

LOOP

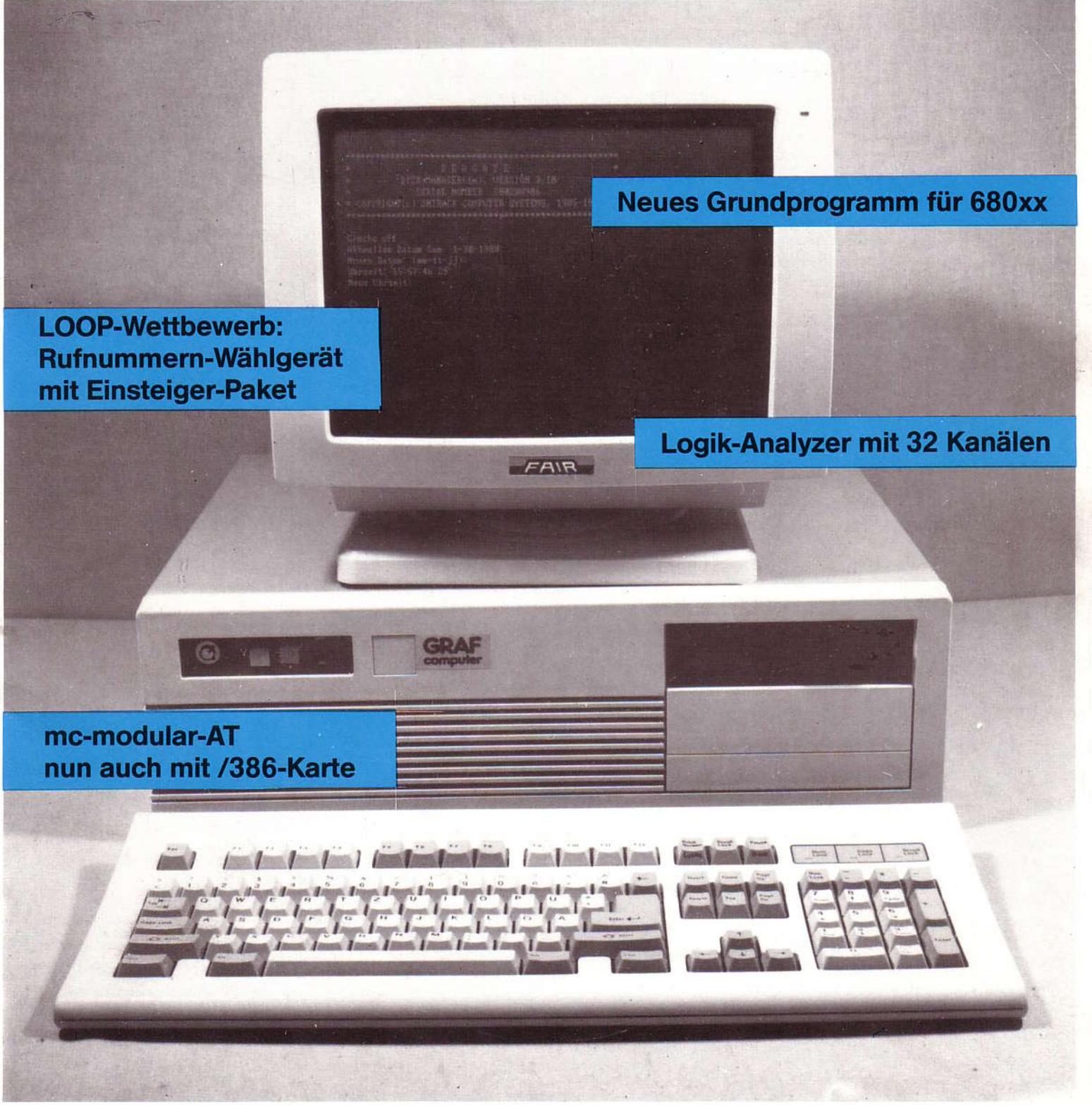
19. März 1988

17

3. JAHRGANG

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

DM 3,-



Neues Grundprogramm für 680xx

LOOP-Wettbewerb:
Rufnummern-Wählgerät
mit Einsteiger-Paket

Logik-Analyzer mit 32 Kanälen

mc-modular-AT
nun auch mit /386-Karte

LOOP: mal wieder zu spät!

Obwohl wir in *LOOP* 16 die Erscheinungstermine bekanntgegeben haben (Auslieferung dieser *LOOP* am 8. 2. 88), haben wir uns um vier Wochen verspätet.

Grund ist nicht der ausschweifende Fasching oder der starke Schneefall, sondern letztendlich der neue Katalog, der noch rechtzeitig zur Hannover Messe – CeBIT – fertig werden sollte. Unser neuer Katalog hat nun über 200 Seiten und etwa den doppelten bis dreifachen Textanteil. Die Arbeit daran, die wir auf etwa vier Wochen eingeschätzt hatten, hat sich auf vier Monate vergrößert. Nun, auch dieser Katalog ist bei der Druckerei (ab 30. 3. verfügbar) und die *LOOP* liegt vor Ihnen.

NDR-Computer und OS9

Besonders in den letzten Monaten mehrten sich die Anfragen, das Betriebssystem OS9 auf dem NDR-Computer zu installieren. Wir haben uns die Angelegenheit – auch gemeinsam mit unseren Software-Partnern – reiflich überlegt und sind leider zu dem Schluß gekommen, daß sich eine solche Installation für uns nicht lohnt.

Die Lizenz- und Abnahmebedingungen beim Betriebssystem OS9 sind so, daß wir aufgrund der hohen Stückzahl, die man zunächst abnehmen muß, gezwungen wären, einen Preis für OS9 zu verlangen, der von den NDR-Computer-Kunden sicher nicht toleriert werden würde (um ca. DM 1.000,-). Andererseits ist die Nachfrage bei OS9 bei weitem nicht so groß, als daß sich ein von uns gewünschter Preis von ca. DM 200,- errechnen würde.

Deshalb – und aus einigen anderen Gründen, wie Unterstützung, die wir gewähren müssen –, haben wir uns entschlossen, OS9 nicht zu installieren. Es wird also weiterhin bei den Betriebssystemen JOGIDOS (Einsteiger), JADOS (das „Standard“-Betriebssystem) und CP/M68K bleiben.

Neues Layout der LOOP

Sie haben diese *LOOP* sicher schon durchgeblättert und es gemerkt: wir haben die Einteilung neu aufgebaut. Grund dafür ist der mc-modular-AT, der von nun an in der *LOOP* unterstützt werden soll. Natürlich werden die Artikel ganz allgemeine PC- oder AT-Artikel sein und damit alle Anwender dieser Rechner zufriedenstellen. Wir glauben, daß wir damit dem Trend der Zeit gefolgt sind.

Wir freuen uns auf einen regen Briefwechsel, wobei wir auch gerne Ihre Vorschläge und Ihre Nöte zur weiteren Entwicklung des Systems hören wollen.

Danach kommt die Abteilung Z80. Hierunter haben wir alle Programme und alle Hardware, die unter dem Z80-Rechner laufen, zusammengestellt. Der nächste Punkt ist 680xx – hierunter finden Sie alle Produkte, die zu den 68000-Systemen passen.

Natürlich wird es öfter Beschreibungen geben, die für beide Systeme einsetzbar sind. Wir werden diese dann entsprechend markieren.

Der nächste globale Unterteilungspunkt ist der schon erwähnte mc-modular-AT. Unter Sonstiges finden Sie von Kontaktwünschen bis zur Kleinanzeige alles, was nirgends hineinpaßt.

Der NDR-Computer und der mc-modular-AT

Einige unserer Kunden haben uns angesprochen, ob wir denn nur noch mc-modular-AT bauen und den NDR-Computer „sterben“ lassen wollen.

Diese Frage können wir mit ruhigem Gewissen verneinen. Der NDR-Computer hat sich – nicht zuletzt durch den Einsatz bei der Deutschen Bundespost – einen so breiten Kundenkreis geschaffen, daß wir auch weiterhin den NDR-Computer in der bisherigen Form unterstützen werden. Als kleiner Beweis dienen die neuentwickelten Baurgruppen, wie die BUSTEST oder die schon lange sehnlichst erwartete GDP64HS mit Hardscroll.

Allerdings müssen wir uns – und eigentlich jeder in dieser Branche – Gedanken darüber machen, ob und wie der Schritt zum IBM-Computer erfolgen soll. Wir haben hierzu bereits einige Lösungen im Kopf, die wohl auch schon sehr bald zu entsprechenden Produkten führen werden.

Mit dem C64-Logik-Simulator LogSim haben wir erstmals ein völlig anderes Produkt geschaffen, das auf dem – ach so verschmähten – Commodore C64 läuft. Ein kleines Prospektblatt haben wir dieser *LOOP* beigefügt. Dieses Programm – von Rolf-Dieter Klein entwickelt – sucht seinesgleichen und ist besonders für die Erstausbildung der Digitalelektroniker hervorragend geeignet. Vielleicht haben Sie den einen oder anderen Bekannten mit einem C64, dem Sie diese Information weitergeben können.

Wir hoffen, viele unserer *LOOP*-Leser auf der Hannover Messe – CeBIT (Halle 6, Stand F21 – wie im letzten Jahr) zu treffen und freuen uns über angeregte Diskussionen. Das GES-Team und unsere Software-Partner Folkert Hallenga, Stefan Stiehl und Kei Thomsen stehen für interessante Gespräche bereit. Sagen Sie uns auch Ihre Meinung über die *LOOP*!

Bis zur CeBIT oder zur nächsten LOOP

Rolf-Dieter Klein

Gerd Graf

In eigener Sache	17/2
Titelgeschichte	17/3

Z80

Kompliment an unsere Software-Freaks	17/6
Testprogramm im Monitor des Einsteigerpakets	17/6
ROA256/1M auch für ECB-System.	17/8
TOOL-Programm fürs Einsteigerpaket	17/11
Logikanalyse: Was ist das?	17/12
Logikanalysator mit 32 Kanälen	17/13
Umleitung! – Zwei Programme im gleichen Speicher	17/15
dBasell schreibt groß	17/16
Tips und Tricks bei HEBAS.	17/16
TURBO-PASCAL und die RAM-Floppy	17/17
CP/M2.2: Was nun?	17/17

680xx

680xx-Grundprogramm jetzt unter JADOS	17/19
Sprachausgabe für den NDR-680xx-Computer	17/20
Der COL-EDIT ist da!	17/21
LOW COST Editor für JADOS	17/21
CP/M68K und CPU 68020	17/22

mc-modular-AT

16 oder 32 bit – wie hätten Sie's denn gern!	17/23
mc-modular-AT noch schöner	17/25
Der verrückte Bootsektor	17/25

Aus der Technik

Steckbrief	17/6
Wichtiger Hinweis für alle LOOP-Autoren	17/8
Die Situation Spitzt sich zu	17/11
Briefe	17/27
Die „neue“ GDP 64K kommt	17/28
Kontakte und Kleinanzeigen	17/29

Rubriken

Entprellen des STROBE-Signals der Tastatur	17/29
GDP 64K mit Windows	17/30
System-Reset schlägt zur Festplatte durch	17/31
Technischer Service an Computer-Netzteilen	17/31

1. Preis im LOOP-Wettbewerb vergeben

Rufnummernspeicher und Wählergerät

Einsteigerpaket als intelligenter Telefonzusatz

Der in der LOOP 15 ausgeschriebene Wettbewerb hatte große Anteilnahme bei unseren NDR-Programmierern gefunden. Gesucht wurden sinnvolle Anwendungen mit dem Einsteigerpaket. Hierbei konnten Benutzer ihre eigenen Programme und Zusatzschaltungen an uns einschicken.

Aus mehreren Einsendungen konnte die Jury nun die beste Entwicklung auswählen:

Der erste Preis ging an:

Eberhard Maldener
Leipziger Straße 2
6457 Maintal

der mit seinem Rufnummernspeicher große Begeisterung bei uns auslöste.

Der Rufnummernspeicher kann häufig benötigte Telefonnummern speichern. Auf Tastendruck kann die automatische Wahl (nur bei Nebenstellen-Anlagen erlaubt) erfolgen.

Die kleine Zusatzschaltung (auf dem Lochraster der IOE aufgebaut) besteht lediglich aus zwei Relais und mehreren Leuchtdioden, die das Umschalten und Wählen übernehmen. Die Telefonanlage ist durch ein 4-poliges Steckerkabel mit dem Computer verbunden.

Diese Version darf jedoch nicht am öffentlichen Telefonnetz betrieben werden, dafür fehlt die Genehmigung der Post. An die private Haussprechanlage kann sie aber bedenkenlos angeschlossen werden.

Um auch Ihnen einen Nachbau dieser Entwicklung zu ermöglichen, drucken wir hier das komplette Listing des Programmes neben dem Schaltplan, der Stückliste und einer kurzen Beschreibung.

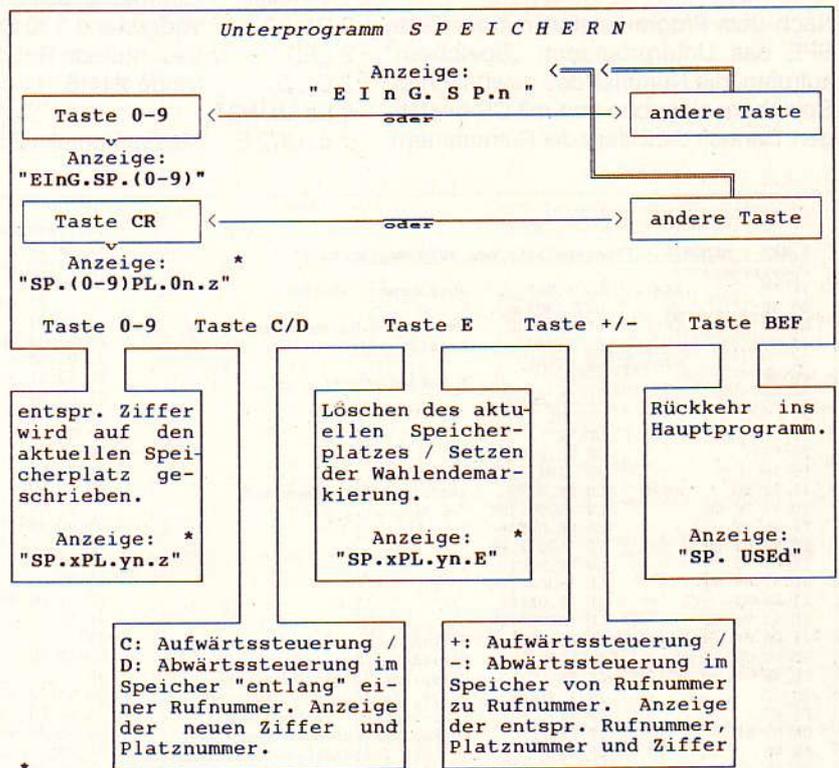
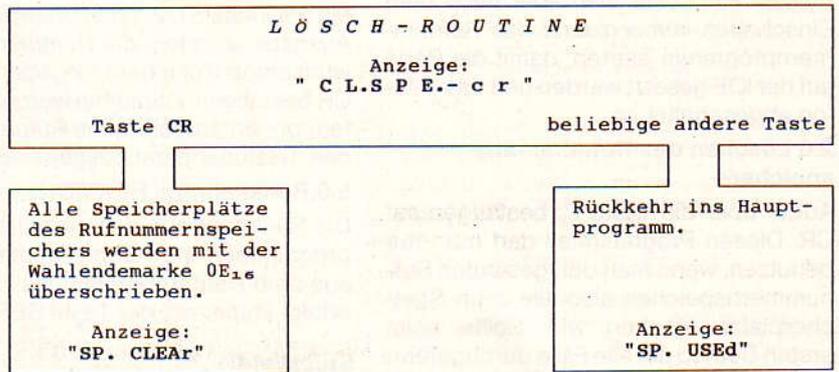
Inbetriebnahme

- IOE-Baugruppe nach dem vorliegenden Schaltplan verdrahten.
- Anschlußleitung des Telefons abstecken und an die IOE-Karte anschließen.
- Verbindung der IOE zum Telefonapparat herstellen.
- Programm auf einer beliebigen Startadresse eingeben (Akkupufferung).
- Programm starten.

Bedienung des Programmes

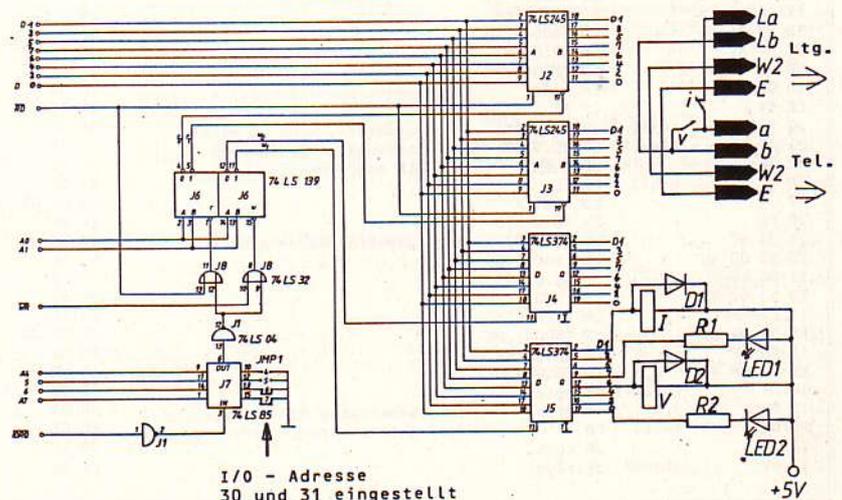
1.0 Programmstart

Wie bei anderen Programmen auch, ist die Startadresse immer die erste Adresse des Programmes.



* x: Rufnummer ; y: Platz innerhalb einer Rufnummer ; z: Ziffer

SCHALTBILD IOE



Hinweis: Der Computer kann natürlich alle anderen geladenen Programme bearbeiten, auch wenn er mit seiner IOE-Karte an ein Telefon angeschlossen ist. In diesem Fall sollte man aber nach dem Einschalten immer zuerst das Rufnummernprogramm starten, damit die Ports auf der IOE gesetzt werden und das Telefon abgeschaltet ist.

2.0 Löschen des Rufnummernspeichers

Aufruf über die Taste C, bestätigen mit CR. Diesen Programmteil darf man nur benutzen, wenn man den gesamten Rufnummernspeicher, also alle zehn Speicherplätze löschen will. Sollte beim ersten Betrieb auf alle Fälle durchgeführt werden.

3.0 Eingabe von Rufnummern

Nach dem Programmstart mit der Taste SPE das Unterprogramm „Speichern“ aufrufen, die Nummer des gewünschten Speichers eingeben und mit CR bestätigen. Danach die Ziffern der Rufnummern,

wie bei einem Testwahlblock, nacheinander eingeben. Die Rückkehr aus diesem Unterprogramm erfolgt mit der Taste BEF.

4.0 Ausgabe von Rufnummern (WAHL)

Mit der Taste START das Unterprogramm Ausgabe aufrufen, die Nummer des gewünschten Speichers eingeben und mit CR bestätigen. Daraufhin werden alle Ziffern der entsprechenden Rufnummer an den Telefonapparat ausgegeben.

5.0 Rückkehr ins HEXMON

Die Rückkehr aus den einzelnen Unterprogrammen ins Hauptprogramm, bzw. aus dem Hauptprogramm ins HEXMON, erfolgt immer mit der Taste BEF.

Stückliste:

1 I-Relais	Günther Nr. 3573 1231 051
1 V-Relais	Günther Nr. 3571 1210 051
2 R1, R2	Widerstand 330 Ohm
2 LED	Leuchtdiode Rot, 2 mm
2 D1, D;	Diode 1N418
8 La,Lb,W2,E	Messerkontakte 2,5x11 mm

Anmerkung der Jury:

Diese Entwicklung hat uns aus verschiedenen Gründen besonders gut gefallen:

- Praxisgerecht. Hier sieht man, daß mit dem Einsteigerpaket vernünftige Lösungen möglich sind.

- Gut dokumentiert. Das A und O eines Programmes.

- Programm sauber dokumentiert.

- Das Programm verwendet nur relative Sprünge, daher kann es unabhängig von der Speicheradresse (Start) eingegeben werden und „verträgt“ sich mit anderen Programmen.

- Am Programm kann man „lernen“, wie man programmiert.

Wir gratulieren Herrn Maldener zu seiner Leistung.

Die anderen Gewinner werden direkt benachrichtigt. Die Beiträge werden wir in loser Folge veröffentlichen.

5_HEX / MEMONIC - Programmliste des Rufnummerngebers

Hauptprogramm:

```

3E FF      start  LD A,OFF      Ausgabeport löschen.
D3 30      OUT (30),A
21 00 9F   test  LD HL,9F00      testen, ob Rufnummernspeicher
1E A0      LD E,0A0      benutzt oder leer.
7E        schl1 LD A,(HL)
FE 0E      CP OE        ( Wahlendemarke : OE16 )
20 06      JR NZ,used
23        INC HL
1D        DEC E
20 F7      JR NZ,schl1
18 56      JR clear
11 92 8C   used  LD DE,8C92      Anzeigepuffer laden mit:
ED 53 00 80 LD (8000),DE  " S P E . U S E d "
11 06 FF   LD DE,OFF06
ED 53 02 80 LD (8002),DE
11 C1 92   LD DE,92C1
ED 53 04 80 LD (8004),DE
11 86 A1   LD DE,0A186
ED 53 06 80 LD (8006),DE
21 00 9F   hptast LD HL,9F00      Startadr. Nr.-Speicher laden.
CD 0C 00   CALL OC      Einlese-u. Anzeigeschleife.
FE 5B      CP 5B      Einsprung in die Programm-
28 57      JR Z,jrspe  teile "Speichern", "Ausgabe",
FE 47      CP 47      "Nr.-Speicher löschen", bzw.
CA D1 OE   JP Z,BEF      Rücksprung ins Betriebssystem
FE 4B      CP 4B      je nach Tastendruck:(SPE,START,
28 54      JR Z,jraus0 C oder BEF)
FE 07      CP 7
20 E9      JR NZ,hptast
11 C6 47   LD DE,47C6      Anzeigepuffer laden mit:
ED 53 00 80 LD (8000),DE  " C L . S P E . - c r "
11 92 8C   LD DE,8C92
ED 53 02 80 LD (8002),DE
11 06 BF   LD DE,0BF06
ED 53 04 80 LD (8004),DE
11 A7 AF   LD DE,0AFA7
ED 53 06 80 LD (8006),DE
CD 0C 00   CALL OC      Einlese-u. Anzeigeschleife.
FE 57      CP 57
20 9A      test1 JR NZ,test      Programmstart, wenn CR nicht,
21 A0 9F   clear LD HL,9FA0      Rufnummernspeicher löschen,
3E 0E      LD A,OE      wenn CR gedrückt.
2D        schl2 DEC L
77        LD (HL),A
20 FC      JR NZ,schl2
11 92 8C   LD DE,8C92      Anzeigepuffer laden mit:
ED 53 00 80 LD (8000),DE  " S P E . C L E A r "
11 06 C6   LD DE,0C606
ED 53 02 80 LD (8002),DE
11 C7 86   LD DE,86C7
ED 53 04 80 LD (8004),DE
11 88 AF   LD DE,0AF88
ED 53 06 80 LD (8006),DE
18 9F      hptast1 JR hptast
18 06      jrspe JR upspe      Zwischensprünge zur Erweiterung
F6 01      test2 OR I      der Relativsprung- Distanz.
18 D1      JR test1
18 6D      jraus0 JRjraus1

```

Unterprogramm Speichern

```

11 86 CF   upspe LD DE,0CF86      Anzeigepuffer laden mit:
ED 53 00 80 LD (8000),DE  " E I n G . S P . n "
11 AB 42   LD DE,42AB
ED 53 02 80 LD (8002),DE
11 FF 92   LD DE,92FF
ED 53 04 80 LD (8004),DE
11 0C AB   LD DE,0AB0C
ED 53 06 80 LD (8006),DE
CD 0C 00   espnr CALL OC      Einlese-u. Anzeigeschleife.
CD 0F 00   cod    CALL OF      Hex- Code bilden.
FE 0A      CP OA      Begrenzen auf die Tasten
30 F6      JR NC,espnr 0 ... 9.
4F        LD C;A      entspr. Ziffer speichern,
CALL 12   umcodieren und
LD (8007),A in Anzeigepuffer laden.
CD 0C 00   CALL OC      Einlese-u. Anzeigeschleife.
FE 57      CP 57      Weiter im Programm bei Taste
20 CC      JR NZ,upspe CR, sonst " E I N G . S P . n "
11 92 0C   LD DE,0C92      Anzeigepuffer laden mit:
ED 53 00 80 LD (8000),DE  " S P . x P L . y . n . z "
16 8C      LD D,8C
79        LD A,C      ( x = S P(eicher)-Nummer )
CD 12 00   CALL 12      ((aus C-Register umcodiert )
5F        LD E,A      ( ins E- Register laden)
ED 53 02 80 LD (8002),DE  (
11 47 C0   LD DE,0C047      ( y = P L(atz)-Nummer )
ED 53 04 80 LD (8004),DE  ((in den Anzeigepuffer laden)
3E 2B      LD A,2B      (z = Ziffer, die auf Platz y
32 06 80   LD (8006),A      (des Speichers x steht (in
1E 0F      LD E,OF      ( den Anzeigepuffer laden)
18 6B      JR spanz
1E 0F      LD E,OF
18 45      JR ladanz
18 15      JR taste
7B        anz  LD A,E      Anzeigeroutine für die
2F        CPL      Platz- Nummer.
B7        OR A
CD 12 00   CALL 12
32 05 80   LD (8005),A
18 0A      JR taste
18 97      hptast2 JR upspe      Zwischensprünge zur Erweiterung
18 8D      jrspe1 JR jrspe  der Relativsprung- Distanz.
18 8D      test3 JR test
18 3C      jraus1 JR jraus2
18 EB      jr anz
CD 0C 00   taste CALL OC      Einlese-u. Anzeigeschleife.
FE 5D      CP 5D      Einsprung in die Programmteile:
28 3B      JR Z, +      - Speichernummer erhöhen
FE 5E      CP 5E
28 3F      JR Z, -      - Speichernummer erniedrigen
FE 47      CP 47
28 EB      JR Z,test3 - Rücksprung ins Hauptprogramm
CD 0F 00   CALL OF
CP 0C      - Platznummer erniedrigen
JR Z,ab   - Platznummer erhöhen
CP 0D
CP 0D      - Platznummer erhöhen
JR Z,jrauf
CP OE      - Wahlendemarke setzen / Ziffer
JR Z,jrlösch löschen.
CP OA      je nach Taste (+,-,BEF,C,D,E).

```

30 DE		JR NC.taste	Begrenzen auf Ziff. 0 ... 9,	FE 47	test?	CP 47	Rücksprung ins Hauptprogramm,
77	lad	LD (HL),A	sonst Ziffer in die bei spanz	28 B1		JR Z.test4	wenn Taste BEF betätigt wird,
CD 12 00		CALL 12	gebildete Adresse laden und	CD OF 00		CALL OF	Taste umcodieren und
32 07 80		LD (8007),A	in den Anzeigepuffer ablegen.	FE OA		CP OA	Begrenzen auf die Tasten
7B		LD A,E	Platznummer aus E- Reg. holen,	30 F2		JR NC,asprn	0 ... 9.
2F		CPL	umrechnen,	4F		LD C,A	Speichernummer ins C- Register
CD 12 00		CALL 12	codieren	CD 12 00		CALL 12	laden,umcodieren und in den
32 05 80	ladanz	LD (8005),A	und in den Anzeigepuffer laden.	32 07 80		LD (8007),A	Anzeigepuffer laden.
2C		INC L	Nächste Adresse bilden,	CD OC 00		CALL OC	Einlese- u. Anzeigeschleife.
1D		DEC E	Platz- Zähler erniedrigen, Ein-	FE 57		CP 57	Auf Bestätigung durch die
20 CB		JR NZ.taste	sprung in Einlese- u. Anzeige-	20 E7		JR NZ,test?	Taste CR warten.
B7		OR A	Schleife.	CD 33 00		CALL 33	Anzeigepuffer löschen.
3E OE		LD A,OE	Wenn letzter Platz erreicht,	11 92 OC		LD DE,OC92	Anzeigepuffer laden mit:
18 02		JR weiter1		ED 53 00 80		LD (8000),DE	"S P. x
18 42		JR jraus3	(Zwischensprung)	79		LD A,C	(x = Speichernummer, die vor
77	weiter1	LD (HL),A	Endmarke auf diesen Platz laden,	CD 12 00		CALL 12	Bestätigung mit CR aktuell war)
21 00 9F		LD HL,9F00	Startadr. Nr.-Speicher laden und	32 03 80		LD (8003),A	
18 B8	test'	JR test3	Rücksprung ins Hauptprogramm.	18 04		JR weiter2	
79		LD A,C	Die Programmteile + und -	18 87	test5	JR test4	Zwischensprünge zur Erweiterung
FE 09		CP 09	ermöglichen das Auf- und Abwärts-	18 B1	braus4	JR aus	der Relativsprung- Distanz.
28 B9		JR Z.taste	wandern durch die Speichernummern	3E 3F	weiter2	LD A,3F	V - Relais setzen. (Apparat wird
OC		INC C	im Bereich von Nr. 0 ... 9.	D3 30		OUT (30),A	während der Wahl überbrückt)
18 06		JR spanz		1E FF		LD E,OFF	Zeitschleife (ca. 500 ms) mit
79		LD A,C		CD 09 00	sch13	CALL 9	Anzeige.(Pause vor der 1. Ziffer)
FE 00		CP 0		CD 09 00		CALL 9	
28 B1		JR Z.taste		1D		DEC E	
OD		DEC C		20 F7		JR NZ,sch13	
79	spanz	LD A,C	Routine zur Berechnung und An-	18 05		JR wahl	
CD 12 00		CALL 12	zeige von Speicher- Platznummer	B7		OR A	
32 02 80		LD (8002),A	und Ziffer.	3E 3F		LD A,3F	
7D		LD A,L		D3 30		OUT (30),A	
E6 OF		AND OF		79	wahl	LD A,C	Speichernummer aus dem C-Register
6F		LD L,A		17		RLA	holen, Adresse berechnen,
79		LD A,C		17		RLA	
17		RLA		17		RLA	
17		RLA		17		RLA	
85		ADD A,L		6F		LD L,A	und ins L- Register laden.
6F		LD L,A		7E		LD A,(HL)	Ziffer ins A- Register holen,
7E		LD A,(HL)		FE OE		CP OE	vergleichen mit Wahlendezeichen,
CD 12 00		CALL 12		28 3E		JR Z.ende	wenn ja, zum Wählende springen.
32 07 80		LD (8007),A		CD 12 00	call	CALL 12	Zu wählende Ziffer umcodieren und
18 95		JR taste		32 07 80		LD (8007),A	in den Anzeigepuffer laden.
18 00		JR 00	Reservesprünge.	7E	holen	LD A,(HL)	(nichtcodierte) Ziffer holen und
18 00		JR 00		OR A		OR A	wenn Ziffer 0,
18 00		JR 00		20 02		JR NZ,ziff	
18 01		JR 01		3E OA		LD A,OA	10 Impulse laden.
80		ADD A,B		57	ziff	LD D,A	Ziffer im D- Register retten.
18 8A	jrtaste	JR taste	Zwischensprünge zur Erweiterung	3E 3C	set	LD A,3C	Impulsrelais setzen.
18 10	jrab	JR ab	der Relativsprung- Distanz.	D3 30		OUT (30),A	
18 1E	jrauf	JR auf		1E 38		LD E,38	Verzögerungswert für 60 ms laden.
18 2C	jrlösch	JR lösch		CD 09 00	sch14	CALL 9	Zeitschleife (60 ms) mit Anzeige.
18 80	jranz1	JR jranz		1D		DEC E	
18 2C	jraus3	JR aus		20 FA		JR NZ,sch14	
18 BE	test4	JR test'		3E 3F		LD A,3F	Impulsrelais zurücksetzen.
18 FO	jrtastel	JR jrtaste		D3 30		OUT (30),A	
18 12	jrauf	JR auf		1E 25		LD E,25	Verzögerungswert für 40 ms laden.
18 9A	jrlad	JR lad		CD 09 00	sch15	CALL 9	Zeitschleife (40 ms) mit Anzeige.
7B	ab	LD A,E	Die Programmteile ab und auf	1D		DEC E	
FE OF		CP OF	ermöglichen mittels der Tasten	20 FA		JR NZ,sch15	
28 F5		JR jrtastel	C und D das "Durchwandern" eines	15		DEC D	Wenn noch nicht alle Impulse aus-
2D		DEC L	Speicherplatzes von der ersten	20 E5		JR NZ,set	gesendet, wiederholen.
1C		INC E	bis zur letzten Ziffer (Platz-	3E FF		LD A,OFF	Anzeigepuffer an der Stelle der
7E		LD A,(HL)	Nummer 0 ... F) und umgekehrt.	32 07 80		LD (8007),A	Ziffernanzeige löschen.
CD 12 00		CALL 12	bei gleichzeitig wandernder	1E FF		LD E,OFF	Zeitschleife (ca 500 ms) mit
32 07 80		LD (8007),A	Anzeige von Platznummer und	CD 09 00	sch16	CALL 9	Anzeige.(Pause zwischen den Ziffern)
18 E4		JR jranz1	Ziffer.	CD 09 00		CALL 9	
7B	auf	LD A,E		1D		DEC E	
FE 00		CP 0		20 F7		JR NZ,sch16	
28 E5		JR Z.jrtaste		23		INC HL	Adresse im Rufnummernspeicher er-
2C		INC L		7E		LD A,(HL)	höhen und nächste Ziffer laden.
1D		DEC E		FE OE		CP OE	Wahlendezeichen abfragen, wenn
7E		LD A,(HL)		20 C2		JR NZ,call	nicht, nächste Ziffer ausgeben.
CD 12 00		CALL 12		3E FF	ende	LD A,OFF	V- Relais zurücksetzen.
32 07 80		LD (8007),A		D3 30		OUT (30),A	
18 D4		JR jranz1		18 99		JR aus4	Ausgabe der nächsten Rufnummer
3E OE	lösch	LD A,OE	Setzen der Wählendemarke.				anbieten.
18 DA		JR jrlad					Hinweis: Port- Adressierungsbefehle sind fett gedruckt.

