

```

002 *****
003 *
004 *          FORMATDRUCK          *
005 *          -----          *
006 *
007 *   DIE FOLGENDE ROUTINE DIENT ZUM AUTOMATISCHEN
008 *   FORMATIERTEN DRUCKEN UND LISTEN.
009 *
010 *   SIE ERLAUBT ES, TEXTE ZU DRUCKEN ODER PROGRAMME
011 *   ZU LISTEN, OHNE AUF SEITENVORSCHUEBE ODER ZEIL-
012 *   LENLAENGEN ZU ACHTEN.
013 *
014 *   SIE ERZEUGT SELBSTSTAENDIG ZEILENVORSCHUEBE
015 *   BEI EVENTUELLER UEBERLAENGE (LIST) MIT EINER
016 *   ZUSAETZLICHEN EINRUECKUNG UM 6 ZEICHEN, SEITEN-
017 *   VORSCHUEBE BEI ERREICHEN DER EINGESTELLTEN
018 *   SEITENZAHL UND GENERIERT BEI BEDARF AUTOMATISCH
019 *   EINE TITELZEILE UND/ODER EINE SEITENNUMMER.
020 *   AUSSERDEM LAESST SICH DIE ERSTE DRUCKPOSITION
021 *   BELIEBIG LEGEN (RAND).
022 *
023 *****
024
025 *****
026 *
027 *   VOR DEM START LINES (>0), CHARS (>0), RAND (>=0)
028 *   UND EVTL. DIE TITELZEILE U. SEITENNR. FESTLEGEN.
029 *   DANACH POKE FLAG,#A = ES WIRD SOFORT EIN TITEL
030 *   AUSGEGEBEN,
031 *   ODER POKE FLAG,0 = KEIN TITEL
032 *
033 *   AUFRUF DES FORMAT-DRUCKENS DURCH POKE #131,4.
034 *   DANACH WIRD JEDER AUSGEGEBENENE TEXT FORMATIERT.
035 *   STEUERZEICHEN UEBER POKE #131,5 AN DEN DRUCKER
036 *   AUSGEBEN, UM DIE ZAEHLER NICHT ZU BEEINFLUSSEN!
037 *   DAS PROGRAMM SOLLTE IMMER UNTERHALB DES HEAP
038 *   LIEGEN, UM ES VOR 'NEW' O.AE. ZU SCHUETZEN.
039 *
040 *   WICHTIG: DIESE ROUTINE MUSS IMMER WISSEN, WO DER
041 *   DRUCKER STEHT, DESHALB KEINE BEWEGUNGEN
042 *   VON HAND DURCHFUEHREN!
043 *
044 *****
045
046
047 BLANK EQU :20
048 OTSW EQU :131      AUSGABE-SCHALTER
049 SEROUT EQU :DD94   AUSGABE AKKU NACH RS-232
050 DISC EQU :0ACF     FUER ANDERES DOS ANPASSEN!
051
052
053
054
055
056 02DD C33003      ORG :2DD      HIERHER SPRINGT DIE AUSGABE-
057
058
059
060
061
062
063

```

064					
065		ORG	:	2EC	BEI BEDARF ANPASSEN
066					
067	02EC	CHARS	RES	1,80	ZEICHEN PRO ZEILE INCL. RAND
068	02ED	LINES	RES	1,66	ZEILEN PRO SEITE
069	02EE	RAND	RES	1,6	ANZAHL BLANKS VOR DER ERSTEN
070					DRUCKPOSITION
071	02EF	LEERZ	RES	1,2	ANZAHL DER LEERZEILEN
072					ZWISCHEN TITEL U. TEXT
073	02F0	CHCNT	RES	1,0	ZEICHENZAEHLER
074	02F1	LCNT	RES	1,0	ZEILENZAEHLER
075	02F2	FLAG	RES	1,0	BIT0=0: WENN DAS NAECHSTE
076					ZEICHEN KOMMT, IST VORHER
077					DER RAND AUSZUGEBEN
078					BIT1=1: TITELZEILE IST AUS-
079					ZUGEBEN
080					BIT2=1: EINRUECKUNG IST AUS-
081					ZUGEBEN (Z.B. BEIM LISTEN
082					BEI UEBERLAENGE)
083					BIT3=1: TITEL IST GEWUENSCHT
084	02F3 30303031	PAGCNT	ASC	'0001'	NAECHSTE AUSZUGEBENDE SEI-
085					TENNUMMER ALS ASCII, VIER-
086					STELLIG
087	02F7	ETAB	RES	1,0	ANZAHL BLANKS ZWISCHEN RAND-
088					ENDE UND TITELTEXT
089	02F8	TEXT	RES	56,0	TEXT DER TITELZEILE
090					MUSS MIT 0 ENDEN, WENN IR-
091					GENDWO '&' STEHT, WIRD DORT
092					DIE SEITENNR. EINGEFUEGT
093					
094	0330 F5	START	PUSH	PSW	ZEICHEN RETTEN
095	0331 3A3101		LDA	OTSW	
096	0334 FE04		CPI	4	
097	0336 DA4403		JLS	DISC1	FUER AUSGABEN ZUR FLOPPY
098	0339 CA4C03		JE	FORMAT	
099	033C FE06		CPI	6	
100	033E DA4803		JLS	CHROT1	AUSGABE VON NICHTDRUCKENDEN
101					STEUERZEICHEN, DIE NICHT AUF
102					DEM BILDSCHIRM ERSCHEINEN
103					SOLLEN, AN DEN DRUCKER
104	0341 F1		POP	PSW	FREI FUEER SPRUNG AUF ANDERE
105	0342 C9		RET		ROUTINEN UND WEITERE AUS-
106	0343 00		NOP		WERTUNG VON OTSW
107	0344 F1	DISC1	POP	PSW	ZEICHEN WIEDER IN AKKU
108	0345 C3CF0A		JMP	DISC	ZUR FLOPPY-ROUTINE
109	0348 F1	CHROT1	POP	PSW	
110	0349 C394DD	CHROUT	JMP	SEROUT	AUSGABE DES AKKUS NACH RS232
111					ODER ZU EINER PARALLELDRECK-
112					ROUTINE
113					
114	034C F1	FORMAT	POP	PSW	ZEICHEN ZURUECK
115	034D EF		RST	5	AUSGABE AUF SCREEN
116	034E 03		DATA	3	
117	034F FE0D		CPI	:D	NEUE ZEILE?
118	0351 CA8B03		JE	CRET	
119	0354 FE0C		CPI	:C	NEUE SEITE?
120	0356 CAAC03		JE	FORMF	
121	0359 E5		PUSH	H	
122	035A F5		PUSH	PSW	
123	035B 3AF202		LDA	FLAG	
124	035E 1F		RAR		
125	035F D4CC03		CNC	TAB	BIT0=0

126	0362	1F		RAR		
127	0363	DCFD03		CC	TITEL	BIT1=1
128	0366	1F		RAR		
129	0367	DCED03		CC	TABEXT	BIT2=1
130	036A	F1		POP	PSW	
131	036B	CD4903		CALL	CHROUT	
132	036E	F5		PUSH	PSW	
133	036F	21F002		LXI	H,CHCNT	
134	0372	34		INR	M	ZEICHENZAehler+1
135	0373	3AEC02		LDA	CHARS	
136	0376	BE		CMP	M	
137	0377	CC7D03		CE	CRETEX	ZEILENLAENGE UEBERSCHRITTEN
138	037A	F1		POP	PSW	
139	037B	E1		POP	H	
140	037C	C9		RET		ENDE FORMATDRUCK
141						
142	037D	3E0D	CRETEX	MVI	A,:D	ERZWINGT NEUE ZEILE UND
143	037F	CD8B03		CALL	CRET	BEREITET EINRUECKUNG VOR
144	0382	3AF202		LDA	FLAG	
145	0385	F605		ORI	5	SETZE BIT0 UND 3
146	0387	32F202		STA	FLAG	
147	038A	C9		RET		
148						
149	038B	F5	CRET	PUSH	PSW	BEREITET NEUE ZEILE VOR UND
150						MACHT BEI BEDARF SEITENVORSC
151	038C	E5		PUSH	H	
152	038D	CD4903		CALL	CHROUT	
153	0390	AF		XRA	A	
154	0391	32F002		STA	CHCNT	ZEICHENZAehler AUF 0
155	0394	3AF202		LDA	FLAG	
156	0397	E6FA		ANI	:FA	SETZTE RAND-FLAG
157	0399	32F202		STA	FLAG	
158	039C	21F102		LXI	H,LCNT	
159	039F	34		INR	M	ERHOEHE ZEILENZAehler
160	03A0	3AED02		LDA	LINES	
161	03A3	BE		CMP	M	
162	03A4	CCAA03		CE	FORMFX	ERZWINGE SEITENVORSCHUB
163	03A7	E1		POP	H	
164	03A8	F1		POP	PSW	
165	03A9	C9		RET		
166						
167	03AA	3E0C	FORMFX	MVI	A,:C	
168	03AC	F5	FORMF	PUSH	PSW	GIB SEITENVORSCHUB AUS
169						UND SETZTE DIE FLAGS
170	03AD	CD4903		CALL	CHROUT	
171	03B0	AF		XRA	A	
172	03B1	32F002		STA	CHCNT	ZEICHENZAehler AUF 0
173	03B4	32F102		STA	LCNT	ZEILENZAehler AUF 0
174	03B7	3AF202		LDA	FLAG	
175	03BA	E608		ANI	8	TITEL GEWUENSCHT?
176	03BC	3AF202		LDA	FLAG	
177	03BF	C4C903		CNZ	SETB1	BEREITE TITELAUSGABE VOR
178	03C2	E6FA		ANI	:FA	SETZTE RAND-FLAG
179	03C4	32F202		STA	FLAG	
180	03C7	F1		POP	PSW	
181	03C8	C9		RET		
182	03C9	F602	SETB1	ORI	2	D.H. VOR DEM NAECHSTEN
183						ANKOMMENDEN ZEICHEN WIRD
184	03CB	C9		RET		DER TITEL AUSGEGEBEN
185						
186	03CC	F5	TAB	PUSH	PSW	GIB LOCHERRAND AUS
187	03CD	3AEE02		LDA	RAND	

188	03D0	32F002		STA	CHCNT	
189	03D3	B7		ORA	A	RAND=0?
190	03D4	CAE103		JZ	AUS	
191	03D7	6F		MOV	L,A	
192	03D8	3E20	SPACE	MVI	A,BLANK	
193	03DA	CD4903	TBLOOP	CALL	CHROUT	
194	03DD	2D		DCR	L	
195	03DE	C2DA03		JNZ	TBLOOP	
196	03E1	3AF202	AUS	LDA	FLAG	
197	03E4	F601		ORI	1	RAND NICHT MEHR AUSGEBEN
198	03E6	E6FB		ANI	:FB	AUCH KEINE EINRUECKUNG
199	03E8	32F202		STA	FLAG	
200	03EB	F1		POP	PSW	
201	03EC	C9		RET		
202						
203	03ED	F5	TABEXT	PUSH	PSW	GIB EINRUECKUNG AUS BEI
204						ZEILENUEBERLAENGE
205	03EE	CDCC03		CALL	TAB	NORMALER RAND PLUS
206	03F1	2E06		MVI	L,6	EINRUECKUNG BEI UEBERLAENGE
207	03F3	3AF002		LDA	CHCNT	
208	03F6	85		ADD	L	
209	03F7	32F002		STA	CHCNT	SETZTE ZEICHENZAEHLER
210	03FA	C3D803		JMP	SPACE	
211						
212						
213						GIB DIE TITELZEILE AUS
214	03FD	17	TITEL	RAL		BIT0 VON FLAG IST CARRY
215	03FE	DCCC03		CC	TAB	WENN DER RAND NOCH NICHT
216						AUSGEBEN, DANN JETZT
217						AUSGEBEN
218	0401	F5		PUSH	PSW	
219	0402	3AF202		LDA	FLAG	
220	0405	E6FD		ANI	:FD	TITEL-FLAG RUECKSETZEN
221	0407	32F202		STA	FLAG	
222	040A	CD1304		CALL	HAUPT	GIB TEXT AUS
223	040D	F1		POP	PSW	
224	040E	D4CC03		CNC	TAB	WENN RAND SCHON AUSGEBEN
225						WAR, DANN GIB IHN NEU AUS
226	0411	1F		RAR		
227	0412	C9		RET		
228	0413	E5	HAUPT	PUSH	H	GIB TITEL U. SEITENNR. AUS
229	0414	D5		PUSH	D	
230	0415	21F702		LXI	H,ETAB	EXTRA SPACES RAND/TEXT
231	0418	7E		MOV	A,M	
232	0419	B7		ORA	A	=0?
233	041A	CA2704		JZ	TITOUT	
234	041D	57		MOV	D,A	
235	041E	3E20		MVI	A,BLANK	
236	0420	CD4903	TTLOOP	CALL	CHROUT	
237	0423	15		DCR	D	
238	0424	C22004		JNZ	TTLOOP	
239	0427	23	TITOUT	INX	H	JETZT TITELZEILE
240	0428	7E		MOV	A,M	ZEICHEN
241	0429	FE26		CFI	'&'	SEITENNUMMER AUSGEBEN?
242	042B	CA4904		JE	NUMMER	
243	042E	B7		ORA	A	=0, ALSO ENDE DER ZEILE?
244	042F	CA3804		JZ	ZENDE	
245	0432	CD4903		CALL	CHROUT	
246	0435	C32704		JMP	TITOUT	NAECHSTES ZEICHEN
247	0438	3AEF02	ZENDE	LDA	LEERZ	
248	043B	3C		INR	A	TITELZEILE ABSCHLIESSEN
249	043C	57		MOV	D,A	UND LEERZEILEN AUSGEBEN

```

250 043D 3E0D          MVI  A,:D      ZEILENVORSCHUB
251 043F CD8B03      LZTT  CALL  CRET
252 0442 15          DCR  D
253 0443 C23F04      JNZ  LZTT
254 0446 D1          POP  D
255 0447 E1          POP  H
256 0448 C9          RET
257 0449 E5          NUMMER PUSH  H      GIB SEITENNUMMER AUS
258 044A 21F302      LXI  H,PAGCNT
259 044D 1604      MVI  D,4      MAX. 4 ZIFFERN
260 044F 1E30      MVI  E,'0'
261 0451 7E          NULL  MOV  A,M      SUCHE ERSTE NICHT-NULL
262 0452 23          INX  H
263 0453 BB          CMP  E
264 0454 C25C04      JNE  ZIFFER   >0
265 0457 15          DCR  D
266 0458 C25104      JNZ  NULL
267 045B 14          INR  D      EVT. 4*NULL
268 045C CD4903      ZIFFER CALL  CHROUT  GIB ZIFFER AUS
269 045F 7E          MOV  A,M      NAECHSTE ZIFFER
270 0460 23          INX  H
271 0461 15          DCR  D
272 0462 C25C04      JNZ  ZIFFER   NOCH EINE?
273 0465 CD6C04      CALL  PLUS1   ERHOEHE SEITENNUMMER ALS ASC
274 0468 E1          POP  H
275 0469 C32704      JMP  TITOUT   WEITER IM TITEL
276
277 046C 1604      PLUS1  MVI  D,4      4 ZIFFERN
278 046E 2B          DCX  H
279 046F 2B          NEXTZI DCX  H
280 0470 34          INR  M      ERHOEHE ZIFFER
281 0471 7E          MOV  A,M
282 0472 FE3A      CPI  :3A     KEIN UEBERTRAG? ( <='9' )
283 0474 D8          RLS      ALLES KLAR
284 0475 3630      MVI  M,'0'   SETZTE 0, ERHOEHE VORIGE ZIF
285 0477 15          DCR  D
286 0478 C26F04      JNZ  NEXTZI
287 047B C9          RET
288
289 047C          END      KEINE FEHLERMELDUNG BEI UE-
                          BERLAUF! (ZAEHLT MOD 10000)

