127,128,0,146,0,108,0,0
2160 REM -----DESCENDER GROSSBUCHSTABEN-----
2166 DATA 74,0,64,128,3,132,1,136,1,254,0,8
2168 DATA 89,0,128,0,240,8,1,8,1,254,0,8
2999 REM -----STEUERZEICHEN VARIABLE-----
3000 EN$=CHR$(14):REM ENLARGED
3010 DEN$=CHR$(20):REM DEL ENL
3040 COND$=CHR$(27)+"F"+CHR$(15):REM CONDENSED
3050 DCOND$=CHR$(18)+CHR$(27)+"E":REM DEL CONDENSED
3080 PROP$=CHR$(27)+"p"+CHR$(1)
3090 DPROP$=CHR$(27)+"p"+CHR$(0)
3100 EL$=CHR$(27)+"M":REM DEL PROP, ELITE
3110 DEL$=CHR$(27)+"!"+CHR$(8):REM DEL ELITE, SET EMP
3120 EMP$=CHR$(27)+"E"
3130 DEMP$=CHR$(27)+"F"
3140 DOU$=CHR$(27)+"G"
3150 DDOU$=CHR$(27)+"H"
3200 RETURN

```

"ä 123
 "ü 125
 "ö 124

β 126

Und so sieht die selbstdefinierte Schrift aus:

AB EDEIGHIJKL MNOPQRSTU VWXYZ AÖÜ
 abcdefghijklmnopqrstuvwxyzöüß

Die nüchterne Computerschrift kann ersetzt werden durch schreibschriftähnliche Buchstaben!

Anm. der Red.: Der Autor hat hier die Codes der Umlaute verändert, um sie von der Tastatur aus zu erreichen. Für die richtigen Codes bitte die angestrichelten Daten entsprechend ändern! Die Daten für die normale Schrift sind dann natürlich überflüssig!
 (Die erste Zahl jeder Zeile ist jeweils der Zeichen-Code.)

Vor dem Eintippen JME JMI eingeben !!!!!!!

```

10 REM ---- dcrlist X ----
20 REM ////////// memory management
30 IF PEEK(7)=0 THEN CLEAR 10512:POKE #29C,PEEK(#29C)+2:POKE #29E,PE
EK(#29E)-2:POKE 7,#FF
40 CLEAR 10000:START=PEEK(#29B)+(PEEK(#29C)-2)*256
50 POKE #131,1:POKE #CB,0:POKE #C9,0:POKE #62,#5D:POKE #63,#EB
60 ENVELOPE 0 0:ENVELOPE 1 0
70 REM ////////// initialisation parameters
80 MX082=0:REM (0=80 char/line )(1=96 char/line)
90 WIDTH=1:REM (0=13*10 cm )(1=15*10 cm)
100 TITLE=1:REM (0=one title/tape)(1=Aside+Bsidedtitle)
110 FRAMECOLOR=9:TEXTCOLOR=14:WINDOWCOLOR=8:WINDOWTEXTC=15
120 REM ////////// screen ////////// onetime initialisation
130 MODE 0:PRINT CHR$(12):COLORT FRAMECOLOR TEXTCOLOR 0 15
140 CURSOR 15,23:PRINT "* * * D C R L I S T X * * *"
150 POKE #8A,#45:POKE #8B,#BC:POKE #BCCA,#C0+WINDOWCOLOR:POKE #BC44,#
D0+WINDOWTEXTC
160 GOSUB 1220
170 INPUT "Date ";DATE$:PRINT :DATE$=SPC(HCARDLEN-LEN(DATE
$)-8)+"update "+DATE$+" "
180 REM ////////// (re)start
190 DINL=3*FRML:DIM INLAY$(DINL):DIM CARD$(50,1):CARD$(47,0)=DATE$:CA
RD$(47,1)=DATE$
200 CURSOR 0,21:FOR A=0 TO 4:PRINT SPC(59):NEXT
210 CURSOR 0,21:PRINT "Driveno 0-3 ?";
220 DCR=GETC:IF DCR<48 OR DCR>51 THEN 220:PRINT CHR$(DCR)
230 DCR=DCR-48:POKE #11F,DCR:CALLM REW
240 INPUT "Serialno ";NO$:PRINT
250 IF LEN(NO$)>4 THEN 240
252 IF LEN(NO$)=1 THEN NO$="0"+NO$
260 PRINT "Title ";TITLL;" char";:INPUT INLAY$(0,0):PRINT
270 CARD$(0,0,0)=INLAY$(0,0)
280 IF LEN(INLAY$(0,0))>TITLL THEN 260
290 IF TITLE=0,0 THEN CARD$(0,0,1,0)=INLAY$(0,0):GOTO 320
300 PRINT "Title B ";TITLL;" char";:INPUT CARD$(0,0,1,0):PRINT
310 IF LEN(CARD$(0,0,1,0))=0,0 THEN CARD$(0,0,1,0)=CARD$(0,0,0,0)
320 REM ////////// headers
330 CRDCOUNT=2:INLCOUNT=3:NO=1:NR=1:SIDE=0
340 IF LEN(INLAY$(0,0))>INLLEN THEN INLAY$(0,0)=LEFT$(INLAY$(0,0),INL
LEN)
350 CARD$(1,0,0,0)="DCR "+NO$+" A":CARD$(2,0,0,0)=""
360 CARD$(1,0,1,0)="DCR "+NO$+" B":CARD$(2,0,1,0)=""
370 INLAY$(2)=""
371 FOR IL=1 TO INLLEN/2-2
372 INLAY$(2)=INLAY$(2)+"B"
373 NEXT
374 INLAY$(2)=INLAY$(2)+" A "
375 FOR IL=1 TO INLLEN/2-2
376 INLAY$(2)=INLAY$(2)+"B"
377 NEXT
390 CALLM #F000:REM REW
400 REM ////////// get a name
410 CURSOR 0,0:POKE #CB,0:CALLM LOOKNEW:IF PEEK(#CB)=0 THEN 530
420 REM ////////// lastfile side A
430 SIDE=SIDE+1:NR=1:CRDCOUNT=2:IF SIDE>1 THEN 500
440 PRINT "turn cassette,type space"
450 GOSUB 1150:A=GETC:IF A<>32 THEN 450:CALLM #F000:REM REW
460 INLAY$(INLCOUNT)=""
461 FOR IL=1 TO INLLEN/2-2
462 INLAY$(INLCOUNT)=INLAY$(INLCOUNT)+"B"
463 NEXT
464 INLAY$(INLCOUNT)=INLAY$(INLCOUNT)+" B "

```