Unterprogramm Ausgabe

11 88 C1	aus	LD DE,OC188	Anzeigepuffer laden mit:
ED 53 00 80		LD (8000),DE	"A U S G. S P. n"
11 92 42		LD DE,4292	
ED 53 02 80		LD (8002),DE	
11 FF 92		LD DE. 92FF	
ED 53 04 80		LD (8004),DE	
11 OC AB		LD DE,0A0BC	
ED 53 06 80		LD (8006),DE	
CD OC 00	asprn	CALL OC	Einlese- u. Anzeigeschleife.

Hex. - Dump der nachfolgenden 58₁₀ Speicherplätze

Hex-Inhalt																ASCII - Code																																			
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																				
																01	23	45	67	89	AB	CD	EF																												
																65	6E	65	72	20	4C	65	69	70	7A	69	67	65	72	20	53																				
																74	72	2E	32	20	36	34	35	37	20	4D	61	69	6E	74	61																				
																6C	20	20	20	54	65	6C	2E	30	36	31	38	31	20	34	39																				
																37	31	36	30																																
																																																E.Mald			
																																																ener Leipziger S			
																																																tr.2 6457 Mainta			
																																																l Tel.06181 49			
																																																7160			

Neue TOOL-Programme eingetroffen

Kompliment an unsere Software-Freaks!

Mehr als erwartet haben Sie, als LOOP-Leser, auf unseren „Aufruf an alle Software-Freaks!“ in LOOP 15/3 reagiert.

Es wurden uns einige Routinen aus der Feder von NDR-Computer-Anwendern zur Veröffentlichung in TOOL-Disketten zugeschickt. Besonders gefreut haben wir uns über den „Mut“ von einigen „Computer-Einsteigern“, die ihre noch kleinen Programme freigegeben haben.

Bezüglich der Qualität der TOOL-Disketten sind wir nur zusammen stark. Der „alte Hut“ des einen, kann der faszinierte Lesestoff des anderen sein.

Sicher möchte man seine (hart erarbeiteten) Programmteile nur ungern allen zur Verfügung stellen. Trotzdem freut man

sich immer wieder über einen interessanten LOOP-Artikel. Um viele gute LOOP-Artikel und TOOL-Programme anbieten zu können, ist Ihre Mitarbeit so wertvoll. Da wir noch nicht genug Material für eine versprochene TOOL-Diskette haben, bitten wir Sie noch einmal – im Sinne aller Anwender – sich von Ihren (nicht mehr so wichtigen) Programmteilen zu „trennen“.

Interessante Themenbereiche:

- RL-BASIC
- Z80-Burbo Tascal
- Z80-Spiele
- Assemblerprogramme
- mc-modular-AT.

Mein Steckbrief in kurzen Worten:

- Klaus Rumrich,
Goethestraße 53, 6457 Maintal 2,
Telefon: (06109) 63417
- geboren am 1. August 1963
- Studium: Theoretische Physik
in Frankfurt/Main.

Mein erster Rechner war 1980 ein AIM 65. Ende 1984 kaufte ich mir einen NDR-Computer, weil dieser Rechner hervorragend für eigene Hardware-Basteleien und Assemblerprogrammierungen geeignet war und zu diesem Zeitpunkt einer der wenigen, überhaupt erhältlichen Rechner mit 16 Bit CPU war.

Hard- und Software-Entwicklungen:

- Automatische Schwimmwettkampfauswertung,

Steckbrief



- Schach
- 3D-Labyrinth
- Umbau der Detmolder DRAM 128 zur DRAM 512

Bevorzugte Programmiersprache:

- Assembler.

Anmerkung der Redaktion:

Gerade mit seinem Schachprogramm, welches unter JADOS läuft und mit allen CPUs 680xx betrieben werden kann, hat Klaus Rumrich bewiesen, daß er als „Klasse-Programmierer“ bezeichnet werden kann.

Lesen Sie dazu doch einmal in der LOOP 15/3 was ein Schachprogramm „denkt“. Darin läßt sich annähernd nachvollziehen, wie kompliziert das Schreiben eines solchen Programmes sein kann.

Funktioniert Ihr Computer?

Testprogramm im Monitor des Einsteigerpaketes

Im EPROM des Einsteiger-Monitors war noch Platz. Was lag näher, als hier die Testprogramme, die bei der Fertigung verwendet werden, unterzubringen.

In diesem Artikel wollen wir die Programme beschreiben und das Quellisting abdrucken. Die Programme sind, um auch Einsteigern eine nützliche Hilfe zu bieten, geradlinig und ohne „Tricks“ programmiert.

Das Testprogramm, das aus mehreren kleinen Einzelprogrammen besteht, hat die Aufgabe, die vorhandene Hardware zu testen. Es beinhaltet folgende Teile:

- Leuchtdiodentest mit 55/AA-Muster
- Test der 7-Segment-Anzeige

- Tastaturtest
- Test der DIL-Schalter
- Test des Speichers auf der SBC3 und die Ports der IOE (mit zusätzlichem Adapter).

55/AA-Test

Hier werden abwechselnd die Werte 55h und AAh auf die LED's (Port 02h) ausgegeben. Dies bewirkt, daß die Leuchtdioden mit den geraden Nummern und die mit den ungeraden abwechselnd leuchten.

Durch den Aufruf der Betriebssystemroutine CLEAR wird die Anzeige gelöscht. Danach wird der Wert AAh (binär: 10101010) auf die Leuchtdioden (Adresse

02h) ausgegeben. Damit die Leuchtdioden auch sichtbar leuchten, muß der Programmablauf mit einer Warteroutine verzögert werden. Das Ausgeben des Bitmusters 55h (binär: 01010101) auf die Adresse 02h bewirkt, daß nun die anderen vier Leuchtdioden aufleuchten. Danach wird wieder die Warteroutine aufgerufen. Der Befehl „DJNZ AA55“ dekrementiert den Wert im Register B und springt zum Label AA55, wenn der Wert in B nicht Null ist. Wurde das Register bis Null heruntergezählt, so fährt das Programm mit dem nächsten Befehl fort.

Sägezahnfunktion

Zuerst wird die Anzeige gelöscht und alle LED's durch Ausgeben der binären Zahl

„11111111“ gelöscht. Zwischen den Ausgabebefehlen wird jeweils die Warteroutine aufgerufen, damit der Vorgang sichtbar abläuft. Anschließend wird jeweils ein Byte an die LED's ausgegeben, das von rechts her (Bit 0) mit Nullen aufgefüllt wird. Als erstes wird also Bit 0 auf Null gesetzt, bei der nächsten Ausgabe auf den Port 02h folgt das Bit 1 usw., bis der Wert 0000000b an die Leuchtdiode gesendet wird. Dadurch werden die LEDs nacheinander eingeschaltet, bis am Schluß alle LEDs leuchten.

7-Segment-Test

Es werden die Startadressen der Tabellen zur Ansteuerung der Segmente in das HL-Register geladen und durch das Unterprogramm DISP auf alle 8 Anzeigen gebracht. Im Unterprogramm DISP wird zuerst ein Zähler (Reg. B) mit FFh geladen. Dann wird die Betriebssystemroutine PRINT aufgerufen. Dieses Unterprogramm (Sprungadresse 15h) hat die Aufgabe 8 Bytes ab der Adresse, die im Register HL steht, ins ANZFELD zu übertragen. Jetzt muß die Routine ANZEIGE (Adresse 09h) aufgerufen werden, die den Segmentcode, der im ANZFELD steht für ca. 1ms auf die Anzeige bringt. Damit man ein kontinuierliches Bild erhält, ist dieser Aufruf in eine Schleife eingebaut, in der die Routine ANZEIGE 256 mal (ffh) aufgerufen wird. In einer zweiten Schleife wird das Unterprogramm nochmals 256 mal aufgerufen. Der Segmentcode wird also insgesamt 512 mal zur Anzeige gebracht, bevor die Tabelle eines neuen Segmentcodes ins ANZFELD lädt und anzeigt. So werden nacheinander alle 8 Segmente aller acht Anzeigestellen angesteuert.

Tastentest

Zunächst wird eine Tabelle eingelesen, die dann durch die Routine PRINT das Wort „taste“ zur Anzeige bringt.

Die Betriebssystemroutine HOLETASTE wartet auf einen Tastendruck. Danach wird mit dem Befehl cp getestet, um welche Taste es sich handelt. Zu diesem Zweck werden die Tastaturcodes (z.B. 07h für Taste C) mit dem Wert aus der Routine HOLETASTE verglichen und bei Übereinstimmung wird die Tastennummer in Reg. B (von oben links=1 bis unten rechts=24 durchnummeriert), in den Akku geladen und durch das Unterprogramm AUS zur Anzeige gebracht. In diesem Unterprogramm wird die Tastennummer in den Akku und die Anzeigeadresse 8006h in das Register IX geladen. Die Routine PRTAC bringt nun den hexadezimalen Inhalt des Akkus an der Stelle von IX ins ANZFELD. Durch das mehrmalige Aufrufen der Routine ANZEIGE wird dieser Tastaturwert sichtbar gemacht. Danach wird wieder zur Tasteneingabe zurückgesprungen. Das Betätigen der

Taste „—“ beendet dieses Programmteil und es wird mit dem nächsten Test fortgefahren.

Test der DIL-Schalter

Am Anfang dieses Programmteiles wird aus der Tabelle DILTAB die Datenfolge geholt, die durch das Unterprogramm DISP das Wort „dil“ auf die Anzeige bringt. Nach einem Tastendruck wird die Stellung der Schalter auf die Leuchtdioden ausgegeben, die zuerst über die Adresse 02h eingelesen und dann wieder auf die gleiche Adresse ausgegeben wird. Steht z.B. der Schalter 5 auf „ON“, so leuchtet auch die LED-Nr. 5 auf. Der Befehl „cp 0FFh“ vergleicht die Stellung der DIP-Schalter mit dem Wert ffh und der darauf folgende Jump-Befehl springt zum nächsten Test, wenn das Ergebnis des Vergleiches positiv ausfällt, also alle DIP-Schalter auf „OFF“ stehen.

Test der IOE-Karte

Bei diesem Test werden die beiden Ports der IOE-Karte getestet. Dazu wird der Wert 55h auf die Ports F0h und F1h ausgegeben. Anschließend werden die beiden Ports wieder eingelesen und mit dem Wert 55h verglichen. Ist der eingelesene Wert nicht gleich dem ausgegebenen Wert, wird eine Fehleroutine aufgerufen. Ansonsten folgt der nächste Test.

Speichertest

In diesem Programmteil ist nun leider ein Fehler unterlaufen. Aus diesem Grund führt das Programm diesen Test gar nicht aus und zeigt immer ein positives Ergebnis des Testes an. Ist beim Test der IOE-Karte kein Fehler aufgetreten, erscheint auf der Anzeige der Schriftzug „SBC IOE“. Wenn ein Fehler beim IOE-Test aufgetreten ist, wird statt „IOE“ nur ein waagrechtlicher Strich abgezeigt.

Falls Sie dennoch den Speicher Ihrer HEXIO testen wollen, können Sie das korrekte Speichertestprogramm abtippen und starten.

Verbesserter Speichertest

Dieses Programm muß von Hand ab der Adresse 8100h eingegeben werden. Es beschreibt den Speicherbereich von 8200h bis 9FF0h mit den Werten 55h und AAh und liest dann diese Werte wieder aus. Nach dem Programmstart wird die Adresse 8200h (Anfangsadresse des zu testenden Speicherbereiches) in das HL-Register geladen. Die Endadresse des zu testenden Speicherbereiches 9ff0h wird in das Registerpaar BC geschrieben. Der erste Testwert 55h, mit dem der Speicher beschrieben wird, wird in das Register E geladen.

Der Befehl „ld (hl),e“ lädt die Speicherzelle, auf die das Register HL zeigt, mit dem Testwert. Darauf wird der Adreßzeiger HL um den Wert Eins erhöht. Um bei der

nachfolgenden Subtraktion den Wert von HL nicht zu verlieren, wird dieser zuvor auf den Stapel geschrieben. Nun wird von der aktuellen Adresse der Endwert des Testspeichers abgezogen. Ist das Ergebnis Null (Speicher voll geschrieben), so springt das Programm zur Leseroutine. Ist das Speicherende noch nicht erreicht, wird der HL-Wert wieder vom Stapel geholt und an den Anfang der Schreibroutine zurückgesprungen.

Am Anfang der Leseroutine wird zuerst der Adreßzeiger HL wieder auf den Anfangswert gesetzt. Dann wird der Inhalt der aktuellen Speicherzelle in den Akku geschrieben. Ist das Ergebnis der Subtraktion des Testwertes vom Akku nicht Null, d. h. es wurde ein anderer Wert aus der Speicherzelle gelesen, als hineingeschrieben wurde, liegt ein Speicherfehler vor. In diesem Fall wird der Test abgebrochen und gleich zum Programmteil FEHLER gesprungen. Ist der Test der Speicherzelle positiv verlaufen, wird mit der gleichen Methode, wie im ersten Programmteil, geprüft, ob das Ende des Speicherbereiches schon erreicht ist. Wenn ja, dann wird der Programmteil SCHLEIFEND aufgerufen. Ansonsten wird die erhöhte Adresse vom Stapel wieder in HL gespeichert und an den Anfang der Leseroutine zurückgesprungen.

Die Routine SCHLEIFEND vergleicht erst den Testwert mit dem Wert AAh. Nach dem ersten Testdurchlauf mit 55h fällt dieser Vergleich negativ aus und es wird das HL-Register auf den Speicheranfang zurückgesetzt und der zweite Testwert AAh in das Register E geladen. Nun wird an den Anfang der Schreibroutine gesprungen und der ganze Test mit dem Wert AAh nochmals durchgeführt. Nach erfolgreicher Beendigung dieses Testes wird zum Programmteil ERFOLG gesprungen.

Der Programmteil ERFOLG hat die Aufgabe, die Meldung „SBC ok“ auszugeben. Zunächst wird die Adresse der Tabelle, in der die Codes für die Buchstaben stehen, in das HL-Register geladen. Die Betriebssystemroutine PRINT sorgt dafür, daß die Buchstabencodes der Tabelle in das ANZFELD übertragen werden. In einer Endlosschleife wird zum Schluß immer wieder die Routine ANZEIGE aufgerufen, damit die Schrift flimmerfrei auf der Anzeige erscheint.

Der Programmteil FEHLER ist identisch mit ERFOLG, mit dem einen Unterschied, daß hier die Codenummern aus der Tabelle FEHLTAB geholt werden. Demzufolge wird das Wort „FEHLER“ angezeigt. Das Programm endet auf jeden Fall in einer Endlosschleife, in der die entsprechende Meldung angezeigt wird. Die HEXIO kann nach Programmende durch Betätigen der Resettaste wieder in den Ausgangszustand versetzt werden.

Achtung: Bei diesem Test wird der Speicherbereich von 8200h bis 9FF0h völlig überschrieben. Programme oder Daten, die in diesem Bereich gespeichert sind, gehen unwiederbringlich verloren!

Startadressen der einzelnen Programme

Adresse	Programm
1 1D00	AA/55-Test
2 1D15	Sägezahnstest
3 1D65	7-Segment-Test
4 1DCD	Tastentest
5 1E90	DIL-Schalter-Test
6 1EA4	IOE-Test
7 1EBA	Speichertest (nicht lauffähig)
7a 8100	berichtigter Speichertest zum Eintippen

ROA256/1M auch für ECB-Systeme

Unsere neue universelle Speicherbaugruppe, die ROA256/1M ist auch für den ECB-Bus erhältlich. Die Vorzüge dieser Baugruppe (Batteriepufferung, statische RAMs oder EPROMs, 32K x 8 oder 128K x 8, WAIT-Logik, Ausblendlogik) sind damit auch für ECB-Systeme nutzbar. Um allerdings sinnvoll mit dem gesamten Speicher der ROA256/1M arbeiten zu können, benötigt Ihr ECB-System eine BANK-Logik. Das Signal „BANKEN“, das der ECB-Bus normalerweise nicht zur Verfügung stellt, wurde bei der ROA256/1M auf den Pin C13 gelegt; dieser Pin ist sonst frei (nc).

Wichtiger Hinweis für alle LOOP-Autoren!

Die LOOP wird Schritt für Schritt im Satz mit PageMaker gestaltet werden. Dazu bitten wir Sie, falls Ihnen dies möglich ist, folgende Hinweise zu beachten:

Artikel wollen Sie bitte immer im NDR- oder im IBM-Format als Textfile mit einschicken. Selbstverständlich erhalten Sie von uns eine entsprechende leere Diskette wieder zurück. Bitte beachten Sie, daß Sie möglichst keine harten Carriage>Returns innerhalb dieser Datei einfügen, da sonst der Umbruch von PageMaker nicht funktioniert.

Programme wollen Sie bitte, sofern möglich, mit auf die Diskette spielen. Weiter bitten wir Sie, einen Programmausdruck mit möglichst neuem Farbband beizulegen. Achten Sie bitte auch darauf, daß die Anzahl der Zeichen pro Zeile bei Programm-Listings 60 nicht übersteigt, da wir sonst zu weit verkleinern müssen.

Herzlichen Dank für Ihre Mitarbeit.

55/aa-Test

Anzeige löschen
Wiederhole
0aah in Akku laden und auf LED-Port ausgeben
Warten
55h in Akku laden und auf LED-Port ausgeben
Warten
fünf mal

7 - Segment - Test

Anzeige löschen
Tabelle 1 laden (Daten für Segmente)
UP Disp aufrufen (Anzeige der Segmente)
Tabelle 2 laden (Daten für Segmente)
UP Disp aufrufen (Anzeige der Segmente)
Tabelle 16 laden
UP Disp aufrufen (Anzeige der Segmente)

Test der DIL-Schalter

Wiederhole
Wort "dil" anzeigen
Stellung der Schalter einlesen
Schalterstellung der Schalter in Akku auf den LED-Port auslesen
bis alle Schalter aus sind (Wert FFh)

Tastentest

Ausgabe des Wortes "taste"	
Wiederhole	
auf Tastendruck warten	
Taste C gedrückt?	
ja	nein
D=1 Aufruf UP Ausgabe	
Taste D gedrückt?	
ja	nein
D=2 Aufruf UP Ausgabe	
Taste "-" gedrückt?	
ja	nein
D=24 Aufruf UP Ausgabe ENDE Tastentest	

Sägezahnstest

Anzeige löschen
Muster "11111111" in Akku laden
Akku auf LED's ausgeben
Warten
Muster "11111110" in Akku laden
Akku auf LED's ausgeben
Warten
Muster "00000000" in Akku laden
Akku auf LED's ausgeben
Warten

Test der IOE-Karte

55h an Port F0h ausgeben	
55h an Port F1h ausgeben	
Einlesen des ersten Ports	
Wert = 55h ?	
ja	nein
Einlesen des zweiten Ports	Sprung zur Fehleranzeige
Wert = 55h ?	
ja	nein
ENDE	Sprung zur Fehleranzeige

Speichertest

Speicheranfang=8200h	Speicherende=9FF0h	Bitmuster=55
Wiederhole	Wiederhole	
	Bitmuster bei Adresse im Speicher ablegen	
	Adresse um eins erhöhen	
bis Speicherende erreicht		
Wiederhole	Wert aus Speicheradresse auslesen	
	Adresse um eins erhöhen	
	Wert = Testwert?	
	ja	nein
		Aufruf Fehlerroutine
bis Speicherende erreicht		
mit Testwert = AAh		
Aufruf Routine ERFOLG		

MACRO-B0 3.43 27-Jul-81 PAGE 1

```

;*****
; * Speichertestprogramm für S803 *
; * (C) Graf Elektronik Systeme *
; * V 1.1 26. November 1987 *
;*****

```

```

0032      org 8100h
0033      clear equ 0033h ; Anzeige löschen
0034      print equ 0015h ; Tabelle einlesen
0035      anzeige equ 0009h ; Meldung zur Anzeige bringen

8100' 21 8200      start: ld hl,8200h ; HL-Register auf Speicheranfang stellen
8101' 01 9FF0      ld bc,9FF0h ; BC-Register auf Speicherende stellen
8106' 1E 95        ld a,55h ; Testwert 55h in Reg. E speichern

8108' 73        lxi ld (hl),e ; Testwert in Speicherstelle ablegen
8109' 23        inc hl ; Adresse inkrementieren
810A' E5        push hl ; aktuelle Adresse auf Stapel schreiben
810B' E0 47      sbc hl,bc ; Endadresse von aktueller Adresse subtrahieren
810C' 28 03      jr z,lesen ; Schleifenende, wenn Ergebnis = 0
810F' E1        pop hl ; Adresse von Stapel holen
8110' 1B F6      jr lx ; weiter mit nächster Adresse

8112' 21 8200      lesens: ld hl,8200h ; HL-Register auf Speicheranfang stellen
8113' 7E        ld a,(hl) ; Wert aus Speicher lesen
8114' 23        inc hl ; Adresse inkrementieren
8117' 93        sub e ; Testwert subtrahieren
8118' 20 1F      jr nz,fehler ; falls Ergebnis nicht 0 dann Fehlerausgabe
811A' E5        push hl ; aktuelle Adresse auf Stapel schreiben
811B' E0 47      sbc hl,bc ; Endadresse von aktueller Adresse subtrahieren
811C' 28 03      jr z,schleifend ; Schleifenende, wenn Ergebnis = 0
811F' E1        pop hl ; Adresse von Stapel holen
8120' 1B F3      jr ly ; weiter mit nächster Adresse

8122' 76        schleifend: ld a,e ; Testwert -> Akku
8123' FE AA      cp 0aah ; Vergleich mit AAh
8125' 2B 07      jr z,erfolg ; wenn ja, dann Erfolgsmeldung
8127' 21 8200      ld hl,8200h ; HL zurücksetzen
812A' 1E AA      ld e,0aah ; zweiter Testwert AAh
812C' 1B DA      jr lx ; Sprung zu Programmteil Schreiben

812E' 21 8144      erfolg: ld hl,erftab ; Tabellenadresse laden
8131' E0 0015      call print ; Tabelle einlesen
8134' CD 0009      lpl: call anzeige ; Meldung anzeigen
8137' 1B F8      jr lpl ; Endlosschleife

8139' 21 814C      fehler: ld hl,fehltab ; Tabellenadresse laden
813C' CD 0015      call print ; Tabelle einlesen
813F' CD 0009      lp2: call anzeige ; Meldung anzeigen

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

8142' 1B F8      jr lp2 ; Endlosschleife

8144' 92 83 CA FF      erftab: db 92h,83h,0cah,0ffh,0a3h,0b5h,0ffh,0ffh
8148' 43 85 FF FF

814C' 8E 88 89 C7      fehltab: db 08eh,8ah,89h,0c7h,8ah,8afh,0ffh,0ffh
8150' 86 AF FF FF

end

```

MACRO-B0 3.43 27-Jul-81 PAGE 5

Macros:

Symbols:

```

0009 ANZEIGE      0032 CLEAR      812E ERFOLG
8144 ERFTAB      8139 FEHLER      814C FEHLTAB
8111 LESEN       8134 LPI       813F LP2
8106 LY         8115 LV        0015 PRINT
8122 SCHLEIFEND 8100 START

```

No Fatal errors!

MACRO-B0 3.43 27-Jul-81 PAGE 1

```

;*****
; * Testprogramm auf Eprom E160HE12 *
; * (C) Graf Elektronik Systeme *
; * 27. Juli 1981 *
;*****

```

```

0033      clear      equ 0033h ; Anzeige löschen
000C      holetaste equ 000Ch ; auf Tastendruck warten
0015      print     equ 0015h ; Tabelle einlesen
0009      anzeige   equ 0009h ; etwas zur Anzeige bringen
0018      prtac     equ 0018h ; Akkuinhalt ausgeben

org 1000h

;*****
;*** 55/aa - Test ***

1000' CD 0033      start: call clear ; Anzeige löschen
1003' 06 05      ld b,5 ; 5 Durchläufe

```

```

1005' 3E AA      out (02h),a
1007' 03 02      call wait
1009' CD 105B'

```

```

100C' 3E 55      out (02h),a
100E' 03 02      call wait
1010' CD 105B'
1013' 10 F0      djnz AASS

```

```

AASS: ld a,0aah ; AA in Akku laden
      out (02h),a ; Muster in Akku auf LED's ausgeben
      call wait ; Warterroutine aufrufen

ld a,055h ; 55 in Akku laden
out (02h),a ; Muster in Akku auf LED's ausgeben
call wait ; Warterroutine aufrufen
djnz AASS ; Schleifenende nach 5 Durchläufen

```

*** Sägezahntest ***

```

1015' CD 0033

```

```

saeges: call clear ; Anzeige löschen

```

```

1018' 3E FF      out (02h),a
101A' 03 02      call wait
101C' CD 105B'

```

```

ld a,11111111b ; alle LED's ausschalten
out (02h),a ; Warterroutine aufrufen

```

```

101F' 3E FE      out (02h),a
1021' 03 02      call wait
1023' CD 105B'

```

```

ld a,11111110b ; erste LED einschalten
out (02h),a

```

```

1026' 3E FC      out (02h),a
1028' 03 02      call wait
102A' CD 105B'

```

```

ld a,11111100b ; zweite LED einschalten usw.
out (02h),a

```

```

102D' 3E F8      out (02h),a
102F' 03 02      call wait
1031' CD 105B'

```

```

ld a,11111000b
out (02h),a

```

```

1034' 3E F0      out (02h),a
1036' 03 02

```

```

ld a,111110000b
out (02h),a

```

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

```

1038' CD 105B'

```

```

call wait

```

```

103B' 3E E0      out (02h),a
103D' 03 02      call wait
103F' CD 105B'

```

```

ld a,111000000b
out (02h),a

```

```

1042' 3E C0      out (02h),a
1044' 03 02      call wait
1046' CD 105B'

```

```

ld a,110000000b
out (02h),a

```

```

1049' 3E 40      out (02h),a
104B' 03 02      call wait
104D' CD 105B'

```

```

ld a,100000000b
out (02h),a

```

```

1050' 3E 00      out (02h),a
1052' 03 02

```

```

ld a,000000000b
out (02h),a

```

*** Pause bis zum nächsten Test ***

```

1054' 06 05      call rot
1056' CD 1F20'
1059' 1B 0A      jr seg

```

```

ld b,05 ; 05 in Register B laden
call rot ; Aufruf ROT-Interprogramm
jr seg ; Sprung zum nächsten Test

```

*** Warteschleife, damit die Anzeige nicht rast ***

```

105B' 21 0000      wait: ld hl,0000h
105E' 11 0001      ld de,0001h
1061' C3 1F1A'      jp loop1
1064' C9          ret

```

```

; 0000h nach HL-Register
; 0001h nach DE-Register
; Sprung zu Programmteil loop1
; Rücksprung

```

*** 7 - Segment - Test ***

```

1065' CD 0033

```

```

seg: call clear ; Anzeige löschen

```

```

1068' 21 1F29'      ld hl,tab1
106B' CD 1F06'      call disp

```

```

; Tabelle 1 laden
; UP Disp aufrufen

```

```

106E' 21 1F31'      ld hl,tab2
1071' CD 1F06'      call disp

```

```

; Tabelle 2 laden usw.