```

```

*****
* S Y M B O L   T A B L E *
*****

```

AUS	03E1	BLANK	0020	CHARS	02EC	CHCNT	02F0
CHROT1	0348	CHROUT	0349	CRET	038B	CRETEx	037D
DISC	0ACF	DISC1	0344	ETAB	02F7	FLAG	02F2
FORMAT	034C	FORMF	03AC	FORMFX	03AA	HAUPT	0413
LCNT	02F1	LEERZ	02EF	LINES	02ED	LZTT	043F
NEXTZI	046F	NULL	0451	NUMMER	0449	OTSW	0131
PAGCNT	02F3	PLUS1	046C	RAND	02EE	SEROUT	DD94
SETB1	03C9	SPACE	03D8	START	0330	TAB	03CC
TABEXT	03ED	TBLOOP	03DA	TEXT	02F8	TITEL	03FD
TITOUT	0427	TTLOOP	0420	ZENDE	0438	ZIFFER	045C

```

1   REM Textrahmen by K.Peter 28.2.82
2   REM
3   REM Dieses Programm zeichnet in Mode 0 beliebige
4   REM Umrahmungen (z.B. fuer Spiele oder Formulare)
5   REM deren Koordinaten in der DATA-Zeile 400 folgender-

6   REM massen gespeichert sind :
7   REM X linkes unteres Eck, Y linkes unteres Eck,
8   REM X rechtes oberes Eck, Y rechtes oberes Eck,
9   REM Nachdem alle Rahmen so ueber ihre Cursorko-
11  REM ordinaten definiert wurden, wird die Daten-
12  REM folge mit #FF abgeschlossen. Es duerfen belie-
13  REM big viele Rahmen, die sich auch schneiden und
14  REM ueberlappen duerfen, definiert werden.
15  REM Geben Sie bitte IMP INT und das Progamm
16  REM ab Zeile 30 ein. Wenn Sie es starten wird
17  REM das in Zeile 400 definierte Beispiel gezeichnet
18  REM
19  REM VARIABLEN :
20  REM Z(),E(),K(),H()
21  REM I,J,K,Z
22  REM
30  DIM ZX(15,1),EZ(4),KZ(4),HZ(6)
35  HZ(1)=3:HZ(2)=6:HZ(3)=12
40  HZ(4)=9:HZ(5)=5:HZ(6)=10
45  PRINT CHR$(12)
50  FOR I%=1 TO 11
60  READ J%,K%
70  ZX(K%-10,0)=J%
80  ZX(J%,1)=K%
90  NEXT I%
100 FOR I%=1 TO 4
110 READ K%(I%)
120 IF K%(I%)=#FF THEN 340
130 NEXT I%
140 FOR I%=1 TO 4
150 EZ(I%)=#B3DD-2*K%((I%/2 MOD 2)*2+1)+#86*K%(I%/3*2+2)
160 J%=I%:GOSUB 280
170 NEXT I%
180 EZ(2)=EZ(2)+2
190 EZ(4)=EZ(4)-2
200 IF EZ(2)>=EZ(1) THEN 230
210 I%=5:FOR J%=2 TO 4 STEP 2
220 GOSUB 280:NEXT:GOTO 180
230 EZ(4)=EZ(4)-#86
240 EZ(2)=EZ(2)+#86
250 IF EZ(2)>=EZ(3) THEN 100
260 I%=6:FOR J%=2 TO 4 STEP 2
270 GOSUB 280:NEXT:GOTO 230
280 ZX=HZ(I%)
290 K%=PEEK(EZ(J%))
300 IF K%>9.0 AND K%<26.0 THEN ZX=ZX(K%-10.0,0.0) IOR ZX
310 ZX=ZX(ZX,1.0)
320 POKE EZ(J%),ZX
330 RETURN
340 DATA 3,19,5,11,6,18,7,24,9,21,10,10
350 DATA 11,23,12,20,13,25,14,22,15,16
390 REM DATEN FUR GEWUENSCHTE RAHMEN
400 DATA 10,4,50,20,10,6,50,18,23,18,37,20,23,4,50,8,#FF

```

```

1 REM DCR Bandkontrolle
2 REM 08.06.82 rh
3 CLEAR 5000:COLORT 5 15 0 0
4 PRINT CHR$(12):CURSOR 18,16:PRINT "M E M O C O M"
5 CURSOR 12,12:PRINT " B A N D K O N T R O L L E"
6 DIM ZEILE(23):DIM TITEL(23):DIM TITEL$(23)
7 FOR ZPOKE=0 TO 23:ZEILE(23-ZPOKE)=#BFEE-#86*ZPOKE:NEXT

8 FOR ANZ=1 TO 23:READ TITEL(ANZ),TITEL$(ANZ):IF TITEL$(
ANZ)="###" THEN 9:NEXT
9 WAIT TIME 50:PRINT CHR$(12)
10 COLORT 8 0 8 0
11 PRINT CHR$(12);
12 FOR PRT=1 TO 23
13 IF PRT<10 THEN PRINT " ";PRT;CHR$(9);CHR$(9);" ";
TITEL$(PRT)
14 IF PRT>9 THEN PRINT PRT;CHR$(9);CHR$(9);" ";TITEL
$(PRT)
15 NEXT
16 REM WANDERNTER CURSOR
17 POKE #74,3:POKE #75,#20
18 FOR X=23 TO 23-(ANZ-1) STEP -1
19 CURSOR 9,X:PRINT CHR$(9)
20 POKE ZEILE(X)+#86,#D0
21 POKE ZEILE(X),#DA:REM #D(X):X=FARBE
22 POKE ZEILE(X)-#86,#D0
23 FOR WARTE=1 TO 500
24 IF GETC<>0 GOTO 29
25 NEXT WARTE
26 CURSOR 9,X:PRINT CHR$(32)
27 NEXT X
28 GOTO 16
29 REM
30 Y=24-X
31 IF TITEL$(Y)="###" THEN 40
32 IF LEFT$(TITEL$(Y),1)="B" THEN 39
33 IF TITEL$(Y)="zum Bandende" THEN GOTO 38
34 IF Y=TITEL(Y) THEN 37
35 GOTO 11
36 REM
37 PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM laed
das Programm : ";TITEL$(Y);" . ";CHR$(1):GOTO 44
38 PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM spui
t zum Bandende ";CHR$(1):GOTO 51
39 PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM hat
da kein Programm ";CHR$(1):GOTO 42
40 PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM hat
nicht mehr Programme auf Band ! ";CHR$(1):GOTO 42
41 REM
42 WAIT TIME 200:GOTO 11
43 REM
44 CALLM #F000:REM REW
45 FOR LAUF=1 TO TITEL(Y)
46 CALLM #F000:REM SKIP1
47 NEXT LAUF
48 LOAD
49 END
50 REM

```

```

20000 REM ARRAY-SAVER
20010 COLORT B 0 8 0: XMCM$=B$: X=0
20020 CURSOR 0,3: PRINT " R E C O R D E R W A H L "
20030 CURSOR 45,3: PRINT "SAVE in DCR 0";: CURSOR 45,2: PRINT "
SAVE in DCR 1";: CURSOR 45,1: PRINT "SAVE in CAS 1";: CUR
SOR 45,0: PRINT "SAVE in CAS 2";
20040 POKE #74,3: POKE #75,#20: FOR X=3 TO 0 STEP -1
20050 CURSOR 40,X: PRINT CHR$(1);: FOR XZ=1 TO 200: IF GETC<>0
GOTO 20060: NEXT XZ: CURSOR 40,X: PRINT CHR$(32);: NEXT X:
GOTO 20030
20060 XCN=24.0-X: ON XCN-20 GOTO 20070,20140,20200,20270
20070 PRINT CHR$(12): CURSOR 1,2: PRINT CHR$(1); FN1$; CHR$(1); "
saving in DCR 0."
20080 CURSOR 1,1: PRINT " TYPE SPACE": XMC
M=GETC: IF XMCM=0 THEN GOTO 20080
20090 CALLM #F000: REM DCR0: REW1: SKIP
20100 CALLM #F003, XERR
20110 SAVEA XMCM$ FN1$
20120 ON XERR GOTO 20020,20020,20020,20340
20130 RETURN
20140 PRINT CHR$(12): CURSOR 1,2: PRINT CHR$(1); FN1$; " "; CHR$(
1); " saving in DCR 1."
20150 CALLM #F000: REM DCR1: REW1: SKIP
20160 CALLM #F000, XERR
20170 SAVEA XMCM$ FN1$
20180 ON XERR GOTO 20020,20020,20020,20340
20190 RETURN
20200 PRINT CHR$(12): CURSOR 1,2: PRINT CHR$(1); FN1$; " "; CHR$(
1); " saving in CAS 1."
20210 CURSOR 1,1: PRINT "SET RECORD, START TAPE, TYPE SPACE": XM
CM=GETC: IF XMCM=0 THEN GOTO 20210
20220 CALLM #F000: REM CAS1
20230 CALLM #F003, XERR
20240 SAVEA XMCM$ FN1$
20250 ON XERR GOTO 20020,20020,20020,20340
20260 RETURN
20270 PRINT CHR$(12): CURSOR 1,2: PRINT CHR$(1); FN1$; " "; CHR$(
1); " saving in CAS 2."
20280 CURSOR 1,1: PRINT "SET RECORD, START TAPE, TYPE SPACE": XM
CM=GETC: IF XMCM=0 THEN GOTO 20280
20290 CALLM #F000: REM CAS2
20300 CALLM #F000, XERR
20310 SAVEA XMCM$ FN1$
20320 ON XERR GOTO 20020,20020,20020,20340
20330 RETURN
20340 REM MELDUNG SAVING ERROR
20350 PRINT CHR$(12)
20360 CURSOR 0,12
20370 PRINT " "; CHR$(1); " S A V I N G E R R
O R "; CHR$(1)
20380 FOR X=0 TO 6
20390 POKE #BFEE-#86*11, #D3
20400 SOUND 0 1 15 0 FREQ(1000)
20410 WAIT TIME 20: SOUND OFF
20420 POKE #BFEE-#86*11, #DE
20430 SOUND 0 1 15 0 FREQ(500)
20440 WAIT TIME 20: SOUND OFF
20450 NEXT
20460 RETURN: REM ZURUECK ZUM MENU

```



```

19500 REM STEUERUNG MDCR
19505 PRINT CHR$(12)
19510 PRINT :PRINT :PRINT :PRINT :PRINT
19515 PRINT SPC(15);" SKIP . . . . . < S >"
19520 PRINT SPC(15);" REWIND . . . . . < R >"
19525 PRINT SPC(15);" LOOK . . . . . < L >"
19530 PRINT SPC(15);" VERIFY . . . . . < V >"
19535 PRINT SPC(15);" ZUM MENUE . . . . . < M >"
19540 INCH=GETC:IF INCH=0 THEN 19540
19545 IF INCH=ASC("S") OR INCH=ASC("s") THEN 19695
19550 IF INCH=ASC("R") OR INCH=ASC("r") THEN 19640
19555 IF INCH=ASC("L") OR INCH=ASC("l") THEN 19580
19560 IF INCH=ASC("V") OR INCH=ASC("v") THEN 19775
19565 IF INCH=ASC("M") OR INCH=ASC("m") THEN 19000
19570 PRINT :PRINT
19575 GOTO 19540
19580 REM LOOK
19585 PRINT CHR$(12):CURSOR 0,12
19590 PRINT "File-Name : "
19595 CALLM #F000:REM LOOK
19600 PRINT "Weitersuchen ? <J/N>"
19605 GOSUB 19300:REM ABFRAGE OK$
19610 IF OK$="J" THEN 19625
19615 IF OK$="N" THEN 19500
19620 GOTO 19605
19625 PRINT CHR$(12)
19630 CALLM #F000:REM SKIP 1
19635 GOTO 19580
19640 REM REWIND
19645 PRINT CHR$(12):CURSOR 0,12
19650 PRINT "Wieviele Files zurueckspulen ? "
19655 PRINT "Druecke 1 - 9 ; 0=REWIND "
19660 GOSUB 19750
19665 IF LAUF=0.0 THEN 19820
19670 FOR REW=1 TO LAUF
19675 CALLM #F000:REM REWIND 1
19680 NEXT
19685 PRINT CHR$(12)
19690 GOTO 19580
19695 REM SKIP
19700 PRINT CHR$(12):CURSOR 0,12
19705 PRINT "Wieviele Files vorspulen ? "
19710 PRINT "Druecke 1 - 9 ; 0=SKIP"
19715 GOSUB 19750
19720 IF LAUF=0.0 THEN 19830
19725 FOR SKI=1 TO LAUF
19730 CALLM #F000:REM SKIP 1
19735 NEXT
19740 PRINT CHR$(12)
19745 GOTO 19580
19750 LAUF=GETC
19755 IF LAUF<48 OR LAUF>57 THEN 19750
19760 IF LAUF=0 THEN 19750
19765 LAUF=LAUF-48
19770 RETURN
19775 REM VERIFY

```

Liebe DAINamiker !

PROGRAMM : "CHECK-LIST-DATA"

Das Programm listet die Inhalte einer C 60 Kasette auf und schreibt das Inhaltsverzeichnis in das in den Zeilen 1-35 stehende DATA-Feld zurück.

Speichert man das Programm bei Neubeginn einer Kasette an die erste Stelle, dann kann man von Zeit zu Zeit das Inhaltsverzeichnis des sich füllenden Bandes aktualisieren. Das Programm bedient sich der REAL TIME CLOCK und der MLP-CHECK-ROUTINE von Robert SIP.

Zusätzlich wird aus der Zählerstandstabelle, die entsteht, wenn man die Zählerstände beginnend mit der Null alle 30 Sekunden in die Tabelle einträgt, mittels des aktuellen Sekundenwertes der Programmzählerstand ermittelt. Jeder Programmtitel mit Zählerstand wird auf dem Schirm dargestellt, in den DATA-Bereich geschrieben und am Ende nochmals komplett ausgegeben.

Ablauf des Programms:

Das Programm sucht nach dem Start die Adresse des DATA-Speichers. Nach Aufforderung, die Kasette bei Zählerstand Null zu starten, wird die 'REAL TIME CLOCK' gestartet. Hierzu wird der Vektor bei der Adresse #70+#71 auf das Anwenderprogramm umgehängt. Bei ordnungsgemäßer Programmbeendigung wird der ursprüngliche Vektor zurückgeschrieben und damit die Uhr gestoppt.

Danach wird mit der MLP-CHECK -Routine das Band auf einen Programmvorspanntitel durchsucht. Der gefundene Titel wird zusammen mit dem aus der vergangenen Zeit berechneten Zählerstand ausgegeben und in den DATA-Bereich gepokt. Die Programmbeendigung erfolgt durch drücken einer Taste während ein Programmtitel auf dem Schirm dargestellt wird oder falls die Taste zu spät gedrückt wurde nach der Darstellung des nächsten Programmtitels.

Für jeden Kassettenrecorder muß die Interpolationstabelle überprüft werden. Die Tabelle ist entstanden in dem 62 Zählerstände des Zählwerks im Abstand von 30 Sekunden beginnend mit der Null, notiert wurden. Der erste Wert ist wie gesagt als Null anzunehmen und der letzte Wert für 30 Minuten und 30 Sekunden zu schätzen, da der Kassettenrecorder dann steht. Der 62 Wert kann eventuell entfallen.

Besser wäre natürlich die Tabelle durch eine geschlossene Funktion zu approximieren. Mir ist es leider nicht gelungen. Die beste Näherung habe ich nach dem Verfahren von Gauss mit der Minimierung des Fehlerquadrates erzielt bei der Funktion  $f(t)=a*x + b*\text{SQR}(x)$ .

Programmbedingungen:

Das Programm ist für eine C60-Kasette geschrieben auf der maximal 35 Programme stehen dürfen.

Sollten mehr Programme auf der Kassette erwartet werden, so müssen NZ und der DATA-Bereich am Programmanfang vergrößert werden.

Sinnvolle Änderung des Programms:  
Der Start des Recorders sollte automatisch erfolgen und nicht nach Drücken einer Starttaste. (ich besitze keinen steuerbaren Recorder.)

Programmschwäche:  
Wenn sich das Programm in der CHECK-ROUTINE befindet, wartet es auf einen Programmvorspann und ist nur noch durch die BREAK-Taste abzubrechen, falls das Bandende erreicht ist.

Eine normale Beendigung des Programms erfolgt, wenn während des Ausdrucks des Programmtitels eine Taste gedrückt wird. Dasselbe geschieht, wenn dieser Augenblick verpaßt wurde, danach aber noch ein Programmtitel auf dem Band erkannt wird.

Ungenauigkeiten, die durch einen Korrekturfaktor im Programm berücksichtigt werden müssen, entstehen dadurch, daß die Uhr während der Titel gelesen wird, ausgeschaltet ist.

Schreiben Sie Ihre Verbesserungs- und Änderungsvorschläge an die Redaktion.

Viel Spaß beim Debuggen!?