```

465 FOR IL=1 TO INLLEN/2-2
466 INLAY$(INLCOUNT)=INLAY$(INLCOUNT)+"B"
467 NEXT
480 INLCOUNT=INLCOUNT+1:GOTO 410
490 REM ////////// last file side B
500 PRINT "type space to start printing":CALLM SKIP
510 GOSUB 1150:A=GETC:IF A<>32 THEN 510:GOTO 740
520 REM ////////// get number
530 INLAY$(INLCOUNT)=MID$(STR$(NO),1,LEN(STR$(NO))-3.0)+"-"
540 IF LEN(INLAY$(INLCOUNT))<3.0 THEN INLAY$(INLCOUNT)="0"+INLAY$(INLCOUNT)
550 CARD$(CRDCOUNT,SIDE)=MID$(STR$(NR),1,LEN(STR$(NR))-3.0)+"-"
560 IF LEN(CARD$(CRDCOUNT,SIDE))<3.0 THEN CARD$(CRDCOUNT,SIDE)="0"+CARD$(CRDCOUNT,SIDE)
570 REM ////////// read name
580 H1$="":FOR A=#B3EB TO #B463 STEP 2:IF PEEK(A)=#20 THEN NEXT
590 FOR B=#B463 TO A STEP -2
600 PB=PEEK(B):PB=PB IAND #7F:IF PB<32 OR PB=127 THEN PB=46
610 H1$=H1$+CHR$(PB):NEXT:FILETYPE$=LEFT$(H1$,1)
620 HH$=RIGHT$(H1$,LEN(H1$)-1)
630 HH$=FILETYPE$+"-"+TYP$(ASC(FILETYPE$))+"-"+HH$
640 INLAY$(INLCOUNT)=INLAY$(INLCOUNT)+H1$:CARD$(CRDCOUNT,SIDE)=CARD$(CRDCOUNT,SIDE)+HH$
650 LNGTH=INLLEN:IF INLCOUNT>=2*FRML THEN LNGTH=INLLEN-1
660 IF LEN(INLAY$(INLCOUNT))>LNGTH THEN INLAY$(INLCOUNT+1.0)=SPC(3)+RIGHT$(INLAY$(INLCOUNT),LEN(INLAY$(INLCOUNT))-LNGTH):INLAY$(INLCOUNT)=LEFT$(INLAY$(INLCOUNT),LNGTH):INLCOUNT=INLCOUNT+1:GOTO 650
670 IF LEN(CARD$(CRDCOUNT,SIDE))<HCARDLEN-1.0 THEN 710
680 CARD$(CRDCOUNT+1.0,SIDE)=SPC(9)+RIGHT$(CARD$(CRDCOUNT,SIDE),LEN(CARD$(CRDCOUNT,SIDE))-HCARDLEN+2.0)
690 CARD$(CRDCOUNT,SIDE)=LEFT$(CARD$(CRDCOUNT,SIDE),HCARDLEN-2):CRDCOUNT=CRDCOUNT+1:GOTO 670
700 REM ////////// next file
710 CALLM #F000:REM SKIP1
720 CRDCOUNT=CRDCOUNT+1:INLCOUNT=INLCOUNT+1:NO=NO+1:NR=NR+1
730 IF INLCOUNT<93 THEN 410
740 REM ////////// buffer management
750 POKE #131,1:BASEND=PEEK(#2A4)+1
760 BUFLen=33*LEN(LINE$)+57*LEN(KADER$)
770 IF PEEK(#63)<>#EB THEN 800
780 POKE #A2,0:POKE #A3,BASEND:POKE #A4,0:POKE #A5,BASEND
790 POKE #A6,0:POKE #A7,#B3:GOTO 860
800 INPPTR=PEEK(#A5)*256+PEEK(#A4):REM BUFFER FULL?
810 IF INPPTR+BUFLen>#B100 THEN POKE INPPTR,#FF:POKE #A4,0:POKE #A5,BASEND:GOTO 800
820 PRTPTR=PEEK(#C9)*256+PEEK(#C8)
830 NDPTR=INPPTR+BUFLen
840 IF PRTPTR>INPPTR AND PRTPTR<NDPTR THEN 820
850 REM ////////// print inlay
860 POKE #131,2:PRINT CARSET$;LINSPEC5$;CONDNSON$;SPRSCRON$;LINE$
870 PRINT BACK1$;H$;UNDERLON$;INLAY$(0.0);UNDLOFF$;SPC(INLLEN-LEN(INLAY$(0.0)));
880 PRINT H$;INLAY$(FRML);SPC(INLLEN-LEN(INLAY$(FRML)));
890 PRINT H$;INLAY$(2.0*FRML);SPC(INLLEN-1-LEN(INLAY$(2.0*FRML)));H$
900 FOR A=1 TO FRML-1:IF A MOD 20<>6 THEN 920
910 PRINT H$;" ";NO$;SPC(BACKLEN-LEN(NO$)-1);:GOTO 950
920 IF A MOD 20<>5 THEN 940
930 PRINT BACK$;:GOTO 950
940 PRINT BACK1$;
950 PRINT H$;INLAY$(A);SPC(INLLEN-LEN(INLAY$(A)));
960 PRINT H$;INLAY$(A+FRML);SPC(INLLEN-LEN(INLAY$(A+FRML)));
970 PRINT H$;INLAY$(A+2.0*FRML);SPC(INLLEN-1-LEN(INLAY$(A+2.0*FRML)))

```

