

Uwe Koch
Frankenstraße 25
5880 LUDENSCHIED
Tel. (023 51) 2 11 92

2. JAHRGANG

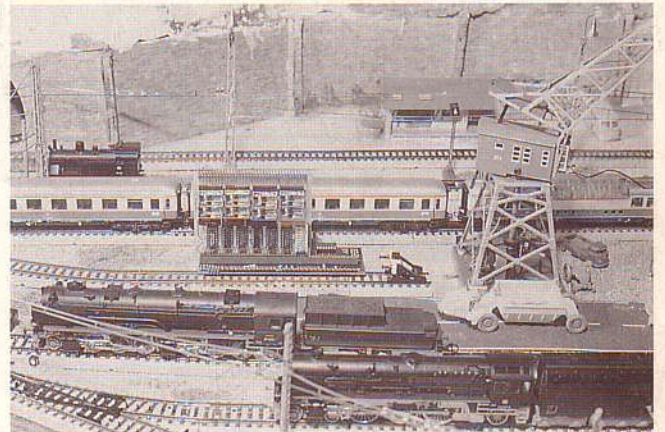
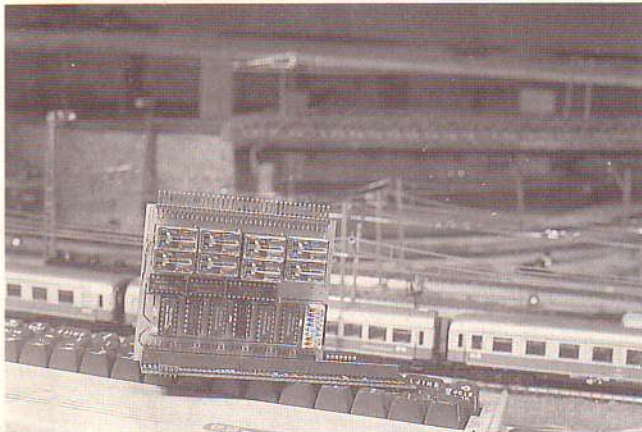
Zeitung für Computer-Bauer, -Anwender, -Programmierer und -Starter

DM 3,-

Eisenbahnsteuerung mit der REL-Baugruppe

Rolf-Dieter Klein

Schon mit dem Einsteigerpaket möglich – Eine REL-Karte steuert 24 Magnete!



Viele Eisenbahnfreunde träumen schon lange von einer computergesteuerten Eisenbahnanlage. Wie man eine Steuerung mit Hilfe des Computers durchführen kann, soll hier an zwei Beispielen gezeigt werden: zum einen für alle HEXMON-Besitzer, die ja leider bei uns bisher viel zu kurz gekommen sind und dann für die Besitzer eines C64-Adapters. Das Programm kann natürlich auch für alle anderen Versionen des NDR-Klein-Computers umgeschrieben werden, daher ist die C64-Version auch in PASCAL geschrieben.

Gesteuert werden soll eine Märklin-Eisenbahn. Märklin arbeitet mit Wechselstrom, daher wird die Lösung für Märklin natürlich auch für Gleichstrom-Eisenbahnen gültig sein.

Eine Weiche besitzt bei Märklin z.B. zwei Magnete. Ein Magnet schaltet auf Geradeausfahrt, ein zweiter auf abbiegen. Die Magnete werden dazu nur für kurze Zeit unter Strom gesetzt und sind im Normalfall stromlos.

Die REL-Baugruppe besitzt 8 Relais, mit je zwei Umschaltkontakten. Nun könnte

man mit dieser REL-Baugruppe bei kurzer Betrachtung eigentlich nur vier Weichen schalten, je ein Relais für einen Magneten. Doch es gibt einen Trick, um sogar 12 Weichen, also 24 Magnete schalten zu können: es wird dazu die sogenannte Multiplextechnik verwendet. Zwei Relais versorgen je eine Gruppe von 6 Relaiskontakten mit Strom. Je zwei Gruppen von Kontakten werden aber von einem Relais geschaltet. Da immer nur eine von beiden Gruppen aktiv ist, kommt man mit 6 Relais für die beiden Gruppen aus. Nur wird für jeden Magnetartikel ein Kontakt verwendet: der Arbeitskontakt für den einen und der Ruhekontakt für den anderen Magneten. Die beiden Auswahlrelais schalten die Umschaltrelais nur kurz ein, dadurch ergibt sich genau das gewünschte Verhalten. Bild 1 zeigt den Verdrahtungsplan.

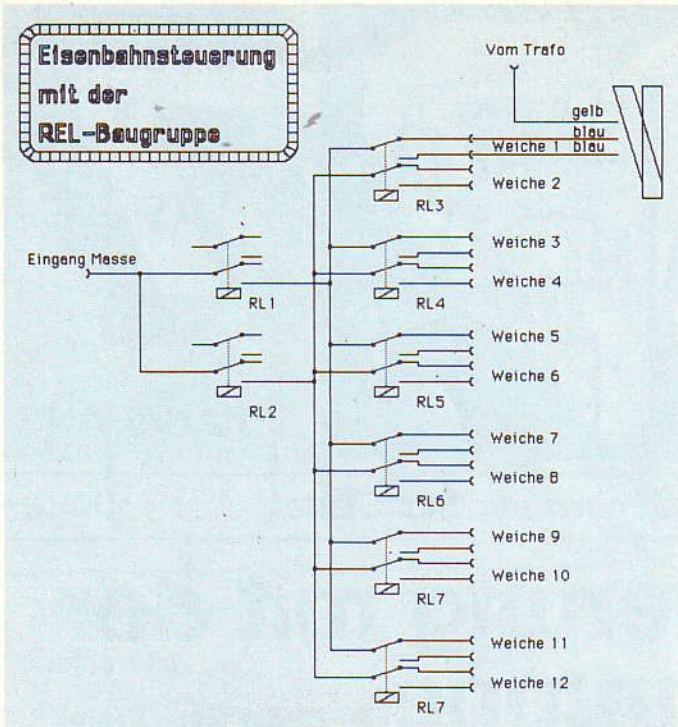
Einen kleinen Nachteil hat das Verfahren auch. Jeweils 6 Magnetartikel werden durch den Stromimpuls gleichzeitig aktiviert. Der Computer muß sich also die Stellung aller Artikel merken (was ihm nicht schwerfällt).

Beispiel für den Programmablauf: Weiche 5 soll gerade gestellt werden. Dazu wird RL5 in die entsprechende Stellung gebracht (Relais an oder aus), alle Relais, RL3, RL4, RL6 und RL7 müssen jetzt auch in die aktuelle Stellung springen, denn die restlichen Weichen werden ja auch nochmals gestellt. Nun wird RL 1 kurz eingeschaltet. Alle Weichen, 1, 3, 5, 7, 9 und 11 werden nun gestellt.

Bild 2 zeigt das Programmlisting für den HEXMON, die REL-Baugruppe wird auf Adresse 50h eingestellt (01010000 0 = Schalter an).

Die HEXIO-Tastatur wird nach dem Programmstart zum Stellen der Weichen verwendet. Übrigens kann man natürlich auch Signale o.ä. mit der Baugruppe betreiben.

Die Tasten 0 bis C steuern die 12 Weichen. Jeder Tastendruck stellt die Weiche in die entgegengesetzte Richtung. Wenn man die Taste BEF drückt, so werden alle Weichen in eine Vorzugslage gestellt.



MACRO-80 3.43 27-Jul-81 PAGE 1

```

;*****
; Eisenbahnsteuerung mit der REL-Baugruppe
; mit Hilfe von HEXMON und HEXID
;
; Rolf-Bieter Klein B&O103 1.1
;
; Mit dem Programm kann man 12 Magnet-
; artikel (Weichen oder Signale mit je
; zwei Magneten) bedienen.
;*****
aseq ; absoluter Code.

0009 anzeige equ 9 ; ins Verzögerung und Anzeige
000C holetaste equ 0ch ; Einlesen einer Taste.
000F tonum equ 0fh ; In Zahl umwandeln.

0050 cal equ 50h ; Mit DIL-Schalter einstellen.
org 8100h ; Start bei HEXID mit HEXMON

; Hauptprogramm - Start.
start:
call clear ; alle Weichen neutral stellen
call schalte ; und ausführen.
schleife:
call holetaste ; Menue und Auswahl.
cp 47h ; Befehl, dann Weichen neutral
jr z,start ; stellen.
cp 4bh ; START (n) dann eigene Tabelle laden
jr z,ladetab ;
cp 5bh ; SPEICH (n) druecken, dann Positionen merke

n
B113 28 35 jr z,meripos
B115 CD 000F call tonum ; Taste 0..9,A..F
B116 38 EC jr c,schleife ; keine Num. Taste, dann loeschen alles.
B11A 21 B1C6 ld hl,weichi ; Adresse der Tabelle laden
B11D FE 0C cp 0ch ; 0..B erlaubt, Rest ungueltig.
B11F 30 E5 jr nc,schleife ; nicht ausführen.
B121 E8 0F and 0fh ; Index darauf addieren
B123 4F ld c,a ;
B124 06 00 ld b,0 ;
B126 09 add hl,bc ; hl=Zeiger auf Weichenstellung
B127 7E ld a,(hl) ; dann komplementieren
B128 2F cpl ; und Wert ablegen
B129 77 ld (hl),a ; Weiche kann man nun ausschalten.
B12A CD B1AF call schalte ; und aktualisieren
B12D 18 D7 jr schleife ; weiter in Schleife.

B12F ; ladetab: ; Positionen von einer Tabelle
CD 000C call holetaste ; Dann 0 bis F druecken
CD 000F call tonum ; als Positionsmarker
38 CF jr c,schleife ; nein, dann zurueck.
CD B162 call null2 ; hl ist dann Startadresse
11 B1C6 ld de,weichi ; und in die aktuelle
06 0C ld b,12 ; Tabelle uebertragen

MACRO-80 3.43 27-Jul-81 PAGE 1-1

B13F lalp:
7E ld a,(hl) ; neuen Wert von Merke-Tabelle holen
12 ld (de),a ; und in aktuelle Tabelle speichern
23 inc hl ; neue Position bilden
13 inc de
10 FA djnz lalp ; Wiederholen bis alle 12 Weichen gestellt.
CD B1AF call schalte ; Weichen stellen.
18 BC jr schleife ; dann zurueck.

```

Wozu dann überhaupt einen Computer? Nun – man kann sich bestimmte Weichenstellungen merken und auf Wunsch abrufen. Dazu stellt man die Weichen mit den Tasten 0 bis C in die gewünschte Stellung und drückt dann SPEICH und eine Taste 0 bis F. Die Position ist gespeichert. Nun kann man die Weichen wieder verstellen

und drückt dann die Taste START und eine Taste 0 bis F. Dabei muß man jetzt die gleiche Ziffer drücken, die man beim Speichern angegeben hat. Dann wird die alte Weichenstellung geladen und alle Weichen dementsprechend gestellt. So kann man z.B. bestimmte Fahrstrecken vorprogrammieren, oder wenn man auch Signale verwendet, einen NOTHALT. Das

```

B14A CD 000C meripos:
B14D CD 000F call holetaste ; Dann 0 bis F druecken
B150 38 B4 jr c,schleife ; als Positionsmarker
B152 CD B162 call null2 ; hl ist dann Startadresse
B155 11 B1C6 ld de,weichi ; und in die aktuelle
B158 06 0C ld b,12 ; Tabelle uebertragen
B15A merip:
1A ld a,(de) ; Wert aus aktueller Tabelle laden
B15B 77 ld (hl),a ; und in Merke-Tabelle ablegen.
15C inc hl
B15D 13 inc de
B15E 10 FA djnz merip
B160 18 A4 jr schleife ; dann zurueck.

B162 null2:
4F ; aktu*12 + Tabellenadresse
06 00 ld c,a
CD 05 ld b,0 ; Index * 12
CD 05 push bc
29 pop hl
09 add hl,hl ; *2
09 add hl,bc ; *2 + x
29 add hl,hl ; (*2 + x) * 2
29 add hl,hl ; (*2 + x) * 2 * 2
11 B1D2 ld de,tabelle ; laden.
19 add hl,de ; Startadresse der Tabelle
C9 ret

B170 clear:
21 B1C6 ld hl,weichi ; alle Weichen neutral stellen
06 0C ld b,12 ; 12 Weichen.
B175 36 00 cpl ;
B177 23 02 ld (hl),0 ; 0=neutral
10 FB inc hl
B17A C9 djnz cpl ;
ret

; Unterprogramm, dass die Weichen stellt.
; Dabei steht im Akku der Wert 1 oder 3
; jenachdem, welche Weichengruppe
; geschaltet werden soll.
B17B F5 pulset: ; A=1 oder 3, je nach Schaltgruppe.
B17C CD B191 push af ;
B17F F1 call del100 ; warten bis andere Relais ein.
pop af

MACRO-80 3.43 27-Jul-81 PAGE 1-2

B180 D3 50 out (rel),a ;
B182 CD B191 call del100 ; 100 ms warten
B185 3E 00 ld a,0 ; RL 0 ausschalten
B187 D3 50 out (rel),a ;
B189 3E 02 ld a,2 ; RL 1 auch
B18B D3 50 out (rel),a ;
B18D CD B191 call del100 ; warten bis aus.
B190 C9 ret ; Unterprogramm Ende

B191 del100:
06 64 ld b,100 ; ca. 100 Millisekunden
B192 3E 00 warte: ;
B193 CD 0009 call anzeige ; verzögert um las
B196 10 FB djnz warte ; 100 Mal durchlaufen
B198 C9 ret ;

; Unterprogramm stellt die Relais gemäss
; der Tabelle.
; hl=Adresse von 6 Speicherzellen
; die die Stellung (0,offh) beinhalten.
B199 06 06 einstell:
0E 04 ld b,6 ; Anzahl der Relais
ld c,4 ; Stellindex RL 1.. RL 8
einip:
7E ld a,(hl) ; Stellung prüfen
or a ; <0 dann weiter
B19E 87 jr nz,einl ; = 0 dann
B19F 20 03 jr nz,einl ; Relais aus
B1A1 79 ld a,c ;
B1A2 18 03 jr ein2 ; und weiter
B1A4 einl:
79 ld a,c ; <0 , also
C6 01 add a,1 ; Relais an
B1A7 3E 00 or a ; hier weiter.
B1A8 33 50 out (rel),a ; und ausführen
B1A9 0C inc c ; neuer Code
B1AA 0C inc c ; in 2er Schritten.
B1AB 23 inc hl ; neue Position.
B1AC 10 EF djnz einip ; bis alle Positionen aktuell
B1AE C9 ret ;

; Unterprogramm stellt alle Weichen
; auf aktuellen Stand gemäss der Speichertabelle.
B1AF schalte:
21 B1C6 ld hl,weichi ; Relais Gruppe 1 vorbereiten
CD B199 call einstell ;
3E 01 ld a,1 ; Gruppe 1 schalten
B1B7 CD B17B call pulset ;
21 B1CC ld hl,weichi2 ; Relais Gruppe 2 vorbereiten
CD B199 call einstell ;
3E 03 ld a,3 ; Gruppe 2 schalten
B1C2 CD B17B call pulset ;
B1C5 C9 ret ;

MACRO-80 3.43 27-Jul-81 PAGE 1-3

B1C6 Speicher fuer aktuelle Position der einzelnen Weichen.
B1C8 00 00 00 00 ; 0=neutral, 0ffh=gestellt.
B1CA 00 00 weichi:
defb 0,0,0,0,0,0 ; Gruppe 1
B1CC 00 00 weichi2:
defb 0,0,0,0,0,0 ; Gruppe 2
B1D0 00 00 ;

B1D2 tabelle: ; Stellungen, die bei Druecken
; von START eingelesen werden soll.
B1D2 defb 12*16 ; 16 verschiedene Felder moeglich

end

```

```

MACRO-80 3.43 27-Jul-81 PAGE 1-3

B1C6 Speicher fuer aktuelle Position der einzelnen Weichen.
B1C8 00 00 00 00 ; 0=neutral, 0ffh=gestellt.
B1CA 00 00 weichi:
defb 0,0,0,0,0,0 ; Gruppe 1
B1CC 00 00 weichi2:
defb 0,0,0,0,0,0 ; Gruppe 2
B1D0 00 00 ;

B1D2 tabelle: ; Stellungen, die bei Druecken
; von START eingelesen werden soll.
B1D2 defb 12*16 ; 16 verschiedene Felder moeglich

end

```

Programm kann sich 16 Positionen merken.

C64-Adapter

Bild 3 zeigt das Listing in PASCAL. Es wurde hier das OXFORD-Pascal für den C64 verwendet. Das Programm ist aber auch leicht in Pascal/S für den NDR-Klein-Computer umschreibbar.

```

1 PROGRAM BAHN(INPUT,OUTPUT);
2 (* STEUERUNG MIT REL-BAUGRUPPE *)
3 (* VON ROLF-DIETER KLEIN *)
4 (* VERSION 1.1 860105 *)
5 (* IN OXFORD PASCAL *)
6 (* MIT C64-ADAPTER *)
7 CONST
8   IO = 56032; (* BASISADRESSE IO *)
9   REL = 80; (* OFFSET REL AUF 50H *)
10 VAR
11   STELLUNG : ARRAY [1..12] OF INTEGER;
12   TABELLE : ARRAY [1..12,0..9] OF INTEGER;
13   CH : CHAR;
14 (* *)
15 (* *)
16 (* *)
17 PROCEDURE DELAY;
18 VAR I:INTEGER; A:REAL;
19 BEGIN
20   A := 0;
21   FOR I:=1 TO 50 DO
22     A := A * 1.1;
23   END;
24 (* *)
25 (* *)
26 (* *)
27 PROCEDURE PULSE(X:INTEGER);
28 BEGIN
29   DELAY;
30   POKE(IO+REL,X); (* RELAIS AN *)
31 DELAY;
32   POKE(IO+REL,0); (* RL 1 AUS *)
33   POKE(IO+REL,2); (* RL 2 AUS *)
34 DELAY;
35 END;
36 (* *)

```

```

37 (* *)
38 (* *)
39 PROCEDURE EINSTELL(N:INTEGER);
40 VAR I,J : INTEGER;
41 BEGIN
42   J := 4; (* STARTWERT *)
43   FOR I:=1 TO 6 DO BEGIN
44     IF STELLUNG[I+J]=0 THEN
45       POKE(IO+REL,J);
46     ELSE
47       POKE(IO+REL,J+1);
48     J := J + 2; (* NEUER CODE *)
49   END;
50 END;
51 (* *)
52 (* *)
53 (* *)
54 PROCEDURE SCHALTE;
55 BEGIN
56   EINSTELL(0);
57   PULSE(1);
58   EINSTELL(6);
59   PULSE(3);
60 END;
61 (* *)
62 (* *)
63 (* *)
64 PROCEDURE CLEAR;
65 VAR I: INTEGER;
66 BEGIN
67   FOR I:=1 TO 12 DO STELLUNG[I]:=0;
68 END;
69 (* *)
70 (* *)
71 (* *)

```

```

72 PROCEDURE HOLE(K:INTEGER);
73 VAR I: INTEGER;
74 BEGIN
75   FOR I:=1 TO 12 DO
76     STELLUNG[I]:=TABELLE[I,K];
77   END;
78 (* *)
79 (* *)
80 (* *)
81 PROCEDURE SPEICHERE(K:INTEGER);
82 VAR I:INTEGER;
83 BEGIN
84   FOR I:=1 TO 12 DO
85     TABELLE[I,K]:=STELLUNG[I];
86   END;
87 (* *)
88 (* *)
89 (* *)
90 PROCEDURE UMSCHALT(I:INTEGER);
91 BEGIN
92   IF STELLUNG[I] = 0 THEN
93     STELLUNG[I] := 1;
94   ELSE
95     STELLUNG[I] := 0;
96   END;
97 (* *)
98 (* *)
99 (* *)
100 BEGIN
101   CLEAR;
102   WRITELN('BEREIT');
103   SCHALTE;
104   REPEAT
105     REPEAT
106       CH := GETKEY
107     UNTIL CH (<) CHR(0);

```

```

108 CASE CH OF
109   'A' : UMSCHALT(1);
110   'S' : UMSCHALT(2);
111   'D' : UMSCHALT(3);
112   'F' : UMSCHALT(4);
113   'G' : UMSCHALT(5);
114   'H' : UMSCHALT(6);
115   'Z' : UMSCHALT(7);
116   'X' : UMSCHALT(8);
117   'C' : UMSCHALT(9);
118   'V' : UMSCHALT(10);
119   'B' : UMSCHALT(11);
120   'N' : UMSCHALT(12);
121   : CLEAR;
122   : BEGIN
123     REPEAT
124       CH := GETKEY
125     UNTIL CH (<) CHR(0);
126   WRITELN('SPEICHERN', ORD(CH)-ORD('0'));
127   SPEICHERE(ORD(CH));
128   REPEAT
129     CH := GETKEY
130   UNTIL CH (<) CHR(0);
131   WRITELN('LADEN', ORD(CH)-ORD('0'));
132   HOLE(ORD(CH));
133   REPEAT
134     CH := GETKEY
135   UNTIL CH (<) CHR(0);
136   UMSCHALT(ORD(CH));
137   END;
138 END;
139 SCHALTE;
140 UNTIL CH = '=';
141 END.

```

Hier wird die C64-Tastatur als Eingabe verwendet. Die Tasten A, S, D, F, G, H, Z, X, C, V, B, N dienen der Weichenumschaltung. Mit der Leertaste kann die Grundstellung einstellen. Mit ", " wird eine Weichenstellung abgespeichert und mit ", " kann man sie wieder abrufen. Danach gibt man jeweils eine Zahl 0 . . . 9 ein, dementsprechend kann sich das Programm 10 Stellungen merken. Da die Relais eine Verzögerung zum Schalten benötigen, wird hier eine Warteschleife mit dem Namen DELAY benötigt. Die Warteschleife sollte ca. 1/2 bis 1/10 Sekunden lang sein.