```

```

1074' 21 1F39'      ld hl,tab3
1077' CD 1F06'      call disp

```

```

ld hl,tab4
call disp

```

```

107A' 21 1F41'      ld hl,tab4
107D' CD 1F06'      call disp

```

```

ld hl,tab5
call disp

```

```

1080' 21 1F49'      ld hl,tab5
1083' CD 1F06'      call disp

```

```

ld hl,tab6
call disp

```

```

1086' 21 1F51'      ld hl,tab6
1089' CD 1F06'      call disp

```

```

ld hl,tab7
call disp

```

```

108C' 21 1F59'      ld hl,tab7
108F' CD 1F06'      call disp

```

```

ld hl,tab8

```

```

1092' 21 1F61'      ld hl,tab8

```

```

call disp

```

```

1098' 21 1F69'      ld hl,tab9
109B' CD 1F06'      call disp

```

```

ld hl,tab10
call disp

```

```

10A1' 21 1F71'      ld hl,tab10
10A4' CD 1F06'      call disp

```

```

ld hl,tab11
call disp

```

```

10A7' 21 1F79'      ld hl,tab11
10AA' CD 1F06'      call disp

```

```

ld hl,tab12
call disp

```

```

10AD' 21 1F81'      ld hl,tab12
10B0' CD 1F06'      call disp

```

```

ld hl,tab13
call disp

```

```

10B3' 21 1F89'      ld hl,tab13
10B6' CD 1F06'      call disp

```

```

ld hl,tab14
call disp

```

```

10B9' 21 1F91'      ld hl,tab14
10BC' CD 1F06'      call disp

```

```

ld hl,tab15
call disp

```

```

10C2' 21 1FA1'      ld hl,tab15
10C5' CD 1F06'      call disp

```

```

ld hl,tab16
call disp

```

MACRO-B0 3.43 27-Jul-81 PAGE 1-2

```

1095' CD 1F06'

```

```

call disp

```

```

1098' 21 1F69'      ld hl,tab9
109B' CD 1F06'      call disp

```

```

ld hl,tab10
call disp

```

```

109E' 21 1F71'      ld hl,tab10
10A1' CD 1F06'      call disp

```

```

ld hl,tab11
call disp

```

```

10A4' 21 1F79'      ld hl,tab11
10A7' CD 1F06'      call disp

```

```

ld hl,tab12
call disp

```

```

10AA' 21 1F81'      ld hl,tab12
10AD' CD 1F06'      call disp

```

```

ld hl,tab13
call disp

```

```

10B0' 21 1F89'      ld hl,tab13
10B3' CD 1F06'      call disp

```

```

ld hl,tab14
call disp

```

```

10B6' 21 1F91'      ld hl,tab14
10B9' CD 1F06'      call disp

```

```

ld hl,tab15
call disp

```

```

10BC' 21 1F99'      ld hl,tab15
10BF' CD 1F06'      call disp

```

```

ld hl,tab16
call disp

```

; *** Pause bis zum nächsten Test ***				IEB1' 00 21 800h	ld ix,800h	
1DC8' 06 03		ld b,03		IEB5' CD 001B	call prtac	
1DCA' CD 1F20'		call rot		IEB8' CD 0009	jp: call anzeige	; nächste Taste
				IEBB' 10 FB	djnz jp	
; *** Tastentest ***				IEBD' C3 1003'	jp taster	
1DC0' 21 1FB9'	taste:	ld hl,tast	; HL-Reg. auf Tabelle Taste stellen	; *** Test der DIL - Schalter ***		
1DD0' CD 0015		call print	; Wort "taste" ausgeben	IE90' 21 1FA9'	dil:	ld hl,diltab
1DD3' CD 000C	taster:	call holetaste	; auf Tastendruck warten	IE93' CD 1F06'		call disp
1DD6' 16 01		ld d,1	; wird Taste C gedrückt,	IE96' CD 000C		call holetaste
1DD8' FE 07		cp 07h	; so wird die 1 ausgegeben	IE99' 08 02	dill:	in a,(02h)
1DDA' CA 1E7E'		jp z,aus		IE9B' 03 02		out (02h),a
1DDD' 16 02		ld d,2	; - wie oben -	IE9D' FE FF		cp 0ffh
1DDF' FE 17		cp 17h	; jedoch Taste D	IE9F' 28 03		jr z,io
1DE1' CA 1E7E'		jp z,aus		IEA1' C3 1E99'		jp dill
1DE4' 16 03		ld d,3		; *** Test der IOE - Karte ***		
1DE6' FE 27		cp 27h	; Taste E	IEA4' 3E 55	io:	ld a,55h
1DE8' CA 1E7E'		jp z,aus		IEA6' 03 F0		out (0f0h),a
1DEB' 16 04		ld d,4		IEA8' 03 F1		out (0f1h),a
1DED' FE 37		cp 37h	; Taste F	IEAA' 08 F0		in a,(0f0h)
1DEF' CA 1E7E'		jp z,aus		IEAC' FE 55		cp 055h
1DF2' 16 05		ld d,5		IEAE' 28 03		jr z,io1
1DF4' FE 47		cp 47h	; Taste DEF	IEB0' C3 1EE5'		jp faul1
MACRO-80 3.43	27-Jul-81	PAGE	1-3	IEB3' 08 F1	io1:	in a,(0f1h)
				IEB5' FE 55		cp 55h
				IEB7' 28 20		jr z, ende
				; *** Speichertest ***		
1DF6' CA 1E7E'		jp z,aus		IEB9' 13 35		jr faul2
1DF9' 16 06		ld d,6		; *** Speichertest ***		
1DFB' FE 57		cp 57h	; Taste DR	IEB9' 21 8200		ld hl,8200h
1DFD' CA 1E7E'		jp z,aus		IEBE' 1E 55		ld e,55h
1E00' 16 07		ld d,7		IEC0' 01 9FFF		ld bc,9fffh
1E02' FE 0B		cp 0bh	; Taste B	IEC3' 73	ix:	ld (hl),e
1E04' CA 1E7E'		jp z,aus		IEC4' 23		inc hl
1E07' 16 08		ld d,8		IEC5' 7C		ld a,h
1E09' FE 1B		cp 1bh	; Taste 9	IECA' 90		sub b
1E0B' CA 1E7E'		jp z,aus		IEC7' 20 FA		jr nz,ix
1E0E' 16 09		ld d,9		IEC9' 21 8200		ld hl,8200h
1E10' FE 28		cp 28h	; Taste A	IECC' 5E	iy:	ld e,(hl)
1E12' CA 1E7E'		jp z,aus		IECD' 23		inc hl
1E15' 16 10		ld d,10h		IECE' 06 55		sub 55h
1E17' FE 38		cp 38h	; Taste B	IED0' 20 29		jr nz,faul3
1E19' CA 1E7E'		jp z,aus		IED2' 7C		ld a,h
1E1C' 16 11		ld d,11h		IED3' 90		sub b
1E1E' FE 4B		cp 4bh	; Taste START	IED4' 20 FA		jr nz,ly
1E20' CA 1E7E'		jp z,aus		IEB6' C3 000C		call holetaste
1E23' 16 12		ld d,12h		; *** Testende ***		
1E25' FE 58		cp 58h	; Taste SPE	IEB9' 21 1FB1	ende:	ld hl,tabelle
1E27' CA 1E7E'		jp z,aus		IEBC' CD 0015		call print
1E2A' 16 13		ld d,13h		IEBD' CD 0009	sch:	call anzeige
1E2C' FE 0D		cp 0dh	; Taste 4	IEE2' C3 1EDF'		jp sch
1E2E' CA 1E7E'		jp z,aus		; *** UP faul bei IOE-Fehler ***		
1E31' 16 14		ld d,14h		IEE5' 21 1FC1'	faul1:	ld hl,ng1
1E33' FE 1D		cp 1dh	; Taste 5	IEE8' CD 0015		call print
1E35' CA 1E7E'		jp z,aus		IEEB' CD 0009	jp1:	call anzeige
1E38' 16 15		ld d,15h		IEEE' 1B FB		jr jp1
1E3A' FE 2D		cp 2dh	; Taste 6	IEF0' 21 1FC9'	faul2:	ld hl,ng2
1E3C' CA 1E7E'		jp z,aus		IEF3' CD 0015		call print
1E3F' 16 16		ld d,16h		IEF6' CD 0009	jp2:	call anzeige
1E41' FE 3D		cp 3dh	; Taste 7	IEF9' 1B FB		jr jp2
1E43' CA 1E7E'		jp z,aus		IEFB' 21 1FD1'	faul3:	ld hl,ng3
1E46' 16 17		ld d,17h		IEFE' CD 0015		call print
1E48' FE 4D		cp 4dh	; Taste STEP	IEF1' CD 0009	jp3:	call anzeige
1E4A' CA 1E7E'		jp z,aus		IEF4' 1B FB		jr jp3
1E4D' 16 18		ld d,18h		; *** Tastennummer ausgeben ***		
1E4F' FE 5D		cp 5dh	; Taste +	IF06' 06 FF	disp:	ld b,0ffh
1E51' CA 1E7E'		jp z,aus		IF08' CD 0015		call print
1E54' 16 19		ld d,19h		IF0B' CD 0009	ip:	call anzeige
1E56' FE 0E		cp 0eh	; Taste 0	IF0E' 10 FB		djnz ip
				IF10' CD 0009	xxx:	call anzeige
MACRO-80 3.43	27-Jul-81	PAGE	1-4	MACRO-80 3.43 27-Jul-81 PAGE 1-6		
1E58' CA 1E7E'		jp z,aus		IF13' 10 FB		djnz xx
1E5B' 16 20		ld d,20h		IF15' C9		ret
1E5D' FE 1E		cp 1eh	; Taste 1	IF16' 19	loop:	add hl,de
1E5F' CA 1E7E'		jp z,aus		IF17' 30 FD		jr nc,loop
1E62' 16 21		ld d,21h		IF19' C9		ret
1E64' FE 2E		cp 2eh	; Taste 2	IF1A' 19	loop1:	add hl,de
1E66' CA 1E7E'		jp z,aus		IF1B' 30 FD		jr nc,loop1
1E69' 16 22		ld d,22h		IF1D' C9		ret
1E6B' FE 3E		cp 3eh	; Taste 3	; *** Pause bis zum nächsten Test ***		
1E6D' CA 1E7E'		jp z,aus		IF1E' 06 08		ld b,08
1E70' 16 23		ld d,23h		IF20' CD 105B'	rot:	call wait
1E72' FE 4E		cp 4eh	; Taste DPT	IF23' CD 105B'		call wait
1E74' CA 1E7E'		jp z,aus		IF26' 10 FB		djnz rot
1E77' 16 24		ld d,24h		IF28' C9		ret
1E79' FE 5E		cp 5eh	; Taste -	; ***** Tabellen für Anzeigen *****		
1E7B' CA 1E9D'		jp z,dil	; bei Minus Sprung zum DIL-Test	IF29' 7F 7F 7F	tab1:	db 07fh, 07fh, 07fh, 07fh, 07fh, 07fh, 07fh, 07fh
1E7E' 06 FF	aus:	ld b,0ffh		IF2D' 7F 7F 7F		
1E80' 7A		ld a,d	; Ausgabe des Wertes in Reg. D			

```

IF31' FE FE FE FE      tab2: db 0feh, 0feh, 0feh, 0feh, 0feh, 0feh, 0feh, 0feh
IF35' FE FE FE FE
IF39' FD FD FD FD      tab3: db 0fdh, 0fdh, 0fdh, 0fdh, 0fdh, 0fdh, 0fdh, 0fdh
IF3D' FD FD FD FD
IF41' FB FB FB FB      tab4: db 0fbh, 0fbh, 0fbh, 0fbh, 0fbh, 0fbh, 0fbh, 0fbh
IF45' FB FB FB FB
IF49' F7 F7 F7 F7      tab5: db 0f7h, 0f7h, 0f7h, 0f7h, 0f7h, 0f7h, 0f7h, 0f7h
IF4D' F7 F7 F7 F7
IF51' EF EF EF EF      tab6: db 0efh, 0efh, 0efh, 0efh, 0efh, 0efh, 0efh, 0efh
IF55' EF EF EF EF
IF59' DF DF DF DF      tab7: db 0dfh, 0dfh, 0dfh, 0dfh, 0dfh, 0dfh, 0dfh, 0dfh
IF5D' DF DF DF DF
IF61' BF BF BF BF      tab8: db 0bfb, 0bfb, 0bfb, 0bfb, 0bfb, 0bfb, 0bfb, 0bfb
IF65' BF BF BF BF
IF69' 00 FF FF FF      tab9: db 0h, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
IF6D' FF FF FF FF
IF71' FF 00 FF FF      tab10: db 0ffh, 0h, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
IF75' FF FF FF FF
IF79' FF FF 00 FF      tab11: db 0ffh, 0ffh, 0h, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
IF7D' FF FF FF FF
IF81' FF FF FF 00      tab12: db 0ffh, 0ffh, 0ffh, 0h, 0ffh, 0ffh, 0ffh, 0ffh
MACRO-80 1.43 27-Jul-81 PAGE 1-7
IF85' FF FF FF FF
IF89' FF FF FF FF      tab13: db 0ffh, 0ffh, 0ffh, 0ffh, 0h, 0ffh, 0ffh, 0ffh
IF8D' 00 FF FF FF
IF91' FF FF FF FF      tab14: db 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0h, 0ffh, 0ffh
IF95' FF 00 FF FF
IF99' FF FF FF FF      tab15: db 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0h, 0ffh
IF9D' FF FF 00 FF
IFA1' FF FF FF FF      tab16: db 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0h
IFA5' FF FF FF 00
IFA9' AL EF E7 FF      ditab: db 0alh, 0c7h, 0c7h, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
IFAD' FF FF FF FF
IFB1' 92 83 C6 FF      tab17: db 92h, 83h, 0c6h, 0ffh, 0ffh, 0fch, 0c0h, 086h
IFB5' FF FC C0 86
IFB9' 87 88 92 87      tab18: db 87h, 88h, 92h, 87h, 86h, 0ffh, 0ffh, 0ffh
IFBD' 86 FF FF FF
IFC1' 92 83 C6 FF      ng1: db 92h, 083h, 0c6h, 0ffh, 0ffh, 0ffh, 0ffh, 0bth
IFC5' FF FF FF BF
IFC9' 92 83 C6 FF      ng2: db 92h, 083h, 0c6h, 0ffh, 0ffh, 0ffh, 0bth, 0ffh
IFCD' FF FF BF FF
IFD1' BF BF BF FF      ng3: db 0bth, 0bth, 0bth, 0ffh, 0ffh, 0c7h, 0c7h, 086h
IFD5' FF CF CD 86
end
MACRO-80 1.43 27-Jul-81 PAGE 5
Macros:
Symbols:
1005' A855          0009' ANZEIGE      1E7E' AUS
0033' CLEAR        1E90' DIL          1E99' DILL1
1FA9' DILTAB       1F06' DISP         1ED9' ENDE
1EE5' FAUL1        1EF0' FAUL2        1EFB' FAUL3
000C' HOLETASTE    1EA4' IO           1EB3' I01
1E8B' JP           1EEB' JP1          1EFA' JP2
1F01' JPS          1F16' LOOP         1F1A' LOOP1
1F0B' LP           1EC3' LK           1ECC' LY
1FC1' NG1          1FC9' NG2          1FD1' NG3
0015' PRINT        0018' PRNAC        1F20' ROT
1D15' SAEGE        1EBF' SCH          1B65' SEG
1D00' START        1F29' TAB1         1F71' TAB10
1F79' TAB11        1F81' TAB12        1FB9' TAB13
1F91' TAB14        1F99' TAB15        1FA1' TAB16
1F31' TAB2         1F39' TAB3         1F41' TAB4
1F49' TAB5         1F51' TAB6         1F59' TAB7
1F61' TAB8         1FA9' TAB9         1FD1' TABELLE
1F89' TAST         1DCD' TASTE        1003' TASTER
1D5B' WAIT         1F10' XI
No Fatal error(s)

```

Viele NDR-Baugruppen werden nun vom Einsteigerpaket unterstützt:

TOOL-Programme für das Einsteigerpaket

SER, CENT, AMPEL, SOUND, SPRACHE, AD8/16, AD10/1, DA, REL mit dem Einsteigerpaket

Aufgrund der stetig steigenden Popularität des Einsteigerpaketes haben wir uns entschlossen, einige TOOL-Programme für die Anfänger unter uns anzubieten.

Diese TOOLS bestehen aus fertigen Werkzeugen, mit deren Hilfe es möglich ist, weitere Zusatzbaugruppen des NDR-Computers anzusteuern.

Erweiterungen wie etwa die Ausgabe des Speicherinhaltes über die CENT (parallele Schnittstelle) auf den Drucker sind nun fester Bestandteil des Monitors EINDHEX V1.2, der nach wie vor Standardausrüstung bei unserem Einsteigerpaket ist. Die jeweiligen Unterprogrammroutrinen sind nun durch Anspringen der jeweiligen Adresse zu erreichen. Natürlich können diese Programme nicht nur für sich eigenständig laufen, sondern können (und sollen) von Ihren Programmen aufgerufen werden.

Parallel wird ein Handbuch mit einführender Beschreibung der Treiberprogramme von uns angeboten. Neben umfang-

reicher Dokumentation der Wirkungsweise und der Funktion der Programme (Flußdiagramme), beinhaltet das Begleitheft auch die ausgedruckten Listings (PRN Files – Assemblertext mit übersetztem Hexcode).

Unterstützt werden folgende Baugruppen. SER, CENT AMPEL, SOUND, SPRACHE, AD8/16, AD10/1, DA und die REL-Baugruppe.

Die Situation spitzt sich zu:

RAM-Bausteine werden immer teurer: Auswirkung des japanisch/amerikanischen Handelskrieges:

Es geht uns wie vor einigen Jahren: RAM-Bausteine werden gehandelt wie Heizöl und sind nicht/oder nur zu extrem überhöhten Preisen zu bekommen. So hat sich der Verkaufspreis des Standard 256 K x 1 dynamischen RAM's von ca. DM 7,- auf mittlerweile DM 35,- (!) erhöht. Bestellungen über RAM's, die wir in größeren Mengen an unsere Lieferanten geben, werden uns derzeit für die Lieferzeit 1. Quartal 1989, Preis offen, bestätigt.

Was ist der Grund? Er liegt im amerikanisch/japanischen Handelskrieg. Vor ein-

bis zwei Jahren überschwemmten die Japaner mit ihrer qualitativ hochwertigen und kostengünstig hergestellten Speicherware den amerikanischen Markt. Die Amerikaner kannten – statt wie im normalen Marktgeschehen üblich, mit gleicher Qualität und mit besserer Fertigungsaubeute nachzuziehen – leider nur einen restriktiven Weg. Sie zwangen die japanischen Hersteller, ihre Produkte einzuschränken. Dies taten sie auch; die Amerikaner, wie Texas Instruments, zogen sich schon 1985 aus dem unrentablen Geschäft zurück.

Nun, nachdem dieses Geschäft auch für die Japaner unrentabel geworden war, stellten sie ihre Produkte vorwiegend auf 1- oder 4-MB-Chips um. Gleichzeitig erlebten die IBM-PC's und Kompatiblen – wie auch der mc-modular-AT – aufgrund der gefallen Preise einen extremen Boom. Was passierte: Die Ware wurde knapp. Sie wird so knapp, daß derzeit hier das gleiche passiert wie vor zwei Jahren: amerikanische Einkäufer kommen mit Flugzeugen nach Japan, Singapur, Taiwan oder Korea, kaufen Speicherbausteine in großen Mengen, stecken diese in den Koffer, bezahlen bar und fliegen am gleichen Tag wieder zurück.

Derzeit können wir nur abwarten und hoffen, daß sich die Situation bis Ende des Jahres wieder entspannt. Bis dahin sind leider Speicherpreise nur Tagespreise.

Anwendung eines Logik-Analysators:

Logikanalyse: Was ist das?

74193 – alles klar!

Logikanalysatoren sind ein sehr nützliches Hilfsmittel bei der Erprobung und Inbetriebnahme digitaler Schaltungen, speziell von Computern.

Im Unterschied zu Oszilloskopen sollen sie nicht die Kurvenform der Eingangssignale möglichst genau wiedergeben, sondern den zeitlichen Verlauf der Eingangssignale, wobei nur zwischen logisch 0 und logisch 1 unterschieden wird. Dazu benötigen sie einen Speicher, in den die Eingangssignale mit möglichst hoher Abtastrate eingeschrieben werden und einen auswertenden Teil, der u.a. die gespeicherten Informationen in ein Bild umsetzt. Übliche Abbildungsformen sind das Impuls- und Zeitdiagramm.

Ein Logikanalysator sollte in der Regel 16 oder 32 Kanäle besitzen, um auch mal den Daten- bzw. Adreßbus eines „normalen“ 8- bzw. 16-Bit-Prozessors abbilden zu können. Die Frequenz, mit der die Kanäle abgetastet werden, muß für ein Zeitdiagramm deutlich über der Frequenz der Eingangssignale liegen, damit man noch erkennen kann, was „zwischen den Flanken“ steht. Für die zahlenmäßige Darstellung wird man üblicherweise die Abtastfrequenz benutzen, die der Frequenz des schnellsten Kanales entspricht.

Von besonderer Bedeutung ist die Triggerung eines Logikanalysators, d. h., wann er damit beginnt, die Eingangsdaten in seinen Speicher zu schreiben. Was nützt es einem schließlich, wenn die Messung wegen vollem Speicher beendet ist, noch bevor der interessante Bereich angefangen hat. Die Triggerverfahren lassen sich grundsätzlich in zwei Gruppen einteilen.

- Triggerung zu einem bestimmten Zeitpunkt.

- Triggerung bei einer bestimmten Pegelkombination (= Bitmuster) an den Kanälen, „Triggerwort“ (oder -byte) genannt.

Genau genommen ist die zweite Gruppe eine Untergruppe der ersten, denn auch ein Triggerwort repräsentiert einen bestimmten Zeitpunkt.

Ein Beispiel

Der Logikanalysator ist somit ein vielseitiges Meßgerät, das hauptsächlich zur Fehlersuche in Computer- bzw. Peripherieanlagen eingesetzt wird. Er kann aber auch, so zeigt dieses Beispiel, als Hilfsmittel zur Einführung in die digitale Schaltungstechnik dienen.

Als Meßobjekt soll hier ein IC der TTL-Technik untersucht werden, genauer gesagt das IC 74193. Dieses beinhaltet einen Zähler und gehört somit zu der zweiten Gruppe der TTL-Bausteine.

Bei dem Baustein 74193 handelt es sich um einen binären 4 Bit synchronen Vor- bzw. Rückwärtszähler mit programmierbarem Zählbeginn. Die Bedeutung dieser Bezeichnung sei hier nun etwas aufgeschlüsselt:

- 4 Bit bedeutet, daß der Zähler 4 Datenausgänge besitzt, welche einen Zählbereich von 0 – 15 (binär: 0000 – 1111) zulassen.

- Programmierbarer Zählbeginn bedeutet, daß man dem Zähler auf 4 Datenleitungen mitteilen kann, ab welchem Zählwert er beginnen soll zu zählen.

Bild 1 zeigt den Zähler IC 74193 mit den Setzeingängen A bis D und den Zählausgängen QA bis QD. Aus dem Schaltbild

(Bild 2) erkennt man, daß der Zähler einen Rücksetzeingang und je einen separaten Takteingang für Vorwärts- bzw. Rückwärtszählen besitzt.

Es wurde nun folgender Versuchsablauf mit Hilfe des Logikanalysators durchgeführt. Dazu mußte er mit den auf Bild 3 angezeigten Meßbedingungen (Takt: extern, Trigger: intern) programmiert werden. Nun kann der Versuch beginnen und der Logikanalysator wird jede Reaktion des Zähler-ICs 74193 „mitschreiben“. Die nun folgenden Beschreibungen können Sie auf Bild 4 mitverfolgen.

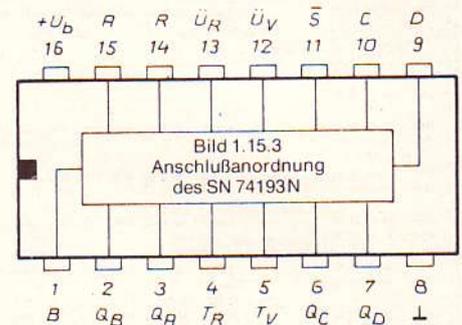


Bild 1

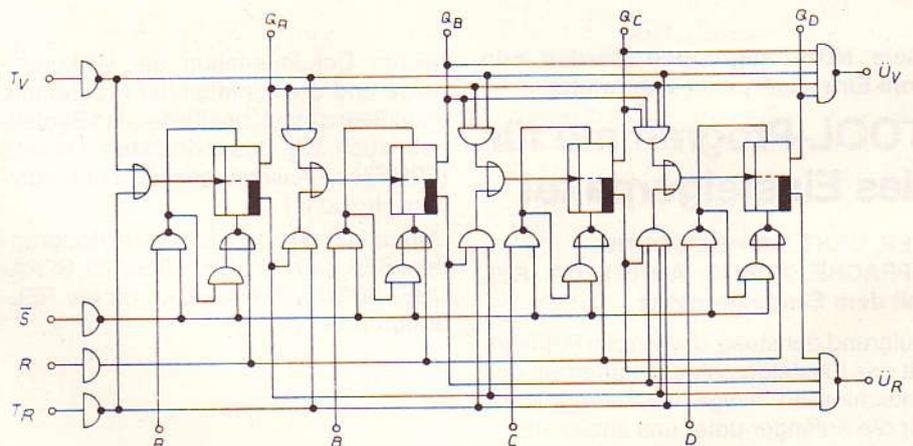


Bild 2

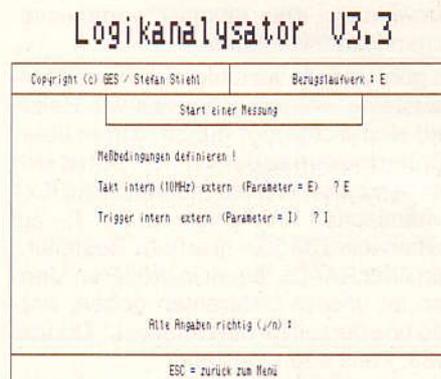
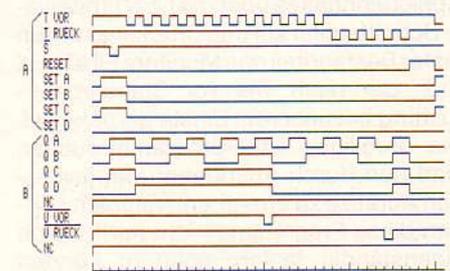


Bild 3



Seite Nr. : 0 Meßfile : AHBT-01.MES
 Takt Nr. : 20 + : Seite vor ? : Hilfe Tabelle
 Byte A : 05 - : Seite zurück ESC : zurück zum Menü
 Byte B : F3

Bild 4

Auf die Takteingänge Tv und Tr wurden abwechselnd ein niederfrequentes Rechtecksignal gegeben. Zuerst jedoch wurde das Signal auf den „Vorwärts“-Takteingang gegeben.

Nun wurde eine binäre 7 (= 0111) auf die Setzeingänge des Zählers gegeben. Durch Aktivieren der Setzleitung (S=0) wird die 7 als Zählbeginn definiert. Durch Takten des Tv-Eingangs beginnt der Zähler

bei 7 aufwärts zu zählen. Bei dem Zählerstand 15 (= 1111) aktiviert der Zählerbaustein den „Vorwärts“-Übertrag (U VOR). Ab Zählerstand 3 beginnt der Zähler rückwärts zu zählen und angekommen beim Zählerstand 0 aktiviert er den „Rückwärts“-Übertrag-Ausgang. Angeworfen beim Zählerstand 15 wurde die Resetleitung aktiviert und die Ausgänge gehen auf 0-Signal.

Wie man an diesem Beispiel erkennen

kann, wird hier der Logikanalysator als Schreiber benutzt und eignet sich somit auch zum Austesten von ICs.

Er ist somit ein universelles Meßgerät, welches nicht nur in Profi-Reparaturwerkstätten gehört, sondern auch ein ausgezeichnetes Lehrmittel, das dem (noch) Laien den Einstieg in die Hardware-Computer-Technologie wesentlich erleichtern kann.

Einfach zwei LOG16 koppeln:

Logikanalysator mit 32 Kanälen

die preiswerte Alternative für viele Anwendungen.

LOG16 hat aufgrund seines einfachen Aufbaues und der ausgefeilten Software schon viele Freunde gefunden. Wir haben nun die Möglichkeit, einfach zwei Baugruppen zu koppeln, um damit 32 Kanäle gleichzeitig zu bearbeiten. Damit eröffnen sich für den Logikanalysator ganz neue Anwendungsgebiete. Übrigens: Die Grenzfrequenz (10 MHz) bleibt natürlich gleich!

Haben Sie schon einmal versucht, mit dem Oszilloskop die Vorgänge auf einem Datenbus zu verfolgen, mehrere digitale Meßpunkte einer Schaltung zu beobachten oder das Timing einer Schnittstelle (z.B. SCSI – Bus) zu analysieren? Der Logikanalysator LOG32 nimmt die logischen Zustände an verschiedenen Meßpunkten mit einer bestimmten Abtastrate auf und erlaubt dann die graphische Darstellung auf dem Bildschirm.

Mit dem LOG32 wird aus dem NDR-Computer in der CP/M 2.2-Version (Z80, Floppy) ein 32-Kanal Logikanalysator mit einer maximalen Abtastfrequenz von 10 MHz.

Der Logikanalysator LOG32 bietet grundlegende Logikanalysefunktionen zu einem erschwinglichen Preis. Er unterstützt Status- bzw. Timinganalyse sowie die Analyse des Z80-Mikroprozessors. Seine Vielseitigkeit macht ihn zum idealen Schulungsgerät, um z. B. Schüler und Studenten mit den zahlreichen Aspekten des digitalen Testens und Messens vertraut zu machen.

Für die Timinganalyse bietet der LOG32 32 Datenkanäle, einen externen Clock-Eingang, eine externe Triggermöglichkeit, einen internen Takt von 10 MHz sowie eine interne Triggermöglichkeit und eine Triggermöglichkeit auf ein 16-Bit-Wort, wobei die Triggerbedingung für 16 der 32 Kanäle als logisch 1 oder logisch 0 spezifiziert werden kann.

Für die Mikroprozessor-Analyse werden jeweils 8 Datenkanäle zu einem Byte zu-

sammengefaßt. Die Disassemblierungs-Darstellung (Bild 1) zeigt die von LOG32 erfaßten Daten, als auch den disassemblierten Instruktionscode für den Mikroprozessor. So kann der Anwender einfach die Assemblierungs-Sprachinstruktionen (für die Software-Analyse) und auch die aktuellen Bitzustände (für die Hardware-Analyse) ablesen. Die Daten bei der Disassemblierungs-Darstellung können hexadezimal oder als Diagramm dargestellt werden. Desweiteren besitzt die 100 KB (!) große Software des LOG32 umfangreiche Möglichkeiten zur Darstellung und Analyse der gemessenen Daten (Bild 2 und 3 zeigen die Hauptmenüs). Es

kann ein hexadezimaler oder binärer Dump der Daten erstellt werden (Bild 4), sowie Teile bzw. ganze Messungen können miteinander verglichen werden. Und es können Bitkombinationen in den Meßdaten gesucht werden (Bild 5). Hier unterstützt der LOG32 auch die Analyse von seriellen Schnittstellen.