```

;H$
980 NEXT
990 REM ////////// photocard
1000 PRINT SUPSCROFF$;LINSPC10$;CONDNSON$;KADER$
1010 FOR SIDE=0 TO 1
1020 PRINT H$;SPC(CARDLEN);H$:PRINT CONDNSOFF$;
1030 PRINT H$;ENLARGON$;CARD$(0,SIDE)
1040 PRINT SPC(CARDLEN*2/7-LEN(CARD$(1,SIDE)));CARD$(1,SIDE)
1050 PRINT ENLARGOFF$;CONDNSON$;
1060 PRINT H$;SPC(CARDLEN);H$
1070 FOR A=2 TO 20:PRINT H$;SPC(2);CARD$(A,SIDE);
1080 PRINT SPC(HCARDLEN-LEN(CARD$(A,SIDE)));CARD$(A+19,SIDE);
1090 PRINT SPC(HCARDLEN-LEN(CARD$(A+19,SIDE)));H$
1100 NEXT:PRINT KADER$:NEXT SIDE
1110 REM ////////// bufferend
1120 POKE PEEK(#A4)+PEEK(#A5)*256,#EF:REM ENDFLAG
1130 IF PEEK(#63)=#EB THEN CALLM INITSPoolER
1140 POKE #131,1:GOTO 190
1150 REM ////////// dingdong
1160 ENVELOPE 0 15,3;3,3;
1170 FOR I=1 TO 2:SOUND 0 1 15 0 FREQ(1046.0):WAIT TIME 5
1180 SOUND 1 1 15 0 FREQ(1175.0):WAIT TIME 5:NEXT
1190 SOUND 2 1 15 0 FREQ(1760.0):WAIT TIME 30
1200 WAIT TIME 50:SOUND OFF :RETURN
1210 REM ////////// parameter setting
1220 GOTO 1330
1230 ADRES=START
1240 READ BYTE:IF BYTE<#100 THEN POKE ADRES,BYTE:ADRES=ADRES+1:GOTO 12
40
1250 IF BYTE=#FFF THEN RETURN
1260 READ LBYTE,HBYTE:POKE ADRES,BYTE IAND #FF
1270 ON BYTE/#100 GOTO 1280,1300
1280 LBYTE=LBYTE+(START IAND #FF):CARRY=LBYTE SHR 8
1290 POKE ADRES+1,LBYTE IAND #FF:HBYTE=HBYTE+(START SHR 8)+CARRY:GOTO
1320
1300 LBYTE=LBYTE+(STARTB IAND #FF):CARRY=LBYTE SHR 8
1310 POKE ADRES+1,LBYTE IAND #FF:HBYTE=HBYTE+(STARTB SHR 8)+CARRY
1320 POKE ADRES+2,HBYTE:ADRES=ADRES+3:GOTO 1240
1330 STARTB=START:GOSUB 1230:INITSPoolER=START+#44:REW=START+#C:SKIP=S
TART+#F
1340 LOOKNEW=START+6:START=#F800:GOSUB 1230
1350 REM ////////// formatting
1360 DIM TYP$(255.0)
1370 READ FBYT$,FTYP$:IF FTYP$="END" THEN 1390
1380 TYP$(ASC(FBYT$))=FTYP$:GOTO 1370
1390 FRML=31:IF MX082=1 THEN CARDLEN=105:INLLEN=24:BACK$="+-----":BACK
LEN=5:GOTO 1410
1400 CARDLEN=70:INLLEN=24:BACK$="+-----":BACKLEN=5
1410 CARDLEN=CARDLEN+CARDLEN*2/13*WIDTH
1420 HCARDLEN=CARDLEN/2-1:TITLL=25+4*WIDTH
1430 FOR A=1 TO CARDLEN:KADER$=KADER$+"-":NEXT:KADER$="+"+KADER$+"+"
1440 FOR A=1 TO INLLEN:LINE$=LINE$+"-":NEXT
1450 LINE$=BACK$+"+"+LINE$+"+"+LINE$+"+"+LINE$+CHR$(8)+"+"
1460 DIST$=SPC(INLLEN)+CHR$(124)
1470 ESCAPE$=CHR$(27):AAN$=CHR$(1):OFF$=CHR$(0)
1480 CARSET$=ESCAPE$+"R"+CHR$(0)
1490 UNDERLON$=ESCAPE$+"-"+AAN$:UNDLOFF$=ESCAPE$+"-"+OFF$:CONDNSOFF$=C
HR$(18)
1500 CONDNSON$=CHR$(15):LINSPC5$=ESCAPE$+"A"+CHR$(5):LINSPC10$=ESCAPE$
+"A"+CHR$(10)
1510 SPRSCRON$=ESCAPE$+"S"+AAN$:SUPSCROFF$=ESCAPE$+"T"+AAN$
1520 H$=CHR$(124):DIST1$=RIGHT$(DIST$,LEN(DIST$)-1)

```

```

1530 BACK1$=H$+SPC(LEN(BACK$)-1):ENLARGON$=ESCAPE$+"W"+AAN$:ENLARGOFF$
=ESCAPE$+"W"+OFF$
1540 RETURN
1550 REM -----MLPDATA-----
1560 DATA #1C3,#DA,#0,#1C3,#6A,#0,#1C3,#12,#0,#1C3,#44,#0,#1C3,#C1,#0,
#1C3
1570 DATA #B6,#0,#F5,#C5,#D5,#E5,#21,#0,#0,#1,#FF,#0,#1CD,#22,#0,#C3
1580 DATA #74,#F7,#F5,#5E,#23,#AF,#57,#CD,#4B,#F3,#1C2,#27,#0,#F1,#CD,
#00
1590 DATA #FB,#C3,#FB,#F1,#CD,#1F,#F3,#21,#CB,#0,#36,#FF,#21,#6,#0,#39

1600 DATA #F9,#C3,#4D,#C1,#2A,#A2,#0,#22,#C8,#0,#121,#3,#0,#22,#62,#0
1610 DATA #21,#F9,#FF,#36,#80,#F3,#F5,#3A,#5F,#0,#F6,#1,#32,#5F,#0,#32

1620 DATA #FB,#FF,#F1,#FB,#21,#31,#1,#36,#1,#C9,#F5,#3A,#0,#FD,#E6,#8
1630 DATA #1CA,#7B,#0,#3A,#F3,#FF,#E6,#10,#1C2,#8D,#0,#21,#F9,#FF,#36,
#80
1640 DATA #FB,#F1,#E1,#C9,#21,#5D,#EB,#22,#62,#0,#1C3,#80,#0,#21,#CA,#
0
1650 DATA #7E,#B7,#1CA,#9C,#0,#3E,#A,#36,#0,#1C3,#AB,#0,#2A,#C8,#0,#7E