Manfred Burmeister

CHECK-LIST-DATA

1	DATA & 3	Ø.....
2	DATA & 4	Ø.....
3	DATA & 5	Ø.....
4	DATA & 6	Ø.....
5	DATA & 8	Ø.....
6	DATA &119	Ø.....
7	DATA &120	Ø.....
8	DATA &121	Ø.....
9	DATA &122	Ø.....
10	DATA .123	Ø.....
11	DATA .124	Ø.....
12	DATA .125	Ø.....
13	DATA .126	Ø.....
14	DATA .127	Ø.....
15	DATA .127	Ø.....
16	DATA .128	Ø.....
17	DATA .130	Ø.....
18	DATA .131	Ø.....
19	DATA .132	Ø.....
20	DATA .133	Ø.....
21	DATA .134	Ø.....
22	DATA .134	Ø.....
23	DATA .135	Ø.....
24	DATA .136	Ø.....
25	DATA .137	Ø.....
26	DATA .138	Ø.....
27	DATA .139	Ø.....
28	DATA .139	Ø.....
29	DATA .140	Ø.....
30	DATA .141	Ø.....

```

31 DATA .          0.....
32 DATA .          0.....
33 DATA .          0.....
34 DATA .          .....
35 DATA .          .....

36 REM ENDE DER PROGRAMM TITEL
10000 POKE #29C,3: CLEAR 4000
10001 NZ=35: REM MAXIMALE PROGRAMME
10002 DIM TIT$(NZ): V7L%=PEEK(#70): V7H%=PEEK(#71)
10003 FOR I=1.0 TO NZ: REM DATA-BEREICH UEBERSPRINGEN
10004 READ T$
10005 NEXT I
10010 REM REAL TIME CLOCK - MLP EIN-POKEN
10020 FOR TX=0 TO 11: READ D%
10030 FOR T1%=0 TO 15: READ D1%
10040 IF D1%>=#100 THEN D1%=(PEEK(#2A6) IAND #FE IOR #E)+D1%
      -#100
10050 POKE D%,D1%: D%=D%+1: NEXT: NEXT
10060 DATA #300,#05,#05,#E5,#F5,#21,#B9,#03,#06,#0A,#0E,#06,
      #16,#00,#1E,#32,#34
10070 DATA #310,#7B,#BE,#C2,#57,#03,#72,#23,#34,#78,#BE,#C2,
      #5E,#03,#72,#23,#34
10080 DATA #320,#79,#BE,#C2,#5E,#03,#72,#23,#34,#78,#BE,#C2,
      #5E,#03,#72,#23,#34
10090 DATA #330,#79,#BE,#C2,#5E,#03,#72,#23,#34,#78,#BE,#C2,
      #5E,#03,#72,#23,#34
10100 DATA #340,#2B,#23,#3E,#02,#BE,#C2,#5E,#03,#2B,#3E,#04,
      #BE,#C2,#5E,#03,#36
10110 DATA #350,#00,#23,#36,#00,#03,#5E,#03,#F1,#E1,#D1,#C1,
      #03,#A9,#D9,#3A,#EF
10120 DATA #360,#101,#FE,#7A,#C2,#57,#03,#21,#BA,#03,#7E,#C6,
      #30,#32,#F1,#100,#23
10130 DATA #370,#7E,#C6,#30,#32,#F3,#100,#23,#7E,#C6,#30,#32,
      #F7,#100,#23,#7E,#C6
10140 DATA #380,#30,#32,#F9,#100,#23,#7E,#C6,#30,#32,#FD,#100,
      #23,#7E,#C6,#30,#32
10150 DATA #390,#FF,#100,#3E,#FF,#32,#EC,#100,#32,#EE,#100,#32,
      #F0,#100,#32,#F2,#100
10160 DATA #3A0,#32,#F4,#100,#32,#F6,#100,#32,#F8,#100,#32,#F,
      #A,#100,#32,#FC,#100,#32
10170 DATA #3B0,#FE,#100,#32,#00,#101,#00,#03,#57,#03,#1A,#00,
      #00,#00,#00,#00,#00
10199 REM DATENBEREICH FUER MLP-CHECK
10200 DATA #F5,#05,#05,#E5,#01,#40,#00,#11,#B1,#80,#21,#9E,#E6,
      #CD,#CE,#02,#E1,#D1,#C1,#F1,#C9
10205 REM TABELLE 62 WERTE VON 0. SEKUNDE BIS 1830. SEKUNDE ALLE
      30 SEKUNDEN
10206 REM ZAHLERSTAENDE ALLE 30 SEKUNDEN NOTIEREN
10210 DATA 62,0,12,25,37,48,60,71,82,92,103,113,123
10220 DATA 133,142,152,161,170,179,188,197,205,214,222,230,239,247,255
10230 DATA 263,270,278,286,293,301,308,315,322,329,337,344,351,357,364,371,378,384,391,397,404
10240 DATA 410,417,423,429,436,442,448,454,460,466,472,478,484,490

```

```

10570 REM DATA-BEREICH IM SPEICHER SUCHEN
10580 FOR I=PEEK(#2A0)*256+PEEK(#29F) TO PEEK(#2A4)*256+PEEK
(#2A3)
10590 IF "&"=CHR$(PEEK(I)) THEN IU=IA:IA=I:REM & WIRD GESUCH
T!!!
10592 IF IA-IU=#36 THEN IP=IU+1.0:GOTO 10610:REM ABSTAND MUS
S #36 SEIN !!!
10600 NEXT I
10601 PRINT "ADRESSE DES DATA-BEREICHS NICHT GEFUNDEN",HEX$(
IP):STOP
10602 REM KASSETTEN LISTING
10610 PRINT CHR$(12):AD$=""
10611 REM MPL-CHECK SPEICHERN
10620 FOR A=1.0 TO 21.0:READ B$:AD$=AD$+CHR$(B%):NEXT
10621 REM
10630 AD=PEEK(VARPTR(AD$)) IOR (PEEK(VARPTR(AD$)+1) SHL 8)+1

10631 REM FELD FUER INTERPOLATIONS - RECHNUNG EINLESEN
10640 READ N:DIM T%(N):DIM IT(3.0)
10642 FOR I=0.0 TO N-1.0:READ T%(I):NEXT
10644 CPT=1.0:Z=23.0:TIT$(CPT)=""
10646 PRINT " KASSETTE AUF ANFANG SPULEN , TASTE DRUECKEN"
10647 IF GETC=0.0 THEN 10647:REM START FUER UHR
10648 POKE #71,3:POKE #70,0:REM VEKTOR MIT NEUER STARTADRESS
E
10650 I=0.0:CORSOR 1,23:CALLM AD:REM BAND-TITEL SUCHEN
10651 S=PEEK(#3BA)+10*PEEK(#38B)+60*PEEK(#3BC)+600*PEEK(#3BD
):REM SEKUNDEN BERECHNEN
10652 Z=Z-1.0:GOSUB 10710
10654 CORSOR 1,Z:IF Z=1 THEN Z=23.0
10660 PRINT CPT;"-";IT$:;REM BILDSCHIRMAUSGABE?
10661 REM ZAEHLERSTAND POKEN
10662 FOR J=1.0 TO 3.0
10663 T%=MID$(IT$,J-1,1)
10664 T=ASC(T%)
10665 POKE IP+(CPT-1)*#36+I,T
10666 I=I+1.0:NEXT
10667 FOR A=#BFEB TO #BFA5 STEP -2.0:PRINT CHR$(PEEK(A));;RE
M TITEL AUS 1.ZEILE LESEN
10668 TIT$(CPT)=TIT$(CPT)+CHR$(PEEK(A)):REM ABSCHLUSSTEXT
10669 POKE IP+(CPT-1)*#36+I,PEEK(A):REM TITEL IN DEN EDIT-BU
FFER FUELLEN
10670 I=I+1.0:NEXT
10680 CORSOR 1,23:PRINT "
"
10685 CORSOR 1,23:IF GETC<>0 THEN 10700
10690 CPT=CPT+1.0:TIT$(CPT)=""
10695 IF CPT=N-1.0 THEN 10700:REM ENDE
10696 GOTO 10650:REM SCHLEIFE
10699 REM PROGRAMM ENDE
10700 POKE #70,V7L%:POKE #71,V7H%:REM VEKTOR ZURUECKAENDERN
10701 PRINT CHR$(12):CORSOR 1,22
10702 FOR I=1.0 TO CPT
10703 PRINT TIT$(I):REM AUSDRUCK DER PROGRAMM-TITEL +ZAEHLER

10709 NEXT I:END
10710 REM INTERPOLATION DER ZWISCHENZAEHLERSTAENDE AUS DEN S
EKUNDEN
10720 IF S>=1830.0 THEN PRINT "FELD T% UEBERLAUF, UHR FALSCH
GESTELLT":STOP
10730 ZW=T%(INT(S/30.0))+T%(INT(S/30.0)+1.0)-T%(INT(S/30.0)

```

```
    ))*FRAC(S/30.0)
10740 IZW=IZW+CPT*0.7:REM KORREKTUR FUER REALTIMECLOCK PAUSE

10744 IZW=INT(ZW):IT$=STR$(IZW)
10745 IT$=" "+IT$
10746 IT$=RIGHT$(IT$,5):IT$=LEFT$(IT$,3)
10747 TIT$(CPT)=IT$+" "+TIT$(CPT):REM STRING UM ZAEHLER VERL
AENGERN
10749 RETURN
```

Das Programm ist wie folgt aufgebaut:

300-318 Initialisierung der BASIC-Pointer und Sprungadressen  
 320-338 Test des eingegebenen Zeichens  
 340-362 Auto-Routine mit der Ausgabe der Zeilennummern  
 368-372 Verschiedene Rücksprünge  
 380-391 PRINT CHR\$(12)  
 393-39E Ausgabe von SYNTAX ERROR  
 3A0-3E6 Einlesen der Zahlen hinter 'A'  
 3F0-42D Unterprogramm zum Einlesen einer Zahl  
 430-44D Test einer beendeten Zeile auf 'A', 'K', 'C'  
 450-46C Ausgabe von NUMBER OUT OF RANGE  
 470-4E7 Einlesen der Zahlen und Zeichen hinter 'K'  
 4E8-54D Berechnung der Adressen der zu löschenden Zeilen  
 54E-57F Ändern der BASIC-Pointer und Löschvorgang  
 580-597 Ausgabe UNDEFINED LINE NUMBER

#### V A R I A B L E N

2EC-2EF Arbeitsspeicher für Math.-AKKU  
 2F0 Flag für Autonumber 1=on 0=off  
 2F1-2F2 nächste auszugebende Zeilennummer für Autonumber  
 2F3-2F4 Schrittweite zwischen den Zeilennummern  
 2F5-2F6 erste zu löschende Zeile (Nummer od. deren Adresse)  
 2F7-2F8 letzte zu lö. Zei. (Nr. od. Adr. der nächsten Zeile)  
 2F9-2FA Schleifenvariable (Adr. der untersuchten Zeile)  
 2FB-2FC altes Ende der Symbol Table

### Das Programm wird mit CALLM#300 gestartet. ###

Folgende Eingaben sind möglich:

\*A 10 5

bedeutet: Die Nummerierung startet bei 10 und läuft in Fünferschritten weiter. Beide eingegebenen Zahlen dürfen nicht gleich Null sein. Zwischen den Zahlen muß sich mindestens ein Blank befinden. Blanks können sonst beliebig auftreten.

\*A 100

bedeutet: Die alte Schrittweite bleibt erhalten. Die nächste auszugebene Zeilennummer ist 100.

\*A

schaltet die automatische Zeilennummerierung ab. Eine Zeilennummer wird nach Drücken der SPACE Taste ausgegeben. Dies jedoch nur, wenn der Cursor sich neben dem Prompt am Zeilenanfang befindet.

\*C

bewirkt das Löschen des Bildschirms

\*K 10-30

bewirkt das Löschen der angegebenen Zeilen. Wenn die Zeilennummern nicht im Programm auftreten, wird als untere Grenze die nächstgrößere der Ersten und als obere Grenze die nächstkleinere der Zweiten Zahl angenommen.

\*K 10 löscht nur diese Zeile

\*K -30 löscht bis Zeile 30

\*K 30- löscht ab Zeile 30

Beim Starten des Programms wird ein NEW ausgeführt. Falls schon ein BASIC Programm im Speicher steht, muß man selbst initialisieren. 29B -00 -06; CLEAR #100; R; 2E1 -20 -03; 296 -01; Dann ist das Programm startklar. !!kein CALLM#300!!  
 Falls eine Floppy oder DCR betrieben wird: POKE #321,2  
 Die Speicherstellen #323,#324 müssen auf den ursprünglichen Wert von #2E1,#2E2 gesetzt werden.



Autonumber und Loeschen von BASIC-Zeilen

```

300 E5 F5 21 00 06 22 9B 02 CD B8 DE 21 20 03 22 E1
310 02 3E 01 32 96 02 F1 E1 C9 00 00 00 00 00 00
320 E6 00 C2 00 00 3A 18 01 B7 C2 E2 D1 CD E2 D1 D8
330 FE 20 CA 40 03 FE 0D CA 30 04 A7 C9 00 00 00 00
340 3A F0 02 A7 CA 69 03 E5 EF 0C 2D C2 68 03 C5 D5
350 2A F1 02 CD B4 EF 2A F3 02 EB 2A F1 02 19 22 F1
360 02 D1 C1 00 00 00 00 00 00 E1 3E 20 C3 3A 03 C1 3E
370 0D C3 3A 03 00 00 00 00 00 00 00 00 00 00 00
380 C5 0C CD D2 DD FE 0D C2 E0 03 3E 0C CD 60 DD C3
390 DD 03 00 CD 5E DD 21 23 DC CD D4 DA 06 FF C9 00
3A0 E5 0C CD D2 DD FE 0D C2 B1 03 97 32 F0 02 C3 DD
3B0 03 21 F1 02 22 2B 04 CD F0 03 3E FF B8 CA DD 03
3C0 48 CD E0 DD FE 0D CA DD 03 21 F3 02 22 2B 04 CD
3D0 F0 03 3E FF B8 CA DD 03 3E 01 32 F0 02 CD 5E DD
3E0 3E 0D E1 C1 C3 3A 03 00 00 00 00 00 00 00 00
3F0 41 CD 0D DE D2 93 03 0C CD E0 DD FE 20 CA 0D 04
400 FE 0D CA 0D 04 FE 20 CA 0D 04 C3 F1 03 CD D2 DD
410 79 48 47 CD 24 C0 21 EC 02 E7 0F 2A EE 02 7D 6C
420 67 A7 C2 2A 04 7D A7 CA 50 04 22 F1 02 C9 00 00
430 C5 0E 01 CD D2 DD CD 0D DE DA 6E 03 FE 41 CA A0
440 03 FE 43 CA 00 03 FE 4B CA 70 04 C3 6E 03 00 00
450 CD 5E DD 21 55 DC CD D4 DA 3E 20 CD 60 DD 21 0D
460 DC CD D4 DA 21 DD DC CD D4 DA 06 FF C9 00 00 00
470 E5 0C CD D2 DD CD 0D DE DA A6 04 FE 2D C2 E0 03
480 2A 9F 02 23 22 88 04 2A 01 07 7C 65 6F 22 F5 02
490 0C 21 F7 02 22 2B 04 CD D2 DD CD F0 03 3E FF B8
4A0 CA 75 05 C3 E8 04 21 F5 02 22 2B 04 CD F0 03 3E
4B0 FF B8 CA 75 05 48 CD E0 DD FE 2D CA C7 04 2A F5
4C0 02 22 F7 02 C3 E8 04 0C CD D2 DD FE 0D C2 D9 04
4D0 21 FF FF 22 F7 02 C3 E8 04 21 F7 02 22 2B 04 CD
4E0 F0 03 3E FF B8 CA DD 03 2A 9F 02 22 F9 02 23 22
4F0 F7 04 2A F5 02 EB 2A 1C 07 7C 65 6F CD 14 DE CA
500 15 05 D2 15 05 2A F9 02 7E CD 30 DE 23 7E A7 CA
510 00 05 C3 EB 04 2A F9 02 22 F5 02 23 22 24 05 2A
520 F7 02 EB 2A 2A 07 7C 65 6F CD 14 DE CA 35 05 DA
530 35 05 C3 48 05 2A F9 02 7E CD 30 DE 23 7E A7 22
540 F9 02 CA 48 05 C3 1B 05 2A F9 02 22 F7 02 D5 2A
550 F5 02 EB 2A F7 02 42 4B CD 1A DE EB 2A A1 02 CD
560 1A DE 22 A1 02 2A A3 02 22 FB 02 CD 1A DE 22 A3
570 02 2A F7 02 EB 2A FB 02 23 CD 4F DE D1 C3 DD 03
580 CD 5E DD 21 03 DC CD D4 DA 21 82 DB CD D4 DA 21
590 55 DC CD D4 DA C3 DD 03 00 00 00 00 00 00 00
    
```

```

*****
*
*   geschrieben im Juni 1982
*
*   von Stefan Goller , Meckenheim
*
*****
    
```

# Assembler System

Assembler System ist ein einfacher One-Line-Assembler.  
Es ist fuer den Assembler-Anfänger gedacht.

Es besitzt 3 Moden:

- 1) Auswahl Modus
- 2) Assembler Modus
- 3) Disassembler Modus

Jeder dieser Moden hat eigene Befehle:

## 1) Auswahl Modus:

'E'xit = Programmende  
'A'ssembler = Starte Assembler Modus  
'D'isassembler = Starte Disassembler Modus  
'P'oke = Schreibe in eine Speicherzelle  
'?'Peek = Gebe Inhalt einer Speicherzelle aus  
'C'allm = Rufe Maschinenprogramm auf  
'M'over = Schreibt Programme von einem Speicher-  
bereich in einen anderen um.  
(Alle 3-Byte-Befehle werden angepasst!)  
'H'ex/Dez. Wandler

## 2) Assembler Modus:

'E'xit = Starte Auswahl Modus  
'D'isassembler = Starte Disassembler Modus  
'! <neue Adresse>' = ändern der Anfangsadresse

## 3) Disassembler Modus:

'A'ssembler = Starte Assembler Modus  
'B'reak = Halte an und warte auf Tastendruck  
'E'xit = Starte Auswahl Modus

Anmerkung: Der Disassembler listet automatisch eine Bild-  
schirmseite, hält dann an und wartet auf einen  
Tastendruck. Fügt man zur Startadresse noch ein  
'#' an, so werden neben den Befehlen noch die  
entsprechenden Hex-Codes ausgegeben!

Viel Spass beim 'assemblern'.

Markus Rahlff

### Änderung an NP 10 (Autonumber)

Da das Zeichen 'C' zum Bildschirm löschen verwendet wird, werden Kommandos wie 'CHECK' nicht mehr angenommen. Es ist daher ratsam ein Zeichen zu verwenden, das in keinem anderen Kommando als erstes Zeichen vorkommt. zB.: 'Q'  
also >S442 43-51

Es liegt ein weiterer Fehler vor: >S380 C5-E5

Der Programmstart kann auch noch verändert werden >S389 B8-CA  
Nun kann ein sich bereits im Speicher befindliches Basic-Programm auf eine beliebige über #59F liegende Adresse geschoben werden. Nach dem Laden und Starten des Autonumberprogramms wird #29B (Start Basic) automatisch der richtigen Wert zugewiesen.

!Ein NEW wird beim Starten also nicht mehr ausgeführt!