Achtung, wenn man diese Schleife auf den NDR-Klein-Computer überträgt, so wird sie nicht mehr lang genug dauern, da 50 Additionen auf dem NDR-Klein-Computer schneller ablaufen als auf dem C64. Die Relais-Baugruppe wird auch auf Adresse 50h eingestellt und JMP6 auf dem C64-Adapter ist eingesetzt.

Achtung, wenn man eine Taste drückt, die nicht in der CASE-Anweisung vorkommt, wird das Programm mit einem Fehler unterbrochen. Doch Erweiterungen überlasse ich dem Leser.

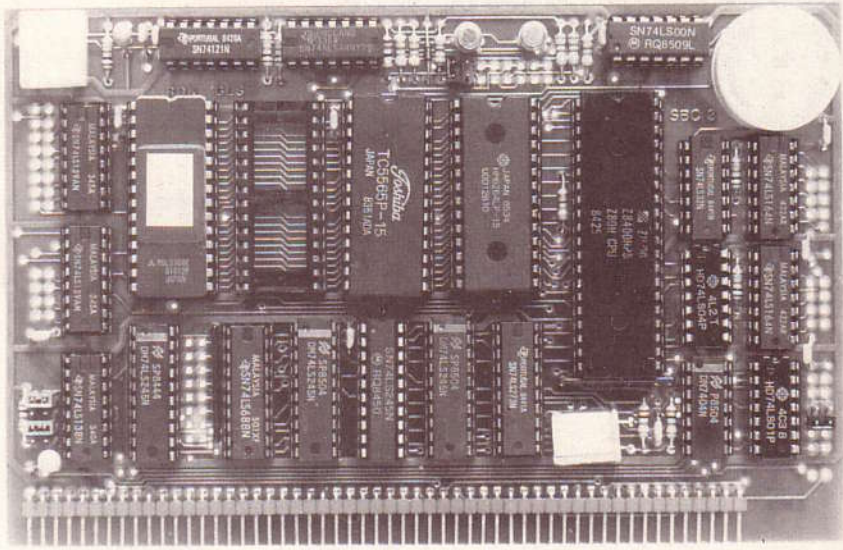
Hier noch ein wichtiger Hinweis. Da alle Magnete starke Störungen beim Ein- und Ausschalten hervorrufen können, muß man besonderen Wert auf gute Leitungsführung und Erdung legen. Man sollte auch kleine Kondensatoren (ca. 100 nF) über alle Kontakte schalten, um die Funkenverzögerung so klein wie möglich zu halten. Die Software sorgt übrigens dafür, daß Funken nur bei den Relais RL 1 und RL 2 auftreten können. Wenn man nicht ausreichend entstört hat, kann es vorkommen, daß beim Schalten der Weichen der Computer auch gleich aussteigt. Das Programm sollte man daher zunächst einmal ohne Weichen, nur mit der REL-Baugruppe testen und dann unbedingt irgendwo abspeichern, bevor es ernst wird.

Hinweis: Akku-Pufferung der SBC2 in LOOP 5, Seite 2!

Der C64-Adapter wird in der nächsten LOOP beschrieben!

Hinweise und Tips zum Entstörproblem bitte an die LOOP-Redaktion.

Die SBC3 ist lieferbar



Die SBC3 ist eine universelle CPU- bzw. SBC-Baugruppe vom Einsteiger-Paket bis zum professionellen CP/M-Computer mit Floppy-Laufwerken und Winchester.

Dabei kann sie zum einen als Single Board Computer, wie die SBC2, verwendet werden, allerdings mit 16 ROM und 16 K RAM in der Grundeinstellung. Andererseits kann die SBC3 auch als CPU-Baugruppe mit Boot-Logik für CP/M verwendet werden. Dann ersetzt sie die Baugruppe CPU Z80 und Bankboot. Diese beiden Betriebsarten können mit 3 Brücken (JUMPER) ein- bzw. umgestellt werden.

Die SBC3 hat aber noch ein paar Besonderheiten, die die SBC2 und die CPU Z80 mit BANKBOOT nicht haben:

1. Auf der SBC3 kann die schnellste Z80 CPU eingesetzt werden (Z80H mit 8 MHz).
2. Es sind verschiedene RAM- und EPROM-Typen einsetzbar
 - RAM: 8K und 2K (z.B. 6264 und 6116)
 - EPROM: 4K, 8K, 16K und 32K (2732, 2764, 27128 und 27256)

3. Akkupufferung der RAMs: Dies ist besonders vorteilhaft, wenn man im Grundprogramm, Basic, GOSI usw. Programme geschrieben hat und diese auch nach dem Ausschalten erhalten bleiben. Damit erspart man sich das lästige Speichern auf Kassettenrekorder. Der Akku hält die Speicherinformation ca. 1 Jahr; er wird aber im Betrieb ständig nachgeladen.

4. WAIT-Logik für Speicherzugriff und I/O getrennt: Bei der Z80H mit 8 MHz können manche Speicher und I/O Einheiten (z.B. FLO2) unter Umständen zu langsam sein. Für diesen Fall können jeweils bis zu 4 WAIT-Zyklen eingestellt werden.

Die Preise:

SBC3H	Handbuch	10,00 DM
SBC3P	Leiterplatte	49,50 DM
SBC3B	Bausatz	187,60 DM
SBC3F	Fertiggerät	258,00 DM

Lieferzeit: Ab Lager!



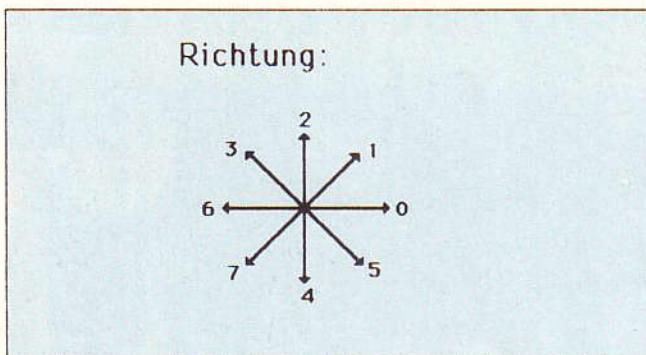
Der FIGUR-Befehl mit dem Z80

von Uwe Koch

Dieses kleine Programm ist eine Erweiterung für das Grundprogramm. Das Programm bildet den FIGUR-Befehl des 68008-Grundprogramms (wie in Schritt für Schritt 2 beschrieben) für den Z80 nach. Um das Programm möglichst einfach und kurz (und damit schnell) zu machen, habe ich allerdings auf die Umcodierung der Vektorrichtungen ver-

zichtet. Es werden hier also die Richtungsnummern des GDP verwendet. (Siehe Skizze neben Listing). Das Programm erwartet in HL die Anfangsadresse der Vektortabelle. In der Tabelle dürfen die Werte 0 bis 7 für die acht Richtungen, eine 8 für unsichtbar weiterzeichnen, eine 9 für widersichtbarzeichnen oder eine 10 für Ende stehen. (10/O Ah).

Werte, die größer als 10 sind, werden durch die OR-Verknüpfung mit F8H zwar in den Bereich 0-7 gebracht, jedoch sollte man sie wegen der lästigen Umrechnung nicht verwenden. Das Programm zeichnet in der jeweils angegebenen Richtung einen Kurzvektor von 3 Punkten Länge. Damit lassen sich sehr schnell auch „vielseckige“ Figuren zeichnen.



```

; TITLE FIGUR ROUTINE
; *****
; ORG 8800H
; WAIT EQU 003BH ;WARTE BIS
; ; GDP FERTIG
;
; LD HL,TABELLE
; CALL FIGUR
; RET
FIGUR: LD A,(HL) ;ERSTEN WERT
; LADEN
; INC HL ;ZEIGER UM 1
; CP 08H ;ERHOEHEN
; ; UNSICHTBAR?
;
8800 21 3A 88
8803 CD 07 88
8806 C9
8807 7E
8808 23
8809 FE 08
    
```

```

880B 20 06 JR NZ,A ;NEIN DANN
880D CD 32 88 ;OR 07H
8810 C3 07 88 ;JA DANN
8813 FE 09 ;STIFT HOCH
8815 20 06 ;ZURUECK
8817 CD 2A 88 ;STIFT
881A C3 07 88 ;SENKEN?
881D FE 0A ;JA DANN
881F CB ;STIFT
8820 F6 F8 ;SENKEN
8822 CD 38 00 ;ZURUECK
8825 D3 70 ;ZURUECK INS
8827 C3 07 88 ;HAUPTPROG.
; Vektors
; GDP FERTIG?
; AUSGABE AN
; GDP
; ZURUECK ZU
; FIGUR
;
; SENK:
882A 3E 00 LD A,00H ;SCHREIB-
882C CD 3B 00 CALL OUT ;FUNKTION
882F D3 70 OUT ;SETZEN
8831 C9 RET
;
; HEB:
8832 3E 01 LD A,01H ;LOESCH-
8834 CD 3B 00 CALL OUT ;FUNKTION
8837 D3 70 OUT ;SETZEN
8839 C9 RET
;
; TABELLE:
883A 00 00 00 02 02 02 03 DB 0,0,0,02,02,02,03
8841 07 04 04 04 00 00 00 DB 07,04,04,04,0,0,0
8848 08 DB 08
8849 04 04 04 06 06 DB 04,04,04,06,06
884E 09 DB 09
884F 05 07 03 01 DB 05,07,03,01
8853 0A DB 0AH
;
;
;
;
885B WAIT
8813 A
882A SENK
883A TABELLE
8807 FIGUR
881D B
8832 HEB
    
```

ComFood – Günstige CP/M 2.2-Software

Vor einiger Zeit flatterte uns der Katalog der Firma ComFood ins Haus. ComFood liefert sogenannte „Public Domain“-Software, speziell für CP/M 2.2. Besonders interessant – die Programme werden bereits im NDR-Computer-Format oder im mc-CP/M-Format (ECMA 70) geliefert. Die Disketten kosten alle um die 20,-DM und enthalten so vielversprechende Titel wie Adventure, Pascal-Tools, Musicraft, Games, Forth Utilities, Wordstar Footnotes usw.

Dies nur als Hinweis, wir haben noch keines dieser Programme ausprobiert. Wer Erfahrungen gemacht hat, wende sich bitte an die LOOP-Redaktion!

Die Adresse von ComFood:
Flaßkamp 24, 4400 Münster,
Telefon: 0251/719768



Hannover-Messe CEBIT und GES

Hannover CEBIT – der NDR-Computer gleich in zwei Hallen!

Wir stellen in Hannover CEBIT in zwei Hallen aus: Einmal unser Firmenstand in Halle 13, Stand 317 und dann in Halle 14 im „Computer-Camp“. Besuchen Sie uns – wir zeigen interessante Neuigkeiten; was, wird noch nicht verraten! Natürlich ist die LOOP-Redaktion ebenfalls vertreten!

Besserer Service bei GES: Telefondienst erweitert!

Unter unserer Nummer 0831/6211 können Sie uns anrufen:

Montag, Dienstag, Donnerstag:
7.30 – 12.00 Uhr, 13.00 – 18.00 Uhr;
Mittwoch:
7.30 – 12.00 Uhr, 13.00 – 20.00 Uhr;
Freitag: 7.30 – 12.00 Uhr.

Technische Fragen stellen Sie bitte nur nachmittags und abends, ab 13.00 Uhr.

Am Freitag machen wir nicht etwa ein langes Wochenende, sondern sitzen alle in unserem Schulungsraum zur internen Schulung beisammen – um Sie noch besser beraten zu können.

**Hannover Industrie –
wir sind auch dabei!**

 **Hannover
Messe '86**
INDUSTRIE 9.-16. 4. 1986

Die Hannover Messe ist ab diesem Jahr zweigeteilt. Natürlich sind wir auch auf der Industrie-Messe, die vom 9. – 16. April stattfindet, dabei. Wo? Wie im Vorjahr, in der **microtronic-Halle 12, 2. Obergeschoß, Stand 2321!**

BASIC-Ausgaben von Zeichen an beliebiger Stelle und in verschiedenen Größen

Oft wird unter dem NDR-BASIC-Interpreter gewünscht, Zeichen an bestimmten X-, Y-Koordinaten auszugeben.

Da das BASIC eine solche Anweisung (noch) nicht kennt, muß man den GDP direkt auf die Koordinaten setzen.

Dies ist leicht möglich – der Baustein hat zwei Registerpaare, eins für den X-, eins für den Y-Wert. In diese Registerpaare kann mit einem OUT-Befehl direkt von BASIC ein Wert eingetragen werden. Der GDP zeichnet oder schreibt dann ab diesem Wert.

In ein weiteres Register (CSIZE) kann die Größe der Zeichen eingegeben werden.

Wie das gemacht wird, geht aus dem absichtlich etwas umständlich programmierten BASIC-Listing hervor.

```

10 REM Ausgabe von Buchstaben an einer beliebigen Stelle
12 CLRS
15 REM U=Schriftbreite
16 REM Z=Schrifthoehe
20 REM X = X-Koordinate
30 REM Y = Y-Koordinate
35 REM B = Anzahl der ASCII Zeichen
40 REM A = ASCII-Konstante
50 REM S = Speicher
60 S = 49152
70 INPUT "String eingeben";A#
80 PRINT:PRINT
90 INPUT "X - Koordinate";X
100 PRINT:PRINT
110 INPUT "Y - Koordinate";Y
111 PRINT:PRINT
112 INPUT "Schrifthoehe";U
114 PRINT:PRINT:INPUT "Schriftbreite";Z
116 OUT 115,16#Z+U
120 CLRS
125 LET B=LEN (A#)
130 FOR C=1TOB
140 LET B#= MID$( A#,C,1)
150 LET A=ASC(B#)
155 POKE S,A :REM Buchstabe speichern
160 LET S=S+1
170 NEXT C
180 LET S=49152
200 IF Y > 255 THEN LET Y=255
210 IF X > 511 THEN LET X=511
300 OUT 122,0 :REM MSB-Register der Y-Koordinate auf Null setzen
310 OUT 123,Y :REM LSB-Register der Y-Koordinate setzen
315 REM MSB-Register muss bei der X-Koordinate beruecksichtigt werden
320 IF X > 255 THEN GOTO 400
330 OUT 120,0 :REM sonst MSB-Register loeschen
340 OUT 121,X :REM LSB-Register der X-Koordinate setzen
350 GOTO 445 :REM wenn X < 255 dann Sprung nach 445
400 LET X=X-256 :REM MSB-Register wird auf 1 gesetzt d. entspricht 255
410 OUT 120,1 :REM MSB-Register der X-Koordinate auf 1 setzen
420 OUT 121,X :REM LSB-Register der X-Koordinate wird gesetzt
445 FOR F = 1 TO B
450 LET A=PEEK(S):REM Buchstabe aus dem Speicher holen
460 OUT 112,A :REM Buchstabe an GDP ausgeben
470 LET S=S+1
480 NEXT F
490 FOR F=1TO1000
500 LET G=1/F
510 NEXT F
515 OUT 115,17 :REM Normale Schriftgroesse einstellen
520 END
    
```

Tips und Tricks bei HEBAS, Nr. 3

Dr. Hans Hehl

In *LOOP* Nr. 5 wurde im Artikel *Tips und Tricks bei HEBAS* zur Analyse und zum Ändern von BASIC-Programmen mit dem Grundprogramm (im EPROM auf Adresse 2000H) ein Verschiebeprogramm erläutert. Für die neuen *LOOP*-Abonnenten ist der „HEX-DUMP“ dieses Programmes als Bild Nr. 4 nochmals aufgeführt.

Nun soll der Aufbau von BASIC-Programmen im Speicher an einem kleinen Musterprogramm erläutert werden, das Bild Nr. 1 zeigt.

Geben Sie das Musterprogramm ein und speichern Sie es ab. Mit einem Debugger erhält man den in Bild Nr. 2 abgebildeten HEXDUMP des ab Adresse 45D8H abgelegten Programmes, da das Programm im Speicher erhalten bleibt.

Die Erklärung der einzelnen Bytes zeigt Tabelle Nr. 1. Die Basic-Befehle wie z.B. INPUT sind in die zugehörigen Tokens (Byte-Befehl) umgewandelt.

Speichert man das Programm dagegen mit dem BASIC-Befehl SAVE „Dateiname“, A als ASCII-Text ab, erhält man den in Bild Nr. 3 gezeigten HEXDUMP. Jede Programmzeile schließt jetzt mit Carriage Return und Linefeed ab. Am Schluß befindet sich das Dateieinde-Zeichen 1AH (Control-Z).

So abgespeicherte BASIC-Programme lassen sich mit Textverarbeitungsprogrammen im sog. Programm-Modus bequem bearbeiten.

Der BASIC-Befehl NEW

Untersuchen Sie nun die Wirkungsweise des BASIC-Befehls NEW. Nachdem Sie das Musterprogramm mit LIST aufgezeigt haben, geben Sie den Befehl NEW ein. Das Programm läßt sich nicht mehr mit dem Befehl LIST anschauen. Verlassen Sie HEBAS mit dem BYE-Befehl, starten Sie einen Debugger (z.B. DDT.COM) und vergleichen Sie die ersten 2 Bytes ab Adresse 45D8H mit den Bytes in Bild

```

1 ' Musterprogramm
3 INPUT A
5 PRINT A
7 A = A + 5
9 PRINT A
11 END

```

Bild Nr.1: Das zu analysierende Musterprogramm

```

45D7 00 ED 45 01 00 A0 20 4D 75 73 74 65 72 70 72 6F
45E7 67 72 61 6D 6D 00 F5 45 03 00 84 20 41 00 FD 45
45F7 05 00 95 20 41 00 0D 46 07 00 41 20 CD 20 41 20
4607 C0 20 12 20 83 00 15 46 09 00 95 20 41 00 1B 46
4617 0B 00 AF 00 00 00

```

Bild Nr.2: HEXDUMP des Musterprogrammes (Tokenform)

```

0100 31 20 27 20 4D 75 73 74 65 72 70 72 6F 67 72 61 1 ' Musterprogra
0110 6D 6D 0D 0A 33 20 49 4E 50 55 54 20 41 0D 0A 35 mm..3 INPUT A..5
0120 20 50 52 49 4E 54 20 41 0D 0A 37 20 41 20 3D 20 PRINT A..7 A =
0130 41 20 2B 20 35 0D 0A 39 20 50 52 49 4E 54 20 41 A + 5..9 PRINT A
0140 0D 0A 31 31 20 45 4E 44 0D 0A 1A 00 00 00 00 00 ..11 END.....

```

Bild Nr.3: HEXDUMP des Musterprogrammes (Langform für Textverarbeitung)

```

8800 21 00 20 11 00 A0 01 00 20 ED B0 3E B0 D3 C8 21
8810 00 A0 11 00 20 01 00 20 ED B0 C3 00 20 00 00 00

```

Bild Nr.4: HEXDUMP des Verschiebeprogrammes aus LOOP Nr.4

45D7	00	Erstes Byte immer 0
45DB - 45D9	LD 45	2. und 1. Byte ergeben die Adresse der nächsten Programmzeile.
45DA	01	Low-Byte, Zeilennummer 1
45DB	00	High-Byte
		3. und 4. Byte enthalten die Zeilennummer (Low-Byte + High-Byte * 256)
45DC	A0	Token für ' (Rem in Kurzform)
45DD	20	Leerzeichen (blank)
45DE - 45EB		ASCII-Text: Musterprogramm 4D 75 73 74 65 72 70 72 6F 67 72 61 6D 6D
45EC	00	Ende der Programmzeile
45ED - 45EE	F5 45	Adresse der nächsten Basic-Zeile
45EF - 45F0	03 00	Zeilennummer 3
46F1	84	Token für INPUT
46F2	20	Leerzeichen
46F3	41	ASCII-Zeichen für A = 41h
46F4	00	Ende der Programmzeile
45F5 - 45F6	FD 45	Adresse der nächsten Basic-Zeile
45F7 - 45FB	05 00	Zeilennummer 5
45F9	95	Token für PRINT
45FA	20	Leerzeichen
45FB	41	ASCII-Zeichen A = 41h
45FC	00	Ende der Programmzeile
45FD - 45FE	0D 46	Adresse der nächsten Basic-Zeile
45FF - 4600	07 00	Zeilennummer 7
4603	CD	Token für =
4607	C0	Token für +
4609 - 460B	12 20 83	Schlüssel für die Zahl 5
460D - 460E	15 46	Adresse der nächsten Basic-Zeile
460F - 4610	09 00	Zeilennummer 9
4611 - 4613	95 20 41	Token für PRINT, Leerzeichen, ASCII-Zeichen A = 41h
4614	00	Ende der Programmzeile
4615 - 4616	1B 46	Adresse der nächsten Basic-Zeile
4617 - 4618	0B 00	Zeilennummer 11
4619	AF	Token für END
461B - 461C	00 00	Ende des Programmes, da keine Adresse mehr angegeben.

Tabelle Nr.1: Aufbau und Ablage eines BASIC-Programmes im Speicher

Nr. 2. Es fehlt die Adresse der nächsten BASIC-Programmzeile. Es wird also nur der Zeiger auf die nächste Programmzeile gelöscht, nicht das Programm. Leider kann jetzt kein Warmstart von HEBAS durchgeführt werden, da HEBAS zum Teil vom Debugger überschrieben wird.

Wer nun experimentieren möchte, sollte BASIC-Programme ab Adresse 8800 ablegen (in LOOP Nr. 5 beschrieben: HEBAS.COM mit DDT.COM abändern, ab Adresse 46BA müssen die Bytes D8 45 durch 00 88 ersetzt werden), dann kann man mit dem Grundprogramm das im Speicher abgelegte BASIC-Programm

abändern. Wird der Zeiger wieder hergestellt, d.h., die Bytes 15 und 88 ab Adresse 8800 wurden mit dem Grundprogramm eingegeben (das Byte 87FF muß den Wert 0 besitzen), so zeigt nach dem Reset und dem Warmstart von HEBAS (siehe Handbuch HEBAS) der BASIC-Befehl LIST wieder das Programm.