1660 DATA #B7,#1FA,#F2,#0,#23,#22,#C8,#0,#32,#F6,#FF,#FE,#D,#1C2,#7B,#
0
1670 DATA #32,#CA,#0,#1C3,#7B,#0,#21,#3,#FE,#36,#9B,#2B,#36,#2,#1C3,#C
9
1680 DATA #0,#21,#3,#FE,#36,#9B,#2B,#36,#1,#3A,#1F,#1,#87,#87,#F6,#3
1690 DATA #32,#1,#FE,#121,#0,#0,#22,#6E,#0,#C9,#F5,#3A,#0,#FE,#E6,#22
1700 DATA #EE,#2,#1CA,#EE,#0,#CD,#1F,#F3,#21,#7B,#D5,#22,#6E,#0,#F1,#C
3
1710 DATA #7B,#D5,#3C,#1C2,#84,#0,#2A,#A2,#0,#22,#C8,#0,#1C3,#7B,#0,#F
FF
1720 DATA #C5,#CD,#8F,#D9,#CD,#B4,#F3,#FB,#CD,#BB,#F3,#F3,#2C2,#34,#0,
#CD
1730 DATA #79,#F3,#6,#F,#CD,#94,#F4,#1C2,#07,#0,#6,#96,#CD,#94,#F4,#E6

1740 DATA #80,#2CA,#34,#0,#CD,#D5,#F3,#1DA,#4,#0,#CD,#A3,#F4,#CD,#E0,#
F3
1750 DATA #1DA,#F6,#0,#C1,#C9,#FFF
1760 REM -----FILETYPES-----
1770 DATA 0,BAS,1,OBJ,2,ARR,#,DNA,&,SPL,*,PAS,*,END

```


4.7.1983

Folgendes Programm Wandelt einen Speicherbereich in DATA Zeilen um

```
10 REM SPEICHERBEREICH IN DATA UMWANDELN
20 CLEAR 20000
30 DIM A(255,2):REM RESERVIERE 2K FÜR DEN EDITOR
40 REM SETZTE EDITBUFFER
50 I=VARPTR(A(0,0))
60 POKE SA2,I IAND $FF:POKE SA3,I SHR 8:REM START BUFFER
65 POKE SA4,I IAND $FF:POKE SA5,I SHR 8:REM CUREND
70 I=I+2048
80 POKE SA6,I IAND $FF:POKE SA7,I SHR 8:REM END BUFFER
90 INPUT "ADR-MIN,ADR-MAX,ZEILE ";XMIN,XXM,ZMIN
100 POKE $131,2:REM PRINT TO EDIT BUFFER
110 Z=ZMIN:X=XMIN
120 GOSUB 500
130 PRINT Z&," DATA $";HEX&(XMIN);", $";HEX&(XXM)
140 Z=Z+10
150 GOSUB 500
160 PRINT Z&," DATA $";HEX&(PEEK(X));
170 X=X+1
180 IF X>XXM GOTO 300
190 FOR K=2 TO 10
200 PRINT " $";HEX&(PEEK(X));
210 X=X+1:IF X>XXM GOTO 300
220 NEXT K
230 PRINT
240 GOTO 140
300 PRINT :PRINT CHR&(0)+CHR&(0);
310 POKE $131,1
320 STOP
500 Z&=STR&(Z)
510 FOR K=0 TO LEN(Z&)-1
520 IF MID&(Z&,K,1)="." GOTO 550
530 NEXT K
540 RETURN
550 Z&=LEFT&(Z&,K)
560 RETURN
```

\$=# &=\$

Nach Ablauf dieses Programmes stehen die DATA-Zeilen im EDIT-Buffer.
Löschen sie das momentane Programm mit NEW und holen die DATA-Zeilen
mit POKE \$135,2 aus dem EDIT-Buffer.



(Oliver Ebenhöhn)

NP45

```

40  REM # READLIND
41  REM - READLINE 12
60  REM 6 OKTOBER 1983
110 DATA #C5,#D5,#E5,#F5,#23,#23,#56,#23,#5E,#EB,#CD,#EA,#E6
111 DATA #22,#24,#01,#21,#23,#01,#36,#FF,#F1,#E1,#D1,#C1,#C9
130 REM #####
131 REM # Zwei DATA-Zeilen erlauben Ihnen einen      #
132 REM # READ-Auftrag zu geben mit einem Zeilennummer #
133 REM # dazu.                                         #
134 REM # RESTORE und abzaehlen von DATA sind ueber-  #
135 REM # fluessig! READ fangt an, wo Sie wollen!     #
136 REM # Diese beide DATA-Zeilen müssen aber die beide #
137 REM # ersten im Programm sein.                     #
138 REM # Der READ-Instruktion ist dann:              #
139 REM # <integer variable> = <Zeilennummer> :      #
140 REM #           CALLM #B200, <integer variable> #
141 REM # Zum Beispiel:                                #
142 REM # G%=3000:CALLM #B200,G%                        #
143 REM # READ faengt nun an ab Zeile 3000 DATA zu lesen.#
144 REM # Sehe weiter das Programmbeispiel:           #
145 REM #####
146 REM # Jan Blokker-Postfach 12-NL 1530 AA Niederlande #
147 REM #####
200 REM - EINLESEN DER READLINE-ROUTINE
210 FOR A=#B200 TO #B21F:READ CZ:POKE A,CZ:NEXT
300 REM - PROGRAMMBEISPIEL:
310 PRINT "BEISPIEL 'READLINE': "
320 DATA 1,2,3,4,5,6,7,8,9,10
330 FOR A=1 TO 10:READ BZ:PRINT BZ;:NEXT:PRINT
340 DATA 11,12,13,14,15,16,17,18,19,20
350 FOR A=1 TO 10:READ BZ:PRINT BZ;:NEXT:PRINT
360 DATA 21,22,23,24,25,26,27,28,29,30
370 FOR A=1 TO 10:READ BZ:PRINT BZ;:NEXT:PRINT
380 PRINT "BEISPIEL: "
390 LINEZ=360:CALLM #B200,LINEZ
400 FOR A=1 TO 10:READ BZ:PRINT BZ;:NEXT:PRINT
410 DATA 31,32,33,34,35,36,37,38,39,40
420 DATA 41,42,43,44,45,46,47,48,49,50
430 DATA 51,52,53,54,55,56,57,58,59,60
440 PRINT "EIN WEITERES BEISPIEL: "
450 ZEILEZ=430:CALLM #B200,ZEILEZ
460 FOR A=1 TO 10:READ BZ:PRINT BZ;:NEXT:PRINT
600 END

```

BEISPIEL 'READLINE' :

```

 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36

```

BEISPIEL:

```

 21 22 23 24 25 26 27 28 29 30

```

EIN WEITERES BEISPIEL:

```

 51 52 53 54 55 56 57 58 59 60

```

END PROGRAM

```

50000 MODE 0:PRINT CHR$(12):PRINT
50010 PRINT "      F U N K T I O N S - D A R S T E L L U N G":PRINT
50020 PRINT "      ERLAEUTERUNGEN":PRINT
50030 PRINT "      Mit diesem PROGRAMM koennen Sie beliebige FUNKTIONEN"
50040 PRINT "      in einem festzulegenden Bereich darstellen."
50050 PRINT "      Nach Druecken der BREAK-Taste sind die FUNKTIONS-"
50060 PRINT "      GLEICHUNGEN unter den STATEMENTS-NR. 61100, fuer die"
50070 PRINT "      1. FUNKTION, 61200, fuer die 2. FUNKTION, ... und"
50080 PRINT "      62000, fuer die 10. FUNKTION in BASIC einzutasten "
50090 PRINT "      und mit ':RETURN' abzuschliessen."
50100 PRINT "      Durch einen erneuten RUN-Befehl starten Sie das"
50110 PRINT "      PROGRAMM wieder und nach Druecken der SPACE-Taste"
50120 PRINT "      werden die gewuenschte FUNKTIONS-Nr. sowie der"
50130 PRINT "      darzustellende Bereich eingegeben.":PRINT
50140 PRINT "      EINGABE VON FUNKTIONS-GLEICHUNGEN - BREAK-TASTE"
50150 PRINT "      FUNKTIONS-DARSTELLUNG - SPACE-TASTE"
50160 Q=GETC:IF Q=0.0 GOTO 50160
50170 IF Q=32.0 GOTO 60000
50180 GOTO 50160
60000 CLEAR 3000:MODE 0:PRINT CHR$(12):PRINT "      F U N K T I O N S -
D A R S T E L L U N G":PRINT
60005 INPUT "      FUNKTION ";N:PRINT :PRINT
60006 GOSUB 60010
60007 GOTO 60050
60010 ON N GOTO 60015,60020,60025,60030,60031,60032,60033,60034,60035,
60036
60012 GOTO 60040
60015 LIST 61100:GOTO 60040
60020 LIST 61200:GOTO 60040
60025 LIST 61300:GOTO 60040
60030 LIST 61400:GOTO 60040
60031 LIST 61500:GOTO 60040
60032 LIST 61600:GOTO 60040
60033 LIST 61700:GOTO 60040
60034 LIST 61800:GOTO 60040
60035 LIST 61900:GOTO 60040
60036 LIST 62000:GOTO 60040
60040 CURSOR 0,CURY+1:PRINT "      "
60045 CURSOR 0,CURY+2:PRINT "      ":PRINT :PRINT :PRINT
60048 RETURN
60050 PRINT :PRINT :GOSUB 63000:PRINT :INPUT "      ANFANGSWERT  XA = ";
XA:PRINT
60055 INPUT "      ENDWERT      XE = ";XE:PRINT
60057 PRINT
60060 PRINT "      TYPE SPACE"
60065 Q=GETC:IF Q=0.0 GOTO 60065
60070 IF Q=32.0 GOTO 60150
60075 GOTO 60065
60150 PRINT :PRINT :PRINT "      MOMENT BITTE ! ! !"
60152 XD=XE-XA:PX=INT(336.0/(XD+2.0)):Y0=1E-10:YU=1E10
60155 FOR X=XA TO XE STEP 0.1
60160 ON N GOSUB 61100,61200,61300,61400,61500,61600,61700,61800,61900
,62000
60163 IF Y>20.0 OR Y<(-20.0) GOTO 60168
60165 IF Y>Y0 THEN Y0=Y:GOTO 60168
60167 IF Y<YU THEN YU=Y
60168 NEXT
60170 IF Y0>0.0 AND YU<=0.0 THEN YD=Y0-YU:PY=INT(212.0/(YD+2.0)):Y0=10
.0+PY+ABS(YU)*PY:GOTO 60195
60180 IF Y0<0.0 AND YU<0.0 THEN YD=ABS(YU):PY=INT(212.0/(YD+2.0)):Y0=2
12.0-25.0:GOTO 60195
60190 IF Y0>0.0 AND YU>0.0 THEN YD=ABS(Y0):PY=INT(212.0/(YD+2.0)):Y0=1
0.0
60195 PRINT CHR$(12):COLORG 7 0 15 2:MODE 5A

```