Bild 6 zeigt den LOG32 bei der Diagramm-Darstellung, wobei der Anwender gerade eine Messung gestartet hat und nun auf deren Beendigung wartet.

Als Meßspeicherbereich stehen maximal 2048 Meßwerte pro Kanal zur Verfügung, so daß sich bei 32 Kanälen eine Meßkapazität von 8 KByte ergibt.

Der LOG32 benötigt 2 Steckplätze auf der Busplatine und besteht aus 2 LOG16-Bausteinen, die lediglich mit 3 Leitungen untereinander verbunden werden müssen.

Technische Daten:

Steuerbausteine:	4 x Z80A PIO
Meßspeicher:	4 x 2KB RAM (2048 Meßwerte pro Kanal, 4 Byte Abtastbreite = 32 Kanäle)
Anschluß:	zwei Steckplätze direkt auf der Busplatine
Abtastfrequenz:	intern: 10 MHz konstant extern: 0 .. max. 14 MHz
Threshold:	TTL
Eingangsimpedanz:	TTL (=) = 500 k Ohm, (= 10pF)
Triggermöglichkeit:	intern: manuell extern: Strobe Breite:) = 40ns
Stromaufnahme:	0.8 A

Der Umbau

Die beiden LOG16-Baugruppen müssen geringfügig modifiziert werden. Dazu wird nach dem Verfahren des Master und Slave vorgegangen, d. h. eine LOG16-Baugruppe ist die Master-, die andere ist die Slave-Baugruppe. Die LOG32-Software ist auf folgende Portadressen voreingestellt:

Master: Portadresse \$F0

Slave: Portadresse \$E0

Aus diesem Grund müssen Sie auch die Adresse der Baugruppe 2 (Master \$F0, Slave \$E0) ändern. Weiter ist am Slave der IC8, Pin19 mit dem IC8, Pin1 des Masters zu verbinden. Am IC12 des Slaves sind Pin6 und Pin3 mit den selben Pins des Masters zu verbinden. Hier ist vor allem darauf zu achten, daß sich keine abisolierten Kabelstellen mit den Leiter-

bahnen der Baugruppen berühren. Und noch ein Tip zu den Verbindungskabeln: verwenden Sie eine Kupplung zwischen den 3 Kabeln, so daß Sie später auch die beiden Baugruppen als LOG16 verwenden und diese unkomplizierter auf den Bus stecken können (Bild 7).

Neben den Verbindungen sind noch einzelne Unterbrechungen vorzunehmen: Auf dem Master die Leiterbahn zwischen IC1, Pin13 und IC8, Pin1. Auf der Slave-Baugruppe ist lediglich noch der Pin3 und Pin6 des IC12 hochzubiegen. Dazu ziehen Sie das IC aus dem Sockel, biegen beide Pins nach oben und stecken das IC wieder ein. Pin3 und Pin6 hängen jetzt in der Luft. Als letztes sind noch die beiden 40poligen Flachbandkabel mit dem Meßobjekt zu verbinden.

Jetzt müßte die Hardware stehen und es kann eine erste Probemessung durchgeführt werden. Sie booten ganz gewöhnlich unter CP/M 2.2 und starten, nachdem sich die Programmdiskette im Laufwerk befindet, mit dem File LOG32. Nun erscheint ein Menü auf dem Bildschirm. Wählen Sie den Unterpunkt „S“ an und triggern Sie jeweils intern (Eingabe I). Nach Wiedererscheinen des Menüs wählen Sie die Funktion „D“, die die gemessenen Signale auf dem Bildschirm graphisch darstellt. – Und nun viel Spaß mit Ihrem neuen LOG32!

Logikanalysator V4.0

Copyright (c) GES / Stefan Strieth	Bezugslaufwerk : A	32 Kanal
Es stehen Ihnen folgende Funktionen zur Auswahl :		
Zurück zum CP/M (Quit)	Q	
Start einer Messung	S	
Grafische Darstellung einer Messung	D	
Laden einer Messung aus dem Maßbuffer	L	
Einrichtung des Anzeigebuffers	BO	
Hardup der Anzeigebuffers	HO	
Vergleichen einer Messung mit dem Anzeigebuffer	V	
Disassemblieren des Anzeigebuffers	LD	
Finden von 1,2 oder 3 Byte Werten	F	
RETURN = Menü 2		
Bitte geben Sie die Buchstaben der gewünschten Funktion ein :		

Bild 2

Logikanalysator V4.0

Copyright (c) GES / Stefan Strieth	Bezugslaufwerk : A	32 Kanal
Anzahl der zu suchenden Bytes (1..3) ? 3		
Suchen in der Kanalgruppe: 0		
Byte(folge) 13 mal gefunden !!!		
SPRCE = Halt/weiter N = Neustart ESC = Abbruch		

Bild 4

Logikanalysator V4.0

Copyright (c) GES / Stefan Strieth	Bezugslaufwerk : B	32 Kanal
(15) 500F 09	EDX	1V
(16) 5010 3E 03	LD A,03	1>
(18) 5012 CD 4A 52	CALL 524A	1M,R
(21) 5015 1E 00	LD E,00	1
(23) 5017 DD 7E 00	LD A,(IX)	1UR
(26) 501A FE 00	CP 00	1RU
(28) 501C 20 04	JR NZ,5022	1
(30) 501E 1E 80	LD E,80	1
(32) 5020 18 06	JR 5028	1
(34) 5022 FE FD	CP FD	1RU
(36) 5024 20 0A	JR NZ,5030	1
(38) 5026 1E C0	LD E,C0	1B
(40) 5028 DD 7E 02	LD A,(IX+02)	1UR
(42) 502B 08	EX AF,AF	1
(44) 502D DD 23	INC IX	1UR
SPRCE = Halt/weiter N = Neustart ESC = Abbruch		

Bild 1

Logikanalysator V4.0

Copyright (c) GES / Stefan Strieth	Bezugslaufwerk : A	32 Kanal
Es stehen Ihnen folgende Funktionen zur Auswahl :		
Parameter auf Diskette schreiben	PS	
Parameter von Diskette lesen	PL	
Ab speichern einer Messung aus dem Anzeigebuffer	MS	
Laden einer Messung in den Anzeigebuffer	ML	
Beschriften der Kanäle	K	
Disketten Inhaltsverzeichnis	I	
Bezugslaufwerk wechseln	W	
Programmbeschreibung (Hilfe)	?	
RETURN = Menü 1		
Bitte geben Sie die Buchstaben der gewünschten Funktion ein :		

Bild 3

Logikanalysator V4.0

Copyright (c) GES / Stefan Strieth	Bezugslaufwerk : A	32 Kanal
Finden von 1,2 oder 3 Byte Werten		
Anzahl der zu suchenden Bytes (1..3) ? 3	Mr.	Text Mr.
1. zu findendes Byte = 1010000X	1	17
2. zu findendes Byte = 100000001	2	33
3. zu findendes Byte = 1X1010001	3	164
	4	187
	5	299
	6	324
	7	400
	8	471
	9	614
	16	711
SPRCE = Halt/weiter N = Neustart ESC = Abbruch		

Bild 5

MASTER

Anhang E: Layout Lötseite

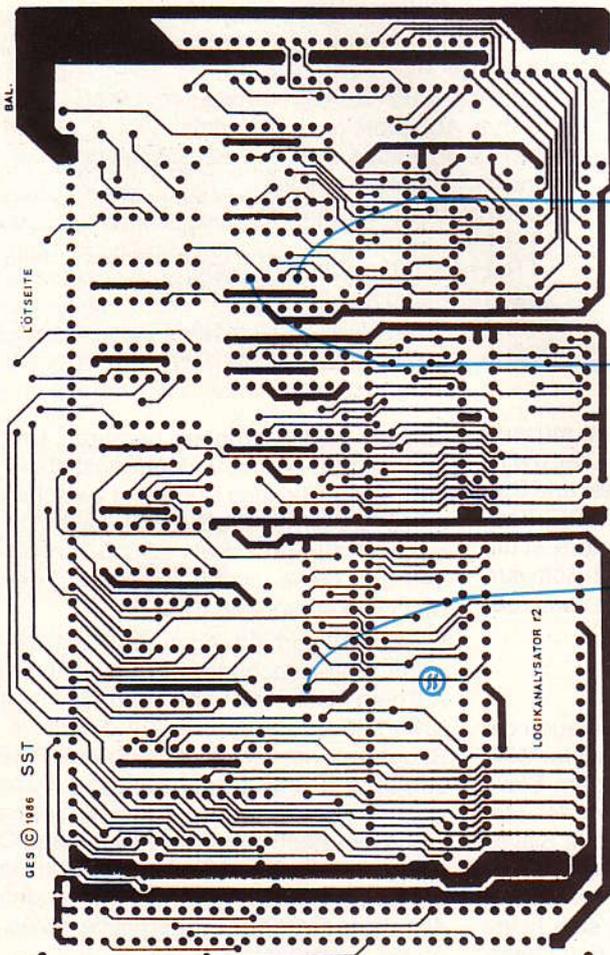


Bild 6

SLAVE

Anhang E: Layout Lötseite

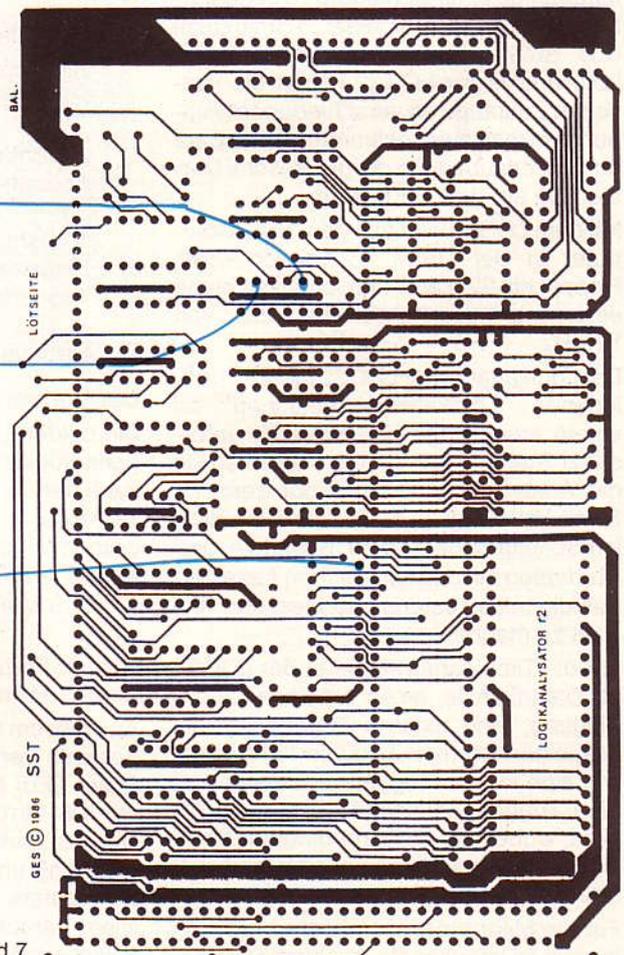


Bild 7

Z80-Grundprogramm und CP/M2.2 gleichzeitig:

Umleitung! – Zwei Programme im gleichen Speicher

von Günter Renner, Schloßbühlstr. 11,
7206 Emmingen-Liptingen

Immer wieder gibt es Menschen, die einfach alles haben wollen. Unter den NDR-Klein-Computer-Besitzern, die einen Z80 ihr eigen nennen, regt sich immer wieder der Wunsch, sowohl das Betriebssystem CP/M als auch das EPROM-residente Grundprogramm von Rolf-Dieter Klein benutzen zu wollen. Beides gleichzeitig zu tun, das bleibt allzuoft ein Wunschtraum. Denn es gibt Konflikte in der Speicherbelegung: CP/M braucht 64 KB RAM, in die man tunlichst keine EPROMs hineinstecken sollte. Und auf der BANKBOOT residiert ab Adresse 0h der unentbehrliche FLOMON. Im RAM sind die ersten 256 Bytes vom System belegt, und die Programme beginnen bei der TPA-Adresse 100h.

Diese Verhältnisse scheinen mit dem RDK-Grundprogramm nicht vereinbar zu sein. Und schon gar nicht mit dem EZASS-Zeilenassembler, der ja ganz besonders für CP/M-Leute sehr attraktiv wäre. Was also tun?

Ein Ausweg könnte darin bestehen, jene Grundprogrammversion zu verwenden, die ab Adresse 2000h nach dem FLOMON auf der BANKBOOT Platz finden kann. In der LOOP wurden bereits früher Programme beschrieben, um EPROM-residente Programme von der BANKBOOT in die RAM-Bank zu laden. Hier soll nun einmal ein anderer Weg aufgezeigt werden, der die Möglichkeit bietet, auch unter CP/M, von eigenen Programmen aus auf das RDK-Grundprogramm zuzugreifen. Der Clou der Sache ist, daß im RAM ab Adresse 3h eine Sprungleiste angelegt wird, die sozusagen um ein paar Ecken herum Unterprogrammaufrufe umlenkt

zum RDK-Grundprogramm, das auf der BANKBOOT ab Adresse 2000h steht. Dadurch kann man von CP/M-Programmen aus die Funktionen desselben benutzen, und es eröffnet sich sogar die Möglichkeit, mit dem EZASS CP/M-gerechte Programme zu schreiben, die bei der TPA-Adresse von 100h beginnen.

In der vorgeschlagenen Lösung liegt das Programm, das man dazu braucht, auf 0EC00h und dürfte damit gerade noch zwischen BIOS und Sektorpuffer Platz finden. In den meisten Fällen hat es dort seine Ruhe und wird nicht etwa gleich überschrieben, sobald man irgendetwas mit CP/M macht.

Seine Funktion besteht darin, zunächst einmal den untersten RAM-Bereich zu retten, damit Daten, die CP/M dort hinterlassen haben könnte, nicht verlorengehen. Dann wird eine Sprungleiste angelegt, die die Grundprogrammaufrufe über eine Zwischenstation, in der die BANKBOOT aktiviert wird, an das Grundprogramm weiterleitet.

Um den ursprünglichen Zustand wieder herzustellen, gibt es ebenfalls einen kleinen Programmteil. Dieser belegt den untersten RAM-Bereich wieder mit seinem ursprünglichen Inhalt, so daß man die Arbeit mit CP/M weiter fortsetzen kann. Ehe Grundprogrammfunktionen aufgerufen werden können, muß erst „initgru“ ausgeführt werden. Das selbe gilt umgekehrt. Ehe wieder mit CP/M gearbeitet wird, muß mit „entfgru“ erst umgeschaltet werden.

Bei Operationen dieser Art tut man sehr gut daran, dem Stack besondere Aufmerksamkeit zukommen zu lassen. Be-

sonders dann, wenn man von CP/M auf Grundprogrammbetrieb wechseln will, muß man damit rechnen, daß er unterhalb der Adresse 8000h liegt. Ein Aktivieren der BANKBOOT hätte selbstverständlich fatale Folgen – man könnte einen Kasten Bier wetten, daß das System abstürzt. Aus diesem Grund legt das Programm „initgru“ einen eigenen Stack an. Es ist deshalb unbedingt notwendig, beim Aufruf die Rücksprungadresse im Register de zu übergeben. Das sieht etwa so aus:

```
ld de,weiter
jp initgru
weiter: ...
```

Nur auf diese Weise ist unabhängig von der Lage des ursprünglichen Stacks gewährleistet, daß der Rechner sicher aus „initgru“ zurückkehren kann.

Zum Schluß noch ein weniger gefährlicher Aspekt. Ein Schönheitsfehler sei nicht verhehlt: Das Grundprogramm enthält feste Symboldefinitionen für die Einsprünge zu den einzelnen Funktionen. In der Version für Adresse 2000h sind die Werte derselben natürlich auch den entsprechenden Einsprungadressen angepaßt. „initgru“ initialisiert aber die Einsprünge ab Adresse 0h, man kann deshalb diese vordefinierten Symbole nicht mehr verwenden.

Die Möglichkeit der Umschaltung auf das Grundprogramm können hier nur in den Grundzügen beschrieben werden. Sie ließe sich aber sicher leicht in CP/M-Programme einbinden. Und vielleicht wird eines Tages gar ein BIOS verfügbar sein, das diese Möglichkeit bietet ...

Anmerkung der Redaktion: LOOP-Leser, jetzt aber los! Wer kann ein solches BIOS erstellen?

```

;program umleitung
;RDK-Grundprogramm auch unter
;cp/m verwenden auf dem
;NDR-Klein-Computer

;(C) Guenter Renner - 2.2.1988

stack:=0eefeh      ;fuer passageren Stack
safe:=0ef00h       ;fuer cp/m-RAM-Inhalt
bankpo:=0c8h       ;1/c-Adresse BANKBOOT
grund:=2003h       ;erster GRU-Aufruf

org 0ec00h

initgru:
jp anfang          ;installieren

entfgru:
jp ende            ;restaurieren

open:
push af
ld a,80h

out (bankpo),a    ;BANKBOOT aktivieren
pop af
ret

close:
push af
xor a
out (bankpo),a    ;RAM aktivieren
pop af
ret

fu1:
call open
call grund
jr close

fu2:
call open
call grund+3
jr close

fu3:
call open
;soviele Funktionen

```

```

call grund+6      ;vorhanden sind      ld hl,0
jr close         ;Return in 'close'    ld de,save      ;sicherer Bereich
; ... usw.      ldir                          ld bc,256      ;fuer cp/m-RAM
;
leiste:         ;Sprungleiste      call open
jp fu1         ;nach oben          ld hl,leiste   ;Sprungleiste
jp fu2         ;gesondert nach     ld de,3h      ;im unteren RAM
jp fu3         ;Grundprogramm-    ld bc,anfang-leiste ;anlegen
jp fu4         ;Funktionen        pop hl
jp fu5         ;wird ab 3h         pop bc
jp fu6         ;im RAM abgelegt     ret
; ... usw.
;
anfang:        ;Stack sichern      ende:
ld (stack),sp  ;und neu anlegen     call open      ;alles wieder
ld sp,stack    ;ist Ruecksprung fuer RET ld hl,save    ;zurueckschreiben
push de        ;fuer cp/m        ld de,0
push bc        ;Stack wiederherstellen ld bc,256
push hl        ;und zurueck      ldir
ret

```

dBASE II schreibt groß

von Ernst van Aaken, Sperberweg 18,
3258 Aerzen 1, Tel.: (05154) 1810

In dBASE II sind die Graphikbefehle mit Flomon ähnlich wie in Hebas.

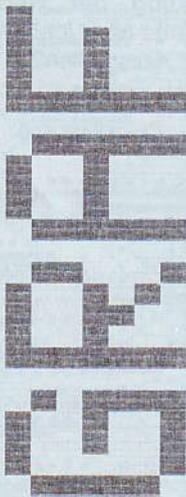
Dieses ist ein kleines Beispielprogramm, um Graphik in eine Programmdatei als Unterprogramm einzubinden. Man sollte es ruhig einmal verändern, um die Graphikmöglichkeiten zu sehen, die auch dBASE II bietet.

```

erase
SET FORMAT TO SCREEN
$ 5,5 SAY "Geben Sie die Breite und Hoehe
fuer einen Grafikbalken ein"
INPUT TO A
INPUT TO B
erase
? chr(27)+chr(27)+"G"
? "M",0,0
? "B3",0,0
? "M",0,50,"B","GRAF"
? "M",1,20
? "R",A,B,1
wait to ANTW
erase
? "ZPO","A"
return

```

Das Ergebnis :



Tips und Tricks bei HEBAS

Nr. 10

von Dr. Hans Hehl

1. WordStar, HEBAS und Plotter für den NDR-Rechner

Mühsam zeichnet man Buchstabe für Buchstabe mit Schablone und Tusche-füller auf Tageslichtprojektor-Folien für Unterrichtszwecke. Mit dem Drucker wird es nichts, da dazu spezielle wischfeste Farbbänder notwendig sind. Ein Plotter kann das viel besser und einfacher, man denke nur an beliebige Schriftneigung (Kursivschrift).

Es gibt einen hervorragenden Bausatz für hohe Ansprüche. Die Firma W. Kopisch in 7712 Blumberg, Buchbergstraße 37, Tel. (07702) 3246 vertreibt einen DIN A3-Selbstbau-Plotter, Typ PL 22/B mit integriertem Z80-Co-Prozessor und intelligenter Betriebssoftware (16 KByte im 27128), der schon in der Zeitschrift c't, Nr. 9/84 beschrieben war. Ich habe diesen Plotter seit einem Jahr in Betrieb und kann ihn nur empfehlen. Welcher Plotter kann schon selbständig interpolieren, Polygonzüge legen, 3-D-Darstellungen drehen u. a. Man kann die Mechanik allein auch erhalten.

Das BASIC-Programm in Bild 1 ermöglicht nun die Ausgabe eines Textes aus einer WordStar-Datei zum Plotter. Natürlich können auch andere Plotter verwendet werden. Die BASIC-Zeilen mit PRINT=2 sind dann dem jeweiligen Befehlsmodus anzupassen. Als Erstes muß die WordStar-Datei als Programm-Datei

(N-Befehl) abgespeichert werden. Ebenso können BASIC-Programme, die vorher im ASCII-Modus abgespeichert wurden (SAVE „Dateiname“,A), ausgegeben werden.

```

10 REM Plotterprogramm von ASCII-Dateien mit HEBAS Verz 0.1.1
20 REM für Plotter Bauatz PL22/B der Fa. Kopisch 07702/3246
30 CLEAR 2000:DIR #1:200
40 CLR
50 PRINT "Plotterprogramm für Textdateien DIN A3"
60 PRINT INPUT "Dateiname mit Extension eingeben":TX
70 IF NOT LOOKUP(LIST 4 "L" + TX) THEN PRINT "nicht vorhanden":GOTO 70
80 CLR
90 REM Bausatz öffnen
100 OPEN#1:"L" + TX + ".T" :REM Daten öffnen
110 ON LOC GOTO 200
120 INPUT LINE#1,LINE1 :REM Daten lesen
130 I = 1
140 GOTO 150
150 PRINT INPUT "Text richtig?":A$ :REM Text prüfen
160 IF A$ = "N" THEN 40
170
180 REM HEBAS: n: Plotterparameter
190 OPTIMIZE "M" :REM Plotter und sein CTRL
200 PRINT#2:"M" :REM Plotterparameter
210 PRINT#2:"B3" :REM langsam schreiben
220 REM Zeichensatz umschalten
230 INPUT "Zeichensatz Deutsch = 1, ASCII = 2":Z$ :REM Z
240 IF Z$ = "1" THEN LFS = "LF"
250 PRINT#2:L$ :REM Zeichensatz
260
270 REM ab selbiger Zeile schreiben
280 REM 2D-Koordinaten: x = xx
290 PRINT#2:INPUT "schreiben ab Zeile 1 oder 11":Y
300 IF Y = 0 THEN Y = 1
310 IF Y = 200 THEN Y = 22 :REM 200er Plattendr FF
320
330 REM Ausgabe auf Plotter
340 FOR Z = 0 TO 1
350 FOR I = 0 TO 1
360 PRINT#1:INPUT#2:THEN INPUT#2:PRINT#1:PRINT#2
370 PRINT#2:PRINT#2
380 NEXT I
390 NEXT Z
400 REM
410 CLOSE#1:END

```

Bild 1: Textausgabe mit dem Plotter

TURBO-PASCAL und die RAM-Floppy

PASCAL initialisiert RAM-Floppy

von Dr. med. H. Kaiser,
Feuerbornstraße 3, 4830 Gütersloh

Das erste Programm ist als Prozedur geschrieben und kann mit der Include-Funktion in beliebige PASCAL-Programme eingebunden werden. In der ersten Inline-Programmzeile wird die Steprate des Floppylaufwerks erhöht. Danach folgt die Initialisierung der RAM-Floppy. Es wird die Routine „bank“ des Flomons verwendet, wobei die ersten 32 K Speicher der Bank 1, auf denen das Directory der RAM-Floppy abgelegt wird, mit dem Wert E5h überschrieben werden.

Das zweite Programm erleichtert die Verwendung der RAM-Floppy, indem es zunächst mit der oben genannten Prozedur die Initialisierung durchführt und verschiedene Dateien, die für eine bestimmte Anwendung benötigt werden, nach E: kopiert. Danach wird Laufwerk E: selektiert und die erste angegebene Datei gestartet. Im hier gezeigten Beispiel werden die für das TURBO-PASCAL benötigten Dateien nach E: kopiert und TURBO.COM wird gestartet. Das Programm hat sich insbesondere auch bei dBASE-Anwendung bewährt, da bei entsprechender Erweiterung des Titel-Arrays neben den dBASE-Systemdateien auch die CMD-Dateien automatisch nach E: geladen werden können. Hierbei gibt es einen deutlichen Geschwindigkeitsvorteil!

```

PROGRAMM TWEL
...
PROGRAM START
...
...

```

Die wichtigsten Grundlagen im Umgang mit Disketten und Laufwerken

CP/M2.2: Was nun?

von Christoph Köhler

Dieser Artikel richtet sich an alle stolzen „Neubesitzer“ eines Floppy-Laufwerkes und des Z80-Betriebssystems CP/M2.2. Es wurde besonders darauf geachtet, die Beschreibung leicht verständlich zu halten, um diese Grundlage allen CP/M-Anwendern zu vermitteln.

Übrigens sind diese Zeilen für alle anderen Betriebssysteme bzw. Prozessoren (CP/M68K, JADOS . . .) ebenfalls grundsätzlich zutreffend.

Im Einzelnen ist beschrieben:

1. Umgang mit Disketten
2. Die Wahl: 3 1/2" oder 5 1/4"
3. Wie legt man eine Sicherheitskopie an
 - 3.1 neue Diskette formatieren
 - 3.2 kopieren mit einem oder zwei Laufwerken
 - 3.3 übertragen der Systemspuren auf die neue Diskette
4. Diskettenwechsel
5. Der Schreibschutz
6. Die RAM-Floppy
7. Tips zur Dokumentation

1. Der Umgang mit Disketten

Die wichtigsten Regeln zum Umgang mit Disketten sind:

- Disketten nie ohne Hülle ablegen, da kleinste Staubkörnchen diese zerstören können.
- Niemals die braune Diskette im Schreib-Lese-Schlitz selbst angreifen, da das wohl der „Abschieds-Händedruck“ wäre.
- Vermeiden Sie ein Verbiegen, Knicken und hohe Temperaturen für Ihre Disketten.
- Legen Sie diese nicht in die Nähe eines Magnetes bzw. eines magnetischen Feldes (Monitor, eiserner Schraubendreher, Telefonleitung) da die Disketten sehr empfindlich sind.
- Lösen Sie beim Aus- und Einschalten Ihres Computers den Verriegelungshebel des Laufwerkes, da sonst ein ungewollter Zugriff des Computers auf die Diskette zu Störungen führen könnte.

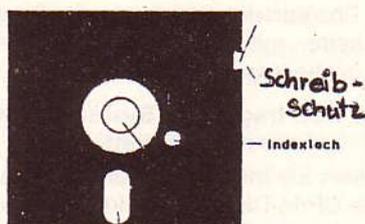


Bild 1

2. Die Wahl: Diskettenformat 3 1/2" oder 5 1/4"?

Bei den mechanisch sehr stabilen 3 1/2" (= Zoll = 2,54 cm) Disketten müssen Sie den großen Vorzug der bequemen Handhabung bei gleicher Kapazität (ca. 800 KByte bei 80 Spuren) mit einem höheren Preis gegenüber der 5 1/4"-Diskette bezahlen.

Sie sollten von Anfang an Laufwerke mit 80 Spuren verwenden, da Sie mehr Platz für Programme auf der Diskette haben und die Bearbeitung derselben wesentlich schneller als mit 40 Spuren erfolgt.

Es können auch 8"-Laufwerke eingesetzt werden.

Praxis: Da man sich oft eine Sicherheitskopie anlegt, haben sich derzeit die meisten NDR-Anwender für das 5 1/4"-Laufwerk entschieden.

3. Wie legt man eine Sicherheitskopie an?

Bevor eine neue Diskette vom Computer beschrieben und gelesen werden kann, muß diese speziell formatiert werden (UFORM.COM). Anschließend kann diese mit einem Kopierprogramm (z.B. PIP.COM, COPY80V.COM) beschrieben werden.

Da alle Programme von uns ohne die CP/M-Systemspuren ausgeliefert werden (muß mit CP/M2.2 erworben werden), müssen diese noch auf die neue Diskette übertragen werden (SYSGEN-80.COM). Die Systemspuren von CP/M2.2 ermöglichen ein selbständiges Booten von der Diskette.

3.1 Neue Diskette formatieren

Zunächst einmal müssen Sie eine neue **Diskette formatieren**. Auf Ihrer CP/M-Diskette ist zu diesem Zweck bereits ein Formatierprogramm vorhanden. Dies können Sie aufrufen, indem Sie „UFORM“ eingeben. Nach dem Druck auf die „RETURN“-Taste erscheint folgendes Menü auf dem Bildschirm: Eingabe: RETURN-Taste

Universal Formatierer V1.2,
Rolf-Dieter Klein (C) 1984, München

- Bitte Systemdiskette herausnehmen und zu formatierende Diskette einlegen -

Minilaufwerke (5 1/4")	= 1
Maxilaufwerke (8")	= 2
Std 8" SD SD 77 Spur	= 3
EMCA 70 DD SD 40 Spur	= 4
NDR- DD DS 80 Spur	= 5
ENDE	= 6

Jetzt: Systemdiskette herausnehmen und unformatierte Diskette in Laufwerk A einlegen!

Hier tippen Sie eine „5“ ein. Jetzt erscheint ein weiteres Menü auf dem Bildschirm: Eingabe: 5

- FLO2-Maxi/Mini (0COH) = 1
- FLO1-Maxi (40H) = 2
- FLO1-Mini (30H) = 3

Nach der Eingabe der Ziffer „1“ erscheint ein drittes Menü auf dem Monitor:

- Laufwerk A = 1
- Laufwerk B = 2
- Laufwerk C = 3
- Laufwerk D = 4
- Laufwerk A Rückseite A = 5
- Laufwerk B Rückseite B = 6
- Laufwerk C Rückseite C = 7
- Laufwerk D Rückseite D = 8
- NEUSTART = 9
- ENDE = 0

Hier wählen Sie die „1“ an. Jetzt erscheint noch eine Zusammenfassung der eingegebenen Daten. Diese sollte wie folgt aussehen.

- Mini-Laufwerk FLO2 MFM
- Anzahl Spuren = 80
 - Anzahl Sektoren = 5
 - Bytes/pro Sektor = 1024
 - Laufwerk A

beide Seiten werden formatiert
Seite 1 mit SS0, Seite 0 normal
Speicherkapazität: 800 KBytes

ACHTUNG Diskette wird formatiert
Start = „J“

Jetzt noch ein „J“ eingeben und der Formatiervorgang beginnt.