Neuer Prospekt – Bitte um Mitarbeit!

Wir planen einen umfangreichen, vierfarbigen Katalog über den NDR-Computer, den mc-CP/M-Computer und unsere sonstigen Produkte zu erstellen. Hier wollen wir speziell auf den Anfänger eingehen und in diesem Katalog die nicht ganz einfach zu erklärende Hardware und Software beschreiben.

Sie waren einmal Anfänger – schreiben Sie uns doch bitte, was Sie gern in einem solchen Katalog lesen würden. Haben Sie Ideen? Wie könnte man den komplexen NDR-Computer (welche Baugruppe mit welcher CPU, welche Software mit welcher Hardware, wenn Einsteiger, dann ...) grafisch und textlich darstellen?

Vielen Dank im Voraus für Ihre Mitarbeit!

Biorhythmus in BASIC

oder: wie geht's mir heute?

Gerhard Kallmeyer
Im Hesppe 25, 3008 Garbsen 4

Mit diesem kleinen BASIC-Programm, das schon mit der SBC2 und dem BASIC-Interpreter läuft, ist es möglich, den Biorhythmus auszudrucken.

Für die, die es nicht wissen: Beim Biorhythmus geht man davon aus, daß jeder Mensch in den Bereichen körperlich, seelisch und geistig Höhe- und Tiefpunkte hat. Der Verlauf dieser Punkte entspricht dem von Sinuskurven, deren Periodendauer unterschiedlich ist und die bei der Geburt mit bestimmten Werten starten.

Das vorliegende Programm berechnet nun (nach Eingabe des Geburtstages)

diese Kurven und stellt sie am Bildschirm dar.

Für 68000-Anwender gibt es auch ein Bio-Rhythmus-Programm EBIO, das über die Bibliotheksverwaltung aufgerufen werden kann.

Da, falls wir das Programm setzen lassen, Fehler schon vorprogrammiert sind, übernehmen wir im Original die Aufzeichnungen des Autors. Das Programm kann natürlich noch erweitert werden – z.B. könnte man die GDP auf unterschiedliche Strichdarstellung umprogrammieren. Hinweis: OUT 114, 1, 2, 3 oder 0, siehe auch LOOP 4, „Graphik auf dem NDR-Computer“.

```

10 REM B I O R H Y T H M U S Programm und Graphik in Basic mit SBC2
20 REM Eingabe Geburts Datum und Vorhersage Datum
30 CLRS
40 PRINT "B I O R H Y T H M U S"
50 PRINT:PRINT
60 INPUT "Geburts Tag :";A
70 INPUT "Geburts Monat:";B
80 INPUT "Geburts Jahr :";C
90 GOTO 110
100 CLRS
110 PRINT
120 INPUT "Vorhers.Tag :";D
130 INPUT "Vorhers.Monat:";E
140 INPUT "Vorhers.Jahr :";F
150 G=A+B*30.4+C*365.25
160 H=D+E*30.4+F*365.25-G

200 REM Ausgabe der errechneten Biorhythmus-Werte
210 CLRS
220 PRINT TAB(15);"## B I O R H Y T H M U S ##"
230 PRINT:GOSUB 1010
240 PRINT TAB(5);"Geburts Dat. : ";A;" ";B;" ";C;TAB(45);"Maerperl. : ";L;" %"
250 I=28:GOSUB 1010
260 PRINT TAB(5);"Vorhers.Dat. : ";D;" ";E;" ";F;TAB(45);"Seelisch : ";L;" %"
270 I=33:GOSUB 1010
280 PRINT TAB(45);"Geistig : ";L;" %"
290 PRINT
300 PRINT "0 = Ende 1 = neues Geb.Datum 2 = neues Vorh.Datum 3 = Graphik"
310 GOSUB 1310
320 GOTO 610

400 REM Ausgabe der Graphik
410 PRINT
420 PRINT TAB(9);"%";TAB(20);"Vorhersage fuer Monat : ";E;" ";F
430 FOR M = 100 TO 0 STEP -20
440 PRINT:PRINT TAB(7);M
450 NEXT
460 PRINT TAB(11);0;TAB(31);10;TAB(51);20;TAB(71);30;TAB(91);"Tag"
470 Page 0,0

```

1399

974

```

480 MOVETO 76,140:DRAWTO 76,38:DRAWTO 448,38
490 GOSUB 1210
500 I=23:GOSUB 1110
510 I=28:GOSUB 1110
520 I=33:GOSUB 1110
530 POKE HEX("87CS"),0

```

671

```

600 GOSUB 1310
610 CLRS:END

```

652

```

1000 REM Errechnung des Biorhythmus
1010 K=SIN((H-INT(H/1))*1)/1*2*3.14)
1020 L=50+INT(50*K+1.4)
1030 RETURN

```

562

```

1100 REM Graphische Darstellung der Kurven
1110 H=E*30.4+F*365.25-G
1120 GOSUB 1010
1130 MOVETO 76,L+40
1140 FOR X=1 TO 62
1150 H=X/2+E*30.4+F*365.25-G
1160 GOSUB 1010
1170 DRAWTO X+6*76,L+40
1180 NEXT:RETURN

```

393

```

1200 REM Tagesmarkierung
1210 FOR N=45 TO 135 STEP 10
1220 MOVETO 76+12*D;N:DRAWTO 76+12*D;N+5
1230 NEXT:RETURN

```

311

```

1300 REM Programmwahl
1310 INPUT P
1320 IF P=1 THEN 30
1330 IF P=2 THEN 100
1340 IF P=3 THEN 410
1350 RETURN

```

230

Farbe für den NDR-Computer – speziell für Z80-Anwender

Teil 2

Rolf-Dieter Klein

Im letzten Teil wurden Unterprogramme zum Betrieb mit dem 68008 Grundprogramm besprochen. Hier sollen die Unterprogramme für die Z80-Besitzer behandelt werden.

Beim Z80 wird die COL256 auf Adresse 0C000H bis 0FFFFH eingeblendet, genau wie beim 68008. Jedoch liegt ab Adresse 0F000H beim CP/M-System die Einsprungtabelle für den FLOMON. Man muß daher darauf achten, daß keine Kollisionen auftreten können. Die hier abgedruckten Unterprogramme berücksichtigen das automatisch.

Wenn man die COL256 mit Hilfe des Z80-Grundprogramms testen will, so kann man das auch tun, man muß dazu nur mal das INIT-Programm und die Initialisierungstabelle aus dem Listing abtippen und auf Adresse 8800H schieben. Dann ruft man INIT auf. Nun erst kann man auf dem Bildschirm Punktmuster erkennen. Das Muster ist rein zufällig, nämlich der Inhalt der Speicherzellen, wie er beim Stromeinschalten auftritt. Nun kann man die COL256 testen. Dazu belegt man mit Hilfe des Grundprogramms den IO-Port 0CEH mit dem Wert 80H. Dadurch wird der Speicherbereich ab Adresse 0C000H bis 0FFFFH eingeblendet. Das Grundprogramm kann dabei sowohl von FLOMON aus, als auch ab Adresse 0

(Standard-System) aufgerufen worden sein. SBC2-Besitzer werden allerdings keinen Erfolg haben, da die Baugruppe die notwendigen Adressen nicht herausgeführt hat.

Achtung, beim Betrieb der Vollausbau-CPU ohne Bank-Boot müssen die Adressbrücken, die den Bank-Bereich bestimmen, auf der COL256 herausgezogen sein, sonst erreicht man die Baugruppe nicht vom Grundprogramm aus. Wenn man dagegen eine Bank-Boot-Baugruppe mit FLOMON und Grundprogramm verwendet, müssen alle vier Brücken (siehe Schaltplan in der Bauanleitung), eingesetzt sein.

Nun zu unserem Programm, das eigentlich für die CP/M-Benutzer gedacht ist. Die Routine INIT programmiert den CRT-Controller auf 256 mal 256 Bildpunkte. Das Unterprogramm CLEAR löscht den Bildschirm dunkel. Mit CLRCOL kann man den Bildschirm auch mit einer beliebigen Farbe vorbelegen. Dazu wird der Farbwert im Akku angegeben. Der Code entspricht der Belegung aus LOOP 6.

Das Unterprogramm SETDOT setzt einen Punkt. Dazu wird die x-Koordinate im Registerpaar HL angegeben und die y-Koordinate im Registerpaar DE. Zur Zeit wird der Bereich nicht geprüft, also der Wert von H und D bleibt unberücksichtigt.

Der Bereich für x und y ist damit 0..255. Im

Register B wird die sogenannte Und-Maske angegeben und im Register C die XOR-Maske. Durch diese zwei Angaben ist es möglich, mit nur einem Unterprogramm verschiedene Funktionen zu erfüllen. So möchte man zum Beispiel einen Bildpunkt mit einer bestimmten Farbe belegen. Dann setzt man die Und-Maske auf 0 und gibt den Farbcode im C-Register an. Durch den Wert 0 bei der Und-Maske wird erreicht, daß der alte Wert auf dem Bildschirm keine Rolle spielt. Es wird dann nämlich eine XOR-Verknüpfung mit dem Ergebnis der Und-Verknüpfung, also mit 0 durchgeführt und das Ergebnis im Bildpunkt gespeichert. Wenn man zum Beispiel den Inhalt des Bildspeichers invertieren möchte, um ein Fadenkreuz darzustellen, so gibt man OFFH als Und-Maske an und OFFH als XOR-Maske. Wenn der Bildpunkt zuvor den Wert 0 hatte, bekommt er nachher den Wert OFFH und umgekehrt. Ähnliches gilt auch für andere Farben.

Interessanter ist noch das Unterprogramm DRAW. Es zeichnet eine Linie. Im Registerpaar HL steht die Koordinate x1 des Startpunktes. Im Registerpaar DE steht die Koordinate x2 des Endpunktes. Im Registerpaar HL' (zweiter Registersatz) steht die Koordinate y1 und in DE' (zweiter Registersatz) der Wert für y2. Im Register B steht dann noch die Und-Maske (siehe SETDOT) und im Register C die

```

;*****
; Unterprogramme fuer den Betrieb
; der CDL256 mit dem 180 unter CP/M
; Rolf-Dietger Klein 830118 V 1.0
;*****

; alle Unterprogramme sind global definiert
; so dass man sie zu eigenen dazulinken kann.
; Achtung, die Bibliothek darf nur im
; Adressraum 0 bis bfff liegen, sonst
; gibt es Kollisionen mit dem RAM-
; Bereich der Farbtaete.
; Routinen sind fuer ein Bildformat
; 256 x 256 Punkten ausgelegt.

0000 cseq

000C crt equ 0cch ; Adressport 6845
000D crt d equ 0cdh ; Dataport 6845
000E crt b equ 0ceh ; Bankport CDL256

0000 col equ 0c000h ; Adresse nach der Einblendung

0000 start: ; Testprogramm fuer alle Funktionen
0001 call init
0002 ld a,0cch
0003 call clrcol ; Gruenes Bild
0004
0005 ld hl,10 ; y=10
0006 ld de,0 ; y=0
0007 ld bc,000c5h ; in Rot
0008 lpt1: ; Senkrechte Linie
0009 call setdot
000A inc de
000B ld a,e
000C cp 0
000D jr nz,lpt1 ; 0..255
000E
000F ld ix,liste
0010 ld b,anz
0011 lpt2: ; Test-Stern zeichnen.
0012 push bc
0013 ld hl,135 ; xi=135 ; asym. Zentrum
0014 ld e,(ix+0)
0015 ld d,(ix+1)
0016 exx
0017 ld hl,100 ; yi=100
0018 ld e,(ix+2)
0019 ld d,(ix+3)
0020 exx
0021 ld bc,00ffh ; WEISS
0022 call draw
0023 inc ix ; naechste Adresse

```

```

002B dd 23
002F dd 23
0041 dd 23
0043 c1
0044 10 DA
0046 c9

0047
0047 0000 0000
0048 0000 0080
004F 0000 00FF
0053 0080 00FF
0057 00FF 00FF
005B 00FF 0080
005F 00FF 0000
0063 0080 0000
0067 0000 0000
0069

anz equ (4-liste)/4
;
006B init:
006C ld hl,tab ; Init-Tabelle
006D ld b,laenge ; Anzahl Befehle
006E ld c,0 ; Index Adresse
006F ilp:
0070 ld a,c
0071 out (crt),a ; Adresse einstellen
0072 ld a,(hl) ; dann
0073 out (crt d),a ; Datenwert einstellen
0074 inc c
0075 inc hl
0076 djnz ilp
0077 ret

; Bildformat 256 x 256

007D tab:
007E defb 111,64,80,7,77,0,64,70,0,3,0,0,0,0,0
0081
0085
0089
008F

laenge equ 4-tab

; Bildschirm loeschen
008C clear: ; auf schwarz
008D ld a,0
008E clrcol: ; oder farbig (A=Farbcode)
008F ld b,a ; Parameter merken
0090 ld a,80h ; dann alle vier Seiten
0091 call clr ; loeschen.
0092 ld a,81h
0093 call clr
0094 ld a,82h
0095 call clr
0096
0098

```

```

009E 3E 83
00A0 CD 00A4
00A3 C9

00A4 clrp:
00A5 push bc
00A6 out (crtb),a
00A7 ld hl,col ; vorbereiten fuer LDIR
00A8 ld de,col+1
00A9 ld (hl),b ; Loeschfarbe ablegen
00AA ld bc,4000h-1 ;
00AB ldir ; und Loeschen.
00AC xor a ; Bank ausblenden
00AD out (crtb),a ; bevor Stack verwendet wird.
00AE pop bc ; damit Stackpointer beliebig sein kann.
00AF ret

```

```

00BB ; Punkt bearbeiten.
00BC setdot: ; hl=x, de=y, b=andmask c=xorasm
00BD ld a,b
00BE ld (anda),a
00BF ld a,c
00C0 ld (xora),a
; weiter bei setdot

00C0 ; Punkt setzen bei y=hl (0..319), y=de (0..199)
00C1 ; AndMask, forMask.
00C2 ; Adresse = y * 256 + x
00C3 ; Ohne Bereichspruefung.
00C4 push bc
00C5 push de
00C6 push hl
00C7 ld a,e
00C8 cpl ; 0..255 -> 255..0 umrechnen
00C9 ld a,a
00CA or 0C0h ; Adresse phys. c000..ffff
00CB ld h,a ; Damit wrap-around.
00CC ld a,e ; 0..3fff erlaubt
00CD rca ; Msb bestimmt Bank
00CE or 80h
00CF out (crtb),a ; Bank einstellen

00D2 ld a,(xormask) ; Durch Verknuepfung
00D3 ld b,a ; mit AND und IDR
00D4 ld a,(anda) ; lassen sich alle
00D5 and (hl) ; Verknuepfungsarten
00D6 xor b ; ausfuehren.
00D7 ld (hl),a ; Beispiel:
00D8 xor a ; Setzen - AND=0, IDR=Farbe
00D9 out (crtb),a ; Komplementieren - AND = 0ifh, IDR = Wert
00DA pop hl ; Odern - AND = NOT(Wert), IDR = Wert
00DB pop de ; Und - AND = Wert, IDR = 0
00DC pop bc

```

```

00E2 C9 ret

00E3 draw: ; hl=x1, de=y2, hl'=y1, de'=y2
00E4 ; b = andmask, c = xormask
00E5 ld a,b
00E6 ld (anda),a
00E7 ld a,c
00E8 ld (xora),a ; alle Register serken
00E9 ld (x1),hl
00EA ld (x2),de
00EB exx
00EC ld (y1),hl
00ED ld (y2),de
00EE exx
00EF
00F0 ld hl,1
00F1 ld (dx),hl
00F2 ld hl,(x2)
00F3 ld (dx),hl
00F4 ld hl,(x2)
00F5 ld (dx),hl
00F6 ld hl,(x2)
00F7 ld (dx),hl
00F8 ld hl,(x2)
00F9 ld (dx),hl
00FA ld hl,(x2)
00FB ld (dx),hl
00FC ld hl,(x2)
00FD ld (dx),hl
00FE ld hl,(x2)
00FF ld (dx),hl
0100 ld hl,(x2)
0101 ld (dx),hl
0102 ld hl,(x2)
0103 ld (dx),hl
0104 ld hl,(x2)
0105 ld (dx),hl
0106 ld hl,(x2)
0107 ld (dx),hl
0108 ld hl,(x2)
0109 ld (dx),hl
010A ld hl,(x2)
010B ld (dx),hl
010C ld hl,(x2)
010D ld (dx),hl
010E ld hl,(x2)
010F ld (dx),hl
0110 ld hl,(x2)
0111 ld (dx),hl
0112 ld hl,(x2)
0113 ld (dx),hl
0114 ld hl,(x2)
0115 ld (dx),hl
0116 ld hl,(x2)
0117 ld (dx),hl
0118 ld hl,(x2)
0119 ld (dx),hl
011A ld hl,(x2)
011B ld (dx),hl
011C ld hl,(x2)
011D ld (dx),hl
011E ld hl,(x2)
011F ld (dx),hl
0120 ld hl,(x2)
0121 ld (dx),hl
0122 ld hl,(x2)
0123 ld (dx),hl
0124 ld hl,(x2)
0125 ld (dx),hl
0126 ld hl,(x2)
0127 ld (dx),hl
0128 ld hl,(x2)
0129 ld (dx),hl
012A ld hl,(x2)
012B ld (dx),hl
012C ld hl,(x2)
012D ld (dx),hl
012E ld hl,(x2)
012F ld (dx),hl
0130 ld hl,(x2)
0131 ld (dx),hl
0132 ld hl,(x2)
0133 ld (dx),hl
0134 ld hl,(x2)
0135 ld (dx),hl
0136 ld hl,(x2)
0137 ld (dx),hl
0138 ld hl,(x2)
0139 ld (dx),hl
013A ld hl,(x2)
013B ld (dx),hl
013C ld hl,(x2)
013D ld (dx),hl
013E ld hl,(x2)
013F ld (dx),hl
0140 ld hl,(x2)
0141 ld (dx),hl
0142 ld hl,(x2)
0143 ld (dx),hl
0144 ld hl,(x2)
0145 ld (dx),hl
0146 ld hl,(x2)
0147 ld (dx),hl
0148 ld hl,(x2)
0149 ld (dx),hl
014A ld hl,(x2)
014B ld (dx),hl
014C ld hl,(x2)
014D ld (dx),hl
014E ld hl,(x2)
014F ld (dx),hl
0150 ld hl,(x2)
0151 ld (dx),hl
0152 ld hl,(x2)
0153 ld (dx),hl
0154 ld hl,(x2)
0155 ld (dx),hl
0156 ld hl,(x2)
0157 ld (dx),hl
0158 ld hl,(x2)
0159 ld (dx),hl
015A ld hl,(x2)
015B ld (dx),hl
015C ld hl,(x2)
015D ld (dx),hl
015E ld hl,(x2)
015F ld (dx),hl
0160 ld hl,(x2)
0161 ld (dx),hl
0162 ld hl,(x2)
0163 ld (dx),hl
0164 ld hl,(x2)
0165 ld (dx),hl
0166 ld hl,(x2)
0167 ld (dx),hl
0168 ld hl,(x2)
0169 ld (dx),hl
016A ld hl,(x2)
016B ld (dx),hl
016C ld hl,(x2)
016D ld (dx),hl
016E ld hl,(x2)
016F ld (dx),hl
0170 ld hl,(x2)
0171 ld (dx),hl
0172 ld hl,(x2)
0173 ld (dx),hl
0174 ld hl,(x2)
0175 ld (dx),hl
0176 ld hl,(x2)
0177 ld (dx),hl
0178 ld hl,(x2)
0179 ld (dx),hl
017A ld hl,(x2)
017B ld (dx),hl
017C ld hl,(x2)
017D ld (dx),hl
017E ld hl,(x2)
017F ld (dx),hl
0180 ld hl,(x2)
0181 ld (dx),hl
0182 ld hl,(x2)
0183 ld (dx),hl
0184 ld hl,(x2)
0185 ld (dx),hl
0186 ld hl,(x2)
0187 ld (dx),hl
0188 ld hl,(x2)
0189 ld (dx),hl
018A ld hl,(x2)
018B ld (dx),hl
018C ld hl,(x2)
018D ld (dx),hl
018E ld hl,(x2)
018F ld (dx),hl

```

```

014B ld de,(dy)
014C xor a ; sue wird durch max
014D sbc hl,de ; bestimmt.
014E bit 7,h
014F jr z,skd1
0150 ex de,hl
0151 xor a
0152 ld hl,0
0153 sbc hl,de
0154 push hl
0155 ld hl,-1
0156 ld (dx),hl
0157 pop hl
0158 skd1:
0159 ld (dy),hl
015A ;
015B ld hl,1
015C ld (dy),hl
015D ld hl,(y2) ; dy bestimmen
015E ld de,(y1) ; und dy
015F xor a
0160 sbc hl,de
0161 bit 7,h
0162 jr z,skd2
0163 ex de,hl
0164 ld hl,0
0165 sbc hl,de
0166 push hl
0167 ld hl,-1
0168 ld (dy),hl
0169 pop hl
016A skd2:
016B ld (dy),hl
016C ;
016D ld hl,(dx)
016E ld (su),hl
016F ;
0170 ;
0171 s4: ; hl=x1,de=y2,hl'=y1,de'=y2
0172 push de ; unsortieren.
0173 pop de ; fuer setdot.
0174 exa
0175 pop de
0176 call setdot ; hl=x, de=y
0177 pop de
0178 ld a,h ; x1
0179 cp d ; x2
017A jr nz,s5 ; wenn gleich, dann
017B ld a,l ; untersuchen ob Ende
017C cp e ; erreicht.
017D jr z,s7
017E s5:
017F ld bc,(su) ; Fall 1
0180 bit 7,h ; su > 0
0181 jr nz,s6
0182 ld bc,(dx)
0183 add hl,bc

```



```

01B7' E5          push hl
01B8' 2A 01E8'    ld hl,(sum)
01B9' ED 4B 01E6' ld bc,(dy)
01BA' AF          xor a
01BB' ED 42       sbc hl,bc
01BC' AF          xor a
01BD' ED 42       sbc hl,bc
01BE' 22 01EB'    ld (sum),hl
01BF' E1          pop hl
01C0' 18 CE       jr s4
01C1' s4         : Fall 2
01C2' s4         : sum < 0
01C3' D9         exx
01C4' ED 4B 01E2' ld bc,(dy)
01C5' 09         add hl,bc
01C6' E5         push hl
01C7' 2A 01E8'    ld hl,(sum)
01C8' ED 4B 01E4' ld bc,(dx)
01C9' 09         add hl,bc

MACRO-B0 3.43 27-Jul-81 PAGE 1-5

01B2' 09          add hl,bc
01B3' 22 01EB'    ld (sum),hl
01B4' E1          pop hl
01B5' D9          exx
01B6' 18 B7       jr s4
01B7' s4         : Endepuefung
01B8' D9          exx
01B9' E5         push hl
01BA' AF          xor a
01BB' ED 52       sbc hl,de

```

```

01BF' 7C          ld a,h
01C0' 85          or l
01C1' E1          pop hl
01C2' D9         exx
01C3' 20 B0       jr nz,s5 ; nein, dann weiter
01C4' 2A 01EE'    ld hl,(x2) ; Endpunkt laden,
01C5' ED 5B 01EE' ld de,(x2) ; damit Register wieder
01C6' D9         exx ; gueltig sind.
01C7' 2A 01F0'    ld hl,(y2) ; beide Register hl und de
01C8' ED 5B 01F0' ld de,(y2) ; erhalten je den Endpunkt,
01C9' D9         exx ; damit kann man leicht
01CA' 3A 01DE'    ld a,(andmask) ; Polygone zeichnen.
01CB' 47         ld b,a
01CC' 3A 01DF'    ld a,(xorasmk)
01CD' 4F         ld c,a
01CE' C9         ret
01CF'             ;
01D0'             ; lokale Variable und Merker.
01D1' 00         andmask: defb 0
01D2' 00         dxrasmk: defb 0
01D3' 0000       dx: defw 0 ; +/-1
01D4' 0000       dy: defw 0 ; +/-1
01D5' 0000       dx: defw 0
01D6' 0000       dy: defw 0
01D7' 0000       sum: defw 0 ; fuer draw-algorithmus
01D8' 0000       x1: defw 0 ; X,Y-Position
01D9' 0000       y1: defw 0
01DA' 0000       x2: defw 0
01DB' 0000       y2: defw 0
01DC' 0000
01DD' 0000

end start

```

XOR-Maske. Die Unterprogramme INIT, CLEAR, CLRCOL, SETDOT und DRAW besitzen zwei Doppelpunkte hinter der Maske, damit werden sie bei Verwendung des M80-Assemblers zu globalen Marken, die man mit dem Linker (L80) zusammen mit anderen, eigenen Programmen, zusammenbinden kann.