```

60197 GOSUB 60010:CURSOR 0,2:GOSUB 63800
60200 DRAW 5,Y0 XMAX-5,Y0 15
60205 X0=ABS(XA)*PX+10.0+PX:IF XA>0.0 THEN X0=10.0
60210 DRAW X0,5 X0,YMAX-5 15
60220 G=1.0:R=2.0:GOSUB 60380:GOSUB 60400
60230 G=-1.0:R=2.0:GOSUB 60390:GOSUB 60410
60240 G=10.0:R=4.0:GOSUB 60380:GOSUB 60400
60250 G=-10.0:R=4.0:GOSUB 60390:GOSUB 60410
60350 GOTO 60450
60380 FOR I=X0 TO XMAX STEP PX*G:GOSUB 60440:NEXT:RETURN
60390 FOR I=X0 TO XMIN STEP PX*G:GOSUB 60440:NEXT:RETURN
60400 FOR I=Y0 TO YMAX STEP PY*G:GOSUB 60445:NEXT:RETURN
60410 FOR I=Y0 TO YMIN STEP PY*G:GOSUB 60445:NEXT:RETURN
60440 DRAW I,Y0+1*R I,Y0-1*R 15:RETURN
60445 DRAW X0+1*R,I X0-1*R,I 15:RETURN
60450 FOR X=XA TO XE STEP 1.0/PX
60455 ON N GOSUB 61100,61200,61300,61400,61500,61600,61700,61800,61900
,62000
60460 XX=X*PX+X0
60462 YY=Y*PY+Y0
60465 IF XX<0.0 OR XX>XMAX THEN XX=0.0:Z%=0
60475 IF YY<0.0 OR YY>YMAX THEN YY=0.0:Z%=0
60480 DOT XX,YY Z%:Z%=0:NEXT
60485 PRINT :PRINT "      NEUE FUNKTION - SPACE      NEUE EINGABE - W";
60490 Q=GETC:IF Q=0.0 GOTO 60490
60495 IF Q=32.0 GOTO 60000
60500 CURSOR 0,2:GOSUB 63105:CURSOR 0,2:GOSUB 63800:GOTO 60450
61100 Y=A(1.0)*X+A(2.0)
61199 RETURN
61200 Y=A(1.0)*X*X*X*X-A(2.0)*X*X*X-A(3.0)*X*X-A(4.0)*X-A(5.0)
61299 RETURN
61300 Y=(A(1.0)*X*X-A(2.0)*X-A(3.0))/(A(4.0)*X-A(5.0))
61399 RETURN
61400 Y=A(1.0)*X+A(2.0)*X*X
61499 RETURN
61500 Y=(A(1.0)*X-A(2.0))/(X-1.0)/(X-1.0)/(X+A(3.0))
61599 RETURN
61600 Y=A(1.0)*X*SIN(A(2.0)*X)*EXP(-A(3.0)*X)-A(4.0):RETURN
61699 RETURN
61700 Y=A(1.0)*LOG(A(2.0)*X+A(3.0)):RETURN
61799 RETURN
61800 Y=A(1.0)*EXP(A(2.0)*X+A(3.0)):RETURN
61899 RETURN
61900 Y=A(1.0)*SIN(X+A(2.0))
61999 RETURN
62000 Y=(EXP(X)+EXP(-X))/A(1.0):RETURN
62099 RETURN
63000 REM
63100 INPUT "      ANZAHL DER KOEFFIZIENTEN = ";NN:PRINT :DIM A(NN)
63105 FOR I=1.0 TO NN:S=((I-1.0) MOD 3.0)+1.0)*15.0
63107 IF (I-1.0)>0.0 AND (I-1.0) MOD 3.0=0.0 THEN PRINT
63110 CURSOR S,CURY:PRINT "A(";I;")=";:INPUT A(I)
63140 NEXT:RETURN
63800 PRINT "      N = ";NN;
63810 FOR I=1.0 TO NN:S=((I-1.0) MOD 3.0)+1.0)*15.0
63820 IF (I-1.0)>0.0 AND (I-1.0) MOD 3.0=0.0 THEN PRINT
63830 CURSOR S,CURY:PRINT "A(";I;")=";A(I);
63840 NEXT:RETURN

```

Roland Winde
Bundesstraße 8
2000 Hamburg 13
Tel.: 040/41 86 00

INSTRING-FUNKTION FÜR DEN DAI

Aufgabenstellung:

Aus einer größeren Anzahl Strings sollen diejenigen ausgegeben werden, in denen eine bestimmte Zeichenfolge enthalten ist.

Beispiel:

Aus dem gespeicherten Jahres-Inhaltsverzeichnis einer Zeitschrift möchte man alle Beiträge haben, die sich mit dem Thema INTERFACE befassen. Mit der INSTRING-Funktion kann man nun überprüfen, in welchen Strings die Zeichenfolge "INTERFACE" enthalten ist.

Die Strings seien dabei in dem Feld A\$(I) gespeichert. Das Stichwort wird der Variablen FIND\$ zugewiesen.

Laden und Vorbereiten des Programms:

Nach dem Laden des Programms mit READ und POKE oder mit der ML-Einleseroutine muß noch der Pointer auf FIND\$ eingepoked werden - (folgende Adressen gelten bei Startadresse #300):

```
POINTER%=VARPTR(FIND$): INSTR=#300  
POKE#314,POINTER% MOD 256  
POKE#315,POINTER% SHR 8
```

Der Aufruf sieht dann so aus:

```
FIND$="INTERFACE"  
CALLM INSTR,A$(I)
```

In #2FF steht dann die Position, ab der FIND\$ in A\$(I) enthalten ist. Wenn PEEK(#2FF)=0, dann ist FIND\$ nicht in A\$(I) enthalten.

Zur Maskierung von Bit 7 (#325):
Ich habe festgestellt, daß in wenigen Fällen A\$(I) Zeichen mit gesetztem 7.Bit enthielt. Diese Strings wurden natürlich nicht gefunden. Darum die Maskierung von Bit 7!

Zum DAI-Editor, Löschen von Programmzeilen;
(Ausgabe 5/83, FW 9,1).

Einfacher ist folgende Methode :