Nach dem Formatieren erscheint wieder folgendes Menü:

- Laufwerk A = 1
- Laufwerk B = 2
- Laufwerk C = 3
- Laufwerk D = 4
- Laufwerk A Rückseite A = 5
- Laufwerk B Rückseite B = 6
- Laufwerk C Rückseite C = 7
- Laufwerk D Rückseite D = 8
- NEUSTART = 9
- ENDE = 0

Diesmal drücken Sie eine „0“ für ENDE. Es erscheint folgende Schrift auf dem Monitor:

Systemdiskette in Laufwerk A einlegen, dann CR (carriage return) drücken.

Legen Sie die Systemdiskette wieder ein!

Drücken Sie die „RETURN“-Taste. Der Formatiervorgang ist damit beendet. Ihre Diskette ist jetzt zum Aufnehmen von Dateien und Programmen bereit.

3.2 Kopieren einer Diskette mit einem oder zwei Laufwerken

– Ein Laufwerk

Entweder kopieren Sie alle Programme einzeln in die RAM-Floppy (siehe letzter Abschnitt) und dann auf die neue Diskette oder Sie arbeiten mit dem DDT-Programm entsprechend folgender Beschreibung:

Sie werden auf Ihrer Systemdiskette ein Programm „DDT.COM“ finden. Dieses Programm rufen Sie auf, indem Sie die Buchstaben „DDT“ und natürlich „RETURN“ eingeben. Auf dem Bildschirm erscheint jetzt nebenstehender Text:

DDT VERS 2.2

Tippen Sie den Namen des Programmes ein, das Sie kopieren wollen. Vor dem Programmnamen muß aber noch „I“ stehen. Danach geben Sie „RETURN“ ein.

–IPROG.COM

Jetzt geben Sie ein „R“ ein. Auf dem Bildschirm erscheint jetzt nebenstehender Text.

XXXX XXXX steht für die dort erscheinenden Zahlen.

Die Länge des Programmes wird vom DDT in hexadezimaler Darstellung angegeben (nur die vorderen beiden Zahlen sind interessant). Um aber die Programme mit dem DDT wieder abzuspeichern, müssen Sie diese Werte in dezimale Werte umrechnen.

Zum Umrechnen müssen Sie die vom DDT ausgegebenen Werte von der hexadezimalen Form in die dezimale umwandeln. Hierbei können Sie nach folgender Methode vorgehen:

Multiplizieren Sie die erste der beiden Ziffern mit 16 und addieren Sie die zweite zu dieser Zahl. Was die einzelnen Zahlen und Buchstaben für einen Wert haben, entnehmen Sie bitte der nebenstehenden Tabelle.

Zahl	Dezimal	Sedezimal
Null	0	0
Eins	1	1
Zwei	2	2
Drei	3	3
Vier	4	4
Fünf	5	5
Sechs	6	6
Sieben	7	7
Acht	8	8
Neun	9	9
Zehn	10	A
Elf	11	B
Zwölf	12	C
Dreizehn	13	D
Vierzehn	14	E
Fünfzehn	15	F

Tippen Sie jetzt „G0“ (Ge und dann Null) ein. Es erscheint nebenstehendes Zeichen.

Nun wechseln Sie die Disketten.

Jetzt geben Sie „Control-C“ ein (Control- und c-Taste gleichzeitig drücken). Wieder „RETURN“ eingeben. Nun tippen Sie „SAVE XX Prog.com“ ein. XX ist die Länge des zu kopierenden Programms und Prog.com steht für den Programmnamen.

– Zwei Laufwerke

Die Erstellung einer Sicherheitskopie mit zwei Laufwerken ist sehr einfach.

Die schnellste Möglichkeit ist die Verwendung des Befehles COPY80V (wird mit CP/M2.2 geliefert).

Dabei muß nach dem Aufruf mit A)COPY80V entsprechend den Anweisungen gehandelt werden.

Bei der Verwendung des PIP-COM Kommandos muß wie in allen CP/M-Dokumentationen beschrieben, vorgefahren werden.

Z.B. Im LW A PIP aufrufen, dann Quelle in LW A, Ziel in LW B, dann B:=A:*ÄVÜ eingeben. (V steht für Verify bzw. Prüfung).

Tip: Vor dem Kopieren die Quell-Diskette mit Schreibschutz versehen (siehe Kapitel 5).

3.3 Übertragen der Systemspuren auf die neue Diskette

Damit Sie mit Ihrer neuen Diskette ohne die CP/M-Disk booten können, müssen von Ihnen noch die Systemspuren übertragen werden. Dazu verwenden Sie das

–R
NEXT PC
XXXX XXXX

Programm SYSGEN80.COM von der CP/M 2.2-Diskette.

Legen Sie die CP/M-Disk in LW A und ggf. die neue Diskette in LW B. Mit nur einem Laufwerk muß die Diskette später gewechselt werden.

Der Aufruf erfolgt mit: A)SYSGEN80.

Es erscheint nun folgender Text auf dem Bildschirm:

SYSGEN VER 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP) A SOURCE ON A,
THEN TYPE RETURN
FUNKTION COMPLETE

Eingaben:
"A"
"RETURN"
Systemdiskette herausnehmen; formatierte Diskette einlegen

DESTINATION DRIVE NAME (OR RETURN TO REBOOT) A FUNKTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)

Jetzt ist Ihre Backup-Diskette mit den Systemspuren versehen.

4. Diskettenwechsel

Achtung! Beim Betrieb mit CP/M2.2 muß jede neue Diskette vor einem Schreibzugriff „angemeldet“ werden. Dies erfolgt mit der Tastenkombination „CTRL“ und „C“ (Warmstart).

Beachten Sie, daß eine Fehlermeldung (falls Sie das „Anmelden“ vergessen haben) erst dann auftritt, wenn das Programm versucht, etwas abzuspeichern. Man sollte sich angewöhnen, grundsätzlich nach jedem Diskettenwechsel einen Warmstart durchzuführen. (Beim CP/M68K und beim JADOS ist das nicht notwendig.)

5. Der Schreibschutz

Um eine Diskette vor versehentlichem Überschreiben von Programmen zu bewahren, ist diese mit einem Schreibschutz versehen. Bei einer 3 1/2"-Diskette kann das zuständige Fenster mit einer Schiebemechanik geschlossen werden. Bei den 5 1/4"- und 8"-Disketten muß dies durch Aufbringen eines Klebers geschehen. Das Aktivieren des Schreibschutzes ist bei den verschiedenen Formaten unterschiedlich:

	Fenster bei gesetztem Schreibschutz
3 1/2"	ZU
5 1/4"	ZU
8"	auf

Sobald Sie irgendwelche Daten auf die Disketten schreiben wollen, muß natürlich der Schreibschutz entfernt sein!

6. Die RAM-Floppy für NDR-Computer

Wie in dem Handbuch der RAM64/256-Baugruppe beschrieben, kann mit dem NDR-Rechner unter CP/M2.2 eine RAM-Floppy betrieben werden. Dabei verhält sich diese RAM-Floppy wie ein normales Laufwerk und kann mit dem Laufwerk-

kennbuchstaben "E": angesprochen werden.

Wie schon der Name „RAM“ verrät, sind alle Programme nur im Arbeitsspeicher. Der Betrieb der Programme erfolgt wesentlich schneller, hat aber den Nachteil, daß diese vor dem Ausschalten auf eine Diskette „gerettet“ werden müssen.

Sie benötigen beim NDR-Computer einen Arbeitsspeicher von mindestens 256 KByte (1 voll bestückte RAM64/256 oder eine ROA256/1M).

Das Erstellen einer Sicherheitskopie mit einem Laufwerk und der RAM-Floppy geschieht folgendermaßen:

Zuerst wird das Programm PIP.COM in das LW E: kopiert. Von dort können einzelne Programme von der Quellediskette in die RAM-Floppy kopiert werden, bis diese voll ist.

Jetzt wird die neue Zieldiskette (mit Systemspuren) in das LW A eingelegt und ein Warmstart durchgeführt (CTRLC). Mit dem PIP auf LW E können die Pro-

gramme auf die Zieldiskette kopiert werden.

Das wird dann solange durchgeführt, bis sich alle Programme auf der Zieldiskette befinden.

7. Tips zur Dokumentation

Auf fast allen Disketten wird auch eine Beschreibung mitgeliefert. Diese können entweder auf dem Bildschirm angesehen oder auf den Drucker ausgegeben werden.

Mit dem CP/M-Befehl "DIR" können Sie sich das Verzeichnis der Diskette anschauen. Die Ausgabe einer Textdatei erfolgt mit dem Befehl "TYPE". Wenn Sie zuvor die Tastenkombination "CTRL" und "P" eingeben, wird die Textausgabe auf den Drucker umgeleitet. Die Beschreibungen können verschiedene Bezeichnungen tragen:

Bei GES: LIESMICH
sonst oft: README
oder READ.ME

oder Name.TXT (Name kann der jeweilige Programmname sein)

oder Name.DOC

Ihre Eingabe z.B.: A)TYPE LIESMICH

Sie können die Ausgabe mit den Tasten "CTRL S" anhalten und je nach Rechner mit "CTRL S" oder "CTRL Q" weiter ausgeben lassen. Mit den Tasten "CTRL C" wird die Ausgabe abgebrochen.

Beim Einstieg in das Betriebssystem CP/M2.2 brauchen Sie auch ein gut lesbares Begleitbuch.

Für Einsteiger empfehlen wir die Bücher

- „CP/M kompakt (Plate)“ Best.-Nr. 10451 und

- „CP/M Handbuch (Zaks)“ Best.-Nr. 10590

Später empfehlen wir Ihnen als Fortgeschrittenen

- „Vom Umgang mit CP/M (POL)“ Best.-Nr. 10586 und

- „Betriebssystem CP/M (Plate)“ Best.-Nr. 10580.

Nun ist es soweit: Nach knapp einem Jahr Arbeit ist die neue Version des 68008-Grundprogramms fast fertig. Zuerst möchten wir uns für die Vorschläge bedanken, die wir von einigen LOOP-Lesern erhalten haben. Viele dieser Vorschläge konnten wir verwirklichen.

Der Quelltext hat unsere ersten Vorstellungen durch die vielen Wünsche weit übertroffen (es sind jetzt über 550 KByte), und auch der 68K-Code nimmt einen Raum von knapp 50 KByte ein. Dadurch ist es notwendig geworden, auch das 68008-Grundprogramm auf 8 Eproms zu betreiben.

Zum jetzigen Leistungsumfang gehören z.B.:

- Ein einfacher Dis-Assembler für den Einzelschrittmodus, der sich automatisch einschaltet, wenn in der DEBUG-Tabelle nichts gefunden wird.
- Ein Editor nach dem RDK-Muster, der Texte über 64 KByte verarbeiten kann. Er ist schneller geworden und hat einige Zusatzfunktionen bekommen (Block auf Diskette speichern, Block von Disk laden, Sicherheitskopie, Linker Rand, Cursor mit Maus steuerbar, Diskfunktionen unter Jados).
- Die Menüsteuerung ist umschaltbar (Normal / Alles auf einer Seite).

680xx Grundprogramm jetzt unter JADOS:

Das neue Grundprogramm ist bald fertig – von Ralf Dombrowski

- Eine kleine Druckersteuerung mit änderbaren Druckbefehlen wurde installiert.
 - Die meisten Menüpunkte sind verbessert worden.
 - Die serielle Schnittstelle wird besser unterstützt (z. B. Umschaltung CO2).
 - Die Hardcopy-Maus-Karte wird durch 16 (!) über Trap aufrufbare Unterprogramme unterstützt.
 - Eine neue GDP ist auch mit eingeplant (Hardware-Scroll, Flächen Füllen, usw.).
 - Es gibt ein Graphik-Paket für GDP und COL256 gemeinsam (Kreise, Linien, Ellipsen, Rechteck, Quadrat, Textausgabe).
 - Das Grundprogramm insgesamt oder einzelne Menüpunkte können über Trap aufgerufen werden.
 - Alle Trapvektoren sind ins Ram geführt.
 - Alle Unterprogramme wurden optimiert.
 - Einbau der Smart-Watch als Alternative zur Uhrenbaugruppe.
 - Über DIL-Schalter der KEY-Karte kann ein einmaliger Sprung ins BOOT-Menü beim Einschalten angewählt werden.
 - Die Floppy-Routine kann jetzt alle Funktionen des Floppy-Controllers ausführen. Eine Umlenkung der Floppy auf die serielle Karte ist möglich.
 - Der Drucker kann seriell betrieben werden.
 - Die Prommer-Routine beherrscht auch den schnellen Programmieralgorithmus.
- Diese Leistungsübersicht zeigt wohl sehr gut, daß das neue Grundprogramm wirklich einiges zu bieten hat.
- Es wird auf der Hannover Messe CeBit 88 gezeigt und ist ab Mai lieferbar. Es ist ein eigenständiges Produkt, fällt also nicht unter die UPDATE-Bedingungen.
- NEU: Die komplette Quelle ist (unter JADOS) auf Diskette verfügbar.

```

* KURZE ERLÄUTERUNGEN:
* FUER ANFANGER MIT DEM NDR-KLEIN-COMPUTER MIT 68008 CPU
* MIT GRUNDPROGRAMM VERSION 4.3
* JEWEILS 6 WERTE IN WORTBREITE WERDEN PRO ZEILE AUFGELISTET. ZWÜLF
* SOLCHE ZEILEN ENTHALT DIE TABELLE. NACHEINANDER SOLLTEN DIE WERTE
* GEHOLT UND WEITER VERARBEITET WERDEN, DAZU EIGNET SICH DER BEFEHL
* MOVE.X Y(A,X). ZIEL FUER Y STEHT DIE ADRESSDISTANZ, BESCHREIBUNG DES
* BEFEHLS: BRINGE DEN INHALT DER ZELLE, DEREN ADRESSE SICH AUS DEM
* INHALT DES ADRESSREGISTERS PLUS DER ADRESSDISTANZ ERGIBT, IN DIE
* EFFEKTIVE ZIELADRESSE. LETZTERE KANN EIN REGISTER ODER EINE ADRESSE
* SEIN.
* MIT EINZELSCHRITT KANN MAN GENAU BEOBACHTEN, WAS IN DEM PROGRAMM
* PASSIERT. DIE WERTE WERDEN SEDEZIMAL ANGEZEIGT, MAN KANN DIESE MIT REGISTER
* D4 UND MIT DEN WERTEN IM LISTING DER TABELLE NAME1.
* (C) ROLF-DIETER KLEIN -QUELLE: AUSFLUG NACH SIMLAND. IN 68008 AUFBAUPRO-
* GRAMME. FRANZIS-VERLAG. MODIFIZIERT 1985 OEHRLÉ
* WILFRIED OEHRLÉ, PHILOSOPHENWEG 19, 7400 TUBINGEN
* PROGRAMM IN DEN EDITOR LADEN (BZW. ABTIPPEN) AB ADR. $9000
* IM MENUE DEBUG INFO AN AUSWAHLEN. DANN MIT RDK - ASSEMBLER ASSEMBLIEREN.
* NUN EINZELSCHRITTMENUE ANWAHLEN UND DAS WORT START EINTIPPEN.
* NEUEN BEFEHL ABARBEITEN = CR TASTE DRÜCKEN
* **** PROGRAMMIEREN IN ASSEMBLER: TABELLENVERWERTUNG

ORG $A000 * AB DIESER ADRESSE ABLAGE DES PROGRAMMCODES
NAME1: * LISTE DER WERTE IN WORTBREITE, DIE GEHOLT UND ANGEZEIGT WERDEN
DC.W $66,$2,$3,$4,$5,$6
DC.W $4,$8,$9,$8,$7,$5
DC.W $1,$2,$2,$2,$5,$6
DC.W $6,$5,$4,$5,$2,$1
DC.W $9,$8,$7,$1,$5,$4
DC.W $4,$5,$6,$9,$8,$7
DC.W $9,$2,$5,$1,$6,$5
DC.W $98,$87,$65,$32,$12,$65
DC.W $8,$9,$8,$7,$4,$5
DC.W $9,$6,$5,$8,$3,$2
DC.W $5,$8,$9,$5,$4,$3
DC.W $9,$8,$5,$6,$3,$4

anzahl equ $c * ANZAHL DER UNTER NAME1 AUFGELISTETEN WERTZEILEN =12
X1 EQU 0 * ADRESSDISTANZ DER WERTE IN DEN ZEILEN
Y1 EQU 2

Z1 EQU 4
X2 EQU 6
Y2 EQU 8
Z2 EQU 10

START:***** HIER PROGRAMMSTART *****
CLR.L D4 * D4 LOESCHEN
CLR.L D5 * D5 LOESCHEN
LEA NAME1,A1 * ADRESSE DER TABELLE NACH A1
MOVE #ANZAHL-1,D5 * ZAEHLER MIT D5 ERRICHTEN. 12 ZEILEN ABARBEITEN.
SCHLEIFE1:
MOVE X1(A1),D4 * 1. WERT AUS ZEILE HOLEN
JSR ANZEIGE * VERGL. D4 MIT ANZEIGE
MOVE Y1(A1),D4 * ZWEITEN WERT AUS ZEILE HOLEN
JSR ANZEIGE * VERGL. D4 MIT ANZEIGE
MOVE Z1(A1),D4 * 3. WERT AUS ZEILE
JSR ANZEIGE
MOVE X2(A1),D4 * 4. WERT AUS ZEILE
JSR ANZEIGE
MOVE Y2(A1),D4 * 5. WERT AUS ZEILE
JSR ANZEIGE * NACHDEM DIE WERTE AUS DER JEWEILIGEN ZEILE GEHOLT SIND,
MOVE Z2(A1),D4 * MUSS ZUM HOLEN DER WERTE AUS DER NAECHSTEN ZEILE ZUM
JSR ANZEIGE * ADRESSREGISTER A0 DER WERT $ C ADDIERT WERDEN. DADURCH
ADDA.L #5C,A1 * WIRD DIE JEWEILIGE ANFANGSADRESSE DER NAECHSTEN TABELLENZEILE
DBRA D5,SCHLEIFE1 * BERECHNET UND VON DORT EIN WERT GEHOLT.
***** ENDE DES HAUPTPROGRAMMS *****
***** UNTERPROGRAMM ZAHLN AUF BILDSCHIRM ANZEIGEN *****
ANZEIGE:
LEA ANZEIGEBUFFER,A0 * BUFFER ADR NACH A0
MOVE.W D4,D0 * WERT VON D4 NACH D0.
JSR $PRINTX * RDK GRUNDPROGRAMMFUNKTION. NACHZULESEN IM HANDBUCH
LEA ANZEIGEBUFFER,A0
MOVE #22,D0
MOVE #30,D1
MOVE #100,D2
JSR $WRITE * RDK GRUNDPROGRAMMFUNKTION AUSGABE AUF BILDSCHIRM
RTS
ANZEIGEBUFFER: DS.B 20 * PLATZ RESERVIEREN FUER ANZUZEIGENDE ZAHLN
DS 0
END

```

Der NDR-Computer lernt sprechen:

Sprachausgabe für den NDR 680xx-Computer

auch als Subsystem mit V24-Anschluß erhältlich

Seit der Entwicklung von Rechenanlagen war man bestrebt, die Simulation der menschlichen Sprache nachzuvollziehen.

Waren die hardwaremäßigen Voraussetzungen bereits vorhanden, so hinkte die Entwicklung von professioneller Software wie immer etwas nach. Dies war auch die Ursache, weshalb die SPRACHE-Baugruppe für den NDR-Computer etwas vernachlässigt wurde.

Dies ändert sich spätestens mit der Erscheinung zweier wichtiger Programme, die nun für den NDR erhältlich sind.

Zum erstenmal ist es jetzt möglich, Texte, die im RDK-Editor stehen, durch die Sprache-Baugruppe aussprechen zu lassen.

Das Programm VOICE2, das die eben beschriebene Aufgabe erledigt, läuft unter dem Betriebssystem JADOS für alle 68000er CPUs.

Für Anwender, die bisher keine Laufwerke besitzen und die 68008 einsetzen, wird eine Epromversion angeboten.

VOICE2 kann das komplette Alphabet und die Zahlen 0 – 9 mit leicht amerikanischer Artikulation aussprechen.

Großbuchstaben, die alleine stehen oder aufeinander folgen, werden buchstabiert. Umlaute werden ebenfalls toleriert.

Sonderzeichen bzw. Satzzeichen werden ignoriert und nicht ausgesprochen.

Im gleichen Atemzug entwickelte der Autor Jürgen Waffner ein weiteres Programm, an dem sich nun auch Insider und Umsteiger erfreuen können.

ETSS2 unterstützt ein für sich eigenständiges Sprachesubsystem, welches aus Baugruppen des NDR-Computers besteht.

Die Aufgabe dieses Sprachesubsystems besteht darin, Texte, die über die eigene serielle Schnittstelle empfangen werden, in einem internen Pufferspeicher abzulegen und beim Eintreffen eines bestimmten Abschlußzeichens auszusprechen. Die mögliche Länge des Textes wird nur von dem zur Verfügung stehenden Speicher begrenzt.

Möglichkeiten der Sprachausgabe, Vokabular und Akzent sind identisch mit VOICE2.

Die Konfiguration kann sich vorteilhafterweise auf ein Minimum beschränken. So sind minimal folgende Baugruppen erfor-

derlich: CPU68008, ROA64, BUS2, SER SPRACHE und eine Stromversorgung für + 12 und + 5 Volt.

Das Eprom, das sinngemäß nur für die 68008 angepaßt ist, sitzt auf einer ROA64 auf Adresse 0, um nach dem Einschalten der Spannungsversorgung sofort aktiv zu sein.

Die serielle Schnittstelle des Subsystems arbeitet hierbei mit Standardwerten (9000 Baud 8 Datenbits 1 Stopbit keine Parität), die vollständig von der Epromsoftware initialisiert werden.

Der besondere Effekt dieses so aufgebauten Systems besteht darin, das es sich wie ein Drucker verhält und auch als solcher behandelt werden kann: Mit dem kleinen Unterschied, daß die Daten nicht auf Papier gedruckt, sondern in phoneometrische Laute umgeformt werden.

ETSS2 bietet eine attraktive Alternative, Sprachausgabe mit einem Computer zu realisieren. Besonderen Wert wurde bei dieser Entwicklung auf Kompatibilität mit anderen Computersystemen gelegt.

Gelöst wurde dieses Problem mit Hilfe der Eigenständigkeit des Subsystems durch Kopplung über eine serielle Schnittstelle (V24).

Text-Editor nun auch in-Farbe:

Der COL-EDIT ist da!

von Kei Thomsen

Endlich ist er fertig, der Editor für die COL 256. Einige werden sich jetzt fragen: „Ein Editor auf einer solchen Grafikkarte, mit nur 256 x 256 Punkten?“ Ja, aber nicht auf 256 x 256, sondern in IBM-Manier mit 640 x 256 Punkten, ohne sonderliche Hardwareänderungen auf der COL! Erstaunt? Ja. Gleich der nächste Punkt, um den Kritikern der COL den Wind aus den Segeln zu nehmen: Der Bildspeicher ist immerhin 80 KByte groß, den es jedes Mal aufzubauen gilt. Aber wen stört denn das? Doch keinen 68008 oder 68000 und erst recht nicht einen 68020. Durch sehr schnelle Zeichenausgabe und optimierte Löschroutinen ist der Editor sehr schnell geworden. Durch die Möglichkeit von 2 Bildseiten sieht man auch den Bildaufbau nicht mehr. Natürlich besitzt der Editor eine vernünftige Speicherverwaltung, um so wenig wie möglich den Speicher verschieben zu müssen. Es kann Text mit jeder beliebigen Länge editiert werden, es sei denn, der Speicher reicht nicht aus, den Text zu fassen. Sprünge vom Anfang des Textes an das Ende und umgekehrt dauern bei 100 KByte Text selbst beim 68008 maximal $\frac{1}{2}$ Sekunde. Der Bildschirm faßt 80 Zeichen je Zeile und hat 20 Zeilen. Die Zeichen haben eine Größe von 8 x 12 Punkten. Dadurch wirken die Zeichen zwar sehr groß, aber dafür läßt sich der Text erheblich besser lesen als auf der GDP mit nur 5 x 8 Punkten. Lange Tests haben gezeigt, daß weiße Schrift auf dunkelbraunem Hintergrund den besten Kontrast besitzt und ein sehr angenehmes Lesen und Schreiben ermöglicht. Als Cursor wurde ein Underline-Cursor benutzt. Dieser hat den Vorteil, daß er nicht dauernd blinkt und das Zeichen einfach unterstreicht. Zum schnellen Auffinden ist der Cursor rot (nicht politisch gemeint). Eine Statuszeile ist selbstverständlich auch vorhanden. Diese ist ganz oben auf dem Schirm und ist grau mit schwarzer Schrift, um nicht zu sehr aufzufallen. Dort wird angezeigt, welcher Zeichensatz gewählt ist, ob der Einfügemodus ein- oder ausgeschaltet ist, die Zeile und die Spalte der Cursorposition sowie die Textlänge und der Name der bearbeiteten Datei.

Bei Fehlern sowie bei Extrafunktionen, die nicht direkt mit dem Text zu tun haben, werden Fenster, die sogenannten Windows, eingeblendet. Dabei wird das Window grau mit schwarzer Schrift dargestellt. Dieses kommt bei Inhaltsverzeichnis, Hilfe, Fehlern, Suchen, Ersetzen, usw. zur Anwendung. Die Steuercodes sind WordStar ähnlich (ich hasse das Wort kompatibel) und noch weitere neue

Steuerkommandos sind vorhanden. Zur Zeit ist der Editor für JADOS geschrieben, wird aber demnächst auch auf CP/M 68K angepaßt.

Geschwindigkeit:

10 mal Blättern = 100 Zeilen:

68008 = 2 sec.

68000 = 1,2 sec.

68020 = 0,6 sec.

Zusammenfassung der wichtigsten Bestandteile:

- Schnelles Bearbeiten von Texten und Programmen.
- Auch große Texte einfach zu bearbeiten.
- 80 Zeichen je Zeile.
- 20 Zeilen je Seite.
- Große Buchstaben (8 x 12 Punkte) mit echten Unterlängen.
- Weiße Schrift auf dunkelbraunem Untergrund.
- Roter „underline-Cursor“.
- Graue Statuszeile mit schwarzer Schrift.
- Anzeige des Status, Spalte und Zeile des Cursors.
- Anzeige der Textlänge und der bearbeiteten Datei.
- Windows bei Fehlern, Datei- und Suchfunktionen.
- Sehr schneller Bildaufbau.
- Volle Nutzung des COL-Speichers.
- Unterstützung durch JADOS.
- Erzeugen von Backup-Dateien.
- Anzeigen des Inhaltsverzeichnisses der Diskette.
- Laden und Speichern von Texten.
- Seitenweises Blättern.
- Blockfunktionen auf Diskette.

Es wird einigen Besitzern der COL aufgefallen sein, daß der Editor 640 x 256 Punkte benutzt. Der dazu passende COL-Treiber ist schon lange fertig, nur die Anleitung ist noch nicht mal angefangen.

Anmerkung der Redaktion: Kei Thomsen hat inzwischen die Jeans mit der Uniform vertauscht und nun hoffentlich mehr Zeit, die Anleitung zu schreiben . . .

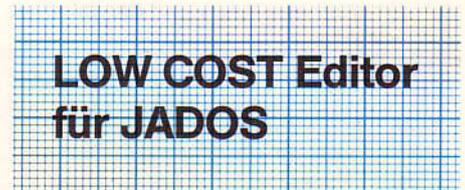
Fernsehen auf der COL

Wer hat schon mal einen Fernsehfilm auf einem Farbmonitor gesehen? So gut wie keiner, nehme ich an. Es gibt eine relativ preisgünstige Möglichkeit dies auf seinem Farbmonitor zu machen. Einige Monitoren vertragen als Farbeingang sogar ein FBAS-Signal (Farb-Bild-Austast-Signal). An diesen kann man einen ca. 200,- DM teuren Fernseh-Tuner von Philips anschließen. Andere Farbmonitoren vertragen nur RGB-Signale. Doch das soll nun auch nicht mehr das Problem sein. Die Firma ELV hat für 175,- DM, inklusive Gehäuse und Platine einen Video-Color-Prozessor herausgebracht. Dieser

ist eigentlich zum Verfälschen von Farben bei Videoüberspielungen gedacht, kann aber auch als FBAS-RGB-Decoder benutzt werden. Und es können auch Bilder aus der COL 256 in Farbe auf Videorecorder aufgenommen werden. Mit den 2 Geräten kann man nun Fernsehen in einer Wahnsinnsqualität sehen und der Ton geht über die HIFI-Anlage. Endlich mal scharfe Bilder im Fernsehen! Wer interessiert ist, kann sich an mich wenden.

Kei Thomsen

Anmerkung der Redaktion: Mit den scharfen Bildern meint der Autor nur die Qualität der Fernsehbilder, nicht deren Inhalt.



Der Programm-Editor JEDI entstand aus dem Wunsch heraus, einen preiswerten und doch leistungsstarken Programm-Editor für die 68000er Serie anzubieten.

Die Bedienung ist eng an den RDK Editor angelehnt, so daß Umsteiger nur die neuen Kommandos erlernen müssen. Der Such- und Ersetzmodus ist schneller und komfortabler geworden. Kommandos zum Aufbrechen und Verschmelzen von Textzeilen sind neu dazugekommen. Bis zu 26 Phrasenmakros (Textpassagen) erleichtern die Arbeit mit dem Assembler, 10 Dateimakros (Textblöcke) machen das Einlesen von ganzen Textdateien per Tastendruck möglich. Die Funktion, Makros abzuspeichern oder im laufenden Editiermodus zu verändern ist ebenfalls erreichbar. Textblöcke können kopiert, gelöscht, von Diskette gelesen oder darauf geschrieben werden. Das Inhaltsverzeichnis kann innerhalb des Editors angezeigt werden.

Dateien sind ohne Verlassen des Editors mit oder ohne Backup abzuspeichern.

Die maximale Größe des zu bearbeitenden Textes ist nur durch die Größe des zur Verfügung stehenden Speichers begrenzt. Auch die Scrollgeschwindigkeit ist wesentlich schneller geworden.

Die eben genannten Eigenschaften, verbunden mit dem attraktiven Preis, machen dieses Produkt besonders interessant.

Einrichten einer lauffähigen Version des CP/M 68 K mit CPU 68020

CP/M 68K und CPU 68020

von Hans-Albrecht May,
Teichstätte 25, 3153 Lahstedt 1,

1. Schreibschutz auf allen Originaldisketten anbringen. Das heißt, die an den Disketten (für den physikalischen Schreibschutz) vorgesehenen Aussparungen überkleben.

2. Im Handbuch der CPU 68020 steht zwar (s. Seite 48), „leider funktioniert das CP/M 68K . . . nicht mit der CPU 68020.“ Doch das ist glücklicherweise nicht ganz richtig. Man kann die Textfiles auf den Disketten mit dem Befehl „Type (Filename)“ durchaus lesen und das ist auch zu empfehlen.