Am Anfang der Unterprogrammssammlung steht übrigens noch ein kleines Testprogramm, das bei der Marke START beginnt. Damit kann man die Unterprogramme einfach testen. Das Unterprogramm löscht den Bildschirm auf Grün. Dann wird eine senkrechte Linie in Rot gezeichnet. Die Linie befindet sich auf der

x-Koordinate mit dem Wert 10. Zum Schluß wird noch das Unterprogramm DRAW getestet. Es zeichnet 8 Linien, die von der Koordinate $x = 135, y = 100$ ausgehen und in den 8 Ecken des Bildschirms enden.

(Fortsetzung folgt)

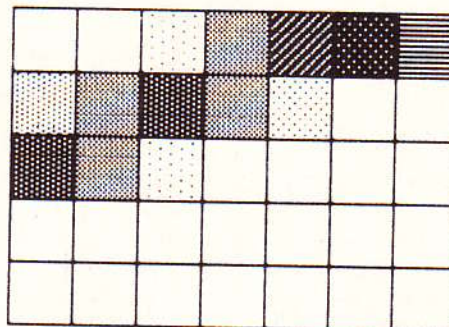
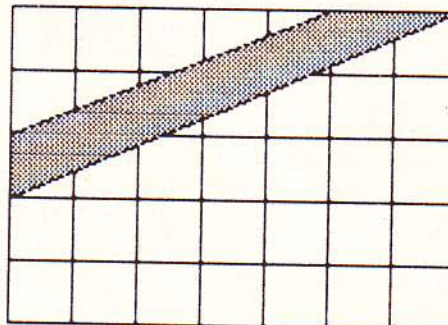
TIPS + TRICKS — TIPS + TRICKS

Antialiasing – oder – wie man das Raster austrickt

Rolf-Dieter Klein

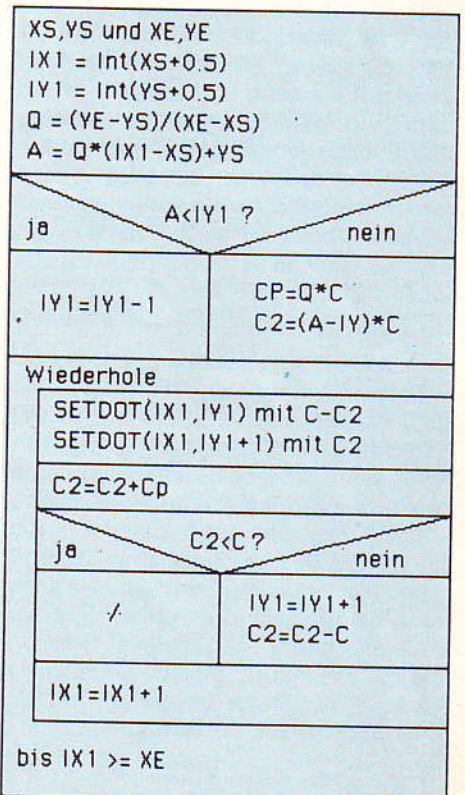
Wer kennt sie nicht, die kantigen und treppenförmigen Linien, die sich über den Bildschirm erstrecken? Bedingt durch das Punktraster lassen sich auf einem Bildschirm keine exakten Linien oder Kreise darstellen. Das soll nun anders werden. Wer eine COL256 besitzt kann das ändern, denn man benötigt Graustufen, um das Übel abzustellen. Die Fachleute nennen das Antialiasing. Dieser Begriff kommt vom umgekehrten Vorgang, dem Aliasing, das immer dann entsteht, wenn man Signale darstellen will und diese mit einem diskreten Zeittakt abtastet. Man erhält dabei hohe Frequenzanteile, die als sichtbares Moiré störend auffallen. Im Studium wird man bei der Fourier-Zerlegung damit konfrontiert. Doch damit genug der Theorie.

Bild 1 zeigt eine Lösungsmöglichkeit. Wenn man eine Linie auf einem Raster darstellen will, so muß man die Linie in Rastereinheiten zerlegen. Man wählt normalerweise die Rasterpunkte aus, die der Linie am dichtesten kommen; der Bresenham-Algorithmus zum Beispiel tut das. Dieser Algorithmus ist auch in unserem Graphik-Prozessor EF9366 eingebaut. In der LOOP6 und in dieser LOOP ist er für die COL256 auch abgedruckt.



Es gibt auch eine andere Möglichkeit, wenn man Graustufen besitzt: man kann die benachbarten Rasterpunkte mit einer überdeckten Fläche entsprechender Helligkeitswerte besetzen. Dadurch entsteht im Auge der Eindruck, daß der Schwerpunkt der Linie genau auf der ursprünglichen Linie liegt. Der Treppeneffekt verschwindet. Ein kleiner Nachteil ist, daß die Linien geringfügig unschärfer wirken.

Der Algorithmus wurde von Akura Fujimoto und Kansei Iwata so aufbereitet, daß er ähnlich schnell wie der Bresenham-Algorithmus ist. Bild 2 zeigt das Strukturprogramm dazu. Hier wurde der Original-Algorithmus abgebildet, der mit REAL-Werten für die Startkoordinaten XS,YS und die Endkoordinaten XE,YE arbeitet.



Algorithmus nach Akura Fujimoto und Kansei Iwata

TIPS + TRICKS — TIPS + TRICKS

Die Differenz 1 entspricht einem Rasterpunkt. Man kann damit Linien sogar um einen Anteil kleiner als einen Rasterpunkt verschieben und die Linie wird auf dem Bildschirm tatsächlich sichtbar, auch um weniger als einen Rasterpunkt verschoben.

ben. Damit ist es auch möglich, Kreise wirklich rund aussehen zu lassen. Der Algorithmus arbeitet in der abgebildeten Form allerdings nur im ersten Oktanten einwandfrei. Man muß ihn abwandeln, will man ihn universell verwenden, indem man z.B. vorher Abfragen und Vertauschungen einbaut.

Bild 3 zeigt das praktisch aufgebaute Programm für die COL256. Es arbeitet in Integer-Arithmetik, da das zunächst einfacher ist.

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

```

0E9C00          *****
0E9C00          * UNTERPROGRAMME ZUM BETRIEB DER FARBIARTE *
0E9C00          * TEIL 2 *
0E9C00          * COL 256 *
0E9C00          * ROLF-DIETER KLEIN 680105 REV 1.0 *
0E9C00          * ANTIALIASING VECTORS *
0E9C00          *****
= FFFFFFFC      CRT EQU $FFFFFFC
= FFFFFFFC      CRTD EQU $FFFFFFC
= FFFFFFFC      CRTB EQU $FFFFFFC
0E9C00
0E9C00          START:          * KLEINER TEST
0E9C00          BSR INIT1        * 6845 INITIALISIEREN
0E9C04          MOVE.B #*00,D0   * HINTERGRUND SCHWARZ
0E9C08          BSR CLRCOL1
0E9C0C
0E9C0C          CLR X1          * VERSCHIEDENE VEKTOREN AUSGEBEN
0E9C10          CLR Y1
0E9C14          MOVE #255,X2
0E9C18          CLR X2
0E9C1C          CLR Y2
0E9C20          SCHLEIFE:      * TESTPROGRAMM
0E9C24          BSR ANTIAL    * NUR 1. OKTAND MOEGLICH.
0E9C28          ADD #5,Y2
0E9C2C          MOVE Y2,D0
0E9C30          CMP #255,D0
0E9C34          BLE SCHLEIFE   *
0E9C38          RTS
0E9C40
0E9C40          INIT1:
0E9C44          BSR INIT2        * REV, KEIN LOESCHEN MEHR DANACH
0E9C48          RTS
0E9C4C
0E9C4C          *****
0E9C4C          * SYNC PARAMETER ETC INITIALISIEREN, 6845
0E9C4C          * HIER WIRD DAS FORMAT 256*256
0E9C4C          * EINGESTELLT
0E9C4C
0E9C4C          INIT2:
0E9C4C          LEA TAB(PC),A0   * TABELLE DER CODES
0E9C4C          CLR D1
0E9C4C          MOVE #LENGTH-1,D3
0E9C52          LPP:
0E9C52          MOVE.B D1,CRT
0E9C58          MOVE.B (A0)+,CRTD
0E9C5C          ADDO #1,D1
0E9C60          DBRA D3,LPP
0E9C64          RTS
0E9C68
0E9C68          TAB:
0E9C68          DC.B 111,64,80,7,77,0,64,70,0,3,0,0,0,0,0
0E9C70          00 00 00 00 00
0E9C74          LENGTH EQU *-TAB
0E9C78          DS 0
0E9C7C
0E9C7C          *****
0E9C7C          * LOESCHEN DES BILDSCHIRMS
0E9C7C

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 2

```

0E9C76          CLEAR1:      * LOESCHEN AUF SCHWARZ
0E9C76          CLR.L D0
0E9C78          CLRCOL1:     * LOESCHEN MIT FARBCODE 0..255
0E9C78          MOVE.B #*80,CRTB * CODE IN D0.B
0E9C80
0E9C80          BSR CLRP
0E9C84          MOVE.B #*81,CRTB
0E9C88          BSR CLRP
0E9C8C          MOVE.B #*82,CRTB
0E9C90          BSR CLRP
0E9C94          MOVE.B #*83,CRTB
0E9C98          BSR CLRP
0E9CA4          CLR.B CRTB
0E9CA8          RTS
0E9CB0
0E9CB0          CLRP:          * UPRG. 16K BYTE SCHNELL LOESCHEN.
0E9CB0          LEA $C000,A0
0E9CB4          MOVE.B D0,D3
0E9CB8          ROL #8,D0
0E9CB8          MOVE.B D3,D0
0E9CB8          MOVE D0,D3
0E9CB8          SWAP D0
0E9CB8          MOVE D3,D0
0E9CC0          MOVE #*1000-1,D3
0E9CC4          LOOP:
0E9CC4          MOVE.L D0,(A0)+
0E9CC8          DBRA D3,LOOP
0E9CCC          RTS
0E9CCE
0E9CCE          *****
0E9CCE          * D0.B = FARBCODE, D1=X ,D2=Y
0E9CCE

```

```

0E9CCE          SETDOT1:      * PUNKT SETZEN, TRANSPARENT.
0E9CCE          MOVE D2,-(A7)
0E9CD0          NOT.B D2      * ACHTUNG NICHT NEG VERWENDEN
0E9CD4          ROR.W #4,D2   * AAAAAAXXXXXX
0E9CD8          OR.B #*80,D2  * AAAAAAXXXXXX
0E9CD8          MOVE.B D2,CRTB * SELEKT DURCHFUEHREN
0E9CE0          LSR.W #2,D2   * 00AAAAA-----
0E9CE4          MOVE.B D1,D2  * 00AAAAAABBBBBBB
0E9CE8          LEA $C000,A0  * BASIS-ADRESSE COL256
0E9CEC          ADDA.W D2,A0   * A0=ADRESSE PIXEL.
0E9D00          MOVE (A7)+,D2
0E9D04          DR.B D0,(A0)
0E9D08          CLR.B CRTB
0E9D0C          RTS
0E9D10
0E9D10          *****
0E9D10          * ANTIALIAS-ALGORITHMUS NACH AKIRA FUJIMOTO,
0E9D10          * KANSEI INATA
0E9D10          * "JAG FREE IMAGES ON A RASTER CRT"
0E9D10          * IM BUCH "COMPUTER GRAPHICS ", HERAUSGEGEBEN
0E9D10          * VON TOSIYASU L. KUNITI, SPRINGER-VERLAG.
0E9D10          * MODIFIZIERT VON ROLF-DIETER KLEIN
0E9D10          * FUER INTEGER WERTE UND FORMELVEREINFACHUNG.
0E9D10          * ACHTUNG, NUR DER 1. OKTAND WIRD
0E9D10          * IN DIESEM BEISPIEL RICHTIG BEARBEITET
0E9D10          * DAS PROGRAMM MUSS ERWEITERT WERDEN,
0E9D10          * WENN MAN ES UNIVERSSELL EINSETZEN WILL.
0E9D10          * ACHTUNG, DIE MAXIMALE INTENSITAET

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 3

```

0E9D14          * BESTIMMT AUCH DEN SCHRITTABSTAND,
0E9D14          * DA HIER NICHT WIE BEI BRESHNAM MIT
0E9D14          * DEN DIFFERENZEN BEARBEITET WIRD.
0E9D14          * ES ERFOLGT DAHER IMMER EINE UMRECHNUNG AUF
0E9D14          * DIE REALE INTENSITAET.
0E9D14          ANTIAL:
0E9D14          MOVE X1,XE      * HIER OKTANDEN UNTERSUCHEN,
0E9D14          MOVE X2,XE      * BEI ERWEITERUNG DES ALGORITHMUS.
0E9D14          MOVE Y1,YE
0E9D14          MOVE Y2,YE
0E9D14          * AB HIER MUSS SELTEN:
0E9D14          * Y<YS UND X<XS
0E9D14          * SOWIE STEIGUNG < 1
0E9D14          MOVE XE,D0
0E9D14          SUB YS,D0
0E9D14          MOVE D0,DX
0E9D14          MOVE YE,D0
0E9D14          SUB YS,D0
0E9D14          MOVE D0,DY
0E9D14          * DERZEIT NUR SCHARZ-WEISS.
0E9D14          LEA BRAUTAB(PC),A1 * FUER INTENS.
0E9D14          LEA GAMMATAB(PC),A2 * FUER KORREKTOR
0E9D14          MOVE XE,IX1      * STARTPUNKT
0E9D14          MOVE YS,IV1
0E9D14          *
0E9D14          MOVE #*80,D0   * INTENS MAX. (32*15)
0E9D14          MULL D0,D0      * VERWENDEN.
0E9D14          DIVS D0,D0      * DANN UP-STARTWERT
0E9D14          MOVE D0,CP      * FARBINCREMENT.
0E9D14          *
0E9D14          MOVE #*80,C2   * STARTINTENS HELL.
0E9D14          *
0E9D14          LFA:
0E9D14          MOVE IX1,D1      * X,Y AUSGEBEN
0E9D14          MOVE IV1,D2      * MIT INTENS MAX-C2
0E9D14          MOVE #*80,D3
0E9D14          SUB C2,D3
0E9D14          ASR #5,D3      * DIV 32, UMRECHNUNG INTENS
0E9D14          MOVE.B 0(A2,D3.W),D3 * GAMMAKORREKTUR
0E9D14          MOVE.B 0(A1,D3.W),D0 * INTENS
0E9D14          BSR SETDOT1
0E9D14          ADD #1,D2      * X,Y+1 MIT INTENS C2
0E9D14          MOVE C2,D3
0E9D14          ASR #5,D3      * DIV 32, UMRECHNUNG INTENS
0E9D14          MOVE.B 0(A2,D3.W),D3 * GAMMAKORREKTUR
0E9D14          MOVE.B 0(A1,D3.W),D0 * INTENS
0E9D14          BSR SETDOT1
0E9D14          *
0E9D14          MOVE C2,D0
0E9D14          ADD CP,D0      * NEUE INTENS
0E9D14          MOVE D0,C2
0E9D14          CMP #*80,D0
0E9D14          BLT SK1      * WENN GROESSER, DANN
0E9D14          ADD #1,IV1    * KORREKTUR, INTENS UND Y-WERT
0E9D14          SUB #*80,C2 *
0E9D14
0E9D14          *****
0E9D14          * DO.B = FARBCODE, D1=X ,D2=Y
0E9D14

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 4

```

0E9D14          SK1:
0E9D14          ADD #1,IX1
0E9D14          MOVE IX1,D0
0E9D14          CMP XE,D0      * BIS ENDPUNKT ERREICHT.

```

TIPS + TRICKS — TIPS + TRICKS

```

0E9DF6 6D00 FF84      BLT LPA * DIESEN ABER NICHT MEHR DARSTELLEN.
0E9DFA 4E75          RTS
0E9DFC
0E9DFC
0E9DFC
0E9DFC
0E9DFC
0E9DFC 00 03 05 06 07
0E9E01 08 09 0A 0B
0E9E06 0C 0D 0E 0F
0E9E08 0F
0E9E0C
0E9E0C
0E9E0C
0E9E0C 00
0E9E0D 40
0E9E0E 80
0E9E0F C0
0E9E10 15
0E9E11 55
0E9E12 95
0E9E13 D5
0E9E14 2A
0E9E15 6A
0E9E16 AA
0E9E17 EA
0E9E18 3F

* GAMMA-KORREKTUR-TABELLE
* HIER GAMMA=1.7
* VI = ((INTENS / INTENSMAX) ^ (1 / GAMMA)) * INTENSMAX
GAMMATAB:
DC.B 0,3,5,6,7,8,9,10,10,11,12,12,13,14,14,15

* GRAUSTUFEN-CODIERUNG
GRAUTAB:
DC.B %00000000
DC.B %01000000
DC.B %10000000
DC.B %11000000
DC.B %00010101
DC.B %01010101
DC.B %10010101
DC.B %11010101
DC.B %00101010
DC.B %01101010
DC.B %10101010
DC.B %11101010
DC.B %00111111
    
```

```

0E9E19 7F
0E9E1A BF
0E9E1B FF
0E9E1C
0E9E1C
0E9E1C
0E9E1C
0E9E1C
0E9E1E
0E9E20
0E9E22
0E9E24
0E9E26
0E9E28
0E9E2A
0E9E2C
0E9E2E
0E9E30
0E9E32
0E9E34
0E9E36
0E9E38
0E9E3B
0E9E38

DC.B %01111111
DC.B %10111111
DC.B %11111111

* SPEICHERZELLEN
X1: DS.W 1
X2: DS.W 1
Y1: DS.W 1
Y2: DS.W 1
XS: DS.W 1
YS: DS.W 1
XE: DS.W 1
YE: DS.W 1
IX1: DS.W 1
IY1: DS.W 1
CP: DS.W 1
C2: DS.W 1
DY: DS.W 1
DX: DS.W 1