Als letzte Änderung biete ich noch die Benutzung der DCR an.

>S372 3A-3C

>S3E5 3A-3C

>S33C 00-A7 00-C3 00-8D 00-F0

Ich hoffe, daß das Programm durch diese Änderungen besser benutzbar wird.

Stefan Goller, Meckenheim

```

1 CLEAR 3000:PRINT CHR$(12);" BITTE WARTEN!";:CURSOR CUR
X-1,CURY
5 FOR I=1.0 TO 60.0:CL%=CL%+" ":NEXT
10 DATA @NOP,2LXI B,@STAX B,@INX B,@INR B,@DCR B,1MVI B,@
RLC,-,@DAD B,@LDAX B,@DCX B,@INR C,@DCR C
20 DATA 1MVI C,@RRC,-,2LXI D,@STAX D,@INX D,@INR D,@DCR D
,1MVI D,@RAL,-,@DAD D,@LDAX D,@DCX D,@INR E,@DCR E,1MV
I E
30 DATA @RAR,-,2LXI H,25HLD,@INX H,@INR H,@DCR H,1MVI H,@
DAA,-,@DAD H,2LHLD,@DCX H,@INR L,@DCR L,1MVI L,@CMA,-,
2LXI SP,2STA
40 DATA @INX SP,@INR M,@DCR M,1MVI M,@STC,-,@DAD SP,2LDA,
@DCX SP,@INR A,@DCR A,1MVI A,@CMC
45 DATA @MOV B.B,@MOV B.C,@MOV B.D,@MOV B.E,@MOV B.H
50 DATA @MOV B.L,@MOV B.M,@MOV B.A,@MOV C.B,@MOV C.C,@MOV
C.D,@MOV C.E,@MOV C.H,@MOV C.L,@MOV C.M,@MOV C.A,@MOV
D.B,@MOV D.C
60 DATA @MOV D.D,@MOV D.E,@MOV D.H,@MOV D.L
70 DATA @MOV D.M,@MOV D.A,@MOV E.B,@MOV E.C,@MOV E.D,@MOV
E.E,@MOV E.H,@MOV E.L,@MOV E.M,@MOV E.A,@MOV H.B,@MOV
H.C
80 DATA @MOV H.D,@MOV H.E,@MOV H.H,@MOV H.L,@MOV H.M,@MOV
H.A,@MOV L.B,@MOV L.C,@MOV L.D,@MOV L.E,@MOV L.H
90 DATA @MOV L.L,@MOV L.M,@MOV L.A,@MOV M.B,@MOV M.C,@MOV
M.D,@MOV M.E,@MOV M.H,@MOV M.L,@HLT,@MOV M.A
100 DATA @MOV A.B,@MOV A.C,@MOV A.D,@MOV A.E,@MOV A.H,@MOV
A.L,@MOV A.M,@MOV A.A
110 DATA @ADD B,@ADD C,@ADD D,@ADD E,@ADD H,@ADD L,@ADD M,
@ADD A
120 DATA @ADC B,@ADC C,@ADC D,@ADC E,@ADC H,@ADC L,@ADC M,
@ADC A
130 DATA @SUB B,@SUB C,@SUB D,@SUB E,@SUB H,@SUB L,@SUB M,
@SUB A
140 DATA @SBB B,@SBB C,@SBB D,@SBB E,@SBB H,@SBB L,@SBB M,
@SBB A
150 DATA @ANA B,@ANA C,@ANA D,@ANA E,@ANA H,@ANA L,@ANA M,
@ANA A
160 DATA @XRA B,@XRA C,@XRA D,@XRA E,@XRA H,@XRA L,@XRA M,
@XRA A
170 DATA @ORA B,@ORA C,@ORA D,@ORA E,@ORA H,@ORA L,@ORA M,
@ORA A
180 DATA @CMP B,@CMP C,@CMP D,@CMP E,@CMP H,@CMP L,@CMP M,
@CMP A
190 DATA @RNZ,@POP B,2JNZ,2JMP,2CNZ,@PUSH B,1ADI,@RST @,@R
Z,@RET,2JZ
191 DATA -,2CZ,2CALL,1ACI,@RST 1,@RNC,@POP D,2JNC,1OUT,2CN
C,@PUSH D,1SUI
192 DATA @RST 2,@RC,-,2JC,1IN,2CC,-,1SBI,@RST 3,@RPD,@POP
H,2JPO,@XTHL
193 DATA 2CPD,@PUSH H,1ANI,@RST 4,@RPE,@PCHL,2JPE,@XCHG,2C
PE,-,1XRI,@RST 5
194 DATA @RP,@POP PSW,2JP,@DI,2CF,@PUSH PSW,1ORI,@RST 6,@R
M,@SPHL,2JM
195 DATA @EI,2CM,-,1CPI,@RST 7
210 DIM A$(255.0):FOR I=0.0 TO 255.0:READ A$(I):NEXT
215 PRINT CHR$(12);" Assembler System"
216 PRINT " M.Rahiff":PRINT " 20.7.1981":FOR I=1.0 TO 60.0
:PRINT CHR$(29);:NEXT:PRINT :PRINT :GOTO 1000

```

```

1450 IF POI>BIS THEN PRINT "** Fertig **":PRINT :RETURN
1455 WO$=A$(PEEK(POI))
1460 IF WO$="-" THEN POKE AB1,PEEK(POI):AB1=AB1+1.0:POI=POI
+1.0:GOTO 1450
1470 IF VAL(LEFT$(WO$,1))=0 THEN WO$="-":GOTO 1460
1480 IF VAL(LEFT$(WO$,1))=1 THEN POKE AB1,PEEK(POI):POKE AB
1+1,PEEK(POI+1):AB1=AB1+2.0:POI=POI+2.0:GOTO 1450
1490 IF VAL(LEFT$(WO$,1))<>2 THEN PRINT "**ERROR IN BASICPR
OGRAMM **":RETURN
1500 POKE AB1,PEEK(POI):AB1=AB1+1.0:POI=POI+1.0
1510 ADR=PEEK(POI)+PEEK(POI+1)*256
1520 IF ADR<BIS AND ADR>AB THEN ADR=ADR+DIFF
1530 HIGH=INT(ADR/256.0):LOW=ADR-HIGH*256.0
1540 POKE AB1,LOW:POKE AB1+1,HIGH:AB1=AB1+2.0:POI=POI+2.0:G
OTO 1450
3000 FOR I=#C000 TO #DFFF
3010 IF PEEK(I)<>#2A OR PEEK(I+1)<>#72 OR PEEK(I+2)<>0 THEN
NEXT
3020 PRINT I:END

```

Ändere Zeile 1520 :

```
IF ADR<=BIS AND ADR>=VON THEN ADR=ADR+DIFF
```

Füge hinzu Zeile 415 :

```
IF LE=5 AND LEFT$(N$,3)="#RST THEN P$=N$:GOTO 450
```

002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045 0300 C5  
046 0301 23  
047 0302 23  
048 0303 E5  
049 0304 C1  
050 0305 CDEDE6  
051  
052 0308 E5  
053 0309 CD39DE  
054 030C E3  
055 030D 7E  
056 030E 23  
057 030F 23  
058 0310 23  
059 0311 E5  
060  
061 0312 DE04  
062  
063

```

*****
**
**      Funktions-Input fuer BASIC-Programme:
**      =====
**
** Dies Maschinenprogramm dient dazu, eine beliebige Funktion via INPUT in ein Basicprogramm einzufuegen.
**
** Dazu muß eine moeglichst lange Zeile freigehalten werden (z.B. REM oder ***), die im Programm als Unterprogramm zur Funktionswertberechnung aufgerufen wird.
**
** Die Zeilennummer wird als Parameter in einer Integer-Variablen an das MLP uebergeben. Z.B.: LINE%=1000:CALLM xxx,LINE%
** Danach erscheint ein Prompt "?" und man gibt nun die Funktion ein, z.B.: Y=SIN(2*X)
** Bei einem Syntaxfehler oder anderem kommt die Fehlermeldung und man kann erneut eingeben.
**
** War die Eingabe korrekt, wird in die Zeile LINE% die Funktion eingefuegt und ein RETURN angehaengt. Der Rest der Zeile wird mit REM aufgefuellt.
**
*****

```

```

GETLIN EQU :E6ED      SUCHE ANF. ZEILE (BC)
ADDMHL EQU :DE39      HL := HL+(HL)+1
FLDEHL EQU :DE7C      FILL (DE)..(HL-1) MIT A
INPFLG EQU :117
INPUT EQU :DD1F       INPUT LINE MIT PROMPT IN A
                        RÜCKKEHR BEI BREAK ODER RET
MOVE EQU :DE4F        BEWEGE BLOCK
SBHLDE EQU :DE1A      HL := HL-DE
BUFBEG EQU :013E      ZEIGER AUF ANFANG COMP.-BUF
LAENGE EQU :0140      L. DER COMP. FUNKTION
ORG :0300             BELIEBIG

FINPUT PUSH B         NUR BC RETTEN REICHT AUS
        INX H
        INX H         HL ZEIGT JETZT AUF ZNR.
        PUSH H
        POP B         FÜR FOLGENDE ROUTINE IN BC
        CALL GETLIN  HL ZEIGT NUN AUF DAS ERSTE
                        BYTE DER ZEILE ZNR.
        PUSH H        AUFBEWAHREN
        CALL ADDMHL   HL NUN ENDE DER ZEILE +1
        XTHL          AUFBEWAHREN U. ANF. ZURÜCK
        MOV A,M       ZEILENLÄNGE
        INX H         ZEILENNUMMER ÜBER-
        INX H         SPRINGEN
        INX H
        PUSH H        BEGINN DES EIGENTLICHEN
                        BEFEHLSTEIL DER ZEILE
0312 SBI 4           LÄNGE-4 KANN FÜR FUNKTION
                        BENUTZT WERDEN, REST
                        FÜR 'RETURN' U. 'REM'

```

064 0314 21BE01	LXI	H, :1BE	ENDE DES INPUT-BUFFERS+1
065 0317 5F	MOV	E, A	BERECHNE HL-A,
066 0318 1600	MVI	D, 0	ALSO ANFANG BUFFER
067 031A CD1ADE	CALL	SBHLDE	
068 031D 223E01	SHLD	BUFBEG	FÜR COMP-ROUTINE MERKEN
069 0320 E5	PUSH	H	UND FÜR MOVE
070 0321 3EFF	MVI	A, :FF	
071 0323 321701	STA	INPFLG	INPUT-FLAG SETZEN
072 0326 3E3F	MVI	A, ' ? '	PROMPT
073 0328 CD1FDD	CALL	INPUT	LINE-INPUT
074 032B 2A3E01	LHLD	BUFBEG	
075 032E 1640	MVI	D, :40	FLAG FÜR PROGRAMMZEILE
076 0330 CF	RST	1	COMPILE
077 0331 00	DATA	:00	
078 0332 DA2603	JC	ERROR	
079			
080			HL ZEIGT NUN AUF ENDE
081 0335 D1	POP	D	COMPILIERTER F. +1
082 0336 E5	PUSH	H	BUFBEG
083 0337 CD1ADE	CALL	SBHLDE	HL:=HL-DE
084 033A 224001	SHLD	LAENGE	ANZ BYTES DER FUNKTION
085 033D E1	POP	H	
086 033E C1	POP	B	ANFANG IN BASIC-ZEILE
087 033F C5	PUSH	B	
088 0340 CD4FDE	CALL	MOVE	AUS BUFFER IN ZEILE
089 0343 C1	POP	B	
090 0344 2A4001	LHLD	LAENGE	
091 0347 09	DAD	B	HL NUN HINTER FUNKTION
092			----- EVTL. AUSLASSEN -----
093 0348 3686	MVI	M, :86	'RETURN'-CODE
094 034A 23	INX	H	
095			
096 034B 36A9	MVI	M, :A9	'REM'-CODE
097 034D 23	INX	H	
098 034E EB	XCHG		
099 034F E1	POP	H	ANFANG NÄCHSTE ZEILE
100 0350 E5	PUSH	H	
101 0351 CD1ADE	CALL	SBHLDE	HL:=HL-DE
02			ALSO WIEVIEL PLATZ NOCH
103 0354 7D	MOV	A, L	
104 0355 3D	DCR	A	-1 FÜR LANGENBYTE
105 0356 12	STAX	D	ZUERST LANGE REM
106 0357 3E2D	MVI	A, ' - '	IRGENDWAS ALS REM
107 0359 13	INX	D	
108 035A E1	POP	H	
109 035B CD7CDE	CALL	FLDEHL	FÜLLE REST DER ZEILE
110 035E AF	XRA	A	
111 035F 321701	STA	INPFLG	INPUT-FLAG RESET
112 0362 C1	POP	B	REGISTER RESTORE
113 0363 C9	RET		ZURÜCK ZUM BASIC
114 0364	END		

\*\*\*\*\*  
 \* S Y M B O L T A B L E \*  
 \*\*\*\*\*

ADDMHL DE39	BUFBEG 013E	ERROR 0326	FINPUT 0300
LDEHL DE7C	GETLIN E6ED	INPFLG 0117	INPUT DD1F
LAENGE 0140	MOVE DE4F	SBHLDE DE1A	

```

59000 REM --- EINLESEN MLP INPUT ---
59010 CLEAR #200
59020 FINPUT=PEEK(#29B)+256*PEEK(#29C)
59030 POKE #29C,PEEK(#29C)+1:POKE #29E,#1
59040 FOR ADR=FINPUT TO FINPUT+#63:READ BYTE
      :POKE ADR,BYTE:NEXT
59050 POKE FINPUT+#33,(FINPUT+#26) IAND #FF
59060 POKE FINPUT+#34,(FINPUT+#26) SHR 8
59070 RETURN
60000 DATA #C5,#23,#23,#E5,#C1,#CD,#ED,#E6
60010 DATA #E5,#CD,#39,#DE,#E3,#7E,#23,#23
60020 DATA #23,#E5,#DE,#04,#21,#BE,#01,#5F
60030 DATA #16,#00,#CD,#1A,#DE,#22,#3E,#01
60040 DATA #E5,#3E,#FF,#32,#17,#01,#3E,#3F
60050 DATA #CD,#1F,#DD,#2A,#3E,#01,#16,#40
60060 DATA #CF,#00,#DA,#00,#00,#D1,#E5,#CD
60070 DATA #1A,#DE,#22,#40,#01,#E1,#C1,#C5
60080 DATA #CD,#4F,#DE,#C1,#2A,#40,#01,#09
60090 DATA #36,#86,#23,#36,#A9,#23,#EB,#E1
60100 DATA #E5,#CD,#1A,#DE,#7D,#3D,#12,#3E
60110 DATA #2D,#13,#E1,#CD,#7C,#DE,#AF,#32
60120 DATA #17,#01,#C1,#C9

```



## SERVO-ANSTEUERUNG MIT DEM DCE-BUS

Von Herrn Willi Sicking, Jägerweg 31, 4423 Gescher erreichte uns folgende Anregung zum gleichzeitigen Ansteuern von bis zu acht Rudermaschinen (auch Servos genannt):

Siehe dazu auch CHIP 8/81 Seite 47ff!

Die Servos werden an Port C des 8255 (ausführliche Beschreibung in einer der nächsten Ausgaben) angeschlossen. Graupner-Servos müssen allerdings über einen Inverter angesteuert werden, da sie mit negativem Signal arbeiten (man könnte die Software natürlich auch entsprechend ändern).

Man kann die Servos z.B. direkt an Masse und 5V des DCE-Bus anschliessen und das Steuersignal von einem Bit des Port C entnehmen, es wird jedoch dringend vor einer sehr leicht möglichen Überlastung der Stromversorgung gewarnt!!!!

Modellbauservos werden mit Impulsen angesteuert, deren Länge den Ruderausschlag bestimmt. Ist z.B. für die Mittelstellung ein Impuls von 1.5msec Länge nötig (ausprobieren), so schlägt der Servo bei einem kürzeren Impuls nach links, bei einem längeren nach rechts aus. Diese Impulse müssen in regelmäßigen Abständen an die Servos ausgegeben werden um sie aufzufrischen. Bei Fernsteuerungen beträgt das Intervall 50msec, dieser Wert ist jedoch sehr unkritisch.

Die Impulse werden durch Setzen und Rücksetzen von Einzelbits des Port C erzeugt und durch eine Grundwarteschleife und eine variable Schleife terminiert.

Maschinenprogramm zum Ansteuern:

```
LXI H,300H ;ADRESSE #300 ENTHALT DEN STEUERWERT
DI ;NÖTIG ZUR GENAUEN ZEITEINHALTUNG
MVA 301H ;ADRESSE #301 ENTHALT DIE SERVO-NUMMER, SIE
;IST =(BIT SHL 1)+1, BIT IST DAS BENUTZTE
;VON PORT C (0..7)

MOV E,M
STA :FE03 ;KONTROLLWORT, SETZE BIT X,PORT C
W: DCR E ;WARTESCHLEIFE JE NACH WERT IN #300
JNZ W
MVI E,100 ;ENTSPRICHT ETWA 1MSEC
G: DCR E ;GRUNDWARTESCHLEIFE
JNZ G
DCR A
STA :FE03 ;KONTROLLWORT LÖSCHE BIT X,PORT C
EI
RET ;ZURÜCK ZUM BASIC
```

Beispielprogramm BASIC:

```
5 CLEAR 1000:REM JETZT AUF KEINEN FALL HEAP BENUTZEN!
10 POKE #FE03,#90:ALLE PORTS AUF OUTPUT
20 FOR X=#302 TO #31C:READ C:POKE X,C:NEXT X
60 DATA #21,#00,#03,#F3,#3A
70 DATA #01,#03,#5E,#32,#03,#FE,#1D,#C2
90 DATA #0D,#03,#1E
100 DATA 96 (muß je nach Servo angepaßt werden!)
110 DATA #1D,#C2,#13,#03,#3D,#32,#03,#FE,#FB,#C9
140 B=1:REM KONTROLLBYTE FÜR BIT 0
150 A=PDL(0)
160 POKE #300,A
170 POKE #301,B
180 CALLM #302:REM STELLE SERVO JE NACH PDL-WERT
190 GOTO 150
```

Mit diesem Programm lassen sich natürlich auch acht Servos nacheinander ansteuern.