Im File „Liesmich“ steht, man soll zunächst mit dem Programm „COPY“ alle Originaldisketten kopieren (natürlich nachdem sie schreibgeschützt sind, siehe unter 1.). Und das geht mit dem System auf der Originaldiskette (Nr.: 1) wirklich nicht. Das Programm „COPY“ läuft nicht, sondern bricht mit einer Fehlermeldung ab. „PIP“ läßt sich dagegen aufrufen. Ich empfehle daher folgende Schritte:

2.1 Formatieren einiger Disketten mit „UFO68020.68K“. Auch das geht mit dem mitgelieferten System auf Originaldiskette 1.

2.2 Kopieren einiger Programme von den Originaldisketten auf die neuformatierten Disketten. Welche Programme man dafür benötigt läßt sich aus den Files *.SUB ablesen, allerdings erst, wenn man die beiden mitgelieferten Manuals gelesen hat! Nein, lieber nicht. Wenn Sie sich einiges über die mit den *.SUB-Files aufgerufenen Programme durchlesen, reicht es, die Befehle zu entschlüsseln.

Sie benötigen mehrere Files von verschiedenen Originaldisketten, deren Nummern ich in Klammer mit angebe, was allerdings nur für die Version auf 5¼ Zoll Disketten (80 Spuren) DS/DD zutrifft.

NDRBUI.S	(Dsk. 1)
und	
AS68.REL, AS68INIT. und RELOC.REL	(Dsk.2)
und	
LO68.REL,	(Dsk.3)
und	
CPMLIB	(Dsk.5)
(und wenn Sie wollen)	
die Files *.SUB	(Dsk.1)
(und; das ist auch optional)	
SIZE68.REL	(Dsk.4)

Es ist allerdings meiner Ansicht nach zu Anfang zu empfehlen, die weitere Bearbeitung der Programme schrittweise, also ohne die Kommandofiles *.SUB vorzunehmen. Dann sieht man, ob diese richtig ausgeführt wurden.

Auch die anderen Files *.68K von der Originaldiskette können mit PIP kopiert werden, wenn auf der Diskette noch genug Platz ist, etwa 100KB als Reserve sollte man sich schon lassen.

2.3 Auf die kopierte Diskette den **Systemlader** von der Originaldiskette **übertragen**, mit folgendem Programmaufruf:

```
XPUTBOOT CPMLDR.SYS B
(wenn in Laufwerk B die kopierte Diskette ist, andernfalls das entsprechende Laufwerk angeben).
```

2.4 Die **Disketten aus den Laufwerken** nehmen und ein RESET geben (C reicht nicht), oder den Rechner kurz ganz abschalten. Den Rechner neu starten und die kopierte Diskette in Laufwerk A einlegen. Sie sehen so, ob sich das CP/M über das Grundprogramm booten läßt. Ist das nicht der Fall, müssen Sie überprüfen, ob Sie den Schritt 2.2 richtig ausgeführt haben und gegebenenfalls wiederholen.

2.5 Assembler AS68.REL initialisieren mit dem Befehl:

```
AS68.REL -I AS68INIT
(danach muß die Datei AS68SYMB.DAT auf der Diskette stehen).
```

2.6 Jetzt können Sie das **BIOS neu übersetzen**, mit dem Befehl:

```
AS68.REL NDRBIOS.S
(danach muß die entsprechende Datei NDRBIOS.O gebildet sein).
```

2.7 Linken eines neuen CPM.REL Files mit dem Linker LO68 mit dem Befehl:

```
LO68.REL -R -UM68010 -UCPM
-OCPM.REL CPMLIB NDRBIOS.O
```

2.8 Herstellen eines neuen Files **CPM.SYS** mit dem Befehl:

```
RELOC.REL -Baaaa CPM.REL
CPM.SYS
aaaa ist dabei die Adresse, auf der das System geladen wird, je nach gewünschter Größe der TPA, der RAMFLOPPY G: und natürlich dem verfügbaren RAM-Bereich.
```

bei 128K also für aaaaa 18000
bei 256K also für aaaaa 38000
bei 512K also für aaaaa 78000

2.9 (Ist **optional**.) Wie Sie mit der Abfrage STAT *.REL leicht feststellen können, ist der Speicherbedarf der Files *.REL ziemlich groß. Sie können ihn auf etwa die

Hälfte reduzieren, wenn Sie diese (relativen) Files in (absolute) Files mit einer festen Startadresse umwandeln. Allerdings machen Sie bitte nachfolgend beschriebene Umformung so nicht mit dem File CPM.REL! Sie rufen das Programm RELOC.REL auf wie folgt:

```
RELOC.REL RELOC.REL RELOC.68K
ebenso die übrigen Files .REL
RELOC (filename. REL) (filename.68k)
dann überzeugen Sie sich mit STAT
*.68k sicherheitshalber noch einmal, welche Files .REL Sie umgeformt haben. Danach können Sie die umgeformten Files .REL mit ERA (Filename) löschen und haben auf der Diskette mehr Platz zur Verfügung.
```

3. Start des neu erstellten Systems und Kopieren der Originaldisketten.

Zunächst, wie unter 2.3 beschrieben, den Rechner neu starten. Sie können nun endlich das Programm „COPY“ starten und wenn Sie alles richtig gemacht haben, meldet es sich mit einigen Fragen. Meldet es sich richtig, brechen Sie es ruhig wieder ab!!!

Wenn Sie zwei Laufwerke haben, ist es praktischer, Sie übertragen das Programm „COPY“ mit „PIP“ auf die RAM-Floppy G: also

```
PIP G:=A:COPY.68KÄVÜ
```

Dann schalten Sie mit „G:“ auf die RAM-Floppy um und starten von dort das Programm „COPY“. Nun habe Sie beide Laufwerke zum Kopieren frei. Machen Sie sich ruhig vorher einen Kaffee zurecht, dann das Kopieren dauert seine Zeit. Ein weiterer Vorschlag: lesen Sie nebenbei das Manual durch. Wenn Sie die Originaldiskette jeweils im Laufwerk A haben und die zum Kopieren vorbereitete formatierte Diskette in Laufwerk B, geben Sie einfach den Befehl

```
COPY ALL A B
```

bis Sie alles kopiert haben. Viel Spaß!

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

mc-modular-AT nun auch mit /386-Baugruppe:

Beschreibung zur CARD AT /386.

Die CPU-Baugruppe CARD-AT /386 kann im mc-modular-AT anstelle der CATE Card eingesetzt werden. Dadurch erhält man ein System mit echtem 32 Bit Prozessor (INTEL 80387). Der Austausch erfolgt dabei auf die denkbar einfachste Weise: alte CPU herausziehen und die neue Karte mit der dazugehörigen Speicherkarte einstecken, anschließend das mitgelieferte Setup-Programm für die neue Karte fahren und schon läuft der Rechner wieder.

Die CPU besteht aus drei Baugruppen (im BUS werden zwei Einsteckplätze belegt):

- die Karte mit dem Prozessor, BIOS, Keyboard-Controller, Steckplatz für math. Co-Prozessor (wahlweise 80287 - 80387), Akku und CMOS RAM
- die Karte mit dem Hauptspeicher (1 MB bestückt, 100 ns); kann auf 2 MB aufgerüstet werden.
- die Buserweiterung auf 32 Bit.

Da der 16 Bit Bus für den 32 Bit breiten Speicher nicht ausreicht, müssen 16 Bit extra verbunden werden. Dazu dient die Buserweiterung, eine kleine Baugruppe mit 3 Steckplätzen, die CPU und Speicher miteinander verbindet.

Achtung! Diese Baugruppe ist polarisiert, das heißt, sie funktioniert nur, wenn sie richtig herum eingesteckt wurde. Die Seite mit der Bezeichnung „F1“ zeigt dabei zum schon vorhandenen großen Bus.

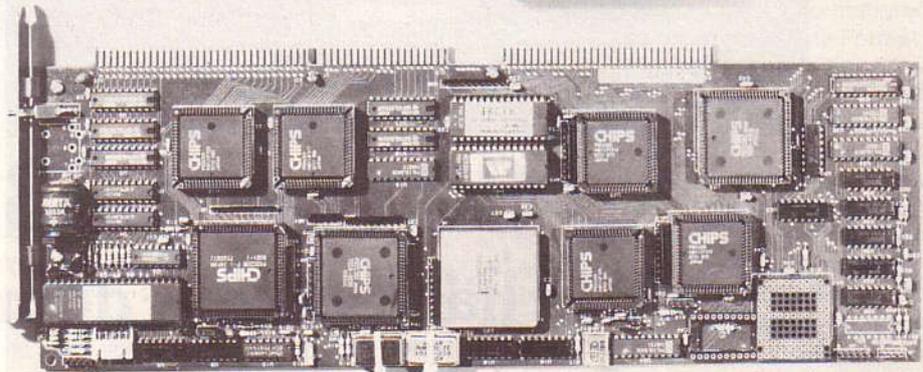
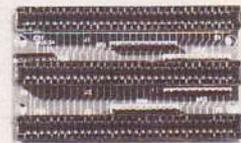
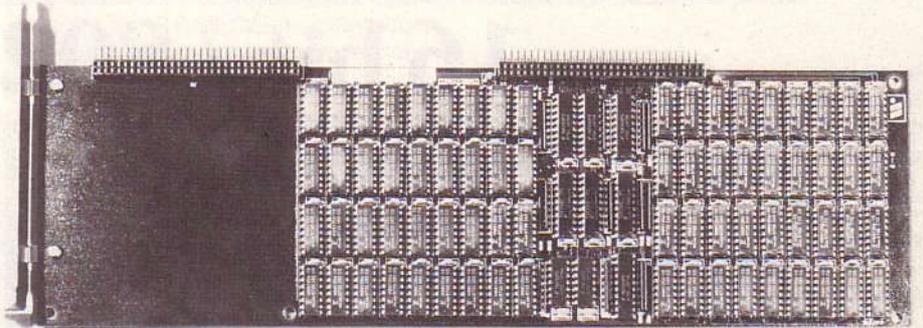
Auf der Speicherkarte sind zwei Banks zu je 1 MB (36 IC's * 256 KB) vorhanden. Es kann nur in 1 MB Schritten bestückt werden. Wenn Sie die Karte vor sich mit den Steckkontakten nach unten halten, ist die Bank links für den Hauptspeicher, die rechts für den Erweiterungsspeicher (extended memory) zuständig. Die Speicher müssen mindestens 100 ns Zugriffszeit aufweisen (oder schneller).

Die CPU hat an der Rückseite (Haltewinkel) einen Umschalter für den verwendeten Bildschirmadapter (Monochrom oder Farbe) genau wie die CATE Karte, nur daß der Schalter für den entsprechenden Adapter in die entgegengesetzte Richtung zeigen muß. Für einen monochrom Adapter nach rechts, wenn Sie die Karte von oben betrachten. Der Tastaturschluß ist identisch zur CATE Karte. Auch die Anschlüsse für Lautsprecher, Betriebsanzeige, Schlüsselschalter und externer Batterie sind vorhanden.

Vom ersten MByte Speicher können nur 640 KB verwendet werden (DOS Hauptspeicher), die restlichen 384 KB sind momentan nicht nutzbar.

16 oder 32 bit -

Wie hätten Sie's denn gerne?

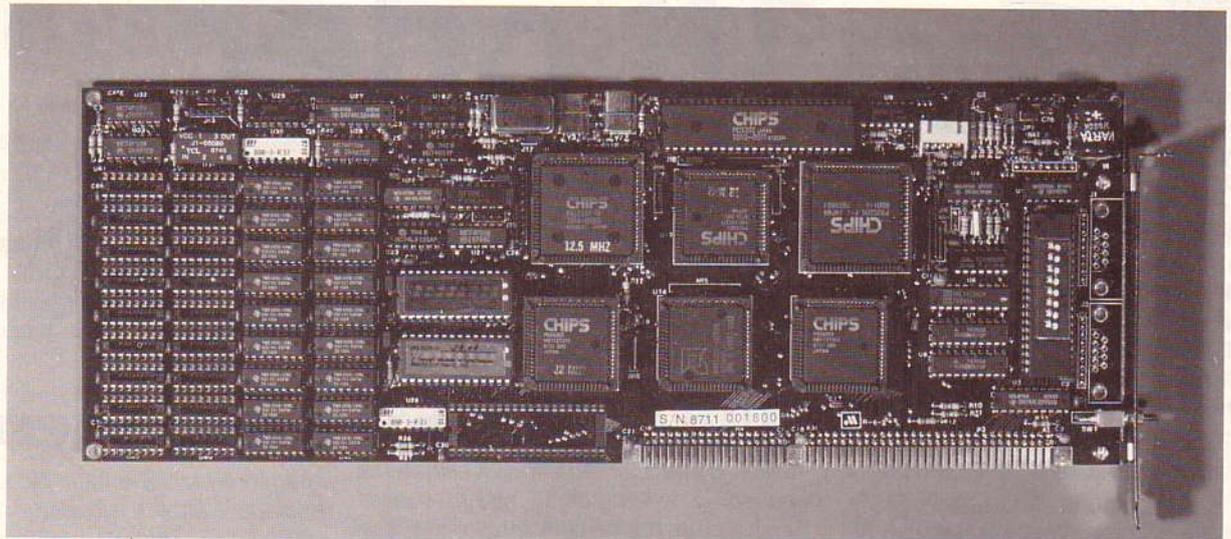
**mc-modular-AT /386****Technische Daten:**

- IBM-AT-kompatibel
- 32 Bit CPU-80386-16
- Clock wählbar über Tastatur 8/16 MHz, Optional 20 MHz
- I/O Busgeschwindigkeit kompatibel mit 8 MHz AT
- Bis zu 16 MB, 32 Bit, 0 wait state memory

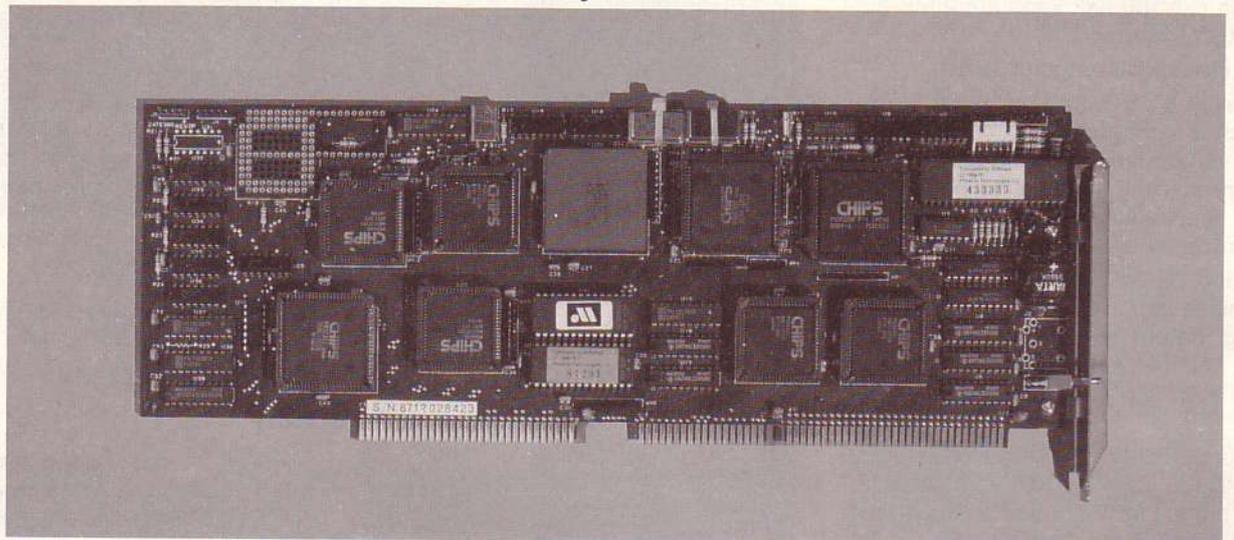
- 32 Bit Speicherkonfigurationen: 2 MB, 4 MB, 8 MB, 10 MB, 16 MB
- Sockel für 80287 und 80387
- Uhr/Kalender, NiCad-Akku
- Option: Tochterkarten für Speicherkarte mit a) 2 seriellen und 2 parallelen Schnittstellen oder b) EGA-Karte, Lev. 3, 640 x 480 Punkte
- Niedriger Stromverbrauch + 5 V, 3,5 Ampere
- Shadow-RAM-Bereich für System- und Bildschirm-BIOS.

See us at
COMDEX/Spring '88
 May 9-12, 1988
 Georgia World Congress Center
 Atlanta, GA

GRAF[®] computer



16 bit / 80286



32 bit / 80386

Wie hätten Sie's denn gerne?

mc - modular - AT

Der mc-modular-AT verfügt nun über zwei CPU-Karten. Die CPU /286 und, **NEU**, die CPU /386. Beide Baugruppen funktionieren mit allen anderen Baugruppen des Rechners. Somit kann der mc-modular-AT heute mit der /286-CPU und morgen – durch einfachen Austausch – mit der /386 CPU bestückt werden.

Umfangreichen Prosket kostenlos – mit anhängender Bestellkarte oder telefonisch (24 Std.) anfordern.

Technische Daten /386 CPU

Besteht aus zwei Baugruppen (CPU/Speicher) und 32-bit BUS-Erweiterung. Zentraleinheit 80386. 16 MHz. AT kompatibel.

I/O-Busgeschwindigkeit kompatibel mit 8 MHz AT. Speicher: 1 MByte (32 bit, 100 ns), 0 Waits im Lieferumfang. Speicher erweiterbar auf 2 MB (durch ICs auf Baugruppe) und weiter bis zu 16 MB. Sockel für 80287 **oder** 80387. Uhr/Kalender. Phoenix-BIOS. CHIPS-Technologie.

Bestell-Nr. 11213 DM 3850,- (freibl., incl. MWSt.)

Anfragen und OEM-Anfragen an:

GRAF ELEKTRONIK SYSTEME GMBH

Magnusstraße 13 • Postfach 1610 • 89 60 Kempten (Allgäu) • Telefon: (08 31) 6211
 Teletex: 831804=GRAF • Telex: 17831804=GRAF • Datentelefon: (08 31) 6 93 30

Filiale Hamburg:
 Ehrenbergstraße 56
 2000 Hamburg 50
 Telefon (040) 38 8151

Filiale München:
 Georgenstraße 61
 8000 München 40
 Telefon (089) 2715858

Händler:

Jörg Korb • 1000 Berlin 33 • Telefon (030) 8211947
 Schweiz: SYSTECH • CH-4104 Oberwil • Tel. (061) 301031
 A: Stecher GmbH • Ing.-Etzel-Str. 75-76 • 6020 Innsbruck

mc-modular-AT noch schöner:

Neues Gehäuse für den mc-modular-AT

Das neue Gehäuse des mc-modular-AT wurde mehr an die technischen Fähigkeiten des Computers angepaßt. Außer den optischen Verbesserungen wurde auch großer Wert auf eine noch komfortablere Montage der mechanischen Bestandteile gelegt. Dieser Beitrag versteht sich als Ergänzung des bisherigen Handbuchs zum mc-modular-AT.

Das neue „Gewand“ des AT haben wir auf dem Titel gezeigt.

Bevor nähere Details des Gehäuses beschrieben werden, lesen Sie die **Verbesserungen** und Änderungen als Zusammenfassung:

- RESET-Schalter auf der Frontseite
- Schlüsselschalter für Tastaturabschaltung
- „TURBO“-Schalter mit Anzeige
- Passiv-BUS steckbar
- Maximal 2 Diskettenlaufwerke möglich
- Keine LED-Montage mehr notwendig
- Größere Abmessungen:
49,5 x 43,5 x 16 cm

Der **RESET-Schalter** auf der Frontseite dient der besseren Handhabung. Anschluß siehe mc-modular-AT-Handbuch.

Der **Schlüsselschalter** dient der Abschaltung der Tastatur bei eingeschaltetem Rechner. Dieser kann jederzeit – auch in jedem beliebigen Programm – entfernt werden. Da dann die Tastatur nicht mehr als Eingabegerät zur Verfügung steht, kann ein nicht ausgeschaltetes Programm vor unbefugtem Zugriff bewahrt werden. Außerdem wird das Gerät mechanisch abgesperrt.

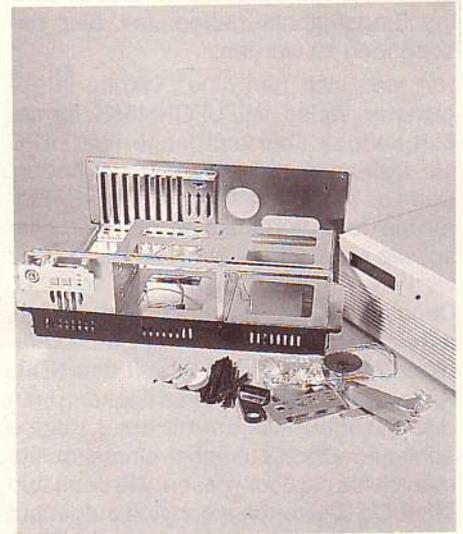
Der **Passiv-BUS** braucht nicht mehr festgeschraubt zu werden, sondern wird einfach aufgesteckt und mit einer Schraube gesichert.

Der **TURBO-Schalter** ermöglicht eine Umschaltung der Taktfrequenz von 8 MHz auf 10 MHz. Die entsprechende Leuchtdiode zeigt die eingestellte Taktfrequenz.

Mit dem neuen Gehäuse sind nur maximal **2 Laufwerke** (3 1/2" oder 5 1/4", auch gemischt) und 2 Festplatten verwendbar. Dabei werden die Festplatten im mittleren Ausschnitt (siehe Bild) montiert.

Bei der **Montage und Verdrahtung** der Leuchtdioden und Tastschalter braucht nicht mehr gelötet zu werden. Die mitgelieferten Kabel sind bereits mit Stecker-Verbindungen versehen. Nur Lautsprecher muß gelötet werden.

Beim Kauf von nur 3 1/2" Laufwerken sollte auch der Zubehörsatz mitbestellt werden, da das dazu notwendige Floppy-Kabel nur in diesem enthalten ist.



Tip: Zum Anschrauben der Baugruppen mit dem Haltewinkel sollten nicht die Schrauben mit den festangebrachten Beilagscheiben verwendet werden. Als negative Nebenerscheinung ergab sich, daß beim Festschrauben auch der Nachbar-Haltewinkel berührt wird. Dies führt dazu, daß beim Lösen einer Baugruppe immer zwei Schrauben gelöst werden müssen.

Auf Wunsch ist noch das „alte“ Gehäuse mit Klappdeckel lieferbar.

Patchwork – 1. Teil

Der verrückte Bootsektor

von Günter Renner

Ein altes Sprichwort sagt, daß überall nur mit Wasser gekocht wird. In der Computerei kann das wohl nur heißen, daß man überall nur mit Bits und Bytes arbeitet. Wenn diese Gebilde nun aber so einheitlich sind, wie man meinen möchte, so sollte es ein Problem, das vielen Anwendern zu schaffen macht, gar nicht geben: die Unmöglichkeit, Daten von einem Computersystem auf ein anderes zu übertragen.

Wer einmal versucht hat, mit dem NDR-Klein-Computer und dessen Floppy-Controller 1797 Disketten zu lesen, die auf einem IBM- oder kompatiblen PC erstellt worden sind, wurde sicher herb enttäuscht: es geht nichts. Nicht einmal den Bootsektor zu laden gelingt, was man leicht nachvollziehen kann, wenn man eine Diskette einlegt, „Floppy-Start“ des 68K-Grundprogramms drückt und an-

schließend den RAM bei \$9800 ansieht. Was tun aber jene Unentwegten, die sich weder von den grundsätzlichen Verschiedenheiten der Prozessorarchitektur, noch von den Problemen der Formatierung abschrecken lassen wollen?

Zunächst einmal besteht eine Inkompatibilität der Floppycontroller. Die Formatierung enthält bestimmte Bytes, die den Beginn von Datenfeldern (z.B. Sektoren) kennzeichnen. Leider ist es nun so, daß der 1797-Controller diejenigen der MS-DOS-Disketten nicht erkennen kann. Umgekehrt aber können die Controller der PCs diejenigen erkennen, die der 1797 verwendet. Will man dem freien Datenverkehr näherkommen, müßte man demzufolge zunächst einmal eine Diskette auf dem NDR-Klein-Computer formatieren – allerdings so, daß sie auf den PCs läuft. Dazu gibt es auch ein geeig-

netes Werkzeug: Der Formatierer UFORM68K kann das geeignete Format, nämlich 40 Spuren, doppelseitig, doppelte Dichte und neun Sektoren zu je 512 Bytes je Spur und Seite erzeugen. Natürlich braucht man dazu noch entweder ein „altmodisches“ 40-Spur-Laufwerk, was inzwischen keine sonderlich teure Investition mehr ist. Oder aber man verwendet ein 80-Spur-Laufwerk, das per Jumper auf 40-Spur-Betrieb umschaltbar ist. Man sollte sich dabei allerdings über etwas im Klaren sein, was schon manchen PC-Benutzer geärgert hat. Die modernen 80-Spur-Laufwerke haben schmalere Schreib-Leseköpfe. Disks, die damit formatiert sind, laufen nur sehr selten auf alten 40-Spuren, weil die Spur so schmal gerät, daß sie die breiteren Köpfe nicht mehr zuverlässig ummagnetisieren kann. Als Faustregel gilt, daß man zusammen mit XT-PCs besser richtige 40-Spur-Drives am NDR-Klein verwendet, während sich zusammen mit den AT-Typen umschaltbare 80-Spur-Scheibendreher empfehlen. Es ginge natürlich auch mit normalen 80-Spur-Drives, wenn man sich einen speziellen Formatierer schreibt, der ein doppeltes step-in

bewerkstelligt. Doch würde dies zu sehr ins Eingemachte gehen, um hier beschrieben zu werden.

Hat man sich nun eine Diskette in geeigneter Weise mit UFORM68K formatiert, kann sie aber noch lange nicht im PC als Datendiskette Verwendung finden. Denn es müssen erst noch einige Sektoren mit Daten vorbelegt werden, und zwar so, wie MS-DOS es will. Und genau hier wird es schon ein klein wenig verrückt. Denn es soll ein Weg aufgezeigt werden, auf einer MS-DOS-Diskette einen bootfähigen Monitor für den 68008 des NDR-Klein unterzubringen. In der Tat kann man diese beiden Fliegen mit einer Klappe schlagen. Doch zunächst einmal müßte man etwas darüber wissen, wie denn nun MS-DOS-Disketten eigentlich aufgebaut sind. Das Format wurde bereits erwähnt, wobei dazu zu sagen ist, daß es das älteste, aber auch das immer noch gebräuchlichste ist, das von diesem Betriebssystem verwendet wird. Es sei mir erlaubt, Möglichkeiten, die z.B. MS-DOS 3.3 bietet, hier bewußt auszuklammern.

Von Haus aus enthalten die MS-DOS-Disketten sozusagen als Grundausstattung folgende Elemente, die wie folgt auf Spur 0 lokalisiert sind:

	Sektor/Seite 0	Sektor/Seite 1
„Bootsektor“:	1	
erste FAT:	2 - 3	
zweite FAT:	4 - 5	
Hauptinhaltsverzeichnis:	6 - 9	1 - 3.

Bei Systemdisketten hat der Bootsektor dieselbe Funktion wie z.B. bei CP/M. Bei reinen Datendisketten enthält er lediglich eine Tabelle von Parametern, die Aus-

kunft geben über das Format der Diskette und die Belegung mit systemrelevanten Daten. Diese Daten sind es, die unsere Disk natürlich auch enthalten muß, und die wir nach dem Formatieren noch anlegen müssen.

Dann gibt es noch eine ganz gefährliche Sache: die FATs, die File-allocation-tables. Grob gesagt handelt es sich hier um Listen, denen man die Sektoren entnehmen kann, die zu einer bestimmten Datei gehören. An Hand derselben liest MS-DOS die richtigen Sektoren in der richtigen Reihenfolge von der Scheibe und setzt sie im RAM zu kompletten Dateien zusammen. Natürlich geht das nicht mehr, sobald diese Tabelle fehlerhaft wird. Aus Sicherheitsgründen sind deshalb gleich zwei davon vorhanden.

An die FATs schließt sich noch der Bereich des Hauptinhaltsverzeichnisses an. MS-DOS bietet die Möglichkeit, Unterverzeichnisse einzurichten, damit der Disketteninhalt übersichtlicher bleibt. Der Diskmonitor, der später noch vorgestellt wird, arbeitet allerdings nur mit dem Hauptinhaltsverzeichnis, das übrigens insgesamt 112 Einträge fassen kann.

Auf der Spur 0, Seite 1 bleiben mithin die Sektoren 4 bis 9 übrig. Diese werden später vom Diskmonitor belegt.

Doch nun zum ersten Element, dem Bootsektor. Wie bereits gesagt, muß er Daten über das Diskettenformat enthalten. Uns soll er aber darüberhinaus auf dem NDR-Klein dienen. Denn das Grundprogramm enthält eine Funktion, mit der man den Bootsektor nach \$9800 einlesen kann. Steht ganz am Anfang noch ein Befehl

'nop' als Kennung, so springt der Prozessor just dorthin und beginnt zu arbeiten.

Das Programmlisting enthält nun alles, was im Bootsektor stehen muß, damit die Disk einerseits auf einem PC als Datendiskette läuft, andererseits auf dem NDR-Klein mit Floppy-Start gestartet werden kann. Wie es sich für einen rechten Bootsektor gehört, lädt er einige Sektoren, die später mit dem Diskmonitor belegt werden, in den RAM. Und wenn alles gutgegangen ist, wird dieser auch gestartet. Damit kann man dann Dateien, die sich auf der Diskette befinden, in den RAM laden.

Voreilige, die das Programm gleich eintippen und assemblieren wollen, können alles mit

```
lea bootsek(pc),a0
move # drive,d4      *!lwd
clr d3               *!Spur 0
moveq # 2,d1         *!Sektor 1
moveq # 2,d1         *!schreiben
move # !floppy,d7
trap # 1             *oder bsr traprts
rts
```

auf eine geeignet formatierte Diskette schreiben. Startet man diese dann mit ‚Floppy-Start‘, handelt man sich garantiert eine ‚Illegal Instruction‘ ein, denn es steht noch nichts Vernünftiges in den Sektoren, die den Monitor aufnehmen. Aber man kann sich das Ganze auf Adresse \$9800 sicher einmal ansehen... In der nächsten Folge geht es um das Anlegen der FATs und des Inhaltsverzeichnisses.