END

0E8E1A Ende-Symboltabelle
00A8BC Ende-Debug-Tabelle
    
```

Bild 4 zeigt die Ausgabe auf dem Bildschirm, ohne den Antialiasing-Algorithmus. In Bild 5 ist er wirksam. Man erkennt immer noch ein störendes Moiré. Dieses

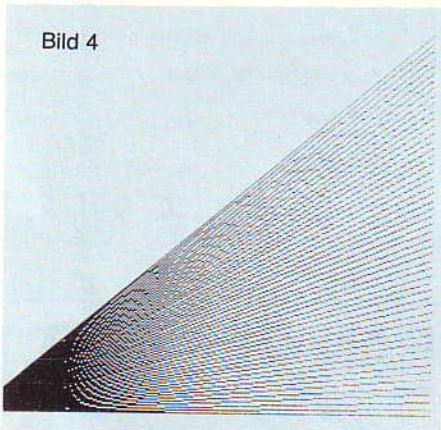


Bild 4

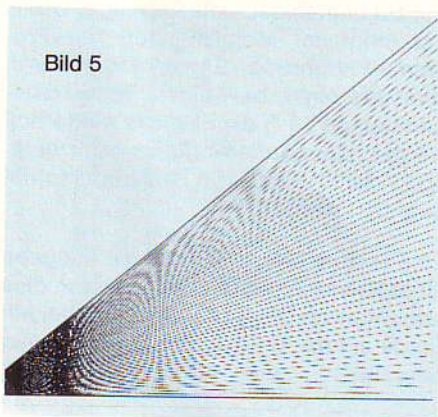


Bild 5

würde man auch auf dem Bildschirm sehen, wenn man nicht noch eine Gammakorrektur einbaut (wie im Listing Bild 3 bereits geschehen). Die Gammakorrektur ergibt Bild 6, die erst auf dem Bildschirm voll wirkt. Was ist die Gammakorrektur? Wenn man auf einem Bildschirm die Helligkeit durch lineare Spannungsänderung vergrößert, entsteht kein linearer Helligkeitsverlauf, da die Elektronenstrahl-Eigenschaften keine linearen Helligkeitsänderungen bewirken. Für jeden Bildschirm gibt es eine spezifische Gam-

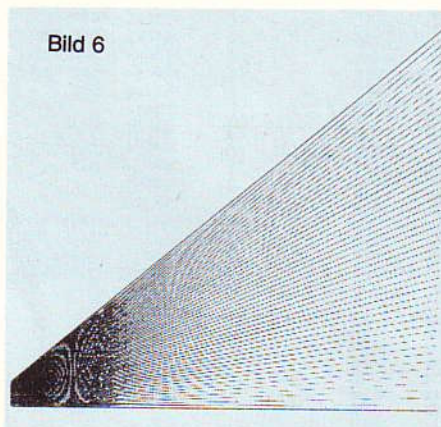


Bild 6

makorrektur. Diese Gammakorrektur ist nichts anderes als eine Tabelle, im Programm die Tabelle GAMMATAB, die die korrigierten Werte enthält. Hier wurde der Wert 1.7 in die im Listing angegebene Formel eingesetzt, der beim verwendeten TOEI-Monitor die besten Ergebnisse liefert und interessanterweise auch auf dem Drucker das beste Bild.

Literatur:

Edited by Tosiyasu L. Kunii,
„Computer Graphics, Theory und Applications“, Springer-Verlag, Tokyo, Berlin, Heidelberg, New York.

GOMOKU

An dieser Stelle sollte eigentlich der Artikel GOMOKU Teil 2 erscheinen. Da aber dazu das Listing nötig wäre, konnten wir hier diesen Artikel nicht veröffentlichen, da dieser den Rahmen der LOOP sprengen würde. Aus diesem Grund haben wir uns entschlossen, das Listing extra (mit Kommentaren) zu veröffentlichen. Dieses Listing können Sie ca. ab Mitte Februar von uns beziehen.

Wir planen auch, dieses Programm in unserer Datenbank zur Verfügung zu stellen. Über einen Zugriff auf die Datenbank (z.B. mit ZEAT) kann dieses Programm in Ihren Rechner geladen werden. In einer der nächsten LOOP's mehr darüber.

Floppy Testprogramm 68K

Rolf-Dieter Klein

Nachdem man eine Floppy mit Hilfe von UFORM formatiert hat, wird sie automatisch Prüfgelesen. Nun ist es aber manchmal ganz praktisch, eine Floppy auch später einmal auf Fehler zu prüfen. Ein kleines Testprogramm hilft da weiter. Das Programm liest alle Sektoren eines 80-Spur Laufwerkes. Dabei wird das Laufwerk B verwendet. Es wird zunächst die Vorderseite und dann die Rückseite gelesen. Im Fehlerfall wird die Spur-, Sektor- und Seitennummer auf dem Bildschirm ausgegeben. Wenn kein Fehler vorliegt, meldet sich das Programm wieder im Menue. Wenn man von Laufwerk A lesen will, so muß man anstelle \$22 den Wert \$21 in das Register D4 laden (Adresse \$E9C36).

Das Programm läßt sich natürlich auch für andere Floppy-Formate umstellen. Die Anzahl der Sektoren wird hier auf Adresse \$E9C60 mit einem CMP-Befehl abgefragt. Hier sind es 5 Sektoren, daher erfolgt ein Vergleich mit 5. Die Nummer der höchsten Spur wird auf Adresse \$E9C76 eingestellt, beim 80-Spurformat zählt man die Spurnummern von 0 bis 79, daher ist der Wert 79 als höchste Spurnummer eingetragen. Wenn man nur eine Laufwerkseite prüfen will, so entfällt die Abfrage auf Adresse \$E9C84 und man setzt dahinter einfach einen RTS-Befehl.

Das Programm ist natürlich noch nicht sehr komfortabel, dafür aber kurz genug, um es schnell mal einzutippen.

TIPS + TRICKS — TIPS + TRICKS

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

```

0E9C00
0E9C00      * FLOPPY TESTPROGRAMM
0E9C00      * ES WERDEN ALLE SPUREN EINGELESEN
0E9C00      * ROLF-DIETER KLEIN 860205 1.0
0E9C00      * PRUEFUNG ERFOLGT AUF LAUFWERK B
0E9C00
0E9C00      START:
0E9C00      MOVE #0,D1 * STEPRATE UND SSD SETZEN
0E9C04      MOVE #80,D3
0E9C08      JSR @FLOPPY
0E9C0E
0E9C0E      *
0E9C0E      MOVE #0,SEITE * START MIT SEITE 0
0E9C12      000E9D06
0E9C16
0E9C16      *
0E9C16      LESALL:
0E9C1A      MOVE #0,SPUR
0E9C1E
0E9C1E      LESFLD:
0E9C22      MOVE #1,SEKTOR
0E9C26
0E9C26      LESSPUR:
0E9C26      MOVE #1,D1 * LESEN
0E9C2A      MOVE SEKTOR,D2
0E9C30      MOVE SPUR,D3
0E9C36      MOVE #22,D4 * DOPP. DICHT, MINI. SEITE B
0E9C3A      MOVE SEITE,D0
0E9C40      DR D0,D4 * UND SEITE EINSTELLEN
0E9C42      LEA TEMP,A0 * DORTHIN LADEN
0E9C48      JSR @FLOPPY
0E9C4E
0E9C4E      *
0E9C4E      BCS FEHLER * MELDUNG AUSGEBEN SONST.
0E9C52      ADD #1,SEKTOR
0E9C56      000E9D04
0E9C5A      MOVE SEKTOR,D0
0E9C60      CMP #5,D0
0E9C64      BLE LESSPUR
0E9C68      ADD #1,SPUR
0E9C6C      000E9D02
0E9C70      MOVE SPUR,D0
0E9C76      CMP #79,D0
0E9C7A      BLE LESFLD
0E9C7E      MOVE SEITE,D0
0E9C84      CMP #0,D0
0E9C88      BNE ENDE
    
```

```

0E9C8C      33FC 00B0      MOVE #80,SEITE * DANN RUECKSEITE
0E9C90      000E9D06
0E9C94      6000 FFB0      BRA LESALL
0E9C98
0E9C98      4E75      ENDE:
0E9C9A      RTS * ENDE , TEST OK
0E9C9A
0E9C9A      FEHLER:
0E9C9A      LEA TXT1,A0
0E9CA0      MOVE #22,D0
0E9CA4      MOVE #10,D1
0E9CAB      MOVE #120,D2
0E9CAC      JSR @WRITE
0E9CB2      LEA BUFFER,A0
0E9CB8      MOVE SPUR,D0
0E9CBE      JSR @PRINT4D
0E9CC4      MOVE.B # ' ',(A0)+
0E9CC8      MOVE SEKTOR,D0
0E9CCE      JSR @PRINT4D
0E9CD4      MOVE.B # ' ',(A0)+
    
```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 2

```

0E9CDB      3039 000E9D06      MOVE SEITE,D0
0E9CDE      4EB9 000E15FC      JSR @PRINT4D * 0. ODER 128
0E9CE4      10FC 0000      MOVE.B #0,(A0)+
0E9CEB      41F9 000E127      LEA BUFFER,A0
0E9CEE      303C 0022      MOVE #22,D0
0E9CF2      323C 000A      MOVE #10,D1
0E9CF6      343C 0032      MOVE #50,D2
0E9CFA      4EB9 000E13B6      JSR @WRITE
0E9D00      4E75      RTS
0E9D02
0E9D02      0000      SFUR: DC.W 0
0E9D04      0000      SEKTOR: DC.W 0
0E9D06      0000      SEITE: DC.W 0
0E9D08
0E9D08      TEMP: DS.B 1024 * SEKTORBUFFER
0EA108
0EA108      4665686C657220      TXT1: DC.B 'Fehler (Spur) (Sektor) (Seite)',0
0EA10F      28537075722920
0EA116      2853656B746F72
0EA11D      29202853656974
0EA124      652900
0EA127
0EA127      BUFFER: DS.B 80
0EA177
0EA177
0EBD9C      Ende-Symboltabelle
    
```

GRAMME · PROGRAMME · PROGRAMME · PRO

Balkengrafik- programm

Fritz Blinne, Niddastr. 9, 2805 Stuhr 2
Telefon: (04 21) 56 24 06

Das zum Jahresbeginn an Sie geschickte Programm habe ich hiermit um eine einzugewandene Grafiküberschrift erweitert.

Mit diesem Programm kann eine persönliche Bundeswahlprognose für das Wahljahr 1987 erstellt werden.

Wenn ein Matrixdrucker angeschlossen ist und die HCOPY/MAUS Baugruppe im NKC ihren Stammplatz hat, dann erlaubt dieses Programm eine schnelle Drucker- ausgabe der Balkengrafik. Damit entsteht für jede teilnehmende Person eine Dokumentation seiner Voraussage für die 1987 kommende Bundestags- wahl.

Im Programm befinden sich Routinen, die den Grafikprozessor auf Kursivschrift und unterbrochene Linien umschalten.

Die Eingabe von Werten über 100 oder unter 1 ist nicht möglich, dafür sorgt eine

andere Routine.

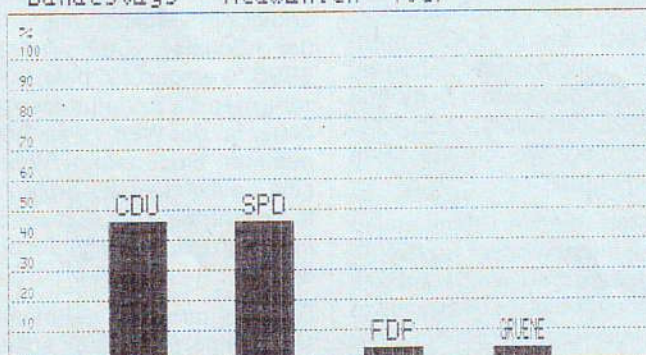
Bei auf dem Bildschirm stehender Grafik hat man vier Möglichkeiten der Pro- grammfortführung. Einmal kann durch Drücken einer beliebigen Taste (aus- genommen ^\$*) die Eingabe wiederholt werden, zum anderen kann nach betä- tigen der '*' Taste die Grafiküberschrift geändert werden.

Eine weitere Möglichkeit ist die Ausgabe einer Hardcopy durch den Drucker, dies erfolgt durch Drücken der '^' Taste. Mit Druck auf die '\$' Taste ist das Programm beendet.

Alles weitere ist aus den Kommentaren im Listing ersichtlich. Auch eine Ände- rung der Routinen, um die Balkengrafik für andere Zwecke zu verwenden, ist leicht möglich.

Für die Ausgabe auf den Drucker muß das Testprogramm von G. Sternberg aus dem Beschreibungsheft der Bau- gruppe HCOPY/MAUS Seite 30 im Spei- cher auf Adresse \$D0000 stehen. Ich habe in meinem NKC auf dieser Adresse einen 8KB statischen Ramchip für spe- zielle Ablagen von Arbeitsprogrammen stecken.

Bundestags - Neuwahlen 1987



GRAMME · PROGRAMME · PROGRAMME · PRO

```
*****
* >> Balken-Grafik << *
* (C) F.Blinne 2.1.1986 *
* Niddastr.9 2805 Stuhr 2 *
* Tel. 0421/562406 *
* das Proamm ist relokativ *
*-----*
* fuer eine Hardcopy muss *
* das Testprogramm von *
* G. Sternberg aus dem *
* Handbuch Seite 30 im *
* Speicher Adresse $D0000 *
* stehen ! *
*****
```

```
ORG $10000
HEAT: DC L $55AA0180
      DC B 'Balkenr'
      DC L START-HEAT
      DC L ENDEA-HEAT
      DC B 1,0,0,0
      DC L 0,0
```

```
CDU EQU $9000
SPD EQU $9010
FDP EQU $9020
ALT EQU $9030
BUFFER EQU $9040
TEXTW EQU $9100
```

```
LINE EQU $FFFFF72
DRUCK EQU $D0010
```

```
MOVETO: MOVE #!MOVETO,D7
        TRAP #1
        RTS
DRAWTO: MOVE #!DRAWTO,D7
        TRAP #1
        RTS
PRINT: MOVE #!WRITE,D7
        TRAP #1
        RTS
EINGB: MOVE #!READ,D7
        TRAP #1
        RTS
WERT: MOVE #!WERT,D7
        TRAP #1
        RTS
LOESCH: MOVE #!CLR,D7
        TRAP #1
        RTS
```

```
PRINTS: MOVE B #!$04,LINIE
        MOVE B (A0)+,D0
        BEQ S BACK
        MOVE #!CO2,D7
        TRAP #1
        BRA S PRINTS
BACK: MOVE B #!$00,LINIE
```

```
START: MOVE #!CLRSCHREIB,D7
        TRAP #1
        MOVE #!CUIROFF,D7
        TRAP #1
        MOVE #!HIDE,D7
        TRAP #1
        LEA COPY(PC),A0
        BSR PRINTS
        LEA BESCH1(PC),A0
        MOVE #!$21,D0
        MOVE #1,D1
        MOVE #240,D2
        BSR PRINT
        LEA BESCH2(PC),A0
        MOVE #!$11,D0
        MOVE #!10,D1
        SUB #22,D2
        BSR PRINT
        LEA BESCHY(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCH2(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCH3(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCH4(PC),A0
        SUB #14,D2
        BSR PRINT
        LEA BESCH5(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCH6(PC),A0
        SUB #14,D2
        BSR PRINT
        LEA BESCH7(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCH8(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCH9(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCHA(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCHX(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCHB(PC),A0
        SUB #12,D2
        BSR PRINT
        LEA BESCHC(PC),A0
        SUB #14,D2
        BSR PRINT
        LEA BESCHD(PC),A0
        SUB #12,D2
        BSR PRINT
        MOVE #!CI,D7
        TRAP #1
        BSR LOESCH
```

```
NEU1: LEA TEXTB(PC),A0
      MOVE #!$21,D0
      MOVE #60,D1
      MOVE #230,D2
      MULU #2,D0
      MOVE D0,FDP
FX4: MOVE #!$12,D0
     MOVE #450,D1
     MOVE #150,D2
     LEA TEXT4(PC),A0
     BSR PRINT
     MOVE #!$22,D0
     SUB #20,D2
     MOVE #3,D3
     LEA BUFFER,A0
     BSR EINGB
     LEA BUFFER,A0
     BSR WERT
     CMP #100,D0
     BGT FX4
     CMP #1,D0
     BLT FX4
     MULU #2,D0
     MOVE D0,ALT
     BSR LOESCH
     CLR L D2
     MOVE B #!$01,LINIE
     MOVE #!10-1,D3
     WDH0: CLR D1
          ADD #20,D2
          BSR MOVETO
          MOVE #512,D1
          BSR DRAWTO
          DBRA D3,WDH0
          MOVE #!WAIT,D7
          TRAP #1
          MOVE B #!$00,LINIE
          CLR D1
          CLR D2
          BSR MOVETO
          MOVE #230,D2
          BSR DRAWTO
          MOVE #511,D1
          BSR DRAWTO
          CLR D2
          BSR DRAWTO
          CLR D1
          BSR DRAWTO
          LEA PROZ(PC),A0
          MOVE #!$21,D0
          MOVE #10,D1
          MOVE #214,D2
          BSR PRINT
          MOVE #!$11,D0
          LEA HUND(PC),A0
          MOVE #202,D2
          BSR PRINT
          MOVE #!$22,D0
          MOVE #14,D1
          MOVE #124,D2
          MOVE #40,D3
          LEA TEXTW,A0
          BSR EINGB
          BSR LOESCH
BEGIN: LEA TEXTA(PC),A0
      MOVE #!$21,D0
      MOVE #60,D1
      MOVE #230,D2
      BSR PRINT
FX1: MOVE #!$22,D0
     MOVE #2,D1
     MOVE #150,D2
     LEA TEXT1(PC),A0
     BSR PRINT
     SUB #20,D2
     MOVE #3,D3
     LEA BUFFER,A0
     BSR EINGB
     LEA BUFFER,A0
     BSR WERT
     CMP #100,D0
     BGT FX1
     CMP #1,D0
     BLT FX1
     MULU #2,D0
     MOVE D0,CDU
FX2: MOVE #!$22,D0
     MOVE #150,D1
     MOVE #150,D2
     LEA TEXT2(PC),A0
     BSR PRINT
     SUB #20,D2
     MOVE #3,D3
     LEA BUFFER,A0
     BSR EINGB
     LEA BUFFER,A0
     BSR WERT
     CMP #100,D0
     BGT FX2
     CMP #1,D0
     BLT FX2
     MULU #2,D0
     MOVE D0,SPD
FX3: MOVE #!$22,D0
     MOVE #300,D1
     MOVE #150,D2
     LEA TEXT3(PC),A0
     BSR PRINT
     SUB #20,D2
     MOVE #3,D3
     LEA BUFFER,A0
     BSR EINGB
     LEA BUFFER,A0
     BSR WERT
     CMP #100,D0
     BGT FX3
     CMP #1,D0
     BLT FX3
```



```

000302 303C 0012 MOVIE #18,00
000304 4E9F 00000028 JSR SLO
000306 4E75 RTS
00030E 303C 0018 FETT: MOVIE RESEC,00 #FETTDRUCK
000312 4E9F 00000028 JSR SLO
000318 303C 0045 MOVIE #E',00
000322 4E75 RTS
000324 303C 0018 FETTAUS: MOVIE RESEC,00 #FETTAUS AUS
000328 4E9F 00000028 JSR SLO
000332 303C 0046 MOVIE #F',00
000338 4E75 RTS
00033A 303C 0018 LINK: MOVIE RESEC,00 #LINKER RAND
00033E 4E9F 00000028 JSR SLO
000344 303C 004C MOVIE #I',00
000348 4E9F 00000028 JSR SLO
00034E 303C 0005 MOVIE #S,00
000352 4E9F 00000028 JSR SLO
000358 4E75 RTS
00035A 303C 0018 ESCN: MOVIE RESEC,00 #UBERSPRINGEN PERFORATION
00035E 4E9F 00000028 JSR SLO
000364 303C 004E MOVIE #N',00
000368 4E9F 00000028 JSR SLO
00036E 303C 0005 MOVIE #S,00
000372 4E9F 00000028 JSR SLO
000378 4E75 RTS
00037A 303C 0018 ESCD: MOVIE RESEC,00 #UBERSPRINGEN PERFORATION AUS
00037E 4E9F 00000028 JSR SLO
000384 303C 004F MOVIE #D',00
000388 4E9F 00000028 JSR SLO
00038E 4E75 RTS
000390 303C 0018 PAP: MOVIE RESEC,00 #PAPIERERKENNUNG AUS
000394 4E9F 00000028 JSR SLO
00039A 303C 0038 MOVIE #Q',00
00039E 4E9F 00000028 JSR SLO

0003A4 4E75 RTS
0003A6 303C 0018 PAPERIN: MOVIE RESEC,00 #PAPIERERKENNUNG EIN
0003AA 4E9F 00000028 JSR SLO
0003B0 303C 0039 MOVIE #R',00
0003B4 4E9F 00000028 JSR SLO
0003BA 4E75 RTS
0003BC 363C 0063 SCHLEIFE:MOVIE #100-1,D1 #VERZUEBERUNGSSCHLEIFE
0003C0 4E9F 00000E38 MARTE: JSR ISYNC
0003C6 6700 FFFF BEG MARTE
0003CA 51CB FFFF DBRA D3,MARTE
0003CE 4E75 RTS
0003D0 4570736FAE2052 TITEL: DC.B 'Epson RX',0 #*TEXTE
0003D6 5800
0003D8 4475737449A572 UNTER: DC.B 'Justierung',0
0003DE 75AE6700
0003E4 40A5AEFD00 MENUEM: DC.B 'Menu',0
0003E8 41202020456A9 ELITEM: DC.B 'a - Elite',0
0003F0 746500
0003F6 62202020506A963 PICAM: DC.B 'b - Pica',0
0003FA 4100
0003FC 63202020427265 SOM: DC.B 'c - Breitschrift eine Zeile',0
000402 697473636872A9
00040A 667420A56A6E6A5
000410 20A6A56A6E6A500
000418 64202020427265 ESCDM: DC.B 'd - Breitschrift',0
00041F 697473636872A9

Rolf-D.Klein a8000/08 Assembler 4.3 (C) 1984, Seite 5

000426 667400
000428 65202020427265 ESCWAUEN:DC.B 'e - Breitschrift aus',0
000430 697473636872A9
000438 667420A1757500
00043E 66202020506A963 SIM: DC.B 'f - Schallschrift',0
000445 40A16C73636872
00044C 69A67400

000450 6720202053A368 STAUSM: DC.B 'g - Schallschrift aus',0
000457 40A16C73636872
00045E 69A67420A175753
000465 00
000466 6620202046A574 FETTM: DC.B 'h - Fettdruck',0
00046D 74447275A3A900
000474 6920202046A574 FETTAUSM:DC.B 'i - Fettdruck aus',0
00047B 74447275A3A920
000482 61757300
000486 6620202046A574 LINKM: DC.B 'k - Linker Rand',0
00048D 68A5722052A16E
000494 6400
000496 6C20202000A265 PERM: DC.B 'l - Überspringen der Perforation',0
00049D 72737072A9A6E7
0004A4 65A6E20A4A57220
0004AB 50A572A6A72A1
0004B2 74A9AFA500
0004B8 60202000A265 PERAUSM: DC.B 'm - Überspringen der Perforation aus',0
0004BE 72737072A9A6E7
0004C5 65A6E20A4A57220
0004CC 50A572A6A72A1
0004D3 74A9AFA50A6175
0004DA 7300
0004DC 6E202020506A170 PAPM: DC.B 'n - Papiererkennung aus',0
0004E3 69A572A572A8A5
0004EA 6E6E756E672061
0004F1 757300
0004F4 6F202020506A170 PAPERINM: DC.B 'o - Papiererkennung ein',0
0004FB 69A572A572A8A5
000502 6E6E756E672065
000509 69A500
00050C 7020202040A5A6 MENM: DC.B 'p - Menü',0
000513 F900
000515 ENDE: END

000DF6 Ende-Symboltabelle

```

IC's beschriften mit GOSI68K

Rolf-Dieter Klein

Mein GOSI68K-Compiler ist bisher arg zu kurzgekommen. Das soll sich nun ändern. Anhand kleiner Programmbeispiele möchte ich es Ihnen schmackhaft machen, mit dem GOSI-Compiler zu arbeiten.

Entscheidend ist, daß der Compiler Maschinen-(Assembler)-Code erzeugt und damit sehr schnelle Programme liefert. GOSI68K besitzt (noch) keine REAL-Arithmetik. Wenn man eine REAL-Arithmetik braucht, kann man z. B. PASCAL/S einsetzen. GOSI soll jedoch nur für schnelle und für Grafikanwendungen verwendet werden. An eine Erweiterung für Gleitkomma ist gedacht, sobald die schnelle FPU (floating point arithmetik unit 68881 mit ca. 2 Mikrosekunden pro Gleitkommaaddition) für eine breitere Personenschicht verfügbar ist.

GOSI ist eine Sprache, die mit LOGO verwandt ist, jedoch nicht die Listenstruktur von LOGO verwendet, sondern die konventionellen Feld-Konzepte.