Angeregt sei noch, daß man die Auffrischung auch über Interrupts steuern könnte. Wir würden uns freuen, gelegentlich Erfahrungsberichte über dieses Thema zu bekommen.

```

1      REM IMP INT
50000 REM =====
50010 REM =      PROGRAMM FÜR WAHLFREIEN DATEIZUGRIFF      =
50020 REM =====
50030 REM =      (C) Bernd Preusing                        29.8.82      =
50040 REM =====
50050 REM Mit den hier vorgestellten, auch im TINYDOS
50060 REM verwendbaren Routinen, kann auf bis zu fuenf
50070 REM gleichzeitig offene Files wahlfrei zugegriffen
50080 REM werden.
50090 REM Dabei koennen jeweils 128 Byte aus einem be-
50100 REM liebigen Record des Files (Zaehlung beginnt
50110 REM bei 1) an eine anzugebende RAM-Adressse gela-
50120 REM den werden. Schreiben ist natuerlich genauso
50130 REM moeglich.
50140 REM Bei Benutzung dieses Programms sollten keine
50150 REM Routinen des $MSTRDOS aufgerufen werden, ins-
50160 REM besondere nicht OPENI o.ae. Nach einer Verwen-
50170 REM dung von SAVEA,LOADA etc. muß immer CALLM
50180 REM RESETD% erfolgen, da danach alle Files ge-
50190 REM schlossen sind.
50200 REM =====
50210 REM Verwendung der Routinen:
50220 REM CALLM RESETD% schliesst alle Files
50230 REM CALLM CLOSE%,FILE$ schliesst den File FILE$
50240 REM . Die Files duerfen auf keinen Fall anders
50250 REM . geschlossen werden!
50260 REM CALLM OPEN%,FILE$ oeffnet den File FILE$
50270 REM CALLM GETREC%,FARM%(0) liest einen Sektor
50280 REM CALLM PUTREC%,FARM%(0) schreibt einen Sektor
50300 REM FILE$ muss immer folgendes Format haben:
50310 REM FILE$="I0FILENAMEEXT", d.h. der File ist oder
50320 REM wird fuer Input geoeffnet, ist in Laufwerk 0
50330 REM und hat den Namen 'FILENAME' und die Spezifi-
50340 REM kation 'EXT'. LEN(FILE$) ist also immer 13,
50350 REM d.h. er muß ggf. mit Blanks aufgefuellt werden.
50360 REM Beispiel: FILE$="01TEST" " (Output, LW1)
50370 REM PARM%(1) muß DIM (2) sein und enthalten:
50380 REM PARM%(0)=VARPTR(FILE$)
50390 REM PARM%(1)=RECNR% ;Nummer des Records (>=1)
50400 REM PARM%(2)=RAMADR% ;Adresse wohin bzw. woher
50410 REM =====
50999 REM Kleines Syntaxdemo, soll nicht laufen.
51000 CLEAR 10000
51010 GOSUB 59000:REM MLP BEREITSTELLEN
51020 DIM PARM1(2),PARM2(2),BUFFER(200)
51030 POKE #9B7,0:CALLM RESETD:REM ZU BEGINN IMMER NOETIG!
51040 FN1$="I0DATEN1 DAT":FN2$="00DATEN1 DAT"
51060 CALLM OPEN,FN1$:GOSUB 58000:REM OEFFNE DATEN.DAT:0 FUER INPUT
51070 CALLM OPEN,FN2$:GOSUB 58000:REM OEFFNE DATEN.DAT:0 AUCH FUER OUT
PUT
51080 PARM1(0)=VARPTR(FN1%):PARM1(2)=VARPTR(BUFFER(0))
51090 PARM2(0)=VARPTR(FN2%):PARM2(2)=PARM1(2)
51100 INPUT "RECORDNR":RECNR:PRINT :PARM1(1)=RECNR:PARM2(1)=RECNR
51110 CALLM GETREC,PARM1(0,0):GOSUB 58000
51130 REM . HIER RECORD BEARBEITEN (POKE,PEEK)...
51140 CALLM PUTREC,PARM2(0):GOSUB 58000
51150 CALLM CLOSE,FN1$:CALLM CLOSE,FN2$
51160 END
38000 REM -----
58010 REM - CHECK FOR DISK ERRORS -
58020 REM -----
58030 ERROR=PEEK(#9B7):IF ERROR=0 THEN RETURN:REM NO ERROR
58040 PRINT :POKE #9B7,0:REM CLEAR

```

```

58050 IF ERROR=#11 THEN PRINT "INVALID RECORD NUMBER"
58060 IF ERROR=#10 THEN PRINT "TOO MANY OPEN FILES"
58070 IF ERROR=#6 THEN PRINT "INVALID ACCESS MODE (I/O)"
58080 PRINT "PRESS SPACEBAR TO CONTINUE":CALLM #D6DA
58090 RETURN
59000 REM -----
59001 REM -- READ MLP AND INITIALIZE VARS --
59002 REM -----
59010 READ NRBYTES:DIM MLP((NRBYTES+5*14)/4)
59020 GETREC=VARPTR(MLP(0))
59030 SUM=0:READ CHKSUM
59040 FOR ADR=GETREC TO GETREC+NRBYTES-1:READ BYTE:POKE ADR,BYTE:SUM=SUM+BYTE:NEXT
59050 IF SUM<>CHKSUM THEN PRINT :PRINT "BAD CHECKSUM IN DATA":END
59060 READ NREL
59070 FOR RELNR=1 TO NREL
59080 READ RELADR
59090 ORG400ADR=PEEK(GETREC+RELADR)+PEEK(GETREC+RELADR+1)*256
59100 NWADR=ORG400ADR+GETREC-#400
59110 POKE GETREC+RELADR,NWADR IAND #FF:POKE GETREC+RELADR+1,NWADR SHR
      8
59120 NEXT RELNR
59130 PUTREC=GETREC+#27:CLOSE=GETREC+#83
59140 OPEN=GETREC+#9E:RESETD=GETREC+#E1
59150 CALLM RESETD:POKE #9B7,0:REM CLEAR FILES AND ERRORS
59160 RETURN
59999 DATA #F8,#5C31
60000 DATA #C5,#CD,#09,#04,#C4,#CE,#05,#C1
60010 DATA #C9,#CD,#56,#04,#C2,#50,#04,#C1
60020 DATA #21,#3E,#04,#16,#58,#E3,#CD,#3D
60030 DATA #04,#C2,#25,#04,#CD,#3A,#05,#0E
60040 DATA #01,#11,#80,#00,#C9,#E1,#C9,#C5
60050 DATA #CD,#2E,#04,#C3,#04,#04,#CD,#56
60060 DATA #04,#C2,#50,#04,#C1,#21,#23,#04
60070 DATA #16,#59,#C3,#15,#04,#CD,#6F,#04
60080 DATA #72,#23,#77,#23,#70,#23,#71,#11
60090 DATA #04,#00,#21,#80,#09,#C3,#94,#04
60100 DATA #C1,#E1,#3E,#09,#B7,#C9,#23,#23
60110 DATA #7E,#23,#E5,#6E,#67,#5E,#23,#56
60120 DATA #E1,#23,#23,#23,#46,#23,#4E,#23
60130 DATA #23,#23,#7E,#23,#6E,#67,#E3,#C5
60140 DATA #E5,#01,#F2,#FF,#21,#30,#05,#3E
60150 DATA #04,#CD,#21,#D1,#C8,#09,#3D,#F8
60160 DATA #C3,#79,#04,#C5,#CD,#8A,#04,#C3
60170 DATA #04,#04,#5E,#23,#56,#CD,#71,#04
60180 DATA #C2,#52,#04,#23,#36,#00,#CD,#6F
60190 DATA #04,#36,#46,#C3,#8E,#04,#C5,#CD
60200 DATA #A5,#04,#C3,#04,#04,#5E,#23,#56
60210 DATA #CD,#71,#04,#C8,#D5,#21,#F9,#04
60220 DATA #01,#0E,#00,#16,#05,#AF,#BE,#CA
60230 DATA #C4,#04,#09,#15,#C2,#B6,#04,#D1
60240 DATA #3E,#10,#B7,#C9,#D1,#13,#E5,#D5
60250 DATA #06,#0D,#1A,#77,#13,#23,#05,#C2
60260 DATA #CA,#04,#E1,#46,#23,#7E,#23,#1E
60270 DATA #0B,#CD,#47,#04,#E1,#C8,#36,#00
60280 DATA #C9,#C5,#CD,#77,#04,#06,#05,#21
60290 DATA #F8,#04,#11,#0D,#00,#73,#23,#36
60300 DATA #00,#19,#05,#C2,#ED,#04,#C1,#C9
60310 REM REL. ADDRESSES
60320 DATA 25
60330 DATA #02,#0A,#0D,#17,#1A,#29,#2C,#2F
60340 DATA #32,#3B,#75,#81,#85,#88,#8E,#91
60350 DATA #A0,#A3,#A9,#AE,#B8,#BD,#D0,#EB
60360 DATA #F4,THIS IS NEVER READ

```

```

1      GOTO 100:REM TERMINAL-PROGRAMM K.HOFFMANN 18.07.82
2      POKE P,V IAND 255:POKE P+1,V SHR 8:RETURN
20     BUFST=#F00:BUFL=#1000:RETURN
100    POKE #2E0,#C3:P=#2E1:V=#992:GOSUB 2
105    CLEAR 2000:DIM A(255):IF PEEK(#29B)+(PEEK(#29C) SHL 8)
<#9B5 THEN LOADA A :CALLM VARPTR(A(0))
110    GOSUB 20:V=BUFST+BUFL:P=#29B:GOSUB 2:CLEAR 10000:POKE
#131,1
120    GOSUB 20:P=#A2:V=BUFST:GOSUB 2:P=#A6:V=BUFST+BUFL:GOSU
B 2
130    CLEAR 10000:DIM IN$(255):PRINT CHR$(12);"Terminal,Edit
,Load,Save,Rewind,Print,Delete old data ?"
140    INPUT A$:PRINT :IF A$="E" THEN POKE #29B,0:CALLM #94C:
GOTO 130
150    IF A$<>"L" THEN 190
160    INPUT "FILENAME ";FN$:LOADA IN$ FN$:POKE #131,2:V=PEEK
(#A4)+(PEEK(#A5) SHL 8)-2:P=#A4:GOSUB 2
170    FOR J=0 TO 255:PRINT IN$(J):IF IN$(J)<>" " THEN NEXT
180    PRINT CHR$(0);:POKE #131,1:GOTO 130
190    IF A$="R" THEN CALLM 768:GOTO 130
200    IF A$="D" THEN GOSUB 20:P=BUFST:V=0:GOSUB 2:P=#A4:V=BU
FST+2:GOSUB 2:GOTO 130
202    IF A$<>"S" THEN 212
204    INPUT "FILENAME ";FN$:IF FN$="" THEN 204
208    POKE #296,1:J=0:FOR I=1 TO (PEEK(#903) SHL 8)+PEEK(902
):A=GETC:IN$(J)=IN$(J)+CHR$(A):IF A=13 THEN J=J+1
210    NEXT:SAVEA IN$ FN$:POKE #296,0
211    GOTO 130
212    IF A$<>"T" THEN 130
215    POKE #29B,0:POKE #FFF5,#84:POKE #131,0:V=(PEEK(#A5) SH
L 8)+PEEK(A4)-1:P=#A4:GOSUB 2
217    CALLM #904:POKE #131,1:GOTO 130
220    POKE #296,1:POKE #131,3:FOR I=1 TO (PEEK(#903) SHL 8)+
PEEK(902):PRINT CHR$(GETC);:NEXT:POKE #131,1
230    GOTO 130

```

\*\*\*\*\*  
 \* M O D E 8 \*  
 \*\*\*\*\*

Das Unterprogramm "MODE 8" dient dazu, auf dem DAI 48K eine Vergrößerung der graphischen Auflösungsfähigkeit, bei, mit dem Basic-Interpreter vergleichbarem Komfort, zu erreichen. Dies wird allein durch geeignete Software erreicht und erfordert keine weiteren Eingriffe.

Zu den Daten des Programms :

der Bildschirm wird in 243 Zeilen zu 528 Bildpunkten aufgeteilt, wodurch sich 128304 einzeln ansprechbare Bildpunkte ergeben. Dies entspricht einer Steigerung der Auflösung gegenüber Mode 6 um 50%. Jedoch ist die Bildschirmaufteilung nicht mehr quadratisch, das heißt, Kreise werden nicht mehr rund. Es liegt der Vierfarbenmodus vor. Ein MODE 8A existiert nicht, es darf also, wenn ein MODE 8-Bild auf dem Schirm ist kein Text erscheinen. Weiterhin stehen 2 Routinen zum Setzen von Bildpunkten und zum Zeichnen von Linien zur Verfügung. Das Löschen des Bildschirms und das Umschalten auf andere MODI kann wie bekannt vorgenommen werden. Die Geschwindigkeit des Programms ist, da es sich um ein Basic-Programm handelt, wesentlich geringer als die der interpretierbaren Programme. Befindet man sich in MODE 8, so stehen noch 14K für Programme Strings und Arrays zur Verfügung. Die MODE 8-Graphik befindet sich im oberen Speicherbereich ab #4047. Die im Programm benutzten Variablen sind :

INT: I , A , B , A1 , A2 , B1 , B2 , C , BY , BI  
 FPT: B1 , B2 , ST

Zur Bedienung des Programms :

Das Unterprogramm wird zu dem gegebenen Programm hinzugefügt (siehe MERGE, deutsches Handbuch). Umgeschaltet auf MODE 8, wird mit >GOSUB 50000< Anweisung. Die Farben müssen also vorher gewählt werden.

Einen Punkt setzt man, indem man die X-Koordinate in die Variable A% einschreibt, und die Y-Koordinate in die Variable B% (  $0 \leq X \leq 527$  ,  $0 \leq Y \leq 242$  ).

Die Farbe wird gewählt, indem man in die Variable C% einen Wert von 1 bis 4 einschreibt, entsprechend der 1. bis 4. Farbe die in der letzten >COLOR%<Anweisung stand. Mit einem >GOSUB 50200< wird der Punkt dann gesetzt.

Eine Linie wird gezeichnet, indem man die Anfangs- und Endkoordinaten in die Variablen A1%, B1% A2%, B2% einschreibt. Wahl der Farbe siehe oben. Nach einem >GOSUB 50400< wird die Linie dann gezeichnet.

Das Programm sollte immer in einer Endlosschleife enden, damit ein Zerstören der Graphik durch die Textzeilen vermieden wird.

Reinhard Kietzmann  
 1000 Berlin

```

IMPINT:LIST
50000 REM *** UNTERPROGRAMM "MODE 8" ****
50020 REM *** von Reinhard Kietzmann, Berlin * 9.12.1981 *
50040 MODE 0:I=FREE:IF I<29400 THEN PRINT :PRINT "OUT OF MEMORY FOR MODE 8":GOTO 65000
50060 FOR I=16500 TO 26500:POKE I,0:NEXT MODE 6
50080 FOR I=49135 TO 26095 STEP -90:POKE I,0:POKE I-1,0:NEXT
50100 POKE 26090,0:POKE 26091,0:POKE 26082,0:POKE 26083,0
50120 FOR I=49135 TO 16707 STEP -134:POKE I,48:POKE I-1,64:NEXT
50140 RETURN
50200 REM *** PUNKT ZEICHEN ****
50220 IF A<0 OR A>527 OR B<0 OR B>242 THEN MODE 0:PRINT "OFF SCREEN":GOTO 65000
50240 IF C<1 OR C>4 THEN MODE 0:PRINT :PRINT "COLOR NOT AVAILABLE":GOTO 65000
50260 BY=16705+INT(134*(A/9)):BI=A MOD 9
50280 ON C GOTO 50300,50320,50340,50360
50300 POKE BY,PEEK(BY) IAND (255-(128 SHR BI)):POKE BY-1,PEEK(BY-1) IAND (255-(128 SHR BI)):RETU
RN
50320 POKE BY,PEEK(BY) IAND (255-(128 SHR BI)):POKE BY-1,PEEK(BY-1) IOR (128 SHR BI):RETURN
50340 POKE BY,PEEK(BY) IOR (128 SHR BI):POKE BY-1,PEEK(BY-1) IAND (255-(128 SHR BI)):RETURN
50360 POKE BY,PEEK(BY) IOR (128 SHR BI):POKE BY-1,PEEK(BY-1) IOR (128 SHR BI):RETURN
50400 REM *** LINIE ZEICHEN ****
50420 IF A1<0 OR A1>527 OR A2<0 OR A2>527 OR B1<0 OR B1>242 OR B2<0 OR B2>242 THEN MODE 0:PRINT
:PRINT "OFF SCREEN":GOTO 65000
50440 IF C<1 OR C>4 THEN MODE 0:PRINT :PRINT "COLOR NOT AVAILABLE":GOTO 65000
50460 IF A2-A1=0 AND B2-B1=0 THEN A=A1:B=B1:GOTO 50260
50470 IF A2-A1=0 THEN ST1=99999999:GOTO 50490
50480 B11=B1:B21=B2:ST1=(B21-B11)/(A2-A1)
50490 IF ST1>1 OR ST1<(-1) THEN 50520
50500 FOR A=A1 TO A2 STEP INT(500/(A2-A1))+0.5:(B=B1+(A-A1)*ST1+0.5):GOSUB 50260:NEXT:RETURN
50520 FOR B=B1 TO B2 STEP INT(500/(B2-B1))+0.5:(A=A1+(B-B1)*ST1+0.5):GOSUB 50260:NEXT:RETURN
65000 REM *** UNTERPROGRAMME ****

```

Im Handel sind kleine, billige Geraete erhaeltlich, die diese Uebungen ebenfalls ausfuehren. Eine solche Applikation auf dem DAI mag deshalb fraglich erscheinen. Die Moeglichkeiten der Darstellung und der Kontrolle sind hier jedoch weit besser. Das Programm soll aber auch ein Beispiel fuer die Bildschirmformatierung und die Zeitmessung mittels 8080-Code sein.