Günter Renner
Schloßbühlstraße 11
7206 Emmingen-Liptingen

```
*Bootsektor fuer Transferdiskette
*wird von IBM-PCs und Kompatiblen
*auf Datendisks akzeptiert,
*kann auf NDR-Klein-Computer mit
*'Floppy-Start' des 68k-Grundprogr.
*gebootet werden und laedt einen
*kleinen Filemonitor von Spur 0
*Seite 1, Sektoren 4 - 9 (=3 kB).
*Diese Version verwendet keinen trap!
```

```
* (C) 1988 Guenter Renner
* Schlossbuehlstr. 11
* 7206 Emmingen-Liptingen
```

```
*Floppy-Start laedt nach $9800+Basis
*ggf. ORG-Anweisung aendern...
```

```
org $9800
```

```
bootsek:
```

```
traprts equ bootsek-$9800+$31d6
handler equ bootsek+$200
drive equ $21      *nach Bedarf...
```

```
nop                *Kennung fuer
bra laden          *RDK-Booter
```

```
dc.b 0,0,0,0,0    *nur z. Auffuellen
```

```
*BIOS-Parameterblock - unbedingt noetig
*fuer PC-DOS, sonst wird Disk nicht
*akzeptiert - muss auf $b beginnen!
```

```
biospar:
dc.b 0,2           *512 Bytes/Sektor
dc.b 2             *Sektoren/Cluster
dc.b 1,0           *reserv. Sektoren
dc.b 2             *Anzahl FATs
dc.b $70,0         *Eintraege root
dc.b $d0,2         *Anzahl Sektoren
dc.b $fd           *Kennung
dc.b 2,0           *Sektoren/FAT
dc.b 9,0           *Sektoren/Track
dc.b 2,0           *zweiseitig
dc.b 0,0           *versteckte Sektoren
dc.b 0,0
dc.b $a            *Laenge des Blocks
```

```
dc.b $df           *Kennung
dc.b 2,$25,2,9,$2a,$ff,$50,$f6,$f,2
```

```
dc.b $cd,$20      *ist INT 20/8086
                        *zurueck ins DOS
```

```
ds 0              *auf gerade Adresse!
```

```
laden:
lea bootsek(pc),a4 *ab hier 68k-Code
                        *ist Grundprog. da?
```

```

suba.l #9400,a4      *Adresse d. Kennung
cmp.l #5aa58001,(a4) *wichtig, weil alles
beq ladenok         *ohne trap geht

clr d0              *sonst zurueck
subq #1,d0          *a la cp/m
rts

ladenok:
lea handler(pc),a0  *jetzt handler laden
move #drive,d4      *lwd
bset #7,d4          *immer Seite 1
clr d3              *Spur 0
moveq #4,d2         *ist erster Sektor
moveq #1,d1         *lesen
lea fflag(pc),a4   *fehlerfrei
clr.b (a4)         *annehmen

lade1:
move #!floppy,d7   *Treiber aufrufen
bsr traprts
bcc lade3          *gings gut?
or.b d0,(a4)      *Fehlerflag updaten

lade3:
adda.l #200,a0     *naechstes Ziel
addq #1,d2         *naechster Sektor
cmp #9,d2          *bis Sektor 9
ble lade1

tst.b (a4)         *wenn fehlerfrei

deq handler        *Handler starten

move #!clrscreen,d7 *sonst
bsr traprts
lea meldf(pc),a4

lade4:
move.b (a4)+,d0    *Fehlermeldung
beq lade5
move #!co2,d7
bsr traprts
bra lade4

lade5:
move #!ci,d7       *trotzdem starten?
bsr traprts
and #5f,d0         *klein/gross
cmp #4a,d0         *ist es 'j'/'J'?
beq handler
rts

fflag: dc.b 0      *fuer Floppy-Fehler

meldf:
dc.b $d,$a,$a,$a
dc.b 'Handler-Ladefehler.'
dc.b 'Trotzdem starten? J = ja!'
dc.b $d,$a,0
ds 0
    
```

Briefe

Kontakte

Kleinanzeigen

Leider habe ich mit RNWINDOW V4.1-02 Probleme: Ich verwende einen NDR-Computer, Flomon 1.5, Hardcopymaus r2 und die Atari-Maus.

Das Programm läßt sich einwandfrei laden. Am oberen Bildschirmrand erscheint die Angabe der Speicherbank 81. Die Statuszeile erscheint, auch die Demos und Setze-Anweisungen lassen sich laden und laufen einwandfrei. Soweit ist also alles klar. Die Maus bringt aber den Ärger.

Ich habe mich genau an die entsprechenden Anweisungen gehalten (SETZE NORM, SETZE MAUSAN, usw.). Ein Fenster zu eröffnen ist dabei aber reine Glückssache. Wenn es dann aber doch einmal geklappt hat, so verschwindet es meistens schon wieder beim Bewegen der Maus (Fadenkreuz). Die Tastenfunktionen der Maus sind nach meiner Feststellung soweit in Ordnung, bis auf „FENSTER VERGRÖßERN“: hier springt beim Druck auf die rechte Taste manchmal das Fadenkreuz nicht nach unten rechts, sondern nach oben links im aktiven Fenster. Wenn das Fadenkreuz ohne direkte Funktion über den Bildschirm bewegt wird, wird dadurch sehr oft ein Bildschirmscroll ausgelöst, oder das Fadenkreuz bleibt an irgend einer Stelle des Bildschirms hängen, obwohl die Maus weiter bewegt wird.

Martin Bohlen, Kanalstraße Süd 18, 2962 Ostgroßefehn 1

Antwort LOOP:

Sollten irgendwelche Funktionen der Maus ausgelöst werden, ohne daß Sie eine Taste gedrückt haben (wenn Sie z.B. das Fadenkreuz über den Bildschirm bewegen und so einen Scroll auslösen), dann ist es trotz allem wahrscheinlich, daß der Fehler in der Hardware liegt. Überprüfen Sie daher noch einmal, ob beide Maustasten korrekt vom Rechner gelesen werden können (evtl. mit einem kleinen Hilfsprogramm oder einem Oszilloskop). Ein öfters festgestellter Fehler ist das Fehlen einer Brücke oder eines Pullup-Widerstandes für eine der beiden Tasten.

Sollten Sie die Beispielenüs (WS, TURBO) nicht gebrauchen können (weil Sie weder WS noch TURBO besitzen), dann entfernen Sie zum Test in einer der beiden Menüdateien alle Zeilen, die mit einem '?' beginnen. Nach SETMENU . . . müßte dann am unteren Bildschirmrand eine Zeile mit verschiedenen Punkten erscheinen. Beim Anklicken einer dieser Punkte müßte ein weiteres Fenster eröffnet werden. Ist dies der Fall, dann arbeitet das Menüsystem korrekt. Für Ihren Eigenbedarf schreiben Sie sich die Me-

nüs dann am besten selbst; der Aufbau einer derartigen Menüdatei müßte aus den Beispielen hinreichend gut hervorgehen.

Wir hatten die LOOP-Leser nach ihren Erfahrungen mit Kassettenrecordern gefragt. Hier einige Antworten:

Thema: Kassettenrecorder

Ich verwende den Data Recorder DR202A von der Firma Sanyo Elektronik, Truderingerstraße 13, 8000 München 80, Preis ca. 130 DM. Dieser Data Recorder ist schon 2 Jahre sehr zuverlässig und mit ADSS, Bandzählwerk, Reset, Save Mute und Fernbedienungstaste ausgerüstet. Es muß lediglich auf der Kassetten-Interfacekarte „Cas 83“ der Widerstand R 4 = 1KR abgetrennt werden, um das Ausgangssignal des EXOR3.CD 4070 anzuheben. Ich bin wirklich sehr zufrieden mit dem Gerät. Schaltungsunterlagen sind kostenlos von der Firma Sanyo erhältlich. Herbert Kanja, DDR

Kassetten-Datenrecorder DMF-C 668:
Best.-Nr. 50.KD **DM 120,-**
zu beziehen über:
-vgs- verlagsgesellschaft
Breite Straße 118/120, 5000 Köln 1

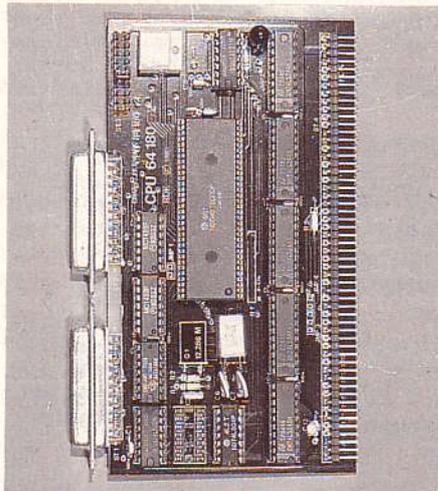
Hardscroll und Hardcopy:

Die „neue“ GDP64K kommt!

Neuer Name: GDP64HS für „Hardcopy + Scroll“

Schon längst erwartet, schon oft angekündigt, jetzt im Musterbau und bald verfügbar: die GDP64HS. HS steht für Hardscroll – damit ist ein Manko der alten GDP65K nun endlich behoben. Der von uns so lang verteidigte Fehler innerhalb des GDP-Prozessors existierte tatsächlich.

Erst die neueren, seit etwa eineinhalb Jahren ausgelieferten GDP-Prozessoren haben diesen Fehler nicht mehr. Wir haben uns deshalb mit Herrn Bulwien, dem Entwickler der Zusatzhardware geeinigt und diese Änderung mit ins Layout für die neue GDP64K genommen.



Die Baugruppe und der Bausatz werden ab Mai 88 erhältlich sein. Wir werden ebenso einen günstigen Umbausatz für die bisherigen GDP64K-Benutzer anbieten, der aus der Platine und den neuen Bausteinen besteht. Mehr hierüber in der nächsten LOOP, sobald wir die GDP tatsächlich liefern können.

Natürlich entspricht die Baugruppe der GES-Norm und läßt sich sowohl beim NDR-Computer als auch bei ECB-Systemen einsetzen. Wir zeigen die Baugruppe auf der Hannover Messe CeBIT.

GDP64HS ist völlig aufwärtskompatibel zur bisherigen GDP64K. Alle Programme laufen also problemlos weiter. Natürlich müssen Scroll und Hardcopy von den Programmen unterstützt werden. Das neue Grundprogramm zum 680xx-System tut dies bereits.

Briefe

Kontakte

Kleinanzeigen

Die Redaktion:

Achtung MODULA2 USER

In der letzten Zeit erhielten wir zahlreiche Anfragen bezüglich des Modula2 Compilers für die 680xxer Serie der Firma P1.

Die Probleme entstanden zum größten Teil durch die für einen Anfänger schwer zu verstehende Dokumentation. Die Beschreibungen der einzelnen Module enthielten zwar alles Notwendige zur Funktion des Compilers, jedoch fehlte es oft an sinnvollen Beispielen.

Ja gut, es gibt zwar mittlerweile reichhaltige Literatur über die Programmierung von Modula2, aber die Kompatibilität der beschriebenen Compiler ist meist an der MSDOS-Ebene angelehnt.

Aus diesen Anregungen heraus möchten wir diese Lücke schließen und appellieren an ihre Mitarbeit.

Als aller erstes erscheint es uns sinnvoll, Material zu sammeln; mit anderen Worten, wir möchten Sie ermutigen, Ihre eigenen Erfahrungen in Form von fertigen Modulen zu unseren Händen zu senden. Gefragt sind keine fertigen Programmprodukte, sondern Module, die eine oft verwendbare Tätigkeit ausführen, eben ganz nach der Philosophie von Modula2.

Die Module sollten, um die Bearbeitungsschwierigkeiten zu erhöhen, in Form eines Datenträgers vorliegen.

Sobald genügend TOOL-Programme vorliegen, wird eine Modula-TOOL-Diskette bei uns verfügbar sein.

Natürlich ist diese Mitarbeit nicht umsonst. Autoren, deren Module veröffentlicht werden, erhalten einen Warengutschein. Als zweiten Punkt möchten wir einen Adressenaustausch zwischen Mo-

dula2-Programmierern ermöglichen. Wer mit der Veröffentlichung seiner Anschrift einverstanden ist, kann sich bei uns telefonisch oder schriftlich melden (H. Pawlowitz). Nach Ablauf eines bestimmten Zeitraumes erhält jeder, der von uns notiert wurde, eine Liste aller uns bekannten Modula-Benutzer.

Also, arbeiten Sie aktiv bei der Gestaltung Ihres NDR-Computers mit und leisten Sie so wertvolle Teamarbeit, die zu guter Letzt eine große Hilfe für Sie sein wird.

In der LOOP 15 fand ich einen Artikel von Herrn Rüdiger Bäcker für „Aufsteiger von CP/M68K nach JADOS“ sowie einen Leserbrief von Herrn Dieter Lindinger, in dem die Vermutung geäußert wurde, daß JADOS „bedeutend besser ist als das fast überall eingesetzte MS-DOS-System. Ich glaube, daß hier ein falscher Eindruck entsteht:

– Derzeit einsatzfähige professionelle Betriebssysteme sind zum Beispiel UNIX oder EUMEL.

– In naher Zukunft wird das Betriebssystem OS/2 eine große Rolle spielen, da es von IBM auf kleinen Rechnern eingesetzt wird und eine Kopplung an Großrechner (natürlich von IBM) gestattet. Laut Werbeaussagen von GES wird OS/2 auch auf dem mc-modular-AT laufen!

– Derzeit wird allorts MS-DOS eingesetzt. Aber: MS-DOS ist tot! (Es wird von OS/2 abgelöst.)

– Auf Rechnern mit dem M680xx-Prozessor (sofern sie nicht unter UNIX laufen) gibt es momentan kein Standard-system.

– Auf dem M680xxer NDR-Klein-Computer ist das einzig derzeit verfügbare professionelle Betriebssystem CP/M68K. Hier gibt es Compiler für diverse höhere Programmiersprachen (insbesondere das p1 Modula2). Dennoch ist CP/M68K, insbesondere bei der Disketten-/Festplattenverwaltung, dem MS-DOS weit unterlegen. (Es gibt zum Beispiel keine hierarchischen Verzeichnisse und kein allgemeines Diskettenformat.)

– JADOS ist bei der Diskettenverwaltung (Festplatten werden nicht unterstützt) dem CP/M68K weit unterlegen. Dafür ist die hardwarenahe Programmierung einfacher, da anders als bei CP/M68K direkt auf alle Grundprogrammrountinen zugegriffen werden kann.

– JOGIDOS gestattet nur eine sehr einfache Diskettenverwaltung und ist eigentlich kein vollständiges Betriebssystem, sondern eher ein kleines Programmpaket zur Diskettenverwaltung. Martin Husemann, Am Ballerbrink 5, 4800 Bielefeld 18, Tel. (0521) 201293

Antwort LOOP:

Das Betriebssystem MS-DOS ist das derzeit weltweit am häufigsten eingesetzte Betriebssystem. Es ist sicher von der Technik her nicht mehr auf dem neuesten Stand, es wäre jedoch verfehlt, dieses Betriebssystem als „tot“ zu bezeichnen. Alle derzeit im Markt befindlichen IBM-Computer mit der 8088/8086-CPU werden weiterhin MS-DOS fahren – und dies sind sicher einige Millionen. OS/2 läuft natürlich ebenfalls auf dem modular-AT. Beim NDR-System werden wir uns auch weiter auf die drei Betriebssysteme JOGIDOS (für Einsteiger), JADOS und CP/M68K konzentrieren.

Verkaufe: NDR-Computer 68008 voll betriebsbereit, Schrofengehäuse, CPU 68, SBC2, IOW, IOE/EX, CENT, GDP64, 2 x ROA64, RAM64/256, CAS, POW5, POW21/26, NETZ, NE2, Frontprommer, 2 x Floppy 3 1/2", 3 x BUS2, Progr. Z80, 68008, Basic, UFORM68K, Datey, Textey, Dosey, Tast- u. Zenit-Monitor. VB 1600,- DM. W. Freerk, Eulenkamp 5, 3007 Gehrden, Tel. 05108/5098

Verkaufe: NDR-Klein-Computer, CPU 68000 (16 bit), 4 x ROA64 + RAM + Akku-Puffer, Key, Dintast, GDP64K Monitor (Zenith), NETZ, Grundprogramm + Buch, Pascal/S, Loop 2-15 (kompl. od. einzeln). Preis: VB. Ralf Rigger, Frauengasse 8, 7407 Rottenburg 1, Tel. 07472/23928

NDR 68008: Verk. hochkomfort. Vokabelprogramm 4000 (!) englische Vok. und Redewendungen, „intelligente“ Eingabekontr., autom. Fehlerwiederh. usw. 64K RAM erforderl. Disk. und ausführl. Anleitung für nur DM 80,- per NN + Porto u. Verp. Info gegen Rückumschl. Kai Ruppert, Kiebitzweg 8, 3171 Osloß, Telefon (05362) 7440.

VERKAUFE: NDR-Computer BUS3, ROA64, ROA16, KEY, Tast 1, Z80-CPU, GDP64K, HEX-I/O, Hext-Tast. Eproms: HEX-MON, Grundprogramm Basic 1.5 mit allen Beschreibungen + Anleitungen. VB 550,- DM, dazu Monitor-Chassis grün 100,- DM, 5 1/4" LW 100,- DM. Jürgen Kömen, Am Butzenberg 8, 6696 Nonnweiler, Tel. 06873/7934.

Verkaufe: SCHACH 1.7 (Maussteuerung möglich, alle 24 WM-Partien 1987 mit auf Diskette) DM 39,- + Porto unter JADOS 2.1. Info: U. Steinhaus-Peers, Schauenburgstraße 52, 2300 Kiel.

Verkaufe aus Zeitmangel NDR-Klein-Computer CPU68K, funktionsfähig im Gehäuse mit grünem Bildschirm, 2 x ROA64 mit Grundprogramm, Tast KL, FLO3, Disklaufwerk 5 1/4", GDP64K, NEC3, SBC2, POW5, CPU Z80, versch. Disketten (JADOS, -TOOL, SCHACH), Literatur und Eproms. Dieter Lindinger, Gutenbergstraße 46, 8502 Zirndorf, Tel. 0911/607606.

Verkaufe: 1 x CPU68000 12 MHz-Vers. mit Präz.-Sockel DM 130,-; 1 x Bankboot DM 45,-; 1 x Banksel DM 20,-, alles fertig aufgebaut und funktionsfähig. Th. Krähmer, Kolpingweg 9, 7344 Gingen/Fils, wochentags Tel. 07572/6428.

Verkaufe: NDR-Computer für Einsteiger, CAS-Platine mit Anschlußkabel, Stecker und dazugehöriger Kassettenrekorder Grundig CR 585a, neuwertig, betriebsbereit DM 175,-; SBC2-Baugruppe DM 60,-. Marc Rasch, Kleine Düwelstraße 3, 3000 Hannover 1, Tel. 0511/852426

VERKAUFE: CPU Z80, 2 x ROA64, DRAM256, KEY, GDP64K, FDC, CENT2, IOE, CAS, BANKBOOT, BUS3, Tastatur, Monitor, Laufwerk, Literatur, Software, zus. DM 700,-. Tel. 07251/41957.

NDR 68000/08: Superprogramme für JADOS 2.1 vom JADOS-Entwickler: DISASSEMBLER DM 50,-; REVERSI DM 39,-; DISKETTENMONITOR nur DM 35,-. Preise + Porto: DM 3,- (Scheck) / DM 6,- (NN) auf Disk 3 1/2" oder 5 1/4" unter JADOS. K. Janßen, Hanninxweg 74, 4150 Krefeld 1

Suche Kontakte zu anderen NDR-Computer-Besitzern! Ich arbeite mit CPU Z80, CPM und dem SPS-Compiler. Wer im Großraum Euskirchen, Köln und Bonn an einem Kontaktaustausch interessiert ist, der ruft doch mal an! Johannes-Wilfried Hennes, 5350 Euskirchen, Gust.-Nachtigall-Str. 4, Tel. 02251/55508. PS. Am besten nach 19.00 Uhr oder am Wochenende.

Wir sind einige NDR-Computer-Freunde in der Schweiz. Alte Füchse oder Neulinge wollen so freundlich sein und sich zwecks **Erfahrungsaustausch** zu uns gesellen: Josef Allgäuer, CH-Nendeln, (075) 32990 oder Heinz Amgwerd, CH-Wohlen, (057) 227642.

Suche Kontakt im Raum Düsseldorf, NDR68008-Einsteiger. Peter Espenhahn, Karolingerstraße 50, 4000 Düsseldorf 1.

Verkaufe: SBC3 (8 MHz-16KByte RAM), HEXIO, HEXMON, KEY, TAST1 und Gehäuse, GDP64K, Gehäuse3, LOOP 1-17; alle Platinen sind mit Handbücher, fertig aufgebaut und funktionsfähig; Verhandlungsbasis: DM 600,-; einzeln ca. 1/2 Bausatzpreis. Oliver Vogel, Tel. (07153) 31699 (von 17.30 Uhr - 20.30 Uhr).

Aus der Technik

Wenn seltsame Zeichen kommen:

Entprellen des STROBE-Signals der Tastatur

Sollten Sie schon einmal ein längeres Kabel an die parallele Tastatur angeschlossen haben, so können Sie vom Prellen der Tastatur wahrscheinlich ein Lied singen. Besonders empfindlich ist dieses STROBE-Signal der Tastatur beim Ein- und Ausschalten von anderen elektrischen Verbrauchern. Ein klassischer Fall ist z. B. das Einschalten des Lichtes; der Effekt ist meist, daß einige willkürliche Zeichen auf dem Bildschirm erscheinen. Physikalisch entsteht dieses Prellen am Lichtschalter: Durch das Einschalten kippt der Schalterkontakt um und federt natürlich leicht nach, was zu Prellsigna-

len führt. Dieses Prellsignale gelangen über die Netzkabel zum Netzteil des Rechners und unter Umständen in die digitale Logik des Rechners (mitunter zum STROBE-Signal). Diese Prellsignale des Schalters können aber auch über die Luft zum Tastatur-Strobe gelangen (hochfrequente Schaltimpulse). Diese beiden Störquellen kann man natürlich durch einen Netzfilter (im NE3 enthalten) und durch entsprechende Abschirmungsmaßnahmen (Tastaturkabel abschirmen, geschlossenes Metallgehäuse - GEH3) relativ gut eindämmen.

Sollten Sie aber ein offenes System haben, hat natürlich jeder Störsender (Schalter, Küchenmaschine, Föhn usw.) Zugriff auf Ihren Rechner. Hier hilft dann meistens nur noch die Holzhammermethode: Durch Einlöten eines Kondensators vom STROBE gegen Masse können solche Störungen herausgefiltert wer-

den. Die Auswahl des Kondensators sollten Sie Ihren Verhältnissen anpassen. Auf der Baugruppe KEY2 ist bereits ein Kondensator (56 pF) eingelötet, doch dieser ist in den meisten Fällen zu klein.

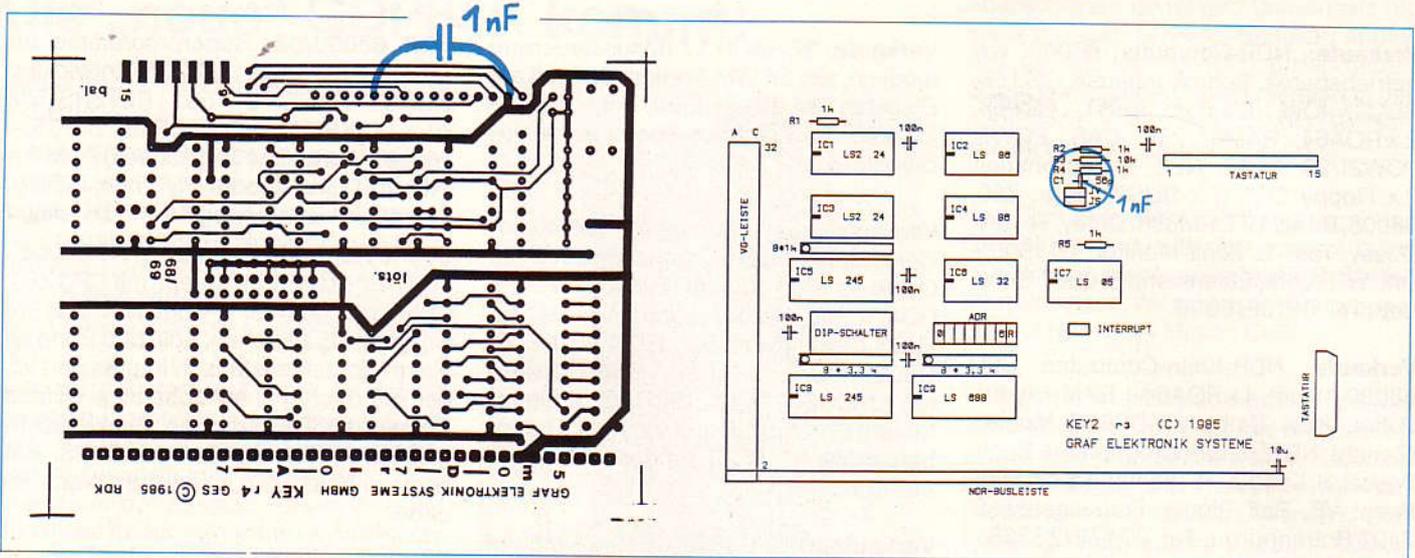
Sie sollten ihn allerdings nicht größer als 1 nF wählen. Durch diesen relativ einfachen Eingriff in die KEY bzw. KEY2 können Sie Ihren Rechner um einiges störstärker machen.

Änderung KEY r1, KEY r2, KEY r3 und KEY r4:

Einlöten eines 1 nF Kondensators an ST1 bzw. STECKER1 zwischen PIN 2 (Masse) und PIN 11 (STROBE), siehe Abb.

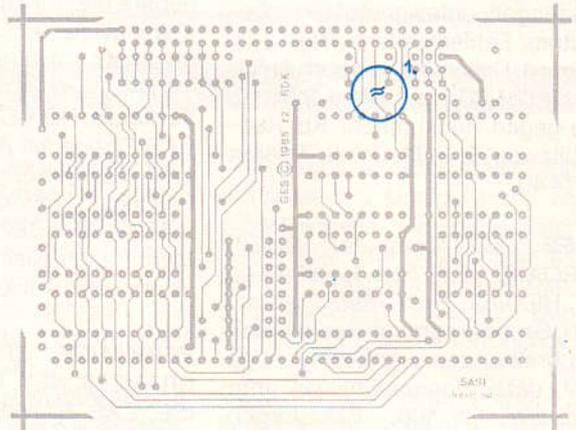
Änderung der Baugruppe KEY2 r2 und KEY2 r3:

Auswechseln des Kondensators C1 (56 pF) gegen einen 1 nF-Kondensator (siehe Abb.).



Druckfehler bei der letzten Sasi-Änderung

Leider ist beim Setzen der LOOP 16 das Transparent auf dem Layout verrutscht. Dadurch wurde die falsche Leiterbahn gekennzeichnet. Hier nun die richtige Änderung der Sasischnittstelle, mit der nun garantiert ein fehlerfreies Arbeiten der Festplatte gewährleistet ist.



Kleiner Hardware-Eingriff nötig: GDP 64K mit Windows

Das Flackern beim Bildschirmscrollen unter der Windowverwaltung ist, wie auch schon in der „Read.me“-Datei beschrieben, durch eine kleine Hardwareände-

rung auf der GDP 64 abzuschalten. Erreicht wird dadurch eine wesentliche Steigerung der Scrollgeschwindigkeit. Es ist also jedem Benutzer zu empfehlen, diese Änderung durchzuführen.

Da die Bezeichnungen der ICs in unseren Handbüchern im Gegensatz zum Elektronikladen unterschiedlich sind, entstan-

den kleine Ungereimtheiten. Deswegen hier die korrekte Verdrahtung der Baugruppe:

Die /DW-Leitung des 9366 auftrennen und auf PIN 1 des 74LS32 führen. PIN 2 und PIN 9 des 74LS32 verbinden. Ausgang 3 nach PIN 3 des 4164 führen (J6).

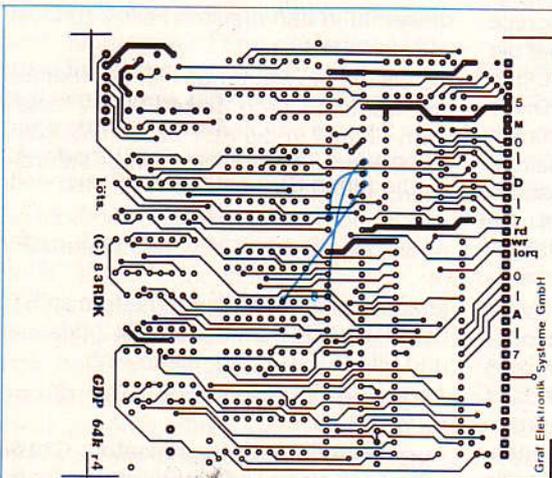


Abb. 7.9.25 Die Lötseite der Leiterplatte GDP64

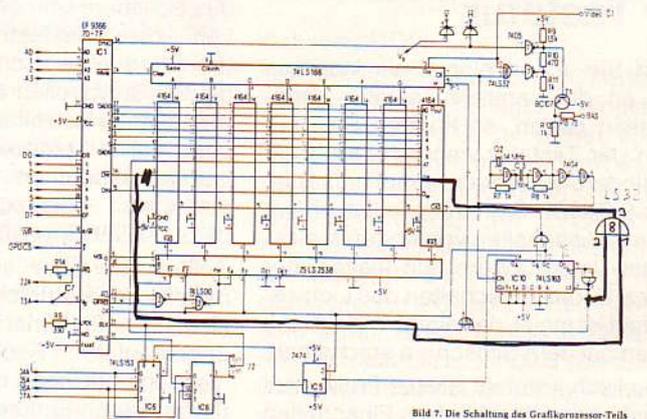


Bild 7. Die Schaltung des Grafikprozessor-Teils

Probleme bei der Festplatte:

System-Reset schlägt zur Festplatte durch

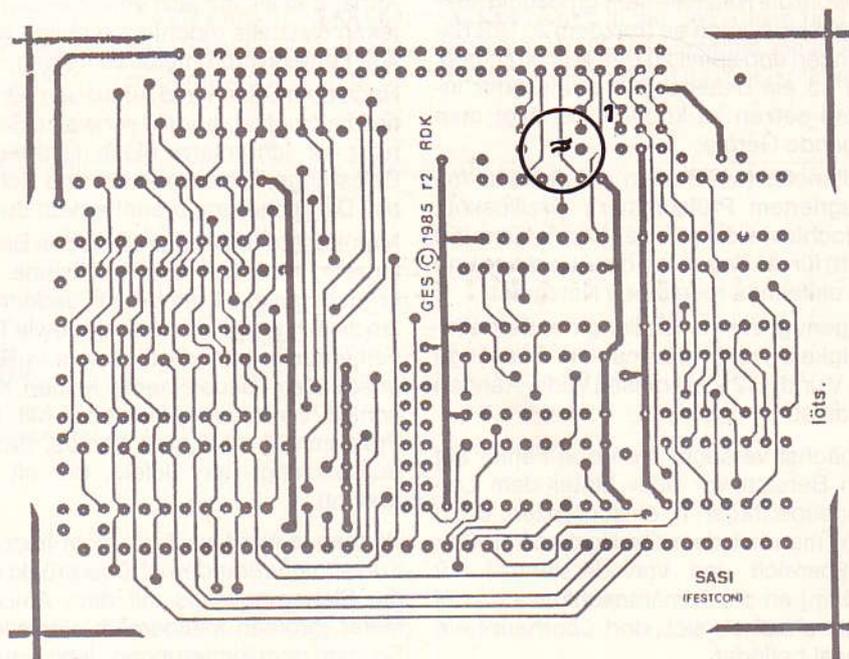
Es kann vorkommen, daß die Festplatte bei häufigem Ein-/Ausschalten „abstürzt“ und das Direktory löscht.