```

lerne testen :n
wenn :n=0 [rk]
blinker 0 :n
dr :n
testen :n-1
ende

```

```

lerne pad :x :y :i
sh aufxy 6*:*x-4 512-20*(y+1)-1 sa
aufkurs 90
wh 2 [vw 20 re 90 vw 18 re 90]
blinker :x :y
dr :i
ende

```

```

lerne gehaeuse :n
sh aufxy (40-10)*6-4+18 512-20*(n+1)-1-10 sa
aufkurs 90
vw :n/2*40 re 90
vw 6*10-9-1
vw 11 re 90 wh 18 [vw 2 re 10]
sh wh 18 [vw 2 re 10] sa li 90 rw 10
vw 6*10-9 re 90
vw :n/2*40 re 90
vw 6*20-18-1 re 90
ende

```

```

lerne ic :n
setze "x 40-10
setze "y 2
setze "i 1
wh :n/2 [
pad :x :y :i
setze "i :i+1
setze "y :y+2
]
setze "x 40+10
setze "y 2
setze "i :n
wh :n/2 [
pad :x :y :i
setze "i :i-1
setze "y :y+2
]
ende

```

```

zfeld "icname 30
zfeld "hilf 80
zfeld "pins 40

```

```

lerne zaehle
setze "count 0
solange :hilf[:count] <> 0
[ setze "count :count+1]
rg :count
ende

```

```

lerne nameein
blinker 40-10 0
setze "icname :eingabe 20
setze "hilf :icname
blinker 40-(zaehle)/2 0
dr :icname
ende

```

```

lerne pinein :n
setze "y 2
wh :n/2 [
blinker 0 :y
setze "pins :eingabe 28
setze "hilf :pins
blinker 28-(zaehle) :y
dr :pins
setze "i :i+1
setze "y :y+2
]
setze "x 40+14
setze "y 2
setze "i :n
wh :n/2 [
blinker :x :y
setze "pins :eingabe 28
blinker :x :y
dr :pins
setze "i :i-1
setze "y :y+2
]
ende

```

```
zfeld "komm 40
```

```

lerne kommentar :n
setze "x 40-7
setze "y 3
wenn :n>2 [
wh :n-2 [

```

```

blinker :x :y
setze "komm :eingabe 15
setze "hilf :komm
blinker 40-(zaehle)/2 :y
dr :komm
setze "y :y+1
]
]
ende

```

```

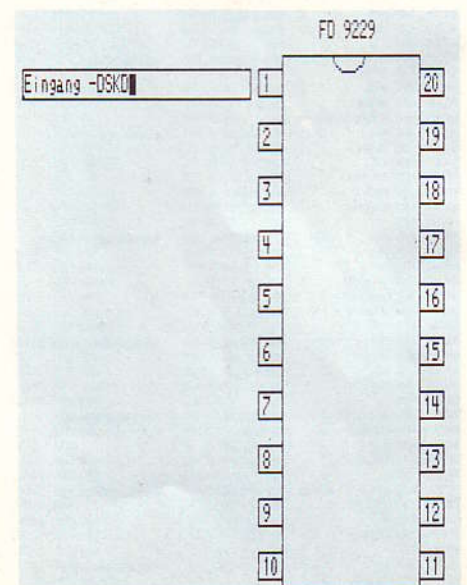
lerne icaus
bild vi seite 0 0
blinker 20 10
dr [Pin-Anzahl eingeben:]
setze "anzahl :zahl 10
wenn :anzahl < 2 [rk]
bild vi seite 0 0
ic :anzahl
gehaeuse :anzahl
nameein
pinein :anzahl
kommentar :anzahl
solange :taste? [ ]
setze "a :taste
icaus
ende

```

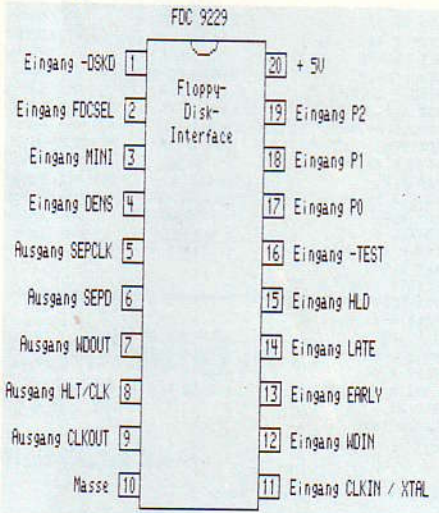
```
icaus
```

Doch nun genug der Theorie. Bild 1 zeigt das Programm. Es hat die Aufgabe, die Umriss eines IC's auf dem Bildschirm zu zeichnen. Dazu gibt man zuerst die Anzahl der Pins an, wie in Bild 2 gezeigt. Dann wird das IC gezeichnet.

Pin-Anzahl eingeben:



Nun kann man das IC beschriften. Dazu erscheint jeweils ein Eingabefenster, in das man den Text eingibt. Bild 3 zeigt einen Ausschnitt aus diesem Vorgang. Am Schluß erhält man z. B. Bild 4. Nun kann man z. B. mit Hilfe des Hardcopy-Programms das Bild auf einem Drucker ausgeben. Das Programm stellt nur einen Anfang einer Idee dar, die man auch noch ausbauen kann. So könnte man alle Beschriftungen speichern und damit eine Bibliothek aufbauen, bei der man nur den IC-Namen eingibt und schon das Anschlußschema auf den Bildschirm bekommt.



Das Programm selbst ist in mehrere Prozeduren geteilt. Das erste Unterprogramm ist nur als Hilfe zur Programm-erstellung gedacht und gibt bei Aufruf eine Zeilennummerierung auf dem Bildschirm aus.

Das Unterprogramm PAD zeichnet ein Rechteck auf dem Bildschirm, in dem eine Ziffer abgelegt wird. Dazu wird die Koordinate x und y, sowie die Pinnummer i angegeben. x und y sind hier nicht in Punktrastereinheiten, sondern in Zeilen und Spalten anzugeben. Dadurch war es möglich, den Befehl BLINKER zum Positionieren bei der Textausgabe zu verwenden. Man kann Texte in GOSI natürlich auch an jeder beliebigen Stelle ausgeben, doch dazu muß man mit CMD Texte direkt ausgeben. Das Unterprogramm GEHÄUSE :n gibt die Umriss des Gehäuses ohne Pins, bei einem n-poligen IC aus.

Das Unterprogramm IC schließlich gibt alle PADs eines n-poligen IC's aus. Auf dem Bildschirm lassen sich allerdings nur maximal 24-polige IC's ausgeben, so wie das Programm jetzt aufgebaut ist.

Jetzt werden drei Felder definiert. Das Zeichenfeld ICNAME enthält den Namen des IC's, HILF dient als Hilfsfeld zur Parameterübertragung in das Unterprogramm ZÄHLE, das die Anzahl der Zei-

chen HILF zählt. Damit kann man bestimmen, wieviele Zeichen durch den Benutzer bei der Beschriftung eingegeben wurden. Denn das Programm gleicht automatisch den Rand aus, also Beschriftung auf der linken Seite vom IC wird rechtsbündig ausgegeben, Beschriftung auf der rechten Seite linksbündig und die Beschriftung in der Mitte ist zentriert.

Das Programm NAMEEIN trägt die Überschrift ein. Mit PINEIN werden die einzelnen Pins beschriftet.

Das Zeichenfeld KOMM ist für die Mittenbeschriftung gedacht. Das Unterprogramm KOMMENTAR trägt damit die Beschriftung in die Mitte des IC's ein.

Das Unterprogramm IC schließlich ruft alle bisherigen Unterprogramme in der richtigen Reihenfolge auf. Wenn man 0 als Pinzahl eingibt, wird dieses Unterprogramm wieder verlassen.

Am Schluß steht noch ICAUS, damit wird das Unterprogramm, das eigentliche Hauptprogramm also, aufgerufen.

Abschließend noch ein Hinweis. Aktuell ist die Version 3.1 des GOSI-Compilers. Wer noch alte Versionen 2.0 besitzt sollte sie schnell umtauschen, denn es haben sich einige Dinge geändert.

Hardcopy von der COL256

9 Graustufen auf einem EPSON-Drucker

von Willi Sicking

Das nachfolgende Programm erlaubt die Hardcopy eines Farbbildes auf einem EPSON-Drucker, wobei 9 Graustufen ausgegeben werden.

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

```

OE9C00
OE9C00
OE9C00
OE9C00
OE9C00
OE9C00
OE9C00
OE9C00
OE9C00
OE9C00
*****
*   HARDCOPYPROGRAMM MIT 9. GRAUSTUFEN   *
*   *                                     *
*   VON WILLI SICKING                     *
*   *                                     *
*****
*   VERS COL256, 1.1 RDK                 *
DRG $EB000
PAGE EDU $FFFFFFCE * 16 KBYTE SEITE DER COL256
*
*   ZEILE INITIALISIEREN
*
OE0000 103C 001B      INI:MOVE.B #27,D0 * ESC
OE0004 4EB9 000E0D28 JSR @LO
OE000A 103C 004C      MOVE.B #'L',D0 * BITIMAGE 960 PUNKTE
OE000E 4EB9 000E0D28 JSR @LO
OE0014 103C 008C      MOVE.B #18B,D0
OE0018 4EB9 000E0D28 JSR @LO
OE001E 103C 0003      MOVE.B #3,D0 * 3X256+188 PUNKTE
OE0022 4EB9 000E0D28 JSR @LO * JE ZEILE
OE0028 4E75          RTS
OE002A
*
OE002A * ZEILENABSTAND
OE002A
*
OE002A ABS:
OE002A 103C 001B      MOVE.B #27,D0 * ESC
OE002E 4EB9 000E0D28 JSR @LO
OE0034 103C 0041      MOVE.B #'A',D0 * ABSTAND
OE0038 4EB9 000E0D28 JSR @LO * AUF
OE003E 103C 000B      MOVE.B #8,D0 * B/72 ZOLL
OE0042 4EB9 000E0D28 JSR @LO * EINSTELLEN
OE0048 4E75          RTS
OE004A
*
OE004A * ADRESSBERECHNUNG
OE004A
*
OE004A * FUER HARDCOPY AUS DEM VIRT. BILDSPEICHER
ADR:VIRT:
OE004A 2E02          MOVE.L D2,D7 * Y
OE004C E19F          ROL.L #8,D7 * #256
OE004E BEB1          OR.L D1,D7 * X
OE0050 0687 000B0000 ADDI.L #80000,D7 * +STARTADRESSE
OE0056 2047          MOVEA.L D7,A0 * =ZIEL
OE0058 4E75          RTS

```

```

OE805A
OE805A
OE805A
OE805A
OE805A
OE805A
OE805A 4BE7 7000      MOVEM.L D1-D3,-(A7) * REV 1.1
OE805E 4601          NOT.B D1
OE8060 4602          NOT.B D2 * Y
OE8062 3602          MOVE.W D2,D3
OE8064 E51B          ROL.B #2,D3 * 2 MS BIT
OE8066 0203 0003      AND.B #3,D3 * MASKIEREN
OE806A 0003 0080      DR.B #80,D3 * SEITE BERECHNET
OE806E 13C3 FFFFFFFCE MOVE.B D3,PAGE * COL256 SEITE ANNAEHLN
OE8074 41F9 0000C000 LEA.L #C000,A0 * STARTADRESSE
OE807A 3602          MOVE.W D2,D3 * Y
OE807C 0243 003F      AND.W #3F,D3 * Y-ADRESSE (MAX 64 ZEILEN)
OE8080 E15B          ROL.W #8,D3 * PRO SEITE BERECHNEN
OE8082 D641          ADD.W D1,D3 * UND MIT X VERBINDEN
OE8084 0243 3FFF      AND.W #3FFF,D3 * MAX 16 KBYTE PRO SEITE

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 2

```

OE808B D0C3          ADDA.W D3,A0
OE808A 4CDF 000E      MOVEM.L (A7)+,D1-D3
OE808E 4E75          RTS
OE8090
*
OE8090 * AUSWERTUNG
AUSW:
OE8090 1E10          MOVE.B (A0),D7 * BILDPUNKT LADEN
OE8092 02B7 000000FF ANDI.L #FFF,D7
OE8098 0C47 0000      CMPI.L #0,D7
OE809C 6700 00D6      BEQ VB
OE80A0 0C47 0040      CMP #201000000,D7
OE80A4 6700 00CE      BEQ VB
OE80A8 0C47 0080      CMP #210000000,D7
OE80AC 6700 00C6      BEQ VB
OE80B0 0C47 00C0      CMP #211000000,D7
OE80B4 6700 00BE      BEQ VB
OE80B8 0C47 0015      CMP #200010101,D7
OE80BC 6700 00B6      BEQ VB
OE80C0 0C47 0055      CMP #201010101,D7
OE80C4 6700 00AE      BEQ VB
OE80C8 0C47 0095      CMP #210010101,D7
OE80CC 6700 0094      BEQ V7
OE80D0 0C47 00D5      CMP #211010101,D7
OE80D4 6700 00BC      BEQ V7

```

```

0EB0DB 0C47 002A CMP #X00101010,D7
0EB0DC 6700 008A BEQ V7
0EB0ED 0C47 006A CMP #X01101010,D7
0EB0EE 6700 006A BEQ V6
0EB0EF 0C47 00AA CMP #X10101010,D7
0EB0F0 6700 0050 BEQ V5
0EB0F1 0C47 00EA CMP #X11101010,D7
0EB0F2 6700 003A BEQ V4
0EB0F3 0C47 003F CMP #X00111111,D7
0EB0F4 6700 002A BEQ V3
0EB0F5 0C47 007F CMP #X01111111,D7
0EB0F6 6700 0012 BEQ V2
0EB0F7 0C47 00BF CMP #X10111111,D7
0EB0F8 6700 0004 BEQ V1
0EB0F9 4E75 RTS

```

* BLEIBT WEISS

```

0EB112 V1:
0EB112 0006 0001 ORI.B #1,D6
0EB113 4E75 RTS * 61
0EB114 V2:
0EB114 0006 0001 ORI.B #1,D6
0EB115 0004 0002 ORI.B #2,D4
0EB116 4E75 RTS * 62
0EB117 V3:
0EB117 0006 0003 ORI.B #3,D6
0EB118 0004 0001 ORI.B #1,D4
0EB119 4E75 RTS * 63
0EB120 V4:
0EB120 0006 0001 ORI.B #1,D6
0EB121 0005 0002 ORI.B #2,D5
0EB122 0004 0001 ORI.B #1,D4
0EB123 0003 0002 ORI.B #2,D3
0EB124 4E75 RTS * 64
0EB125 V5:
0EB125 0006 0003 ORI.B #3,D6
0EB126 0005 0002 ORI.B #2,D5
0EB127 0004 0001 ORI.B #1,D4
0EB128 0003 0002 ORI.B #2,D3
0EB129 4E75 RTS * 65
0EB130 V6:

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 3

```

0EB150 0006 0003 ORI.B #3,D6
0EB151 0005 0002 ORI.B #2,D5
0EB152 0004 0001 ORI.B #1,D4
0EB153 0003 0003 ORI.B #3,D3
0EB154 4E75 RTS * 66
0EB155 V7:
0EB155 0006 0003 ORI.B #3,D6
0EB156 0005 0003 ORI.B #3,D5
0EB157 0004 0003 ORI.B #3,D4
0EB158 0003 0001 ORI.B #1,D3
0EB159 4E75 RTS * 67
0EB160 V8:
0EB160 0006 0003 ORI.B #3,D6

```

```

0EB178 0004 0003 ORI.B #3,D4
0EB17C 0005 0003 ORI.B #3,D5
0EB180 0003 0003 ORI.B #3,D3
0EB184 4E75 RTS * SCHWARZ
0EB186 *
0EB186 *
0EB186 * HAUPTPROGRAMM
0EB186 *
0EB186 H:
0EB186 243C 00000000 MOVE.L #0,D2 * PROGRAMMSTART
0EB18C 4EB9 000E02A JSR ABS * Y:=0
0EB192 4EB9 000E0000 NZ:JSR INI * ZEILENABSTAND
0EB198 223C 00000000 MOVE.L #0,D1 * NEUE ZEILE INITIALIS.
0EB19E CLEAR: * X:=0
0EB19E 4283 CLR.L D3
0EB1A0 4284 CLR.L D4
0EB1A2 4285 CLR.L D5
0EB1A4 2C3C 00000001 MOVE.L #1,D6 * UEBERLAUFFLAG (ZAEHLER)
0EB1AA ROTATE:
0EB1AA 4EB9 000E05A JSR ADR * REV 1.1
0EB1B0 E59B ROL.L #2,D3
0EB1B2 E59C ROL.L #2,D4
0EB1B4 E59D ROL.L #2,D5
0EB1B6 E59E ROL.L #2,D6 * 2 BIT AUFSCHIEBEN
0EB1B8 4EB9 000E090 JSR AUSW * GRAUWERT EINTRAGEN
0EB1BE 0642 0001 ADD #1,D2 * Y ERHOEHEN.
0EB1C2 0C46 0100 CMPI.W #100,D6 * 4 PUNKTE AUSGEWERTET ?
0EB1C6 6D00 FFE2 BLT ROTATE
0EB1CA 0442 0004 SUB.W #4,D2 * WIEDER ZURUECK.
0EB1CE 1003 MOVE.B D3,D0
0EB1D0 4EB9 000E02B JSR @D0
0EB1D6 1004 MOVE.B D4,D0
0EB1D8 4EB9 000E02B JSR @D0
0EB1DE 1005 MOVE.B D5,D0
0EB1E0 4EB9 000E02B JSR @D0
0EB1E6 1006 MOVE.B D6,D0 * 4 PUNKTE
0EB1E8 4EB9 000E02B JSR @D0 * AUSDRUCKEN
0EB1EE 0681 00000001 ADDI.L #1,D1 * X:=X+1
0EB1F4 0C81 000000EE CMPI.L #238,D1 * X=XMAX ?
0EB1FA 6F00 FFA2 BLE CLEAR
0EB1FE 103C 000D MOVE.B #30D,D0
0EB202 4EB9 000E02B JSR @D0
0EB208 103C 000A MOVE.B #30A,D0
0EB20C 4EB9 000E02B JSR @D0 * CR UND LF
0EB212 223C 00000000 MOVE.L #0,D1 * NEUE ZEILE X:=0
0EB218 0642 0004 ADDI.W #4,D2 * Y:=Y+4
0EB21C 0C42 0100 CMPI.W #100,D2 * BILD FERTIG ?
0EB220 6D00 FF70 BLT NZ
0EB224 4239 FFFFFFFE CLR.B PAGE
0EB22A 4E75 RTS
0EB22C

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 4

0EBEE0 Ende-Symboltabelle

Hilfsprogramme für den 68008 Teil 2

Hier ist das Zweite Hilfsprogramm für den 68008. Das Problem: Daten und Programme im Speicher verschieben. Die Lösung: RUBATRANS 68!

Mit der Routine lassen sich Daten und Programm im gesamten Speicherwerk beliebig hin und her transferieren. Das Programm ist wieder relokativ und kann

```

Rubatrans 68
(C) 1985 by Ruediger Baecker

Startadresse ==> $400
Endadresse ==> $1000
Ablageadresse ==> $30000

```

Bild 1: Die Eingabe der Datenbereiche erfolgt in HEX

aus der Bibliothek aufgerufen werden. Es werden nach dem Aufruf die Startadresse, die Endadresse und die Ablageadresse in HEX eingegeben (Bild 1). Dann wird der Transfer mit 'CR' gestartet. Nach erfolgter Verschiebung der Daten kann man mit 'm' wieder in das Grundprogramm zurückspringen.