Der Ablauf:

Nach Eingabe von RUN wird das 1X1-Menue in grosser Schrift angezeigt. Die fuer die Uebung gewuenschten Reihen selektiert man wie folgt:

Zeichentaste	:	Reihe selektiert
Space-Bar	:	Reihe nicht selektiert
Return	:	Ende der Eingabe, Uebung beginnt.

Man erscheint die erste Aufgabe. Die Anzeige der Aufgabe erfolgt in grosser Schrift und als gruenes Rechteck. Ab jetzt wird intern die benoetigte Antwortzeit des Schuelers gemessen. Das Resultat ist wie folgt einzugeben:

Zahlentasten	:	0 bis 9
Cursortaste ←	:	Loeschung einer falschen Eingabe
Return	:	Resultat uebergeben

Nach Return wird die Zeit gestoppt. Unterhalb der Aufgabe erscheint unter Pfeifton oder Brumnton das Wort 'richtig' oder 'falsch'. Die gemessene Antwortzeit erscheint oben als roter Balken, aufwaerts bei richtiger, abwaerts bei falscher Antwort. Nach einer kurzen Pause wird die naechste Aufgabe gestellt.

Das Programm besteht aus 2 Teile

BASIC	1X1-Ablauf
8080	Gross-Schrift und Zeitmessung

Die Adressen des Maschinenprogramms sind so gewaehlt, dass das 1X1-Programm noch erweiterbar ist, aber kein Konflikt mit Editor oder Graphic-Memory entsteht. Das Programm ist fuer die 48k-Version gedacht.

Die Handhabung der allgemein verwendbaren Routinen fuer die grosse Schrift und die Zeitmessung werden im Kapitel SYS3000 erkluert.

BILDSCHIRM-BESCHRIFTUNG FUER GROSSE SCHRIFT

4 OBER 22 ZEILEN ZU 23 ZEICHEN JE NACH MODE A. KEYSRE / JAN. 81

3000 05 05 15 FS P23 PUSH B,D,H,PSW ENTRY FOR CURSOR  
 3E 07 MVI A,#07 GET LINEUMBER FROM  
 23 INX H PARAMETERLIST  
 20 DCR A  
 C2 06 30 INZ PARA  
 7E MOV A,M  
 22 00 30 SHLD #30B0 PARAMETERLIST  
 21 00 00 LXI H,XXXX XXXX IS SET BY M23, POS 0,0  
 01 00 00 LXI B,#30 LINE-INCREMNT  
 FE 00 CPI A,0  
 CA 0F 30 JZ F0SC0L JUMP IF POSITION ON LINE 0

300A 09 NEXTLINE 0A0 B  
 30 0CR A  
 C2 DA 30 JNZ NEXTLINE  
 300F 22 B2 30 POSC0L SHLD #30B2 CURSOR-POINTER  
 2A E0 30 LHLD #30B0 PARAMETERLIST  
 28 DCX H  
 29 DCX H  
 2B DCX H  
 28 DCX H  
 7E MOV A,M  
 2A E2 30 LHLD #30B2  
 FE 00 CPI A,0  
 CA F8 30 JNZ TEXT  
 30F2 28 COL  
 DCX H  
 DCX H  
 DCR A  
 C2 F2 30 JNZ COL

30F8 22 B2 30 TEXT SHLD #30B2 CURSOR-POINTER  
 2A E0 30 LHLD #30B0 PARAMETERLIST  
 23 INX H  
 23 INX H  
 23 INX H  
 56 MOV D,M  
 23 INX H  
 5E MOV E,M  
 EB XCHG  
 5E MOV E,M  
 23 INX H  
 56 MOV D,M  
 EB XCHG  
 22 BA 30 SHLD #30B4 TEXTSTRING-POINTER  
 4E MOV C,M  
 3E 00 MVI A,0  
 09 CMP A,C  
 CA 31 31 JZ ENDE  
 23 INX H  
 EB XCHG  
 2A E2 30 LHLD #30B2 CURSOR-POINTER

BILDSCHIRM-BESCHRIFTUNG FUER GROSSE SCHRIFT

FORTSETZUNG A. KEYSRE / JAN. 81

3118 EB NEXTCHAR XCHG  
 46 MOV B,M  
 23 INX H  
 EB XCHG  
 7E MOV A,M  
 FE 5A CPI HSA  
 C2 2A 31 JNZ OK  
 28 DCX H  
 7E MOV A,M  
 23 INX H  
 FE 40 CPI #40  
 CA 31 31 JZ ENDE

312A 70 OK MOV H,B CHARACTER TO SCREEN  
 2B DCX H  
 28 DCX H  
 00 DCR C  
 C2 18 31 JNZ NEXTCHAR  
 3131 22 B2 30 ENDE SHLD #30B2 CURSOR-POINTER  
 EB XCHG  
 22 B4 30 SHLD #30B4 TEXTSTRING-POINTER  
 C3 7B 30 JHP RETURN  
 00  
 00



# ADRESSEN

## SCHREIBEN . . . . .

Mit dem Programm lassen sich 3 verschiedene Arten von Etiketten oder Adressen schreiben:

1. Ein Kleber von der Grösse 1\*1....19\*54-Zeichen kann mit allen auf dem Drucker verfügbaren Zeichen bedruckt werden. Das Format wird abgefragt.

Folgende Tasten werden zur Eingabe gebraucht: CURSOR, TAB, SPACE  
DEL, RETURN, Zeichen

Mit TAB erzeugt man loer-Schritte.

Mit RETURN wird das Geschriebene ausgedruckt.

Mit DEL springt der CURSOR nach links oben.

Mit den CURSOR-Tasten können Zeichen überschrieben werden, das Darunterliegende wird gelöscht.

2. Es kann eine Adresse nach folgendem Muster geschrieben werden:

Code	Telefon
Vorname/Anrede	
Name/Firma(Breitschrift)	
Strasse Nr.	
Land-Plz <u>Ortschaft</u>	

Alle Daten werden abgefragt. Die Länge des Striches unter 'Ortschaft' wird automatisch berechnet.

3. Es kann ein ganzes File vom Kassettengerät gelesen und ausgedruckt werden. Das File muss folgendes Format haben:

Ein File mit 10 Adressen hat 10 Strings = ADR\$(1..10)  
Im ADR\$ sind die Daten wie folgt geordnet: NAME?VORNAME?  
TELEFON?STRASSE NR?LAND-PLZ ORT?GEB.DATUM?CODE  
Für jeden Teil dieses Strings ist eine gewisse Anzahl  
Zeichen reserviert. In Pgm-Zeilen 1600-1650 sieht man  
wieviele Zeichen nach dem Input Befehl abgeschnitten  
werden. In Zeile 1660 wird dann der String zusammen  
gesetzt.

Um ein solches File herzustellen, kann man leicht ein kleines Programm schreiben, aber es existiert ein grosses umfangreiches Adressverarbeitungs-Programm, wo man auch Mutationen machen kann.

## Adressen schreiben

```

5   REM ***** INITIALISIEREN *****
10  PRINT CHR$(12):CLEAR 15000:POKE #131,I
20  COLORT 7 0 0 0:XX=0.0:YY=0.0
25  REM ***** DATEN EINGEBEN *****
30  CURSOR 15,20:PRINT "Adressenausdruck Programm"
35  CURSOR 15,19:PRINT "=====
40  CURSOR 3,16:PRINT "Moechtest Du : - eine Adresse schreiben ==> A"
45  CURSOR 19,15:PRINT "~ ein File ausdrucken ==> F"
50  CURSOR 19,14:PRINT "~ einen Kleber zeichnen ==> K"
55  CURSOR 19,13:PRINT "~ oder Nichts ==> N"
60  CURSOR 2,8:PRINT "Spanne Etiketten ein und schalte Drucker auf ON LINE"
65  CURSOR 5,4:PRINT "Gib ein A,F,K oder N ein ! ";CHR$(129):CURSOR 32,4
67  T=GETC:IF T=75.0 GOTO 69:IF T=70.0 GOTO 1000:IF T=7B.0 GOTO 4000:IF T=65.0 GOTO 1500:GOTO 67
69  PRINT CHR$(12)
70  CURSOR 3,17:PRINT "Welches Format hat die Etiketle ? (zB:5#34/8#34/8#27)"
75  CURSOR 5,15:INPUT "Anzahl Zeilen ";EY:IF EY>19 THEN PRINT " zu gross ":GOTO 75
80  CURSOR 5,14:INPUT "Anzahl Zeichen ";EX:IF EX>54 THEN PRINT " zu gross ":GOTO 80
85  DIM ETIKETTE(EX,EY)
90  REM ***** ETIKETTE SCHREIBEN *****
100 PRINT CHR$(12);"Etiketten Format : (Ausdruck mit 'RETURN')":CURSOR 27-EX/2,13+EY/2
105 FOR I=0.0 TO EY:FOR J=0.0 TO EX:ETIKETTE(J,I)=32.0:NEXT:GOTO 110
110 FOR IX=0 TO EX+1:PRINT CHR$(255);:NEXT:PRINT
120 FOR IY=0 TO EY-1:PRINT SPC(27-EX/2);CHR$(255);SPC(EX);CHR$(255);:NEXT:PRINT SPC(27-EX/2);
130 FOR IX=0 TO EX+1:PRINT CHR$(255);:NEXT
140 CURSOR 28-EX/2+XX,12+EY/2-YY:WAIT TIME 10
150 ZZ=GETC:IF ZZ=0.0 GOTO 150
160 IF ZZ>31.0 AND ZZ<127.0 GOTO 300
170 IF ZZ=9.0 THEN XX=8.0*(INT(XX/8.0)+1.0):GOTO 310
180 IF ZZ=16.0 THEN YY=YY-1.0:IF YY<0.0 THEN YY=0.0:GOTO 140
190 IF ZZ=17.0 THEN YY=YY+1.0:IF YY>EY-1.0 THEN YY=EY-1.0:GOTO 140
200 IF ZZ=18.0 THEN XX=XX-1.0:IF XX<0.0 THEN XX=0.0:GOTO 140
210 IF ZZ=19.0 THEN XX=XX+1.0:IF XX>EX-1.0 THEN XX=EX-1.0:GOTO 140
220 IF ZZ=8.0 THEN XX=0.0:YY=0.0:GOTO 140
230 IF ZZ=13.0 THEN 400
240 GOTO 140
300 PRINT CHR$(ZZ):ETIKETTE(XX,YY)=ZZ:XX=XX+1.0
310 IF XX>EX-1.0 THEN XX=0.0:YY=YY+1.0:IF YY>EY-1.0 THEN YY=0.0
320 GOTO 140
390 REM ***** ETIKETTE AUSDRUCKEN *****
400 PRINT CHR$(12):CURSOR 8,20
410 INPUT "Wieviele Etiketten moechtest Du (0=Ende) ";ANZAHL
420 IF ANZAHL=0 THEN 10
430 PRINT:PRINT:PRINT "Drucker einschalten , Etiketten einspannen , SPACE druecken"
440 IF GETC<>32.0 THEN 440
450 FOR DRUCKEN=1.0 TO ANZAHL:GOSUB 2000:NEXT
460 GOTO 410
1000 REM ***** ADRESS FILE AUSDRUCKEN *****
1010 PRINT CHR$(12):CLEAR 15000:ADRMAY=200.0:DIM ADR$(ADRMAY,0.0):COLORT 6 0 0 0:PRINT:PRINT
1012 PRINT " Hast Du ein File auf Kassette (J/N)?"
1014 F=GETC:IF F=0.0 THEN 1014
1016 IF CHR$(F)<>"J" THEN 10
1020 PRINT:PRINT " Kassette einlegen , Adress File suchen , Tape starten "
1030 PRINT:PRINT " LOAD ADR$(0,0)":LOADA ADR$:PRINT " loaded"
1040 PRINT CHR$(12):PRINT:PRINT:ANZAHL=1.0:STRICH$="=====
1050 PRINT "Moechtest Du von allen Adressen gleichviele (J/N;E=Ende)"
1055 JN$=CHR$(GETC)

```

```

1060 IF JN$="J" THEN INPUT "Wieviele Stueck moechtest Du von allen (0=Ende)";ANZAHL:GOTO 1100
1065 IF JN$="E" THEN 10
1070 IF JN$(">")"N" THEN 1055
1100 I=1.0:IF ANZAHL=0.0 THEN 10
1105 PRINT :PRINT :PRINT :I$=ADR$(I,0.0)
1110 IF LEFT$(JN$,1)="J" THEN 1150
1120 PRINT CHR$(12); "   Wieviele Etiketten moechtest Du von : "
1130 PRINT SPC(20);LEFT$(I$,36); "   ":INPUT ANZAHL:PRINT
1140 IF ANZAHL=0.0 THEN 1300
1150 FT=18.0:GOSUB 3000:REM *** SUBROUTINE AUSDRUCK ***
1300 I=I+1.0:IF ADR$(I,0.0)=" " THEN 1040
1310 IF I<=ADRMX THEN GOTO 1105
1320 GOTO 1040
1500 REM ***** ADRESSE SCHREIBEN *****
1510 PRINT CHR$(12):COLORT 6 0 0 0:STRICH$="=====":B$=SPC(25)
1520 CURSOR 0,22:PRINT "Die Adresse wird nach folgendem Muster gedruckt:(auf B&34)":PRINT :PRINT :PRINT
1530 PRINT "Code           Telefon"
1540 PRINT "   Vorname/Anrede"
1550 PRINT "   Name/Firma (Breitschrift)"
1560 PRINT "   Strasse Nr"
1570 PRINT "   Land-Plz Ortschaft"
1580 PRINT "   ====="
1600 CURSOR 9,9:INPUT "Name/Firma   :";NA$:N$=NA$+B$:NA$=LEFT$(N$,18)
1610 CURSOR 9,8:INPUT "Vorname/Anrede :";VN$:V$=VN$+B$:VN$=LEFT$(V$,18)
1620 CURSOR 9,7:INPUT "Strasse Nr   :";ST$:S$=ST$+B$:ST$=LEFT$(S$,22)
1630 CURSOR 9,6:INPUT "Land-Plz Ort  :";PO$:P$=PO$+B$:PO$=LEFT$(P$,25)
1640 CURSOR 9,5:INPUT "Telefon      :";TE$:T$=TE$+B$:TE$=LEFT$(T$,16)
1650 CURSOR 9,4:INPUT "Code         :";CO$:C$=CO$+B$:CO$=LEFT$(C$,21)
1660 I$=NA$+VN$+TE$+ST$+PO$+SPC(12)+CO$
1700 CURSOR 3,2:INPUT "Wieviele Adressen moechtest Du (0=Ende) ";ANZAHL
1710 IF ANZAHL=0 THEN 10
1720 CURSOR 3,1:PRINT "Moechtest Du Absenderadressen ? (J/N)"
1730 JN$=CHR$(GETC)
1740 IF JN$="J" THEN FT=15.0:GOTO 1800
1750 IF JN$(">")"N" THEN 1730
1760 FT=18.0
1800 GOSUB 3000:REM *** SUBROUTINE AUSDRUCK ***
1810 PRINT CHR$(18):GOTO 1700
2000 REM ***** SUBROUTINE PRINT *****
2005 PRINT CHR$(12);DRUCKEN;"   Etiketete":PRINT :POKE #131,0
2010 FOR I=0.0 TO EY-1.0:FOR J=0.0 TO EX-1.0
2020 PRINT CHR$(ETIKETTE(J,I));:NEXT
2030 PRINT :NEXT:PRINT
2040 POKE #131,1
2050 RETURN
3000 REM ***** SUBROUTINE AUSDRUCK *****
3010 POKE #131,0:PRINT CHR$(FT);
3020 FOR AZ=1 TO ANZAHL:B=0.0
3030 IF MID$(I$,82+B,2)<>" " THEN B=B+1.0:GOTO 3030
3040 PRINT MID$(I$,111,18);MID$(I$,36,16):PRINT
3050 PRINT SPC(4);MID$(I$,18,18)
3060 PRINT SPC(2);CHR$(14);LEFT$(I$,16)
3070 PRINT SPC(4);MID$(I$,52,22):PRINT
3080 PRINT SPC(1);MID$(I$,74,25)
3090 PRINT SPC(9);LEFT$(STRICH$,B):PRINT
3100 NEXT
3110 POKE #131,1:RETURN
4000 REM ***** ENDE *****