```
EDIT n-m:EDIT x  
BREAK/SPACE
```

n-m ist der zu löschende Bereich. Zeile x muß nicht existieren.

INSTRING-FUNKTION

0300 F5 C5 D5 97 32 F3 02 5E 23 56 1A 32 F4 02 13 D5 074E
0310 32 F5 02 2A F8 99 46 23 B8 F2 24 03 D2 24 03 D1 06E8
0320 D1 C1 F1 C9 1A E6 7F 4E B9 C2 41 03 13 23 05 C2 07D5
0330 24 03 3A F5 02 47 3A F4 02 90 3C 32 F3 02 C3 1F 05A4
0340 03 3A F5 02 3D D1 C3 0E 03 0316

TREIBER-PROGRAMM FUER TYPENRAD-SCHREIBMASCHINE SILVER REED EX-42

02F0 00 00 3C 42 48 13 03 52 2E 57 49 4E 44 45 00 00 02D3
0300 F5 C5 D5 E5 21 03 FE 36 80 2B 36 A0 2B 36 40 2A 0718
0310 F5 02 E9 FE 01 C2 2D 03 21 21 03 22 F5 02 C3 1D 060F
0320 04 32 F2 02 21 13 03 22 F5 02 C3 6C 03 FE 02 C2 056E
0330 4A 03 21 3B 03 22 F5 02 C3 1D 04 32 F3 02 21 13 0404
0340 03 22 F5 02 CD 92 04 C3 6C 03 FE 03 C2 63 03 21 05FB
0350 58 03 22 F5 02 C3 1D 04 32 F4 02 21 13 03 22 F5 04CE
0360 02 3E 0A FE 0A C2 83 03 97 32 F1 02 3E 0C CD 78 05E5
0370 04 3A F0 02 FE 00 CA 1D 04 3D 32 F0 02 CD 87 04 05D2
0380 C3 71 03 FE 0B C2 97 03 3A F1 02 FE 00 CA 1D 04 06B2
0390 47 CD A1 04 C3 1D 04 FE 09 C2 B0 03 3A F0 02 FE 0743
03A0 00 CA 1D 04 3D 32 F0 02 3E 0B CD 78 04 C3 1D 04 04C2
03B0 FE 0D C2 BE 03 CD 53 04 CD 92 04 C3 1D 04 E6 7F 075E
03C0 FE 20 DA 1D 04 11 00 00 21 92 04 FE 60 DA DA 03 05F6
03D0 D6 20 5F 19 7E E6 3F C3 DD 03 5F 19 7E 32 FF 02 06DD
03E0 3A F0 02 47 3A F2 02 B8 C2 01 04 CD 53 04 CD 92 06A3
03F0 04 0E 0A 3E 48 CD 78 04 0D C2 F3 03 3E 0A 32 F0 051A

0400	<u>02</u>	3A	<u>FF</u>	<u>02</u>	CD	<u>78</u>	<u>04</u>	21	<u>F0</u>	<u>02</u>	34	3A	<u>FF</u>	<u>02</u>	FE	13	0619
0410	C2	<u>17</u>	<u>04</u>	35	C3	<u>1D</u>	<u>04</u>	FE	53	C2	<u>1D</u>	<u>04</u>	35	CD	<u>25</u>	<u>04</u>	0555
0420	E1	D1	C1	F1	C9	CD	BB	D6	FE	20	CA	<u>31</u>	<u>04</u>	CD	<u>43</u>	<u>04</u>	09BC
0430	C9	21	00	38	CD	<u>8A</u>	<u>04</u>	CD	BB	D6	FE	20	C8	CD	<u>43</u>	<u>04</u>	07D5
0440	C3	<u>37</u>	<u>04</u>	3A	C4	02	FE	FF	C0	3E	01	32	31	01	97	32	0627
0450	C4	02	C9	<u>3E</u>	09	CD	<u>78</u>	<u>04</u>	21	<u>F1</u>	<u>02</u>	34	3A	<u>F0</u>	<u>02</u>	FE	0691
0460	00	C2	<u>6B</u>	<u>04</u>	21	00	38	CD	<u>8A</u>	<u>04</u>	C9	3D	32	<u>F0</u>	<u>02</u>	21	0530
0470	90	02	CD	<u>8A</u>	<u>04</u>	C3	<u>5C</u>	<u>04</u>	32	00	FE	CD	<u>87</u>	<u>04</u>	3E	00	05D6
0480	32	00	FE	CD	<u>87</u>	<u>04</u>	C9	21	A0	09	A7	2B	7D	B4	C2	<u>8A</u>	076A
0490	<u>04</u>	C9	3A	<u>F3</u>	<u>02</u>	FE	00	C8	FE	48	D0	47	3A	<u>F1</u>	<u>02</u>	B8	0804
04A0	D8	3A	<u>F4</u>	<u>02</u>	90	47	CD	<u>53</u>	<u>04</u>	05	C2	<u>A6</u>	<u>04</u>	97	32	<u>F1</u>	072E
04B0	<u>02</u>	C9	48	78	7C	16	14	68	6C	13	64	58	3F	74	27	1F	04CD
04C0	1B	60	1C	38	3C	30	34	28	2C	20	24	18	5B	67	26	5C	0363
04D0	2A	50	70	7A	6B	73	72	71	76	6A	6E	65	62	66	5A	63	065D
04E0	6F	59	5D	79	75	7E	69	61	77	7D	7F	7B	6D	48	48	48	068E
04F0	56	48															009E

```

002          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
003          x SPRACHAUFNAHME UEBER CASSETTENRECORDER x
004          x MIT CALLM §300. WIEDERGABE UEBER CAS- x
005          x SETTENRECORDER MIT CALLM§400. x
006          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
007
008          ANFTAB EQU :500          ANFANGSADRESSE DER TABELLE
009          ENDHI EQU :A0           ENDADRESSE (HI) DER TABELLE
010          ENDLOW EQU :00          ENDADRESSE (LO) DER TABELLE
011          x DER SPEICHERBEREICH KANN HIER, ODER AN DEN
012          x MIT (((((XX)))) GEKENNZEICHNETEN STELLEN
013          x GEAENDERT WERDEN.
014          ORG :300          LESEPROGRAMM
015 0300 F5          PUSH PSW
016 0301 E5          PUSH H
017 0302 C5          PUSH B
018 0303 D5          PUSH D
019          x SPEICHERBEREICH FREI MACHEN
020 0304 210005          LXI H,ANFTAB ((((( ANFTAB ))))
021 0307 0600          MVI B,0
022 0309 70          LOOP MOV M,B
023 030A 23          INX H
024 030B 7D          MOV A,L
025 030C FE00          CPI ENDLOW (((((ENDLOW))))
026 030E C20903          JNZ LOOP
027 0311 7C          MOV A,H
028 0312 FEAO          CPI ENDHI (((((ENDHI))))
029 0314 C20903          JNZ LOOP
030          x AUF SIGNAL WARTEN
031 0317 0E05          MVI C,5
032 0319 3A00FD          SIG1 LDA :FD00
033 031C E680          ANI 128
034 031E 0680          MVI B,128
035 0320 90          SUB B
036 0321 CA1903          JZ SIG1
037 0324 3A00FD          SIG2 LDA :FD00
038 0327 E680          ANI 128
039 0329 0680          MVI B,128
040 032B 90          SUB B
041 032C C22403          JNZ SIG2
042 032F 0D          DCR C
043 0330 C21903          JNZ SIG1
044          x ZEICHEN FUER LESEBEGINN GEBEN
045 0333 3E00          MVI A,0
046 0335 327700          STA :0077
047          x BEGINN DER SPRACHAUFNAHME
048 0338 210005          LXI H,ANFTAB (((((ANFTAB))))
049 033B 1601          MVI D,1
050 033D 4E          ADSCHL MOV C,M
051 033E 3A00FD          BTSCHL LDA :FD00
052 0341 90          SUB B
053 0342 FA4B03          JM FORTS1 SPRINGE,WENN BIT NICHT GES.
054 0345 79          MOV A,C

```