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

```

000400 ORG $400
000400 *
000400 * RUBATRANS 68
000400 *
000400 * ROUTINE ZUM VERSCHIEBEN VON SPEICHERBEREICHEN
000400 *
000400 * COPYRIGHT (C) 1985 BY RUEDIGER BAECKER - POSTFACH 4111 - 5820 GEVELSBERG
000400 *
000400 KOPF:
000400 55AA01B0 DC.L $55AA01B0 * EINTRAG IN BIBLIOTHEK
000404 52756261747261 DC.B 'Rubatran'
000408 &E
00040C 00000020 DC.L RUBATRANS-KOPF
000410 0000010C DC.L TRANSEND-KOPF
000414 01 DC.B 1
000415 00 00 00 DC.B 0,0,0
00041B 00000000 DC.L 0,0
00041C 00000000
000420
= 000009DA ZIELADR EQU $9DA * ADRESSE EINGABEBUFFER
= FFFFFFFF68 TAST EQU $FFFFFF68
000420
000420 RUBATRANS:
000420 3E3C 0011 MOVE #!CLPG,D7 * BILDSCHIRM LOESCHEN
000424 4E41 TRAP #1
000426
000426 4239 FFFFFFFF69 CLR.B TAST+1 * TATATURPORT ZURUECKSETZEN
00042C 4280 CLR.L D0 * REGISTER LOESCHEN
00042E 4281 CLR.L D1

```

```

000430 4282 CLR.L D2
000432 4283 CLR.L D3
000434
000434 41FA 0006 LEA TEXT9 (PC),#0 * TEXTE AUSGEBEN
000438 103C 0043 MOVE.B #43,D0
00043C 123C 0000 MOVE.B #90,D1
000440 143C 000B MOVE.B #200,D2
000444 3E3C 000A MOVE #!WRITE,D7
000448 4E41 TRAP #1
00044A 0442 002D SUB #45,D2
00044E 41FA 0105 LEA DT2(PC),#0
000452 103C 0032 MOVE.B #432,D0
000456 3E3C 000A MOVE #!WRITE,D7
00045A 4E41 TRAP #1
00045C 0442 002B SUB #40,D2
000460 103C 0021 MOVE.B #821,D0
000464 41FA 00B3 LEA TEXT10(PC),#0
000468 3E3C 000A MOVE #!WRITE,D7
00046C 4E41 TRAP #1
00046E
00046E 0442 0014 SUB #20,D2
000472 41FA 00B9 LEA TEXT11(PC),#0
000476 3E3C 000A MOVE #!WRITE,D7
00047A 4E41 TRAP #1
00047C 0442 0014 SUB #20,D2
000480 41FA 00BF LEA TEXT12(PC),#0
000484 3E3C 000A MOVE #!WRITE,D7
000488 4E41 TRAP #1
00048A
00048A GETW1:
00048A 103C 0021 MOVE.B #821,D0
00048E 143C 0073 MOVE.B #115,D2
000492 123C 00FA MOVE.B #250,D1

```

```

000496 163C 0009 MOVE.B #9,D0
00049A 41ED 09DA LEA ZIELADR(AS),A0
00049E 3E3C 000B MOVE #1,READ,D7
0004A2 4E41 TRAP #1
0004A4 41ED 09DA LEA ZIELADR(AS),A0
0004A8 1810 MOVE.B (A0),D4
0004AA 0C04 006D CMP.B #'m',D4
0004AE 6700 005A BEQ DUMPEND
0004B2 3E3C 001D MOVE #1,MERT,D7
0004B6 4E41 TRAP #1
0004BB 2640 MOVEA.L D0,A3
0004BA
0004BA GETW2:
0004BA 143C 005F MOVE.B #95,D2
0004BE 103C 0021 MOVE.B #21,D0
0004C2 123C 00FA MOVE.B #250,D1
0004C6 41ED 09DA LEA ZIELADR(AS),A0
0004CA 3E3C 000B MOVE #1,READ,D7
0004CE 4E41 TRAP #1
0004D0 41ED 09DA LEA ZIELADR(AS),A0
0004D4 3E3C 001D MOVE #1,MERT,D7
0004D8 4E41 TRAP #1
0004DA 2640 MOVEA.L D0,A4
0004DC
0004DC GETM3:
0004DC 143C 004B MOVE.B #75,D2
0004E0 103C 0021 MOVE.B #21,D0
0004E4 123C 00FA MOVE.B #250,D1
0004E8 41ED 09DA LEA ZIELADR(AS),A0
0004EC 3E3C 000B MOVE #1,READ,D7
0004F0 4E41 TRAP #1
0004F2 41ED 09DA LEA ZIELADR(AS),A0
0004F6 3E3C 001D MOVE #1,MERT,D7
0004FA 4E41 TRAP #1
0004FC 2640 MOVEA.L D0,A5
0004FE

```

* 'm' GEDRUECKT ?
* JA, DANN ENDE

```

0004FE DATTRANS:
0004FE 1ADB MOVE.B (A3)+,(A5)+
000500 89CB CMPA.L A3,A4
000502 6700 0066 BEQ ENDE
000506 8000 FFF6 BRA DATTRANS
00050A DUMPEND:
00050A ENDE:
00050A 4E75 RTS
00050C TRANSEND:
00050C
00050C TEXT9:
00050C 52756261747261 DC.B 'Rubatrans 68',0
000513 6E7320363800
000519 TEXT10:
000519 53746172746164 DC.B 'Startadresse ==',0
000520 72657373652020
000522 202030303E00
00052D TEXT11:
00052D 456E461647265 DC.B 'Endadresse ==',0
000534 737365202020
00053B 202030303E00
000541 TEXT12:
000541 41626C61676561 DC.B 'Ablageadresse ==',0
00054B 64726573736520
00054F 202030303E00
000555 DT2:
000555 28432920310938 DC.B '(C) 1985 by Ruediger Baecker',0
Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 3
00055C 35206279205275
000563 65646967657220
00056A 42616563686572
000571 00
000572
0EBBEC Ende-Symboltabelle

```

Einführung in C Teil 4

Rolf-Dieter Klein

Immer wieder wird gefragt, wie man von C aus den Drucker ansprechen kann. Diese Lösung ist in C beim 68000 sehr einfach. Man eröffnet eine Datei mit dem Namen LST:. Leider ist dieser Mechanismus nicht auf andere C-Systeme übertragbar und so muß man im jeweiligen Handbuch immer nachsehen.

Das Beispiel zeigt ein einfaches Programm, das den Text „Hallo es geht“ auf dem Drucker ausgibt. Dazu wird in Zeile 12 eine Datei eröffnet. Diese Datei bekommt den Namen „LST:“. Wenn man dann etwas auf den Drucker ausgeben will, so verwendet man anstelle der Routine printf einfach die Routine fprintf. Dort gibt man als ersten Parameter die Variable „drucker“, damit fprintf weiß, wohin die Ausgabe gehen soll. Danach folgen die Daten wie gewohnt. Man kann alles, was man mit printf auf den Bildschirm ausgeben kann, mit fprintf auch auf den Drucker ausgeben.

Im nächsten Teil, der dann auch wieder etwas ausführlicher ist, werden wir die Dateiverwaltung im allgemeinen behandeln.

```

1:
2: /* Ansprechen des Druckers unter C */
3: /* Rolf-Dieter Klein 860116 1.0. */
4:
5: #include <stdio.h>
6:
7: main()
8: {
9: FILE *fopen();
10: FILE *drucker;
11:
12: drucker = fopen("lst:","w");
13:
14: fprintf(drucker,"Hallo es geht\n");
15: fprintf(drucker,"-----\n");
16:

```

An die Redaktion der Loop!

Als interessierter Nachbauer des NDR-Computers programmiere ich auch. Damit meine Programme nicht ganz untergehen, schicke ich Ihnen ein kleines, nettes Programm zum Thema Grafik für eine der nächsten LOOP's. Wenn es Ihnen gefällt, so können Sie es abdrucken.

Eins sei noch dazu gesagt: Es entstand aus einem Programm des 68008-Grundprogramm Handbuch (Nr. 17, clpg, eine Uhr, die verkehrt herum geht). Es sind nur zwei oder drei Befehle geändert!

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

```

0E9C00 START:
0E9C00 33FC 0001 MOVE #1,SCHREIBSEITE
0E9C04 000E9C8B
0E9C08 33FC 0000 MOVE #0,LESESEITE
0E9C0C 000E9C8A
0E9C10 3B3C 0000 MOVE #0,D4
0E9C14 SCHLEIFE:
0E9C14 3079 000E9C8B MOVE SCHREIBSEITE,D0
0E9C1A 3279 000E9C8A JBR #NEUPAGE
0E9C20 4EB9 000E0ECC MOVE LESESEITE,D1
0E9C26 4EB9 000E0D6E JBR #ERAFEN
0E9C2C 323C 0100 MOVE #256,D1
0E9C30 343C 00B0 MOVE #128,D2
0E9C34 4EB9 000E0EE2 JBR #MOVETO
0E9C3A 4EB9 000E0D7C JBR #SETPEN
0E9C40 3004 MOVE D4,D0
0E9C42 4EB9 000E1030 JBR #COS
0E9C48 0640 0100 ADD #256,D0
0E9C4C 3200 MOVE D0,D1
0E9C4E 3004 MOVE D4,D0
0E9C50 4EB9 000E1034 JBR #SIN
0E9C56 48C0 EXT.L D0
0E9C5B 81FC 0002 DIVE #2,D0
0E9C5F 0640 00B0 ADD #128,D0
0E9C60 3400 MOVE D0,D2
0E9C62 4EB9 000E0F4C JBR #DRAWC
0E9C68 0A79 0001 EOR #1,SCHREIBSEITE
0E9C6C 000E9C8B
0E9C70 0A79 0001 EOR #1,LESESEITE
0E9C74 000E9C8A
0E9C78 0644 000A ADD #10,D4
0E9C7C 4EB9 000E0FDB JBR #CBTS
0E9C82 6700 FFF6 BEQ SCHLEIFE
0E9C86 4E75 RTS
0E9C8B 0000 SCHREIBSEITE: DC.W 0
0E9C8A 0000 LESESEITE: DC.W 0
0E9C8C
0EBBEC Ende-Symboltabelle

```

Andreas Weiland
Röpers Weide 11, 2000 Hamburg 52
Telefon: (040) 880 10 19

Sonderangebote bei GES – Angebot freibleibend:

13 Netzteile, entsprechen NE 1, jedoch ohne 24 V.

Spannungen:

+ 5 V/6 A

– 5 V/0.5 A

+ 12 V/1 A

– 12 V/1 A

Preis pro Stück

148,— DM

Leiterplatten, zweite Wahl, mit kleinen, meist gekennzeichneten Fehlern.

Ohne Garantie, Pausch-Reparatur gilt nicht, event. ältere Revision:

17 CPU80r3

17 SERr3

43 BANKBOOTr1

38 HEXIOR1

18 OUT 1 (Mmc) r2

87 CPU68Kr4

62 FLO2r2

43 MOTOR3

19 IOE r0

34 KEY r0

4 PROMER r2

23 RAM 16 r0

Alle Leiterplatten pro Stück 9,— DM

Ideal für Bastler, nichts für Anfänger!

Betrifft Fehlermeldung EZEAT – „LOOP“ Nr. 6

Der dort gemeldete Fehler von EZEAT (Modemzugriff mit Floppykarte funktioniert nicht) läßt sich durch einen einfachen Trick umgehen: warten Sie nach dem Aufruf des ZEAT-Menüs etwa 10 Sekunden, bevor Sie eine Eingabe tätigen. Dann läuft EZEAT fehlerfrei.

RAM-Floppy mit 420K

Erweiterung der RAMFLOPPY beim NDR-Computer (Z80-System) auf 420K, von Franz Ehrensperger, Filiale München.

Viele Kunden würden gerne mit einer größeren RFLO beim NDR-Computer arbeiten. Hier ist nun eine relativ einfache Änderung, die ohne Kenntnisse des Assemblers gemacht werden kann. Folgende Adressen im BIOS sind dazu zu ändern:

Adresse:	alt	neu
EA7F	DCEFC5EF	E0EFC5EF
EABB	0F00	0F00
EABD	03	04
EABE	07	0F
EABF	00	01
EAC0	B300	D100
EAC2	3F00	7F00
EAC4	C0	C0
EAC5	00	00
EAC6	1000	2000
EAC8	0000	0000
EC42	E603	E607

Wie man anhand dieser Liste ersehen kann, ist die Erweiterung ganz einfach, doch wie bekommt man sie dauerhaft auf die Bootspur? Dazu geht man nun wie folgt vor:

Zuerst SYSGEN80 aufrufen, für SOURCE-DRIVE A eingeben und RETURN drücken, für DESTINATION-DRIVE RETURN drücken. Es meldet sich nun wieder A), aber das Betriebssystem steht im Arbeitsspeicher. Um es zu ändern, schreiben wir es mit SAVE 35 CPM.SYS als File auf die Diskette. Nun holen wir es, nachdem sich A) gemeldet hat, mit DDT CPM.SYS wieder in den Arbeitsspeicher und nun kann das BIOS mit dem S-Befehl geändert werden. Jetzt stehen die Adressen aber nicht mehr an ihrem ursprünglichen Platz, deshalb hier die Liste der zu ändernden Adressen.

BIOS-ADR	TPA-ADR
EA7F	1FFF
EABB	203B
EABD	203D
EABE	203F
EABF	203F
DAC0	2040
EAC2	2042
EAC4	2044
EAC5	2045
EAC6	2046
EAC8	2048
EC42	21C2

Auf Adresse EACB steht im BIOS die Kaltstart-Meldung, wird diese geändert, so meldet sich der Computer nach einem Kaltstart mit einer anderen Meldung. Hier die Daten für folgenden Schriftzug „NDR-COMPUTER + 420KRFL0“ in HEX. E4 44 52 2D 43 4F 4D 4D 50 55 54 45 52 20 2B 20 34 32 30 4B 20 52 46 4C 4F 20. Diese müssen in der genannten Reihenfolge ab Adresse 204B eingegeben werden. Nachdem alle Bytes geändert sind, starten wir mit G0 oder CTRL C das CP/M mit dem Warmboot, es muß sich nun wieder mit A) melden. Um unsere Änderung vom ursprünglichen BIOS zu unterscheiden, speichern wir es unter einem anderen Namen erst einmal auf die Diskette, z.B. FRLO420K.SYS. Wir müssen also SAVE 35 RFL0240K.SYS eingeben. Nun müssen wir unser neues BIOS noch in die BOOTSPUR bringen. Dazu laden wir es mit dem DDT wieder in den Speicher, also DDT RFL0420K.SYS eingeben, wenn sich der DDT gemeldet hat, mit G0 oder CTRL C einen Warmstart herbeiführen, nun SYSGEN80 aufrufen, für SOURCE-DRIVE RETURN eingeben und für DESTINATION-DRIVE A und RETURN eingeben. Nach dem Initialisieren der RAM-FLOPPY müssen nun mit STATE: 416k zur Verfügung stehen, wenn 8 Banks belegt sind, also 8 * R0A64 oder 2 * RAM64/256 mit 256K bestückt.

Nachtrag

zur „Mehr Speed für CP/M“ aus LOOP 6

Die Stepraten-Erhöhung ist natürlich auch unter dem FLOMON für EZEAT möglich. Hier muß nicht die Adresse 9C58 sondern die Adresse 9CB8 geändert werden!

Zeichenpuffer für PREH-Tastatur

Es besteht die Möglichkeit, die PREH-Tastatur mit einem Puffer von 60-Zeichen zu betreiben.

Dabei muß das READY-Signal der Tastatur nicht auf GND gelegt werden (Zusatzblatt zur Tastatur: Brücke von Pin 12 nach 2), sondern direkt auf das IC 74LS74 Pin 9 der KEY-Baugruppe.

Das Kabel zur Key-Baugruppe muß deswegen 11 statt sonst 10 Leitungen führen.

mc-Plotter am NDR-Computer

Sehr geehrte LOOP-Redaktion,

wer mit seinem NDR-Klein-Computer einen preiswerten, selbst gebauten Plotter ohne selbst geschriebene Software betreiben will, kann sich den Selbstbauplotter aus der Zeitschrift MC 8/1983, Seite 64 – 67 bauen – oder zumindest diesen Beitrag als Inspiration für eigene Hardware-Entwicklungen verwenden.

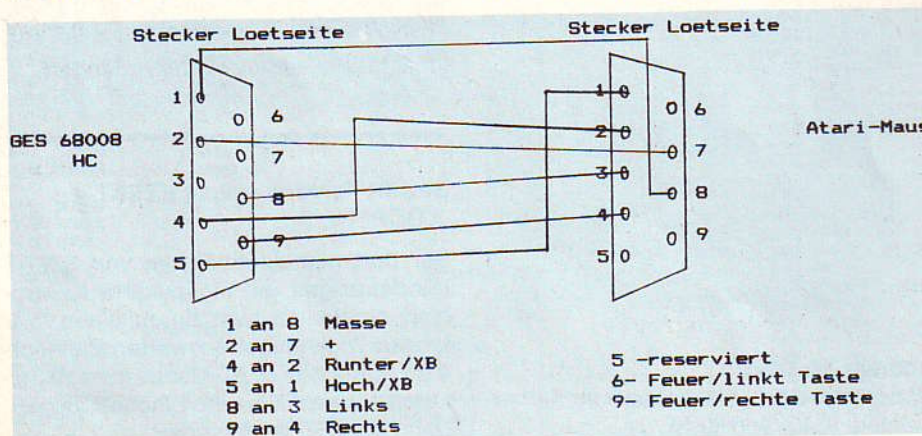
Die beiden Schrittmotoren und der Hubmagnet zur Schreibstiftsteuerung des Plotters werden von einem Z80 EMUF gemäß MC 4/1983, Seite 112 – 114 und MC 2/1984, Seite 88 – 92, gesteuert. Die für den EMUF notwendige Software gibt es z.B. beim Franzis-Softwareservice auf einem Eprom. Ein EMUF ist ein Einplatinen-Microcomputer für universelle Festprogrammmanwendung. Er ist als Bausatz oder Fertigerät zu haben.

Der EMUF wird über den Centronics-Port des NDR-Klein-Computers mit speziellen Befehlen angesteuert. Dazu muß am EMUF das READY-Signal (ARDY von Pin 3) mit Hilfe eines Transistors invertiert werden, bevor es zum NDR-Klein-Computer gelangt. Das Kabel zwischen EMUF und NDR-Klein-Computer enthält elf Leitungen: Bit null bis Bit 7, Busy, Strobe, Masse. Ob alle Signale stimmen, läßt sich vortrefflich mit dem Prüfstift von Rolf-Dietter Klein testen.

Atari-Maus an HCOPY

von Bernd Schumacher,
Hermann-Hesse-Weg 11,
72255 Rutesheim

Die ATARI-Maus (Nr. STM1 1585118042 FCC ID: EBA76FSTM1) arbeitet mit meinem 68008 und der H/C-Gruppe bei folgender Anschlußbelegung:



Zwischen EMUF und Plotter ist ein Treiber (= Verstärker) zu schalten, um die beiden Schrittmotoren und den Hubmagneten zu steuern. In MC 8/1983 sind zwei brauchbare Schaltungen zu finden, in denen jedoch zu jeder Motorwicklung und zum Hubmagneten stets eine Löschiode (z.B. 1N 4006) parallel geschaltet werden sollte, um die Transistoren (BC 337) vor der Induktionsspannung aus den Wicklungen zu schützen (Kathode an plus 24 V, Anode an den Collector des jeweiligen Transistors).

In zwingender Abwandlung zu den angeführten MC-Beiträgen sind Pin 28, Pin 26 und Pin 1 des 31poligen EMUF-Steckers nicht zu verwenden, da Komplikationen bei der Verwendung eines käuflichen EMUF-Busses vermieden werden sollen.

Wer alles selber baut, braucht dies nicht zu beachten.

Wenn alle Verbindungen sorgfältig (!) überprüft sind, kann man in den Editor des NDR-Klein-Computers die Zeichenbefehle eintippen. Ein Beispiel: L,4000,300 läßt den Plotter eine Linie vom gegenwärtigen Standpunkt (Null/Null) zur Position $x = 4000/y = 3000$ ziehen. Nach Verlassen des Editors wird mit dem Optionenmenue die Druckerausgabe ohne Line Feed ausgewählt und anschließend mit dem Menue „Text drucken“ der Inhalt des Editors ohne LF über den Centronics-Port an den EMUF gegeben. Dieser reagiert und steuert dann über die Treiber die Schrittmotoren und den Schreibstift.

Wer will, kann natürlich auch mit dem NDR-Klein-Computer Zeichenprogramme schreiben, anstatt einzelne Zeichenbefehle zur Plotterbedienung zu verwenden.

Alle wesentlichen Informationen, einschließlich der Bezugsquellennachweise, sind den angeführten Beiträgen aus der Zeitschrift MC (Franzis-Verlag) zu entnehmen. MC-Hefte werden übrigens von vielen Bibliotheken, insbesondere von naturwissenschaftlichen Universitätsbibliotheken, gesammelt und können dort fotokopiert werden. Außerdem ist vom Franzis-Verlag ein EMUF-Sonderheft erhältlich.

Ich hoffe, meine Anregung ist einigen NDR-Klein-Computer Anwendern dienlich.



Leser fragen – Fachleute antworten

Stellen Sie auch Ihre Fragen an „loop“

sammenstellung der Programmänderungen und auch der Hardwareänderungen.

Besonders für solche Hinweise, wie die Ihren, sind wir immer sehr dankbar, da doch unwillkürlich etwas vorausgesetzt wird, was nicht sein muß.

Nochmals herzlichen Dank für Ihr Schreiben.