```

Dieses Programm dient zur optischen Bildschirm-  
anzeige fuer "DAInaclock"

```

10 PRINT CHR$(12);POKE #75,32;PRINT CHR$(12);COLORT 9 14
   9 9
20 CLEAR 6000;DIM F%(12.0),F$(12.0),TAG$(6.0),MONAT$(12.0)

   F% fuer Abfrage DAInaclock
   F$ STR$ von F%
   TAG$ fuer Wochentage
   MONAT$ fuer Monate

50 FOR X%=0 TO 6:READ TAG$(X%);NEXT
60 FOR X%=0 TO 12:READ MONAT$(X%);NEXT -Einlesen Tage un
   d Monate

70 GOSUB 180 -Abfrage Uhr
75 GOSUB 230 -Berechnung des Monat aus F%(9) und F%(10)

80 CURSOR 4,20

90 PRINT TAG$(FT%);", ";F%(8.0);F%(7.0);". ";MONAT$(FM%);
   " 1 9";F%(12.0);F%(11.0);POKE #BE5D,#6A
   Anzeige

100 POKE #BC44,#DA;POKE #BB38,#D3
   Vordergrundfarbe aendern

110 Q%=GETC;IF Q%=9 THEN GOTO 280
   Abfrage GETC zum Stellen der Uhr

120 GOSUB 180
   Abfrage Uhr

130 CURSOR 11,14;PRINT F$(1.0);F$(0.0);POKE #BB39,#5A - ae
   ndern der Schriftgroesse

140 IF FALT%=F%(2) THEN 160
   wenn Minutenwert noch gleich, werden Stunden und Zehne
   rminuten nicht geschrieben

150 CURSOR 0,16;PRINT F$(5.0);F$(4.0);": ";F$(3.0);F$(2.0);
   POKE #BC45,#4A;FALT%=F%(2)
   Anzeige Stunden, Minuten

160 IF F%(6.0)>FT% THEN 70
   wenn der Tag sich aendert dann von Zeile 70

170 GOTO 110
   Schleife zur staendigen Abfrage

180 FOR A%=0 TO 12
190 OUT (#33),128;OUT (#30),A%;OUT (#31),3;OUT (#31),1;OUT

```

```
(#33),#90:OUT (#31),9:WAIT TIME 1:X%=INP(#30):OUT (#3
1),0:OUT (#33),#9B
200 F%(A%)=X% IAND #F
210 IF A%=5 THEN F%(5.0)=F%(5.0)-8.0
220 F*(A%)=MID$(STR$(F%(A%)),1,1):NEXT:RETURN
naechere Erklaerung siehe Manual "DAInaclock" und Auszu
g unter F4 der Clubzeitung

230 FT%=F%(6.0)
240 FM1$=STR$(F%(10.0)):FM2$=STR$(F%(9.0))
250 FM$=MID$(FM1$,1,1)+MID$(FM2$,1,1)
260 FM%=VAL(FM$):REM FM%=VAL(RIGHT$(FM$,LEN(FM$)-1))
270 RETURN

280 FOR X%=#B4B3 TO #B3E5 STEP -#B6:POKE X%,#7A:NEXT
290 CURSOR 0,12:INPUT "KANAL";K%:INPUT "ZAHL (ACHTUNG BEI
KANAL 5)";TZ
300 OUT (#33),120:OUT (#30),K%:OUT (#31),7:OUT (#31),5:OUT
(#30),TZ:OUT (#31),#15:WAIT TIME 2:OUT (#31),5:OUT (#
31),0:OUT (#33),#9B
310 PRINT CHR$(12):GOTO 70

320 DATA "SAMSTAG ", "SONNTAG ", "MONTAG ", "DIENSTAG
", "MITTWOCH "
330 DATA "DONNERSTAG", "FREITAG "
340 DATA NULL, JAN, FEB, MAR, APR, MAI, JUN
350 DATA JUL, AUG, SEP, OKT, NOV
360 DATA DEZ
```

von Nils Kay und Wolfgang Schnaak

Das Programm schaltet die Farbe mit Hilfe des Interrupt 7 um, dadurch entsteht ein blinkender Effekt.

Um den Hintergrund blinken zu lassen, muessen die markierten Werte durch folgende ersetzt werden :

#F2/#F6, #B0/#A0, #F2/#F6

```

10  POKE #70,#A9:POKE #71,#D9
    setzt V7 auf Normalwert

20  IF PEEK(#29C)=3 THEN 30:CLEAR #3A:POKE #29D,4
    HEAP setzen

30  FOR A%=#2F0 TO #321:READ W%:POKE A%,W%:NEXT
    Lesen und Einpoken m1-Teil

30  CALLM #313:CLEAR 256
    Initialisierung & Schutz der Blinkroutine

40  COLORT 8 0 8 0:POKE #2EF,15
        H V h v          v2

    h,v & v2= Hintergrund- und wechselnde Vordergrund-
            farben

50  FOR I%=#BA2C-8 TO #BA2C-20 STEP (-2):POKE I%,#FF:NEXT
    schaltet in der 12. Zeile von 0 bis 9 auf Blinken

60  CURSOR 0,12:PRINT " Blinker "
```

Maschinenteil in DATA-Zeilen

```

100  DATA #E5,#C5,#F5,#3A,#C0,#01,#FE,#01
110  DATA #C2,#0D,#03,#3A,#F2,#BF,#47,#3A
    ^^^
120  DATA #EF,#02,#E6,#0F,#F6,#B0,#32,#F2
    ^^^ ^^^
130  DATA #BF,#7B,#32,#EF,#02,#F1,#C1,#E1
140  DATA #C3,#A9,#D9
150  DATA #E5,#21,#22,#03,#22,#9B,#02,#21
160  DATA #F0,#02,#22,#70,#00,#E1,#C9
```

^^^ hier aendern, siehe oben

Dieses Programm ermoglicht die automatische Ausschreibung vorher festgelegter Basicbefehle durch Eingabe von Kuerzel.

Das Programm arbeitet im normalen Eingabe- und im Editmodus.

Die Tabelle der Kuerzel und Basicbefehle kann beliebig geaendert und erweitert werden. Am Ende der Tabelle muss als Kuerzel immer 'END' stehen.

Einlesen des Maschinenprogrammes

```
10 FOR ADRESSE%=#4000 TO #40B0:READ PK%:POKE ADRESSE%,PK%
: NEXT
```

Maschinenprogramm in DATA-Anweisungen

```
20 DATA #F3,#E5,#F5,#3E,#10,#32,#F7,#FF
30 DATA #3A,#F1,#FF,#E6,#40,#CA,#30,#40
40 DATA #3A,#6C,#00,#A7,#FA,#1D,#40,#21
50 DATA #FD,#C6,#C3,#20,#40,#21,#40,#40
60 DATA #22,#6C,#00,#3E,#10,#32,#F7,#FF
70 DATA #3A,#F1,#FF,#E6,#40,#C2,#28,#40
80 DATA #3A,#6D,#00,#32,#E9,#BF,#F1,#E1
90 DATA #C3,#78,#D5,#00,#00,#00,#00,#00
100 DATA #C5,#D5,#E5,#F5,#2A,#BE,#02,#2B
110 DATA #CD,#7A,#40,#7E,#FE,#61,#FA,#73
120 DATA #40,#36,#00,#47,#2B,#CD,#7A,#40
130 DATA #4E,#21,#B0,#40,#16,#00,#7E,#23
140 DATA #B9,#C2,#66,#40,#7E,#B8,#23,#CA
150 DATA #B1,#40,#5E,#19,#23,#7B,#FE,#FF
160 DATA #C2,#5E,#40,#F1,#E1,#D1,#C1,#C3
170 DATA #FD,#C6,#7D,#FE,#B9,#C0,#2E,#BD
180 DATA #C9,#3E,#03,#32,#AC,#40,#3A,#18
190 DATA #01,#A7,#CA,#96,#40,#3E,#2D,#32
200 DATA #AC,#40,#3E,#12,#EF,#2D,#3E,#08
210 DATA #CD,#AB,#40,#5E,#23,#7E,#CD,#AB
220 DATA #40,#1D,#C2,#9C,#40,#F1,#E1,#D1
230 DATA #C1,#E1,#C9,#EF,#03,#C9,#00,#00
240 DATA #00
```

Einlesen 'Kuerzel und Basicbefehle'

```
1000 ADRESSE%=#40B0
1010 READ KUERZEL$,BASICBEFEHL$:IF KUERZEL$="END" THEN 1060
1020 FOR I%=0 TO 1:POKE ADRESSE%,ASC(MID$(KUERZEL$,I%,1)):ADRESSE%=ADRESSE%+1:NEXT
1030 POKE ADRESSE%,LEN(BASICBEFEHL$):ADRESSE%=ADRESSE%+1
1040 FOR I%=0 TO LEN(BASICBEFEHL$)-1:POKE ADRESSE%,ASC(MID$(BASICBEFEHL$,I%,1)):ADRESSE%=ADRESSE%+1:NEXT
1050 GOTO 1010
1060 POKE ADRESSE%,#FF:POKE ADRESSE%+1,#FF:POKE ADRESSE%+2,#FF
1070 POKE #6E,0:POKE #6F,#40:REM V6=4000
1080 END
```

## Tabelle 'Kuerzel und Basicbefehle'

2000 DATA Ab, ABS(, Ai, ASIN(, Al, ALDG(, Ao, ACOS(, As, ASC(, At, ATN  
(, Ca, CALLM

2010 DATA Cg, COLORG, Ch, CHECK, Cl, CLEAR, Co, CONT, Cr, CHR\$(, Cs, C  
OS(, Ct, COLORT

2020 DATA Cu, CURSOR, Cx, CURX, Cy, CURY, Da, DATA, Dr, DRAW, Ed, EDIT  
, En, ENVELOPE

2030 DATA Ex, EXP(, Fi, FILL, Fq, FREQ(, Fr, FRAC(, Ge, GETC, Gg, GOTO  
, Gs, GOSUB

2040 DATA He, HEX\$(, Ia, IAND, In, INPUT, Io, INOT, Ip, INP(, It, INT(,  
, Ix, IXOR

2050 DATA Le, LEFT\$(, Lg, LOG(, Li, LIST, Ln, LEN(, Lo, LOAD, Lt, LOGT  
(, Mi, MID\$(

2060 DATA Mo, MODE, Ne, NEXT, No, NOISE, Pd, PDL(, Pe, PEEK(, Po, POKE  
, Re, READ, Pr, PRINTCHR\$(12)

2070 DATA Ri, RIGHT\$(, Rn, RND(, Rs, RESTORE, Rt, RETURN, Sa, SAVE, S  
c, SCRN(, Se, STEP

2080 DATA Sg, SGN(, Si, SIN(, So, SOUND, Sp, SPC(, Sq, SQR(, Sr, STR\$(  
, St, STOP, Ta, TALK

2090 DATA Tb, TAB(, Tf, TROFF, Tn, TAN(, Tr, TRON, V1, VAL(, Vr, VARPT  
R(, Wa, WAIT

2100 DATA Wt, WAITTIME, Wm, WAITMEM, Xm, XMAX, Ym, YMAX

2110 DATA END, END



Dieses Programm ermöglicht den Ausdruck der Filenamen aller auf einer Cassette befindlichen Programme und Dateien. Es wurde erstellt fuer die MDCR und muß fuer den normalen Cassettenrecorderbetrieb entsprechend geaendert werden.

Dieses Programm ist entnommen aus einem Programm der Clubgruppe Baldur Brock.

```

rh
10 MODE0:PRINT CHR$(12)
20 CLEAR 14000
30 DIM TITELZEILE$(50)

40 DATA #F5,#C5,#D5,#E5,#01,#40,#00,#11,#B1,#B0,#21
50 DATA #9E,#E6,#CD,#CE,#02,#E1,#D1,#C1,#F1,#C9
Maschinenteil in DATA-Anweisungen
60 A$=""

Schleife zum Einlesen der DATA-Anweisungen.
70 FOR A=1 TO 24
80 READ B:A$=A$+CHR$(B):NEXT
Das Maschinenprogramm steht jetzt in der String-
Variablen A$.

90 ADRESSE=PEEK(VARPTR(A$)) IOR (PEEK(VARPTR(A$)+1) SHL 8
)+1
Berechnung der Startadresse des Maschinenprogramms.

100 CPT=1:Z=23:TITELZEILE$(CPT)=""
110 CALLM #F000:REM REW
120 ?CHR$(12)
130 I=0:CURSOR 1,23:CALLM ADRESSE
140 Z=Z-1
150 CURSOR 1,Z:IF Z=1 THEN Z=23
160 PRINT CPT;"-";

170 FOR A=#BFEB TO #BFA5 STEP -2:PRINT CHR$(PEEK(A));
Auslesen der Bildschirmzeile 23 von #BFEB bis #BFA5.

180 TITEL$=TITEL$+CHR$(PEEK(A))
Addieren zum Titel aus Zeile 170.

190 NEXT

200 IF CPT=1 THEN TIT1$=TITEL$
210 IF TITEL$=TIT1$ AND CPT>1 THEN 280
Vergleich zum Abbruch der Checkroutine

220 S$="":IF CPT<10 THEN S$=" "
230 TITELZEILE$(CPT)=S$+LEFT$(STR$(CPT),LEN(STR$(CPT))-2)+
"+TITEL$;TITEL$="
240 TITELZEILE$(CPT)=TITELZEILE$(CPT)+SPC(50-LEN(TITELZEIL
E$(CPT)))
Erstellen einer Titelzeile des Titels, kann nach
Wunsch geaendert werden.
Abspeichern der Titelzeile unter TITELZEILE$(DIM).

250 CURSOR 1,23:PRINT SPC(59):CURSOR 1,23
260 CPT=CPT+1
270 GOTO 130

```

```

30000 REM HILFSPROGRAMME ZUM LADEN UND SPEICHERN (BP110383)
30010 REM VON FILES BELIEBIGEN TYP (1, #, $, etc.)
30020 REM . VIRTUELLES FILEFORMAT:
30030 REM START-LOW, START-HIGH, LAENGE-LOW, LAENGE-HIGH, DATEN
30040 REM
30050 REM DAS FILETYP-BYTE IST IN LOD%+7 UND SAV%+6,
30060 REM ES SOLLTE <= #32 SEIN, DA ES SONST BEI CHECK
30070 REM NICHT ERKANNT WIRD.
30080 REM
30090 REM . AUFRUF-BEISPIELE:
30100 REM POKE LOD%+7, ASC("#"):CALLM LOD%,NAME$
30110 REM
30120 REM POKE SAV%+6, ASC("1"):POKE SAV%+11, BIS% IAND 255
30130 REM POKE SAV%+12, BIS% SHR 8:POKE SAV%+15, VON% IAND 255
30140 REM POKE SAV%+16, VON% IAND 255:CALLM SAV%,NAME$
30150 REM
30160 REM . LADEN AN EINE FESTE ADRESSE:
30170 REM POKE LOD%+20, #21:POKE LOD%+21, ADR% IAND 255
30180 REM POKE LOD%+22, ADR% SHR 8:CALLM LOD%,NAME$
30190 REM
30200 REM FUER DOS-BENUTZER:
30210 REM ALS TYP IMMER #31 BENUTZEN UND IN DIE ADRESSEN
30220 REM #764 BIS #766 DIE EXTENSION EINBRINGEN
30230 REM z.B.: POKE #764, ASC("B"):POKE #765, ASC("L"):POKE #766, ASC("D
")
30240 REM HINTERHER UNBEDINGT WIEDER 'BIN' EINPOKEN!
30250 REM
30400 REM, BEISPIEL SPEICHERN EINES BILDES:
30410 MODE 4: DRAW 0,0 XMAX, YMAX 21: DRAW 0, YMAX XMAX, 0 22
30420 GOSUB 31000: REM MLP EINLESEN
30430 POKE SAV%+6, #31:POKE SAV%+11, #FF:POKE SAV%+12, #BF: REM BILDENEDE
30440 POKE SAV%+15, PEEK(#84):POKE SAV%+16, PEEK(#85): REM ERSTES SCREEN-
BYTE
30450 NAME$="BILD1":CALLM SAV%,NAME$
30460 END
31000 REM EINLESEN MLP'S
31010 DIM LOD%(8), SAV%(11)
31020 LOD%=VARPTR(LOD%(0)):SAV%=VARPTR(SAV%(0))
31030 FOR I%=LOD% TO LOD%+34:READ BYTE%:POKE I%,BYTE%:NEXT
31040 FOR I%=SAV% TO SAV%+47:READ BYTE%:POKE I%,BYTE%:NEXT
31050 RETURN
31060 REM BEI LADEFEHLER: ***BREAK
31100 REM DATEN FUER LADEN
31110 DATA #C5, #5E, #23, #56, #EB, #01, #00, #31
31120 DATA #CD, #CE, #02, #21, #3E, #01, #11, #41
31130 DATA #01, #CD, #D1, #02, #2A, #3E, #01, #CD
31140 DATA #D1, #02, #CD, #D4, #02, #C1, #FB, #D2
31150 DATA #0C, #CB, #C9
31200 REM DATEN FUER SPEICHERN
31210 DATA #C5, #5E, #23, #56, #EB, #3E, #31, #CD
31220 DATA #C5, #02, #11, #00, #00, #13, #21, #00
31230 DATA #00, #D5, #E5, #22, #3E, #01, #21, #3E
31240 DATA #01, #11, #02, #00, #CD, #CB, #02, #E1
31250 DATA #D1, #7B, #95, #5F, #7A, #9C, #57, #CD
31260 DATA #CB, #02, #CD, #CB, #02, #C1, #FB, #C9