```

055 0346 B2          ORA  D
056 0347 4F          MOV  C,A
057 0348 C35003      JMP  FORTS2
058 034B 00          FORTS1 NOP
059 034C 00          NOP
060 034D 00          NOP
061 034E 00          NOP
062 034F 00          NOP
063 0350 7A          FORTS2 MOV  A,D
064 0351 07          RLC
065 0352 57          MOV  D,A
066 0353 D23E03      JNC  BTSCHL
067 0356 71          MOV  M,C
068 0357 23          INX  H
069 0358 7D          MOV  A,L
070 0359 FE00        CPI  ENDLOW  (((ENDLOW)))
071 035B C23D03      JNZ  ADSCHL
072 035E 7C          MOV  A,H
073 035F FE00        CPI  ENDHI   (((ENDHI)))
074 0361 C23D03      JNZ  ADSCHL
075 0364 D1          POP  D
076 0365 C1          POP  B
077 0366 E1          POP  H
078 0367 F1          POP  PSW
079 0368 C9          RET
080
081                  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
081                  ORG  :400          SPRECHPROGRAMM
082 0400 F5          PUSH PSW
083 0401 E5          PUSH H
084 0402 C5          PUSH B
085 0403 D5          PUSH D
086 0404 210005      LXI  H,ANFTAB  (((ANFTAB)))
087 0407 1601        MVI  D,1
088 0409 4E          ADSCH1 MOV  C,M
089 040A 79          BTSCH1 MOV  A,C
090 040B A2          ANA  D
091 040C CA1404      JZ   NULL
092 040F 3E31        MVI  A,:31
093 0411 C31904      JMP  FORT
094 0414 3E30        NULL  MVI  A,:30
095 0416 C31904      JMP  FORT
096 0419 3206FD      FORT  STA  :FD06
097 041C 7A          MOV  A,D
098 041D 07          RLC
099 041E 57          MOV  D,A
100 041F D20A04      JNC  BTSCH1
101 0422 23          INX  H
102 0423 7D          MOV  A,L
103 0424 FE00        CPI  ENDLOW  (((ENDLOW)))
104 0426 C20904      JNZ  ADSCH1
105 0429 7C          MOV  A,H
106 042A FE00        CPI  ENDHI   (((ENDHI)))
107 042C C20904      JNZ  ADSCH1

```

```
108 042F D1            POP   D
109 0430 C1            POP   B
110 0431 E1            POP   H
111 0432 F1            POP   PSW
112 0433 C9            RET
113 0434                END
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx Symbol  Tabelle xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
ADSCH1 0409    ADSCHL 033D    ANFTAB 0500    BTSCH1 040A
BTSCHL 033E    ENDHI  00A0    ENDLOW 0000    FORT    0419
FORTS1 034B    FORTS2 0350    LOOP    0309    NULL    0414
SIG1    0319    SIG2    0324
```

```
0300 F5 E5 C5 D5 21 00 05 06 00 70 23 7D FE 00 C2 09
0310 03 7C FE A0 C2 09 03 0E 05 3A 00 FD E6 80 06 80
0320 90 CA 19 03 3A 00 FD E6 80 06 80 90 C2 24 03 0D
0330 C2 19 03 3E 00 32 77 00 21 00 05 16 01 4E 3A 00
0340 FD 90 FA 4B 03 79 B2 4F C3 50 03 00 00 00 00 00
0350 7A 07 57 D2 3E 03 71 23 7D FE 00 C2 3D 03 7C FE
0360 A0 C2 3D 03 D1 C1 E1 F1 C9
```

```
0400 F5 E5 C5 D5 21 00 05 16 01 4E 79 A2 CA 14 04 3E
0410 31 C3 19 04 3E 30 C3 19 04 32 06 FD 7A 07 57 D2
0420 0A 04 23 7D FE 00 C2 09 04 7C FE A0 C2 09 04 D1
0430 C1 E1 F1 C9
```

IMPINT:LIST

50 REM xxx HEXCODEVORLESER xxx 16.10.1983 xxx
70 REM xxx R. Kietzmann, Berlin xxx
100 PRINT CHR\$(12):CLEAR 1000
150 GOSUB 5000:REM EINLESEN DER ADRESSEN
200 PRINT " VORLESEN DES SPEICHERBEREICHES"
300 INPUT " VON ADRESSE ";ANFANG:PRINT
400 INPUT " BIS ADRESSE ";SCHLUSS:PRINT :PRINT
500 IF RIGHT\$(HEX\$(ANFANG),1)="0" THEN TEXT\$=" ":GOTO 600
550 TEXT\$="\$ "+HEX\$(ANFANG)+" "
600 FOR ADRESSE=ANFANG TO SCHLUSS
700 IF RIGHT\$(HEX\$(ADRESSE),1)≠"0" THEN 1000
800 PRINT TEXT\$:GOSUB 10000:REM TEXT SPRECHEN
900 TEXT\$="\$ "+HEX\$(ADRESSE)+" "
1000 INHALT=PEEK(ADRESSE)
1100 IF INHALT=16 THEN TEXT\$=TEXT\$+"0"
1200 TEXT\$=TEXT\$+HEX\$(INHALT)+" "
1300 NEXT ADRESSE
1400 PRINT TEXT\$:GOSUB 10000:PRINT :PRINT :GOTO 200
5000 REM xxx EINLESEN DER ADRESSEN xxx
5010 DIM FELD(35,1):RESTORE
5020 FOR I=0 TO 35:READ ZEICHEN\$:READ FELD(I,0):READ FELD(I,1):NEXT I
5030 RETURN
10000 REM xxx TEXT\$ SPRECHEN xxx
10100 FOR POS=0 TO LEN(TEXT\$)-1
10200 ZEICHEN\$=MID\$(TEXT\$,POS,1)
10300 IF ZEICHEN\$="" THEN WAIT TIME 10:GOTO 10900
10400 ANFANGSADR\$=FELD(ASC(ZEICHEN\$)-35,0)
10500 SCHLUSSADR\$=FELD(ASC(ZEICHEN\$)-35,1):REM FELDNUMMER=ASCIINUMMER-35
10700 POKE \$405,(ANFANGSADR\$ MOD 256):POKE \$406,(ANFANGSADR\$/256):POKE \$425,(SCHLUSSADR\$ MOD 256):POKE \$426,(SCHLUSSADR\$/256)
10800 CALLM \$400:REM SPRECHEN
10900 NEXT POS
10950 RETURN
10990 REM xxx TABELLE DER ZU DEN ASCII-ZEICHEN GEHOERENDEN ANFANGS- UND ENDADRESSEN xxx
11000 DATA \$,§33DF,§3970
11005 REM NICHT BENUTZTE ASCII-ZEICHEN
11010 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
11020 DATA 0,§1939,§1BA9
11025 DATA 1,§500,§850
11030 DATA 2,§0850,§0B90
11050 DATA 3,§1BA9,§1E40
11060 DATA 4,§1E40,§20E0
11070 DATA 5,§20E0,§24E0
11080 DATA 6,§24E0,§2810
11090 DATA 7,§2810,§2BB7
11100 DATA 8,§0B90,§0E1F
11110 DATA 9,§0E1F,§1195
11115 REM NICHT BENUTZTE ASCII-ZEICHEN
11120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
11130 DATA A,§1195,§148D
11140 DATA B,§165D,§1939
11150 DATA C,§2BB7,§2F30
11160 DATA D,§148D,§165D
11170 DATA E,§2F30,§31C0
11180 DATA F,§31C0,§33DE

§ = #