Fehlerbehebung:

1. Auf der SASI-Baugruppe wird der RESET der SASI zum 50poligen Stecker getrennt.

2. Sollte die Festplatte nicht mehr formatierbar sein, können Sie im Notfall (wenn sonst nichts mehr hilft) die Festplatte mit dem sog. „FACTORY FORMAT“ formatieren. Dazu müssen Sie die definierte Konstante „FORMATC“ (jetzt festgesetzt auf 4) im Formattierer auf den Wert 2 abändern und dann die Festplatte neu formatieren. Dieses „FACTORY“ sollten Sie aber nur durchführen, wenn die Festplatte sonst nicht mehr formatierbar ist.

Achtung: Denken Sie daran: Immer die Dateien der Festplatte auf Floppy sichern!



Fehlersuche leicht gemacht:

Technischer Service an Computer-Netzteilen

Erfahrungen aus der Netzteilreparatur von Alfred Olbrich, GES

In jedem Computer befindet sich ein Hochleistungsnetzteil, das auf den jeweiligen Rechner typ leistungsmäßig abgestimmt ist. Im mc-modular-AT wird mindestens ein 200 Watt Netzteil verwendet, welches im folgenden etwas näher betrachtet werden soll.

Müßte man diese Leistung an geforderten Versorgungsspannungen (+5 V/15 A, -5 V/1 A, +12 V/5 A und -12 V/1 A) mit einem herkömmlichen Transformator übertragen, so benötigte man dazu sicher einen „Mammut-Transformator“, der allein schon das Rechnergehäuse ausfüllen würde – vom Gewicht einmal ganz zu schweigen! Deshalb verwendet man ausschließlich getaktete Netzteile.

Schaltungsbeschreibung

Betrachten Sie die Netzteilschaltung (Abb. 1). Die Eingangsspannung (220 V) durchläuft eine Eingangssiebung zur Verminderung von Störeinstreuungen, bestehend aus R_1 , C_1 , T_1 (Trafo, der als Drossel geschaltet ist), C_2 und C_3 , wobei die

Eingangsspannung mit C_2 und C_3 gegen Masse halbiert wird, (T_2 dient nur der Anpassung an eine 110 V Netzspannung). Diese wird dann zweiweg-gleichgerichtet und mittels R_2 , C_5 , R_3 und C_6 geglättet und ebenfalls gegen Masse halbiert, wodurch an den beiden Leistungstransistoren Q_1 und Q_2 jeweils $\frac{1}{2}U$ Gleichspannung anliegt.

Die Transistoren arbeiten als Schalter im Gegentaktbetrieb und werden vom Impulsbreitenmodulator IC 494 (Abb. 2) über T_3 und T_4 gesteuert.

Im Primärkreis des Leistungsübertägers T_5 kommt somit ein geregeltes (getaktes) Wechselsignal (Abb. 3) zustande.

IC 494 gibt dazu an seinen Ausgängen (8) und (11) ein Sägezahnsignal aus, das in Abhängigkeit zu den Ausgangsspannungen die Regelung des Primärsignals an T_5 bewirkt.

Das Tastverhältnis wird mit C_{11} und R_{12} an (5) und (6) festgelegt. Vom IC wird eine Oszillationsfrequenz von 1 KHz bis 300 KHz zur Verfügung gestellt.

Außerdem überwacht es noch die Betriebsspannung (12) sowie die Ausgangsspannungen mittels zweier integrierter OPs und es gibt ferner die Referenzspannung aus.

Diese ERROR AMPs arbeiten wie ein Komparator, sie vergleichen einen Ist-

wert mit einem Sollwert und werden daraufhin aktiv.

Fällt am Ausgang eine Spannung aufgrund höherer Belastung ab, so wird die Primärseite von T_5 und damit auch die Ausgangsspannung sofort nachgeregelt, indem das Tastverhältnis verändert wird.

Dazu ist vom Ausgang eines jeden Spannungswertes eine Kontrollzuleitung, die über eine bestimmte Schaltlogik miteinander verknüpft ist, zu den OP-Eingängen geführt.

Sekundär sind am Trafo zwei Spannungsabgriffe für die 5 V und 12 V Spannungen gegeben, die gleichgerichtet und gesiebt vier Gleichspannungswerte (bezüglich Masse) liefern.

Die Gleichrichtung beider Abgriffe besteht wiederum aus je einer Vollwellengleichrichtung, die aus paarweise unterschiedlichen Dioden, wegen der verschiedenen Stromstärken, in den positiven und den negativen Zweigen aufgebaut sind. Die Siebung stellt in jedem Zweig einen Tiefpaß dar, der in seiner Dimensionierung den einzelnen Zweigen angepaßt ist.

(Die Emitterschaltung bei PG (Power Good) dient zur Überwachung der am BUS anliegenden Spannungen.)

Die Netzteile sind so beschaffen, daß sie nur unter Belastung die angegebenen Spannungswerte erreichen.

Obwohl die Netzteile sehr großzügig ausgelegt sind, kann es trotzdem zu Überlastungen und somit zu Defekten kommen. Um so ein defektes Netzteil wieder instand setzen zu können, benötigt man folgende Geräte:

Multimeter (am Besten ein digitales mit integriertem Prüfsummer), Oszilloskop, 2 Hochlastwiderstände (ca. 5 Ohm/50 Watt) für die Belastung der Ausgänge und ein einfaches regelbares Netzgerät.

Es genügt zur Feststellung der Betriebsfähigkeit, wenn man nur die Ausgänge +5 V und +12 V mit diesen Widerständen belastet.

Zunächst versucht man den Fehler auf den Bereich vor oder hinter dem Leitungsüberträger T₅ einzugrenzen. Dazu geht man mit dem Oszilloskop (größter Meßbereich und Vorwiderstand) 3.9 MOhm) an die Primäranschlüsse von T₅ und schaut ob sich dort überhaupt ein Signal befindet.

Liegt hier noch ein richtiges Signal an, so untersucht man die Sekundärseite näher. Enthält das Signal Störungen oder ist es gar nicht vorhanden, so ist bereits vorher etwas defekt. Im letzteren Fall beginnt man am Zweckmäßigsten von vorn mit der Überprüfung der Sicherung und der Netzspannung.

Anhand eines mir jetzt vorliegenden, defekten Netzteils möchte ich einige wichtige Fehlersymptome beschreiben.

Nach dem Öffnen und Ausbauen der Platine finde ich eine völlig schwarze Sicherung vor. Ich ersetze diese, klemme die Belastungswiderstände an und schalte ein. Die Sicherung brennt erneut durch.

Man könnte jetzt natürlich jedes Bauteil einzeln auslöten und auf seine Gebrauchsfähigkeit hin prüfen. Jedoch fallen die Eingangswiderstände sowie T₁ als Fehlerquelle aus, da diese als in Reihe geschaltete Bauelemente keinen Kurzschluß verursachen können. Mit dem Prüfsummer stellt sich heraus, daß C₁ durchschlägt (ein Defekt, der oft vorkommt).

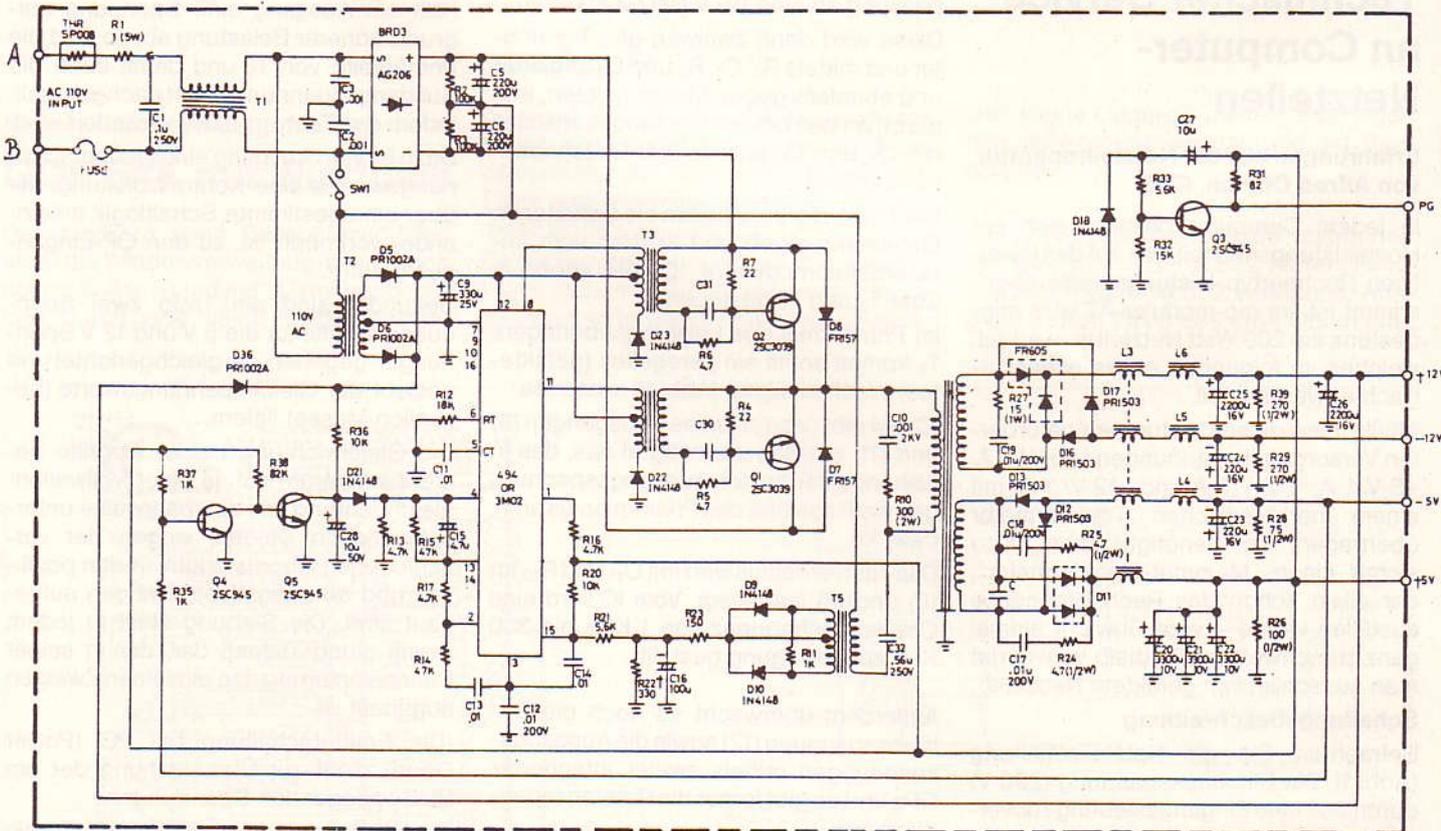
Da man ja nicht weiß, ob nicht noch ein Kurzschluß voranden ist, überbrückt man die Sicherungsclips mit dem Ampere-Meter (größten Meßbereich einstellen!). So spart man Sicherungen. Jetzt schaltet man wieder ein – das Meßgerät zeigt Vollausschlag. Also ist mindestens noch ein derartiger Fehler vorhanden.

Es könnte natürlich ein weiterer Kondensator oder der Gleichrichter sein, jedoch kommt dies erfahrungsgemäß selten vor. Es bietet sich als nächstes die Überprüfung der Leistungstransistoren an, da

diese mit am stärksten beansprucht sind. Zum Testen müssen diese, genauso wie die anderen Bauteile, ausgelötet und mit dem Ohmmeter geprüft werden.

Bei beiden ist die Emitter-Kollektor-Strecke leitend, was den hohen Strom beim Einschaltvorgang erklärt (diese Strecke muß normalerweise einen sehr, sehr hohen Widerstand aufweisen). Nachdem die defekten Teile ausgetauscht wurden, schaltet man wieder ein – das Ampere-Meter zeigt 0,5 A (Achtung, dies ist kein allgemein gültiger Wert eines intakten Gerätes!), es bedeutet, daß man jetzt ohne weiteres eine Sicherung einsetzen kann. Es liegt kein Kurzschluß mehr vor.

Nun vergleicht man das Primärsignal an T₅ mit dem richtigen Signal und hier stelle ich deutliche Störungen in den Anstiegsflanken des Rechtecksignals, in Form von gestrichelten Linien fest. Daraufhin sehe ich mir die Ausgangssignale des ICs an. An (8) und (11) müßte ein exaktes Sägezahnsignal anliegen. Dies ist nicht der Fall. So ersetzt man das IC und sieht sich nochmals die Signale an. Das Signal an T₅ ist nun in Ordnung, aber an den Ausgängen stellt sich keine Spannung ein. Nun klemmt man die Widerstände ab und legt nach und nach an alle Ausgänge eine



NOTE: 1. SW1 ON=110V OFF=220V
2. FUSE 230V 15A, 110V 3A.

entsprechende positive oder negative Prüfspannung (ca. 5 V) mit dem Labornetzgerät an.

Bricht an irgendeinem Zweig diese Spannung zusammen, so ist in diesem Zweig ein Kurzschluß vorhanden, aufgrund defekter Siebelemente oder die Dioden sperren nicht mehr ordnungsgemäß.

Bleibt die Prüfspannung bestehen, so verfolgt man mit dem Voltmeter den Zweig bis zu dem Punkt, an dem keine Spannung mehr anliegt. Damit kann man defekte Drosseln finden.

Im +12 V Zweig stellt sich eine defekte Diode vom Typ FR605 und im +5 V Zweig der DIAC C_{11} als defekt heraus.

Hätte die Prüfspannungsmethode alle Bauteile als funktionsfähig bestätigt, so wären als Fehlerquellen nur noch C_{19} und T_5 übrig geblieben, was aufgrund einschlägiger Erfahrungen recht unwahrscheinlich ist.

Das Netzgerät wäre nun soweit wieder betriebsfähig. Zur Sicherheit überprüft man jetzt noch alle Ausgangsspannungen mit dem Oszilloskop auf Wechselanteile. Es kann nämlich vorkommen, daß die Betriebsspannungen zwar in etwa erreicht werden, die Wechselanteile (auch wenn sie nur gering sind) aber zu Störungen auf dem Bildschirm führen können. Im schlimmsten Fall ist der Rechner nicht voll betriebsfähig. Treten solche Wechselanteile an den Ausgängen auf (am stärksten treten sie in den 12 V Zweigen in Erscheinung), so liegt dies meistens an den Siebelkos C_5 und C_6 nach dem Gleichrichterblock.

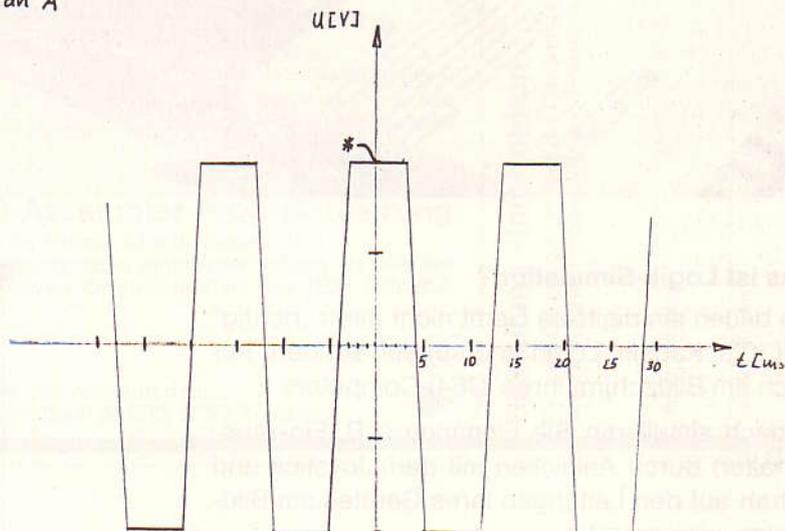
Dieser Artikel umfaßt natürlich keineswegs annähernd die gesamte Materie getakteter Netzteile und deren Fehlersymptome. Dazu sei auf die umfassende Literatur der Leistungselektronik verwiesen. Aber sicher hilft es etwas zum Einstieg in dieses Gebiet und ermöglicht vielleicht den einen oder anderen Fehler selbst zu lokalisieren und zu beheben, da man selten ein defektes Gerät vorfindet, das auf einen Schlag gleich so vielfältige Fehler, wie in diesem Beispiel beschrieben, enthält. Oft sind es nur Kleinigkeiten, wie

eben ein defekter Eingangskondensator C_1 , der die gesamte Schaltung lahm legt. Noch ein Hinweis zu Abb. 3: Wenn der Netzstecker in der Steckdose so steckt, daß die Phase am Eingang auf „B“ liegt, so

erhält man den Kurvenverlauf gemäß Abb. 3. Stecken Sie einfach den Stecker anders herum in die Steckdose und Sie erhalten das Rechtecksignal wie in Abb. 3a aufgezeichnet auf Ihrem Oszilloskop.

Abb. 3a

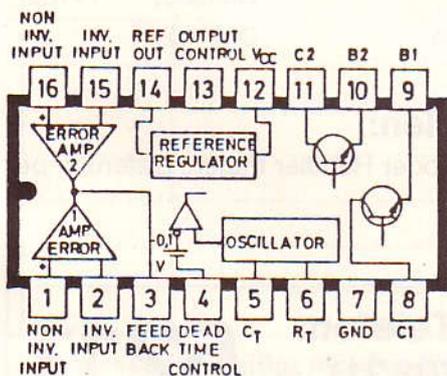
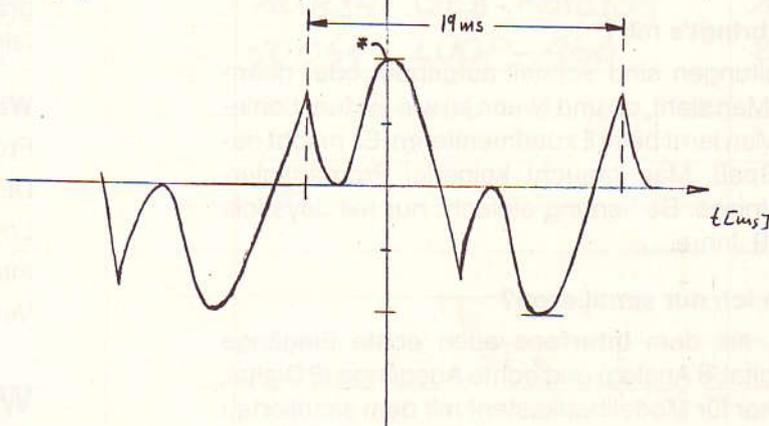
Phase an A

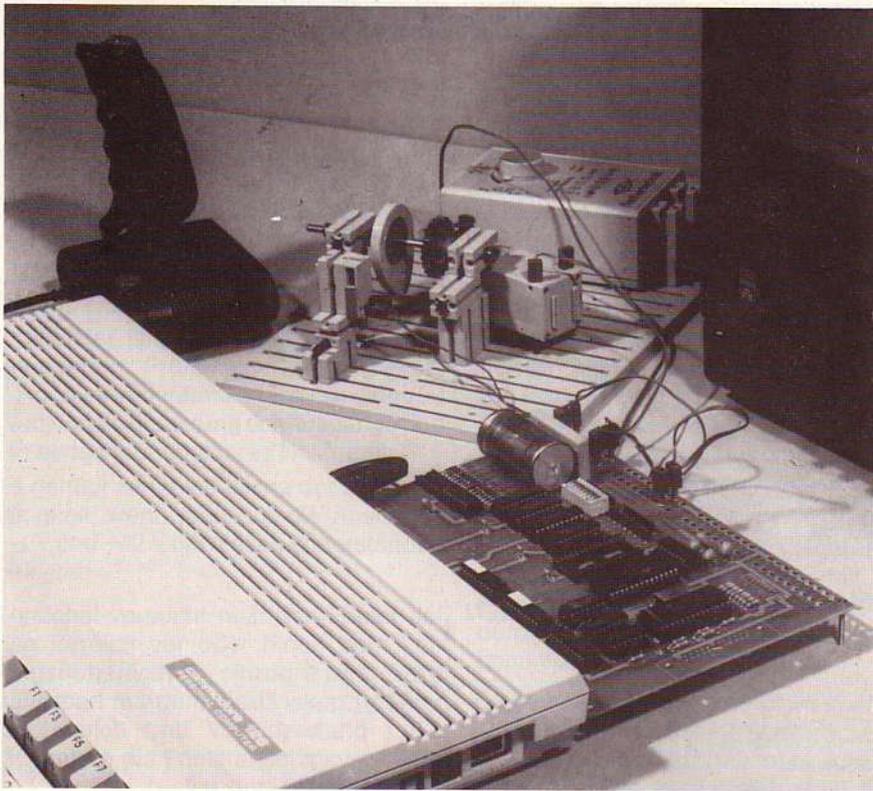


* $U = 20V$ mit $6.8\ M\Omega$ Vorwiderstand!

Abb. 3b

Phase an B





GRAF[®] computer

LogSim C64

Was ist Logik-Simulation?

Sie bauen ein digitales Gerät nicht mehr „richtig“ (mit IC's, Kabeln, Lötens und so) auf, sondern nur noch am Bildschirm Ihres (C64) Computers.

Danach simulieren Sie Eingänge (z.B. Ein-/Aus-schalter) durch Anklicken mit dem Joystick und **sehen** auf den Leitungen Ihres Gerätes am Bildschirm, was passiert.

Welche Geräte kann ich simulieren?

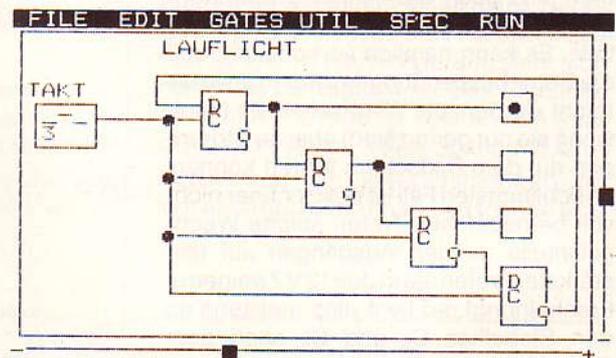
Alle Geräte, die sich auf digitale Grundelemente (AND, OR, NOR, NOT, NAND, D-FLIP-FLOP, COUNTER, TIMER, ANALOG-INPUT usw.) zurückführen lassen.

Was bringt's mir?

Schaltungen sind schnell aufgebaut oder geändert. Man sieht, ob und (wenn ja) wie sie funktionieren. Man lernt beim Experimentieren. Es macht riesig Spaß. Man braucht keinerlei Programmierkenntnisse. Bedienung einfach: nur mit Joystick. 9 - 99 Jahre.

Kann ich nur simulieren?

Nein, mit dem **Interface** auch echte Eingänge (9 Digital, 8 Analog) und echte Ausgänge (8 Digital, 4 Motor für Modellbaukasten) mit dem simulierten Gerät verbinden. Interface funktioniert auch mit BASIC usw.



Was brauche ich?

Einen C64, Joystick, Disk und das LogSim-Programm (nur simulieren) oder das Interface dazu (simulieren und steuern).

Was kostet es?

Prospekt LogSim	kostenl.	Best.-Nr.
DEMO-Disk mit Handbuch	DM 10,-	11188
LogSim Programm	DM 198,-	11187
Interface	DM 398,-	11189
Versandkosten	DM 2,-	

Wir bestellen:

per Postkarte oder Händler fragen. Lieferung per Nachnahme.

GRAF ELEKTRONIK SYSTEME GMBH

Magnusstraße 13 · Postfach 1610 · 8960 Kempten (Allgäu) · Telefon: (0831) 6211
Teletex: 831804=GRAF · Telex: 17831804=GRAF · Datentelefon: (0831) 69330

Filiale Hamburg: Ehrenbergstraße 56 · Tel. (040) 388151 · Filiale München: Georgenstraße 61 · Tel. (089) 2715858

Telefon:
(0831) **6211**

Lehrgänge zum NDR-Computer

Wenn Sie sich intensiv mit der Hard- und Software des NDR-Computers beschäftigen wollen, helfen Ihnen diese Kurse weiter:



Kompakt-Kurs Elektronik

Ca. 200 Seiten im Format A4 mit Experimentiermaterial, Tonbandkassette und Flip-chart zur Einführung in die Grundlagen der Elektrotechnik und Halbleiterphysik.

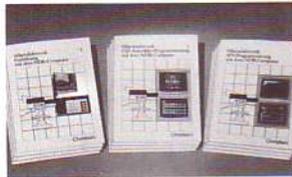
Der Kompakt-Kurs ist ideal für alle, die sich nicht nur mit dem Nachbauen von Schaltungen zufrieden geben, sondern auch selbst Schaltungen ändern, ergänzen oder entwerfen wollen.



Kompakt-Kurs BASIC

Ca. 200 Seiten im Format A4 mit Abschlußtest.

Der Lehrgang behandelt alle gängigen BASIC-Anweisungen und führt Sie anhand von zahlreichen Beispielprogrammen in die ersten Schritte der BASIC-Programmierung ein.



Kompakt-Kurs Mikroelektronik - Einführung

Ca. 240 Seiten Lehrmaterial im Format A4 mit Abschlußtest.

Der Lehrgang zeigt anhand des NDR-Einsteigerpakets, wie Sie Ihren Computer in Maschinensprache programmieren. Jeder Befehl wird erläutert und anhand von Beispielen lernen Sie den Einsatz der Maschinensprache des Z80 lernen.



Kompakt-Kurs Z80-Assembler-Programmierung

Ca. 240 Seiten Lehrmaterial im Format A4 mit Abschlußtest.

Wenn Sie sich in die Assemblersprache einarbeiten wollen, ist dies der richtige Kurs für Sie. Als Hardware-Grundlage dient das ZEAT-Betriebssystem.

Christiani

Bitte Bestellschein abtrennen und einsenden an:
Dr.-Ing. P. Christiani GmbH, Postfach 35000, 7750 Konstanz



Bestellung · Information

Senden Sie mir gleich über die Christiani Kurse ausführliches, kostenloses Informationsmaterial wie Lehrpläne, Probeseiten und Preisliste.

Name, Vorname

Straße, Nummer

PLZ, Ort

35 519

BESTELLKARTE

Ich / Wir bestelle(n) unter Anerkennung Ihrer Geschäfts- und Lieferungsbedingungen folgende Artikel:

Stück	Bestell-Nr.	Bezeichnung	Einzelpreis
	10 834	GES-Katalog	10,-
	10061	LOOP-Abo	20,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift

Bei Minderjährigen die des gesetzl. Vertreters

BESTELLKARTE

Ich / Wir bestelle(n) unter Anerkennung Ihrer Geschäfts- und Lieferungsbedingungen folgende Artikel:

Stück	Bestell-Nr.	Bezeichnung	Einzelpreis
	10 834	GES-Katalog	10,-
	10061	LOOP-Abo	20,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift

Bei Minderjährigen die des gesetzl. Vertreters

Neue Produkte – Neue Preise!

Das neue Grundprogramm für 680xx-Systeme

Best.-Nr.	Beschreibung	Preis DM
11218	68008	95,-
11219	68000	95,-
11220	Quelle auf Diskette, 5 1/4" 80 Spuren . .	60,-
11221	3 1/2" 80 Spuren	60,-

JEDI – Der preiswerte JADOS-Editor

Best.-Nr.	Beschreibung	Preis DM
11192	5 1/4" 80 Spuren	49,-
11193	3 1/2" 80 Spuren	49,-

COL-Edit – Text-Editor für die COL256

Best.-Nr.	Beschreibung	Preis DM
11191	5 1/4" 80 Spuren (JADOS)	98,-
11190	3 1/2" 80 Spuren (JADOS)	98,-

LOG 16 – Der Logik-Analysator, nun auch für 32 Kanäle

Best.-Nr.	Beschreibung	Preis DM
10906	Fertiggerät mit Software	598,-
10902	Bausatz mit SW	498,-
10898	Leiterplatte mit SW	298,-

Die mitgelieferte Software beinhaltet bereits die 32-Kanal-Lösung. Es werden dazu zwei Baugruppen benötigt.

Neues Monitor-Programm für das Einsteigerpaket unterstützt viele Baugruppen

Best.-Nr.	Beschreibung	Preis DM
10876	Monitor-Eprom 2764 (SBC3)	40,-
10912	Monitor-Eprom 2732 (SBC2)	40,-
11184	TOOL-Handbuch mit Listing und Beschreibung der TOOL-Programme . .	20,-
	Update – für Besitzer der alten Monitor-Version (Re.-Nr. angeben) . . .	20,-

Sprachausgabe für den NDR-Computer

Best.-Nr.	Beschreibung	Preis DM
11195	Diskette (JADOS) 5 1/4" 80 Spuren . . .	48,-
11196	Diskette (JADOS) 3 1/2" 80 Spuren . . .	48,-
11194	EPROM für 68008	48,-

Sprachausgabe für ein Subsystem

Best.-Nr.	Beschreibung	Preis DM
11138	EPROM für 68008	48,-

GRAF ELEKTRONIK SYSTEME GMBH
 Magnusstr. 13 · Postfach 1610 · 8960 Kempten
 Telefon (0831) 6211 · Teletex: 831804 = GRAF

Bitte
Porto
nicht
vergessen

**BESTELL-
POSTKARTE**

**GRAF
computer**

Graf Elektronik Systeme GmbH
 Postfach 1610
8960 Kempten

Anschrift:

Lieferform: Nachnahme Vorkasse
 Bankeinzug

Bankeinzug: Hiermit ermächtige ich die Firma GES GmbH, den Rechnungsbetrag für die auf dieser Karte angegebenen Bestellungen von meinem Konto:

BLZ _____ Konto-Nr. _____

Bank: _____
 abzubuchen. Falls mein Konto die erforderliche Deckung nicht aufweist, besteht seitens des kontoführenden Kreditinstitutes keine Verpflichtung zur Einlösung.

Datum _____ Unterschrift _____

Bitte
Porto
nicht
vergessen

**BESTELL-
POSTKARTE**

**GRAF
computer**

Graf Elektronik Systeme GmbH
 Postfach 1610
8960 Kempten

Anschrift:

Lieferform: Nachnahme Vorkasse
 Bankeinzug

Bankeinzug: Hiermit ermächtige ich die Firma GES GmbH, den Rechnungsbetrag für die auf dieser Karte angegebenen Bestellungen von meinem Konto:

BLZ _____ Konto-Nr. _____

Bank: _____
 abzubuchen. Falls mein Konto die erforderliche Deckung nicht aufweist, besteht seitens des kontoführenden Kreditinstitutes keine Verpflichtung zur Einlösung.

Datum _____ Unterschrift _____