```

```

100 REM KALENDER
110 REM NACH 'ELCOMP' NR 11/12 81
120 REM ADAPTIERT AUF DAI VON BP1103B3
130 POKE #131,1:PRINT CHR$(12);
140 PRINT :PRINT SPC(24);"*****"
150 PRINT SPC(24);"* KALENDER *"
160 PRINT SPC(24);"*****"
170 CURSOR 19,15:PRINT "Innerhalb der Grenzen:"
180 CURSOR 19,13:PRINT "1.3.1700 bis 28.2.2100"
200 CLEAR 5000:DIM MO$(11),WT$(6),TAZ(6,5,11)
210 CURSOR 18,10:INPUT "Gib Jahreszahl --> ";JZ%
220 IF JZ%<1700 OR JZ%>2100 THEN 100
230 CURSOR 23,6:PRINT "Bitte warten!!"
240 MZ=14:TX=1
250 GOSUB 1000:REM BER. TAGESZAHL
260 GOSUB 2000:REM BER. WOCHENTAG
270 GOSUB 3000:REM SCHALTJAHR?
280 GOSUB 4000:REM PUFFER FUELLEN
290 GOSUB 5000:REM AUSDRUCKEN
300 GOSUB 6000:REM BER. FEIERTAGE
310 GOSUB 7000:REM FEIERTAGE DRUCKEN
320 POKE #131,1:END
1000 REM . BERECHNUNG DER TAGESANZAHL
1010 TAZ=INT(365.25*(JZ%-1))+INT(30.6*MZ)+TX-621049
1020 IF JZ%<1801 THEN TAZ=TAZ+1
1030 IF JZ%<1901 THEN TAZ=TAZ+1
1040 RETURN
2000 REM . BERECHNUNG DES WOCHENTAGES
2010 WOZ=INT((TAZ/7.0-INT(TAZ/7.0))*7.0+0.5)
2020 RETURN
3000 REM . BERECHNUNG DER SCHALTJAHRE
3010 SJZ=28
3020 IF (JZ%-1700)/4.0-INT((JZ%-1700)/4.0)=0.0 THEN SJZ=29
3030 RETURN
4000 REM . DRUCKPUFFER FUELLEN
4020 DATA JANUAR,FEBRUAR,MARZ,APRIL,MAI,JUNI,JULI,AUGUST,SEPTEMBER,OK
TOBER,NOVEMBER,DEZEMBER
4030 DATA SO,MO,DI,MI,DO,FR,SA
4040 FOR IZ=0 TO 11:READ MO$(IZ):NEXT
4070 FOR IZ=0 TO 6:READ WT$(IZ):NEXT
4110 XZ=WOZ:YZ=0:ZK=0:IZ=1
4120 TAZ(XZ,YZ,ZK)=IZ
4130 IZ=IZ+1:XZ=XZ+1
4140 IF IZ>SJZ AND ZK=1 THEN IZ=1:YZ=0:ZK=ZK+1
4150 IF IZ>30 AND (ZK=3 OR ZK=5 OR ZK=8 OR ZK=10) THEN IZ=1:YZ=0:ZK=Z
K+1
4160 IF IZ>31 THEN IZ=1:ZK=ZK+1:YZ=0
4170 IF XZ>6 AND YZ=0 AND IZ=1 THEN XZ=0
4180 IF XZ>6 THEN XZ=0:YZ=YZ+1
4190 IF YZ>5 THEN YZ=0
4200 IF ZK<12 THEN 4120
4210 RETURN
5000 REM . DRUCKPUFFER AUSDRUCKEN
5010 GOSUB 5500
5020 JZ=0:KZ=0:LZ=0
5030 TBZ=0
5040 FOR IZ=0 TO 2
5050 PRINT TAB(TBZ);
5060 TBZ=TBZ+26
5070 FOR SZ=(10-LEN(MO$(IZ+JZ)))/2 TO 0 STEP -1
5080 PRINT " ";
5090 NEXT
5100 PRINT MO$(IZ+JZ);

```

```

5110 NEXT
5120 PRINT :PRINT
5130 TB%=4
5140 FOR IZ=0 TO 2
5150 PRINT TAB(TB%);
5160 TB%=TB%+26
5170 FOR SZ=0 TO 6
5180 PRINT WT%(SZ); " ";
5190 NEXT
5200 NEXT
5210 PRINT :TB%=4
5220 FOR YZ=0 TO 5
5230 FOR IX=0 TO 2
5240 PRINT TAB(TB%);
5250 FOR XZ=0 TO 6
5260 TA%=STR$(TAX(XZ, YZ, IZ+JZ)); TA%=MID$(TA%, 1, LEN(TA%)-3)
5270 IF TAX(XZ, YZ, JZ+IZ)=0 THEN TA%=" "
5280 IF LEN(TA%)<2 THEN TA%=" "+TA%
5290 PRINT TA%; " ";
5300 NEXT
5310 TB%=TB%+26
5320 NEXT
5330 PRINT :TB%=4
5340 NEXT
5350 JZ=JZ+3
5360 PRINT :PRINT
5370 IF JZ<11 THEN 5030
5380 PRINT CHR$(14);ST%;PRINT
5390 RETURN
5500 REM . DRUCKER EINSCHALTEN
5510 POKE #131,0
5520 ST%="*****"
5530 PRINT CHR$(27);"E";:REM FETTSCHRIFT
5540 PRINT CHR$(14);ST%;PRINT
5590 PRINT CHR$(14);TAB(9);JZ%; " KALENDER ";JZ%
5600 PRINT :PRINT CHR$(14);ST%
5610 PRINT :PRINT
5620 RETURN
5700 REM . DRUCKER ABSCHALTEN
5710 PRINT :PRINT CHR$(14);ST%
5720 POKE #131,1:PRINT CHR$(12)
5730 RETURN
6000 REM . BERECHNUNG DER FEIERTAGE
6010 MX=4:TX=20:JZ%=JZ%+1
6020 GOSUB 1000
6030 TX%=TX%
6040 TA%=INT(INT(TA%/29.575+0.5)*29.575)
6050 GOSUB 2000
6060 TA%=TA%+7-WO%
6070 TX%=TA%-TX%+20
6080 IF TX%<27 THEN TX%=TX%+28
6090 IF TX%>31 THEN TX%=TX%-31:MZ%=MZ%+1
6100 MZ%=MZ%-1
6110 TY%=TX%+19:M1%=5
6120 IF TY%>31 THEN TY%=TY%-31:M1%=6
6130 TZ%=TX%-2:M2%=MZ%
6140 IF TZ%<1 THEN TZ%=TZ%+31:M2%=M2%-1
6150 TT%=TX%+9:M3%=5
6160 IF TT%>31 THEN TT%=TT%-31
6170 MX%=MZ:MZ%=12:TX%=1
6180 GOSUB 1000
6190 GOSUB 2000
6200 TB%=0-WO%

```

```
6210 IF WO%<4 THEN TB%=TB%-7
6220 TB%=TB%+14
6230 TC%=TB%+3
6240 TD%=TB%+7
6250 M%=MX%
6260 RETURN
7000 REM . DRUCKEN DER FEIERTAGE
7010 JZ%=JZ%-1
7020 A$=SPC(20)
7030 PRINT A$;" Feiertage:";PRINT :PRINT
7040 PRINT A$;"Neujahr";TAB(51);"1. ";MO$(0)
7050 PRINT A$;"Karfreitag";TAB(54-LEN(STR$(TZ%)));TZ%;" ";MO$(M2%-1)
7060 PRINT A$;"Ostern";TAB(54-LEN(STR$(TX%)));TX%;" ";MO$(M%-1)
7070 PRINT A$;"Tag der Arbeit";TAB(51);"1. ";MO$(4)
7080 PRINT A$;"Himmelfahrt";TAB(54-LEN(STR$(TT%)));TT%;" ";MO$(M3%-1)
7090 PRINT A$;"Pfingsten";TAB(54-LEN(STR$(TY%)));TY%;" ";MO$(M1%-1)
7100 PRINT A$;"Tag der deutschen Einheit";TAB(50);"17. ";MO$(5)
7110 PRINT A$;"Volkstrauertag";TAB(54-LEN(STR$(TB%)));TB%;" ";MO$(10)
7120 PRINT A$;"Buß- und Betttag";TAB(54-LEN(STR$(TC%)));TC%;" ";MO$(10)
7130 PRINT A$;"Totensonntag";TAB(54-LEN(STR$(TD%)));TD%;" ";MO$(10)
7140 PRINT A$;"Weihnachten";TAB(50);"24. ";MO$(11)
7150 GOSUB 5700
7160 RETURN
```

ich einmal ...

\*\*\*\*\*  
\*MODE 7, 8\*  
\*\*\*\*\*

Programme, um die höchstauflösende Graphik mit 512 \* 240 Bildpunkten auf dem DAI zu generieren, sind schon öfters erschienen. Allerdings gab es immer irgendwelche Einschränkungen (nur MODE 8, DOT, DRAW und FILL nicht benutzbar etc.). Mit den folgenden Programmen hat man alle "Features" der schon vorhandenen Graphikmodi 1 bis 6, und ist wohl auch das Maximale, was man in dieser Sache erreichen kann. (Ausser man schreibt ein neues Basic). Es sind alle Basic-Graphikbefehle verfügbar und es ist sowohl der gesplittete, als auch der reine Graphikmodus möglich. Ebenso funktioniert das "Scrolling" in beiden Richtungen (z.Bsp. von MODE 8 zu 8A und umgekehrt). Beide Programme beginnen bei #300. Für richtige ML-Freaks dürfte es jedoch kein Problem sein, eines der Programme auf einen anderen Adresbereich zu verschieben, um beide Graphikmodi gleichzeitig zur Verfügung zu haben. Ich benutze hauptsächlich den 4-farbenmodus, also MODE 8, 8A, da in MODE 7, 7A das Zeichnen von Linien und Rechtecken schon wesentlich langsamer ist. Vor dem Aufruf mit CALLM#300 oder G300 muss in die Zelle #2FF das Codebyte des jeweiligen Modus eingeschrieben werden. Die Codes sind:

MODE 7 : #0C  
MODE 7A : #0D  
MODE 8 : #0E  
MODE 8A : #0F

(Natürlich reagiert das Programm für MODE 8, 8A nicht auf den Code von MODE 7, 7A u. umgekehrt. Ein Aufruf von MODE 7A sieht in Basic also so aus: 10 POKE #2FF, #0D: CALLM#300)

Das letzte Programm bewirkt ein automatisches Scrolling in den gesplitteten Mode, falls in MODE 7 oder 8 irgendwelche Texte ausgegeben werden. Diese zweite Routine ist unbedingt nötig, weil sonst das Programm dabei aussteigt !!! Alle Textausgaben auf den Bildschirm laufen bekanntlich über die RST5 DATA 3 - Routine. Das Programm überprüft nun, ob diese Routine aufgerufen wird. Wenn ja, wird festgestellt, ob MODE 7 oder 8 gewählt ist, um dann gegebenenfalls auf MODE 7A bzw. 8A umzuschalten. Zur Initialisierung muß man nur den RST5 - Vektor auf die Startadresse des Programms setzen.

Damit dürfte hoffentlich der "Dauerbrenner" MODE 7, 8 zur Zufriedenheit aller erledigt sein.

(Wann gibts die ersten Graphikspiele in MODE 8 ... ?)

Viel Spass beim Ausprobieren !!!

Helge Rebhan

MODE 8, 8A

>D300 43F

```

0300 C5 3A 40 00 F5 3E BF 32 40 00 32 06 FD 3A FF 02
0310 FE 0F CA 4C 03 F5 3A 9D 00 FE 0F CA B5 03 21 00
0320 04 CD 46 E5 CD F3 03 F1 06 06 11 9E 00 CD FC E5
0330 21 4F 42 06 0F CD FC E5 32 9D 00 3E 42 01 F0 30
0340 11 00 00 2A 82 00 CD A4 03 C3 97 03 F5 3A 9D 00
0350 FE 0E CA D3 03 21 20 04 CD 46 E5 21 E7 50 CD 87
0360 E6 11 CF 4E CD FD E1 2A 80 00 F1 06 06 11 9E 00
0370 CD FC E5 21 CF 4E 06 0F CD FC E5 CD A0 03 2A 82
0380 00 01 D4 30 11 00 00 32 9D 00 3E 42 CD A4 03 0E
0390 1C 2A 8E 00 CD A4 03 F1 32 40 00 32 06 FD C1 C9
03A0 F5 C3 2B E5 F5 C5 D5 E5 C3 E6 E5 F5 C3 8B E4 F5
03B0 F5 D5 C3 D2 E4 F1 32 9D 00 21 00 04 CD 46 E5 CD
03C0 F3 03 CD AB 03 21 4F 42 11 9E 00 06 0F CD FC E5
03D0 C3 97 03 21 20 04 CD 46 E5 F1 AF CD AF 03 21 CF
03E0 4E 11 7C 00 06 0F CD FC E5 3E 0F 32 9D 00 C3 97
03F0 03 00 00 00 00 00 2A 90 00 CD 87 E6 2A 80 00 C9
0400 C0 7D B8 0E B0 7D B0 7D E8 7F C0 7D 5B 8C 40 71
0410 00 02 F0 1C B6 00 00 00 00 00 00 00 00 00 00
0420 E8 7F B8 0E 08 6F 18 6F 30 71 40 71 5B 8C B0 7D
0430 00 02 F0 1C B6 00 00 00 00 00 00 00 00 00 00

```

MODE 7, 7A

>S131 03-

>D300 43F

```

0300 C5 3A 40 00 F5 3E BF 32 40 00 32 06 FD 3A FF 02
0310 FE 0D CA 4C 03 F5 3A 9D 00 FE 0D CA B5 03 21 00
0320 04 CD 46 E5 CD F3 03 F1 06 06 11 9E 00 CD FC E5
0330 21 4F 42 06 0F CD FC E5 32 9D 00 3E 42 01 F0 B0
0340 11 FF 00 2A 82 00 CD A4 03 C3 97 03 F5 3A 9D 00
0350 FE 0C CA D3 03 21 20 04 CD 46 E5 21 E7 50 CD 87
0360 E6 11 CF 4E CD FD E1 2A 80 00 F1 06 06 11 9E 00
0370 CD FC E5 21 CF 4E 06 0F CD FC E5 CD A0 03 2A 82
0380 00 01 D4 B0 11 FF 00 32 9D 00 3E 42 CD A4 03 0E
0390 1C 2A 8E 00 CD A4 03 F1 32 40 00 32 06 FD C1 C9
03A0 F5 C3 2B E5 F5 C5 D5 E5 C3 E6 E5 F5 C3 8B E4 F5
03B0 F5 D5 C3 D2 E4 F1 32 9D 00 21 00 04 CD 46 E5 CD
03C0 F3 03 CD AB 03 21 4F 42 11 9E 00 06 0F CD FC E5
03D0 C3 97 03 21 20 04 CD 46 E5 F1 AF CD AF 03 21 CF
03E0 4E 11 7C 00 06 0F CD FC E5 3E 0D 32 9D 00 C3 97
03F0 03 00 00 00 00 00 2A 90 00 CD 87 E6 2A 80 00 C9
0400 C0 7D B8 0E B0 7D B0 7D E8 7F C0 7D 5B 8C 40 71
0410 00 02 F0 1C B6 00 00 00 00 00 00 00 00 00 00
0420 E8 7F B8 0E 08 6F 18 6F 30 71 40 71 5B 8C B0 7D
0430 00 02 F0 1C B6 00 00 00 00 00 00 00 00 00 00

```

0 / \*\*\* RST5-Einsprung \*\*\*

```
1 / testet, ob Mode 7,8 gewaehlt und schaltet evtl. auf
2 / Mode 7A,8A um
3 DW RESTART: C6FD
4 ORG 440
440 E1 5 POPH
441 E3 6 XTHL /hole Databyte
442 F5 7 PUSHPSW
443 7E 8 MOVAM
444 FE 03 9 CPI 3 /Bildschirmausgabe ?
446 CA 4F 04 10 JZ CONT
449 F1 11 ENDE: POPPSW /Stack wieder richten
44A E3 12 XTHL
44B E5 13 PUSHH
44C C3 FD C6 14 JMP RESTART /alte RST5-Adresse
44F 3A 9D 00 15 CONT: LDA 009D /MODE-Code
452 FE 0C 16 CPI C
454 DA 49 04 17 JC ENDE /Es ist MODE 0..6 gewaehlt
457 1F 18 RAR
458 DA 49 04 19 JC ENDE /alles o.k. MODE 7A oder 8A gew.
45B 17 20 RAL
45C 3C 21 INRA /Codebyte bilden
45D 32 FF 02 22 STA 02FF /und abspeichern
460 E5 23 PUSHH
461 D5 24 PUSHD
462 CD 00 03 25 CALL 0300 /Scrolling zu MODE 7A oder 8A
465 D1 26 POPD
466 E1 27 POPH
467 C3 49 04 28 JMP ENDE /zur alten RST5-Routine
```