Sehr geehrte Herren,

um mehrere Sprachen und Funktionen des NDR-Klein-Rechners nebeneinander benutzen zu können, habe ich mir den EFLOMON so umschreiben lassen, daß ich nach dem Einschalten vom Menü her eine ROA-Karte frei wählen kann. Leider gibt es dabei einige Sekundärschwierigkeiten. Der Drucker RX80 erhält nach dem Ausdruck von SPS Kontaktplan in USASCII und Schmalschrift nicht den Befehl zur Normalisierung, wenn ich hinterher einen deutschen Text von einer anderen ROA-Karte gesteuert ausdrucken lassen will, muß ich den Drucker kurz ausschalten. Bei den üblichen IOE-Karten starten die SPS- und Robotprogramme irgendwo mittendrin, wenn man den Rechner einschaltet. Die IC 74LS374 müssen durch 74LS273 ersetzt werden und die e0 sowie e1 müssen statt mit Masse mit Reset verbunden werden. Ich suche das Programm für die Ausführung einer Hardcopy wenn die Hardware mit Z80 vorhanden ist. Wie sieht es mit Programmieranleitung für DA und AD-Karte aus? Bei der Suche nach der Ursache für einen unkonstanten Fehler haben wir 74LS374 in Verdacht, ist darüber etwas bekannt, daß eine Serie dieses Bausteins eine Macke hat?

Günter Lange
Rübker Bruch 11, 2153 Neu Wulmstorf

Antwort LOOP:

Sehr geehrter Herr Lange,
herzlichen Dank für Ihre Anmerkungen. Wir sind dabei, das Hardcopy-Programm

Sehr geehrte Redaktion!

In die allgemeine Lobhudelei möchte ich nicht einfallen, ein so anspruchsvoller Computer braucht eine eigene Zeitung und das ist LOOP. Sicherlich gibt es nichts, was man nicht noch verbessern kann, darum möchte ich nicht versäumen auch meinen Senf dazuzugeben. Was ich gerne hätte:

Eine Zusammenstellung aller benützten I/O-Adressen, auch der nicht erlaubten, weil nicht durchdecodierten.

Eine Zusammenstellung der Programmänderungen Version (X) nach (Y) und wo liegen die Unterschiede?

Programm-Starts unter CP/M ab Adresse 0100 und nicht ab 0000 wie in LOOP 3 ser Ein-/Ausgabeprogramm. Ich bin doch kein Profi und falle auf so etwas herein.

Oft genügen kleine Hinweise, uns Nicht-Profis das Leben zu erleichtern, z.B. LOOP 3, Modem Programm. Auch Leute die schon etwas weiter ausgebaut haben (d.h., Bank-Boot-Karte mit Grundprogramm auf 2000) bekommen das Programm sofort in Gang, wenn man ihnen sagt, schreibt alle Call-Adressen welche ins Grundprogramm gehen von 00xx in 20xx um, denn euer Grundprogramm liegt ja bei 2000.

Zu „Mehr Speed“ in LOOP 6: Wer ältere 8“-Laufwerke benutzt, sollte beim EFLOMON-Programm bleiben. Die Steprate ist zu schnell.

Nun vielleicht auch ein Tip für Leute, die wie ich schon CP/M im Rechner haben, und trotzdem gerne mit dem Grundprogramm arbeiten. Oder Programme aus der LOOP abtippen (z.B.) das Modem-Programm). Aber wie dann abspeichern?

1. Programm erstellen im ram (8800) Länge merken 8000-8xxx
2. Mit Reset zum Menue
3. Mit 1 Floppy-Boot
4. A)DDT
5. m8000,8xxx,0100 (Verschiebe-Befehl)
6. Return dann ^C
7. A)Save n Datei.com
n = Menge 0100...xxxx Datei = Name
z.B. ... Modem
8. Return 0100...0400 n = 4

Zurückholen

1. 1 Floppy-Boot
2. A)DDT Datei.com
3. Return
4. m0100,xxxx,8800
5. Return
7. 3 Grundprogramm 2000
8. Start Ram

Willi Wegemann,
Bredenscheider Str. 28, 4320 Hattingen

Antwort LOOP:

Sehr geehrter Herr Wegemann,
herzlichen Dank für Ihre Anregung. Die Liste der IOE's werden wir in der nächsten LOOP veröffentlichen. Ebenso die Zu-

für den Z80 in ein erweitertes Grundprogramm mit aufzunehmen. Ein normales Z80-Programm ist im Handbuch der Hardcopy-Baugruppe ausgedruckt. Ebenso im Handbuch wird, sobald verfügbar, eine Programmieranleitung für die DA- und AD-Wandler Karten erscheinen. Die Baugruppen sind beim 68000-System bereits im Grundprogramm vorgesehen.

Bei Ihrer Fehlersuche können wir Ihnen speziell bei LS374 nur Bausteine der Firma Texas-Instruments empfehlen. Weiterhin viel Erfolg mit dem NDR-Computer.

Positive Kritik zu Loop:

1. Es ist von Vorteil, wenn es eine eigene Computer-Zeitschrift gibt! Aber für die Einsteiger (Hexmon) ist die „LOOP“ schon zu weit voraus! Es fehlen Anwendungs-Beispiele, -Programme, überhaupt ist die Software und die Programmierung sehr schlecht beschrieben. Bitte denken Sie in Zukunft etwas mehr an die Einsteiger!

2. Weiter fehlt die Information für den Einsteiger und auch Fortgeschrittenen, wie der NDR-Computer ausgebaut werden kann. Ist es sinnvoll den Z80 weiter auf CP/M mit Floppy auszubauen, wenn es schon die echte 16 Bit (68000) und sogar die 32 Bit (68020) Version gibt? Wäre es nicht angebracht eine Gegenüberstellung der verschiedenen Systeme zu machen. Zum Beispiel:

- CPU Z80 / 68000 / 68008 / 68020
- Gegenüberstellung, vor und Nachteile
- Anwendungsmöglichkeiten
- Besondere Eigenschaften
- Betriebssysteme, welche Unterschiede oder Einschränkungen, was gibt es für Systeme (Sinn und Zweck)
- Welche Sprachen, für welches System, für welche Anwendung
- Wie sieht es mit der Software aus?
- Was gibt es an Software, und wo?
- Wie ist der Trend?

Für mich stellen sich diese Fragen, und mit Sicherheit für *viele* andere auch!

Ich interessiere mich für:

- Text.- Verwaltung und Verarbeitung

- über Modem auf Datenbanken zuzugreifen
- zum Messen, Steuern und Regeln
- Rechenprogramme / Sprachkurse
- Unterhaltung (Spiele)

Welcher Ausbau ist für mich geeignet? Im Sinne einer positiven Kritik erbitte ich *dringend* Rat!

Martin Mayländer
Saarwerder Str. 4, 6600 Saarbrücken 6

Antwort LOOP:

Sehr geehrter Herr Mayländer,

wir glauben, daß wir schon in „LOOP“ 6 sehr viel für die Einsteiger getan haben. Das werden wir natürlich weiter tun. Zum Ausbau des NDR-Computers entsteht derzeit ein neuer, umfangreicher Katalog. Hier wird auch die Gegenüberstellung, die Sie erwähnt haben, gegeben.

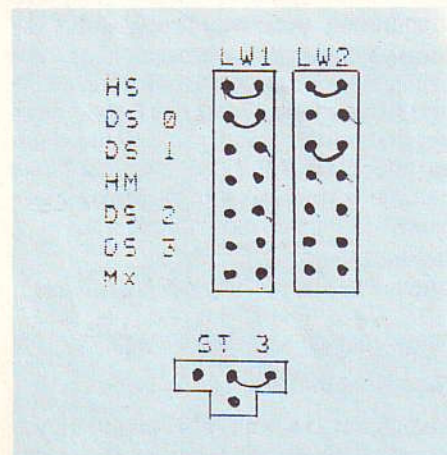
Für den von Ihnen gewählten Interessensbereich würden wir den Ausbau mit Z80 und dem CP/M-Betriebssystem bzw. EZEAT vorschlagen. Bitte beachten Sie auch das Angebot der Firma Com Food – hier in dieser „LOOP“ erwähnt.

Aus der Technik

Lieber Herr Graf,

unser Weihnachtsgeschenk, 2 Floppy-Laufwerke, TEAC 5,25 Zoll, hat uns ein wenig Kopfzerbrechen bereitet. Zu den Laufwerken Typ FD-55 F-03-U wurde uns ein Begleitheft für den Typ FD-55 FV-03 geliefert.

Die Beschreibung der Laufwerk-Einstellung hat uns im Stich gelassen. Statt einer 2 x 4-poligen Stiftleiste enthält das Laufwerk eine 2 x 7-polige. Das Laufwerk läßt also eine größere Selektion zu. Neben Drive Select (DS 0-3) muß an jedem Laufwerk zusätzlich ein Stecker eine Brücke bei HS (Head Select) aufbauen. Beim Betrieb mit Jogi-DOS, Version 2,0, muß die Brücke auf dem Motoron-Stecker (ST 3) auf der FLO 2-Karte von der Mitte nach rechts (nicht von der Mitte nach unten) liegen.



Vielleicht können diese Zeilen als Hinweis für alle dienen, die vor dem gleichen Problem stehen.

Nachdem noch eine kalte Lötstelle auf der FLO 2-Karte beseitigt wurde, läuft alles wunderbar.

Klaus Thiel
Watzmannstr. 1, 8011 Poing

Hinweis: Wir haben festgestellt, daß es unter der gleichen Typenbezeichnung zwei unterschiedliche Laufwerke von Teac gibt, die völlig andere Brücken haben! In Zukunft packen wir den Laufwerken ein Blatt bei, das alle Versionen berücksichtigt.

Laufwerke

Diskettenlaufwerke sind gegen Störungen relativ empfindlich. Sie sollten deshalb in ein Metallgehäuse eingebaut werden.

Wenn die Laufwerke frei stehen, kann schon ein in der Nähe stehender Monitor oder ein Schaltnetzteil zu Störungen führen.

Schließen Sie in jedem Fall auch die Erdungsleitungen an. Bei gemischtem Betrieb mit 8 Zoll und 5 1/4 Zoll Laufwerken ist ein störungsfreier Betrieb ohne angeschlossene Erdungsleitungen kaum möglich.

Achten Sie auf das Abschlußwiderstandsnetz auf den Laufwerken. Der Netzwerkwiderstand soll nur auf dem Laufwerk stecken, an dem das Floppy-

signalkabel endet. Bei gemischtem Betrieb mit 8 Zoll – 5 1/4 Zoll bleibt nur der Netzwerkwiderstand auf dem letzten 8 Zoll-Laufwerk (5 1/4 Zoll ohne R)

Kritische Bausteine auf FLO2 / FLOSASI:
J8 7404 (kein 74LS04)
J10 74LS245

Neues aus den Filialen

NDR-Benutzer treffen sich in Berlin!

Warum soll jeder für sich das Rad allein entdecken . . .?! Das war der Grundgedanke von 4 NDR-Computerbauern in Berlin. Man hatte sich, jeder für sich allein, durch die kleinen Tücken und Probleme des NDR-Computers durchgebissen; bei der Suche nach Programmen und Problemlösungen traf man sich zufällig im Laden in Berlin und tauschte Telefonnummern und Gedanken aus. Um auch andere daran teilnehmen zu lassen, wurde die Idee des „Workshop“ geboren. Da es sich anbot, die Anlaufadresse zu wählen, welche die meisten NDR-Computerbauer kennen, der Laden in Berlin, wurde nur noch kurz über Zeit und ersten Termin diskutiert. Und so war es dann soweit: Der erste Donnerstagabend übertraf die Erwartungen: 14 Interessenten waren erschienen! Das erste „Beschnuppern“ war sehr aufschlußreich: Stellte doch fast jeder fest, daß keiner so richtig perfekt ist und das andere auch Probleme haben. Nach 14 Tagen wuchs die Zahl der Interessenten am

Workshop in Berlin schon auf stolze 18 Leute und am 5. Dezember waren es 22 Personen. Die Informationen strömten schon recht kräftig und auch sehr gute Lösungen der doch etwas „kränkelnden“ Analogschaltungen wie CAS und Analog der GDP 64 wurden als Verbesserungen angeboten. (Die Schaltungen werden in der nächsten Ausgabe unter dieser Rubrik veröffentlicht!) Die Zeit des Workshops (alle 14 Tage um 18.30 Uhr) kommt fast allen Interessierten entgegen und die geplante Zeit von 2 – 2 1/2 Stunden konnte bisher aufgrund der regen Beteiligung noch nie eingehalten werden. (Den letzten Bus haben alle noch bekommen!) Es ist außerdem geplant, ab Januar außerhalb des Workshops, bei bestehendem Interesse, Kurse über Einführung in Pascal und über Grundlagen der Elektronik anzubieten (ca. 10 Teilnehmer). Die Grundgebühr wird ca. 50 – 80 DM für 4 Abende zu je 2 Stunden betragen. Die Termine stehen noch nicht fest.

Kontakt: Jörg Korb Elektronik,
Budapester Straße 39 A, 1000 Berlin 30,
Telefon: (030) 262 74 74

KONTAKTE

Suche Raum Stuttgart:
NDR-Computer-Club oder Kontakt.
Telefon 07021/51495

Suche Erfahrungsaustausch mit ZEAT- und CP/M80-Benutzern.

Andreas Domke, Braunschweig,
Telefon 0531/340298

FLOH MARKT

Verkaufe:

Cherry-Tastatur im Zehnerblock, deutsch und 17 programm. Tasten, passend für NDR-Computer – nicht benutzt! DM 300,-

Andreas Domke, Braunschweig,
Telefon 0531/340298

NDR 68008: Verkäufe 24 K-BASIC-Interpreter, über 100 Befehle. Ab Grundprogramm 4.0 lauffähig. Auf Diskette oder EPROM. DM 149,- per NN.

R. Lobreyer,
Sonnenhalde 27, 7740 Triberg

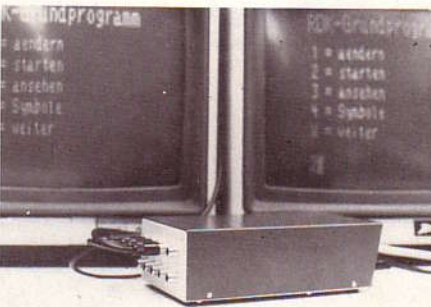
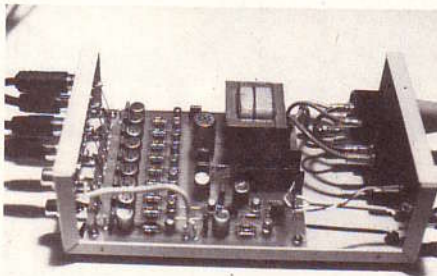
Aussteiger verkauft an Einsteiger NDR-Computer, Z80 Voll, mit Laufwerk, auch einzeln. VB, ab 18.00 Uhr,
Telefon 040/2101903

Monitorverstärker für Schulungs- und Ausstellungszwecke

GES stellt einen 10-fachen Monitorverstärker vor, der folgende technischen Daten hat:

Eingang: VIDEO-BAS-Signal, wie von der GDP geliefert.

Ausgang: 10 BAS-VIDEO-Ausgänge



zur Ansteuerung von 10 getrennten Monitoren geeignet: Das Gerät ist besonders für Schulen, Messen und Ausbildung geeignet. Natürlich können auch alle anderen Computer, die einen BAS-Ausgang besitzen (z. B. Commodore Triumph-Adler), an dieses Gerät angeschlossen werden. Es können auch weniger als 10 Monitore als Ausgang angeschlossen werden. Beliebig viele solcher Geräte lassen sich hintereinander schalten, somit können auch beliebig viele Monitore angesteuert werden.

Der Monitorverstärker kostet DM 729,60 als geprüftes Fertiggerät und ist ab Lager lieferbar.

GES Datenbank – mehr als 500 Benutzer pro Monat

Bald auch Download von Programmen möglich!

Mehr als 500 Anrufe pro Monat auf unserer Datenbank – darauf sind wir schon ein wenig stolz!

Was können Sie denn überhaupt mit unserer Datenbank? Sie können

- aktuelle Informationen abrufen
- Nachrichten an uns oder an alle hinterlassen
- Kontakte knüpfen

- Verkaufs- und Kaufangebote schreiben
- und (bald) Programme von uns in Ihren Rechner laden!

Das erste Programm wird „GOMOKU“ sein, wie in diesem Heft erwähnt. Weitere Programme folgen!

Natürlich können Sie auch Wünsche und Anregungen vorbringen und die LOOP-Redaktion direkt erreichen!

Sie brauchen nur einen Akustik-Koppler, die SER (NDR) oder OUT 1 (mc) und eine Transfer-Software, z.B. das ZEAT-Betriebssystem.

Rufen Sie mal an!

DATENBANK-TELEFON 0831/69330

Berghütte zu vermieten



Die in LOOP 5 vorgestellte Hütte ist noch kurzfristig frei! Die Hütte liegt mitten im Allgäu – von hier aus kann man Langlaufen, Skitouren machen oder alpin fahren. Mietpreis/Woche DM 560,- inkl. aller Nebenkosten.

Näheres bei Alfred Einsiedler, Telefon: 0831/67168



Impressum:

LOOP Zeitung für Computerbauer

Herausgeber: Gerd Graf

Redaktion: Rolf-Dieter Klein, Gerd Graf

Gestaltung und Druck:
Karl-Heinz Rieder, Kempten

Herstellung und Anzeigenverwaltung:
GES GmbH
Magnusstraße 13, 8960 Kempten

Anzeigenpreisliste 1/84

Theorie und Praxis rund um den NDR-Computer

Mikroelektronik Einführung



4 Kursteile (je ca. 70 Seiten im Format A4), DM 38,- je Kursteil

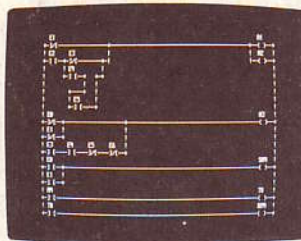
Der Kurs ist auf die HEXIO abgestimmt und ist für alle geeignet, die ihre ersten Schritte in Z 80-Maschinenprogrammierung machen.

Nach diesem Kurs sind Sie in der Lage, eigene Programme zu schreiben und die Arbeitsweise des Z 80 zu verstehen.

Der Kurs ist in verschiedene Fachgebiete aufgeteilt und bringt eine Menge Aufgaben, Beispielprogramme und Übungen.

Aus dem Inhalt: Was ist ein Mikroprozessor? * Inbetriebnahme des Computers * Planung von Programmen * Aufbau der CPU * Speicher und Adressen * Datentransfer * Lauflicht * Breakpoints * Hilfsfunktionen * Logo-Elemente * Strukturiertes Programmieren * Label & Call.

SPS-Programmierung



4 Kursteile (je ca. 70 Seiten im Format A4), DM 38,- je Kursteil

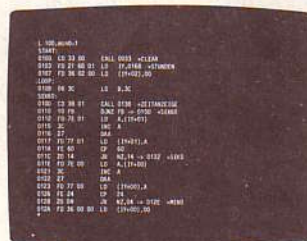
Dieser Kurs zeigt Ihnen, wie SPS programmiert wird, die Normung, die Anwendungsmöglichkeiten und die verschiedenen Darstellungsarten.

Sie lernen spielend leicht, Relais- und Schützensteuerungen in SPS-Programme umzusetzen.

Beispielprogramme, Aufgaben und Übungen geben Ihnen die praktischen Erfahrungen und zeigen, wie SPS professionell eingesetzt wird. Nutzen Sie Ihren NDR-Computer für diese moderne Technik voll aus.

Der Kurs ist in folgende Fachgebiete gegliedert: Steuerungstechnik * Digitaltechnik * Methoden zur Beschreibung von Steuerungsaufgaben * Programmierung * Übungen und Tafeln.

ZEAT-Betriebssystem



Das Betriebssystem beinhaltet in drei EPROMs: Z 80-2-Pass-Assembler, Disassembler, Editor, Debugger, Telefonmodem-Programm, FLOMON 1.5, ausserdem eine ausführliche Dokumentation zum Preis von DM 198,-.

Das Betriebssystem ZEAT benötigt 64-K-RAM (dynamische RAM-Karte). Die EPROMs werden in die BANKBOOT-Karte eingesteckt und sind sofort betriebsbereit. Programmieren Sie Ihren NDR-Computer mit einem Profi-Assembler.

Das Textverarbeitungsprogramm hat volle Bildschirmeditierung und kann neben der Programmierung auch zum Textschreiben eingesetzt werden.

Z 80-Assembler-Programmierung

4 Kursteile (je ca. 70 Seiten im Format A4), DM 38,- je Kursteil

Der Kurs ist auf das ZEAT-Betriebssystem abgestimmt und zeigt Ihnen in leicht verständlicher Art, wie der NDR-Computer in Z 80-Assembler programmiert wird, bringt reichhaltige Übungsbeispiele und Anwendungen. Sie werden erstaunt sein, wie leicht diese Art der Programmierung ist. Und Sie lernen, wie man die serielle Schnittstelle bedient und Daten über Telefon übertragen kann.

Die Fachgebiete dieses Lehrgangs sind: Systembeschreibung * Betriebssystem * Programmierung * Testen * Modemprogramm * Listings, Tafeln und Tabellen.

Christiani

Hier abtrennen und im Umschlag einsenden an: Dr.-Ing. P. Christiani GmbH, Techn. Lehrinstitut und Verlag, Postfach 35 69189, 7750 Konstanz

Bestellcoupon

	Preis je Teil	Gesamtpreis
<input type="checkbox"/> Einführung mit dem NDR-Computer (4 Kursteile)	DM 38,-	DM 152,-
<input type="checkbox"/> Z 80-Assembler-Programmierung (4 Kursteile)	DM 38,-	DM 152,-
<input type="checkbox"/> SPS-Programmierung mit dem NDR-Computer (4 Kursteile)	DM 38,-	DM 152,-
<input type="checkbox"/> Kompakt-Kurs BASIC (angepasst an das RDK-BASIC)	DM 198,-	DM 198,-
<input type="checkbox"/> ZEAT-Betriebssystem (3 EPROMs mit Dokumentation)	DM 198,-	DM 198,-

Name, Vorname

Straße

PLZ, Ort

Datum

Unterschrift

86189