1.4.89

LOOP

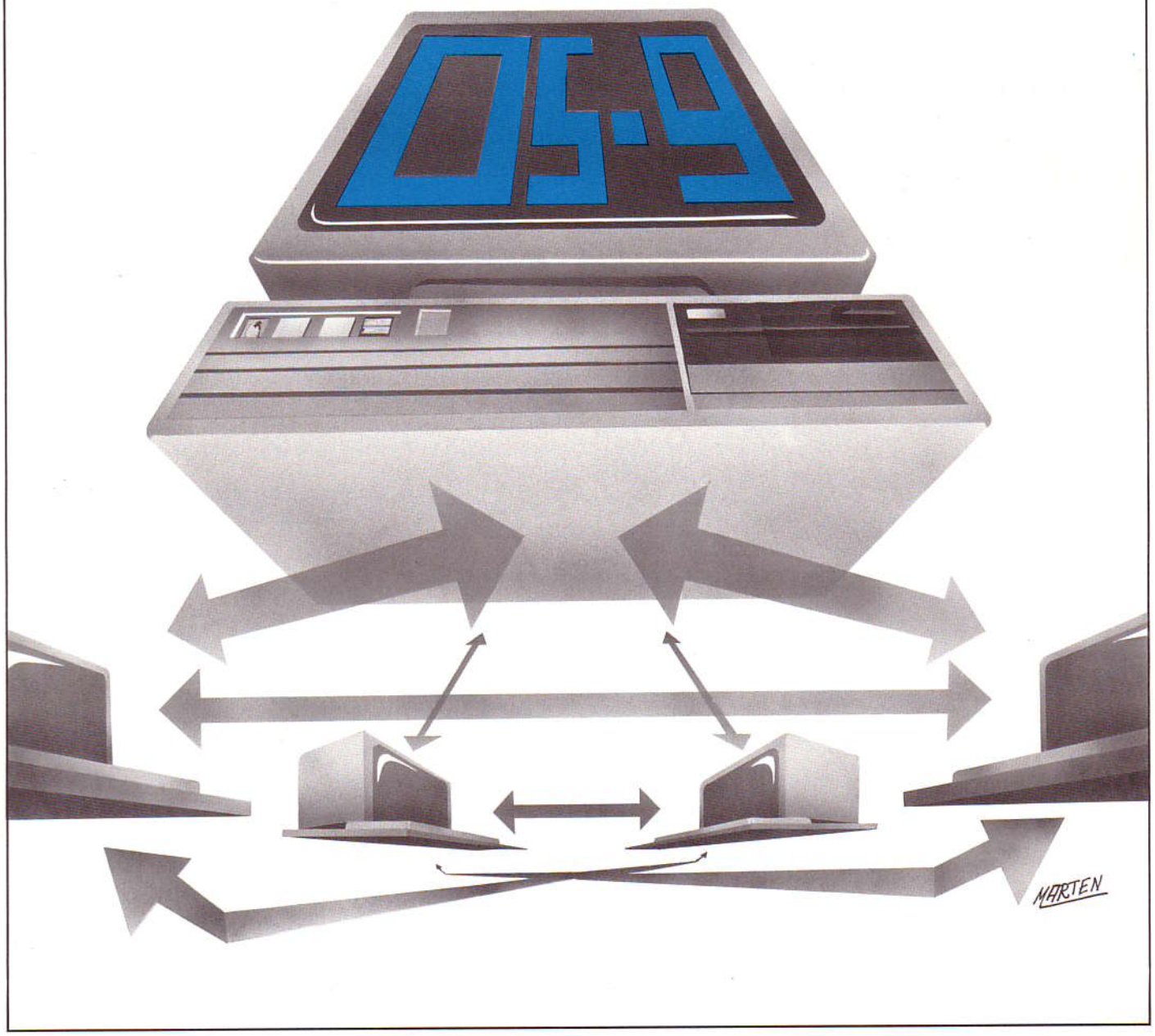
21

März 1989

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

DM 3,50

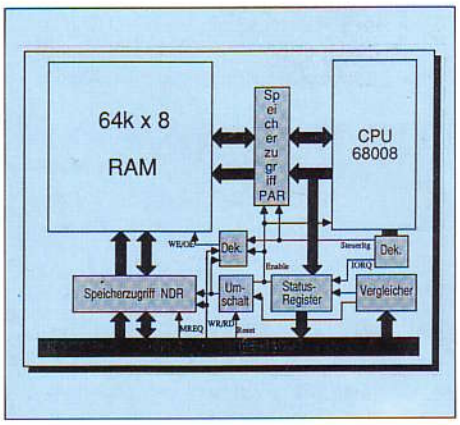
OS/9
Multitasking und Multiuser auf dem NDR Computer



Leitartikel

Parallel-Rechner für die CPU68K

Es ist möglich!
Tobias Thiel entwickelte einen Parallel-Rechner um der CPU68K unter die Arme greifen zu lassen.
So sind Geschwindigkeitssteigerungen bis zum Zweieinhalbfachen möglich.
Die Kosten pro Parallel-Rechner lassen sich dabei auf ca. 200 DM begrenzen.



Blockschaltbild 21/4

Buskopp-Baugruppe
NDR-Systeme als MS-DOS-Rechner 21/7

CPU Z80

Wunschkonzert mit dem NDR-Computer
Das Radio als Experimentierfeld 21/8

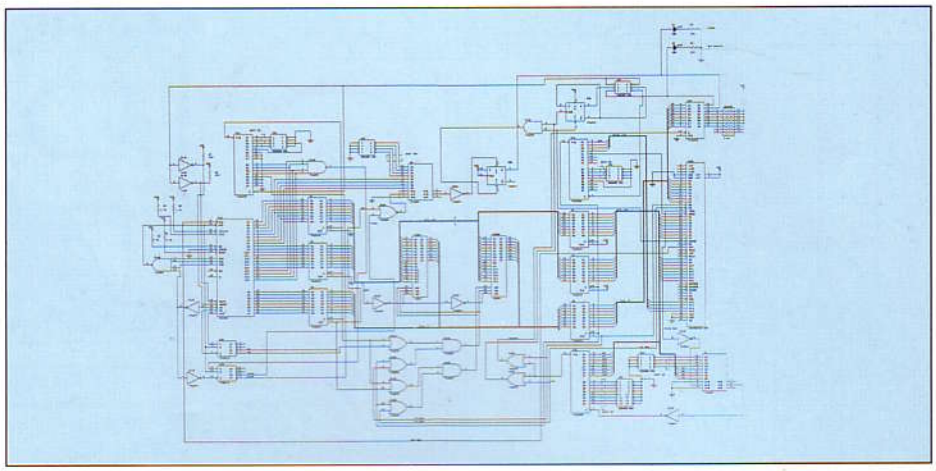
Ökosystem eines Interpreters
Im zweiten Teil gibt es Werkzeug zum Knacken 21/10

Analog/Digital-Wandler für den NDR-Computer
Einfach und preiswert 21/13

NIMM-Spiel
Programmlisting in GOSI 21/14

Der NDR-Computer "bellt"
Mit etwas Grips piepst es auch bei Ihnen 21/16

Grafik mit FLOMON
MBASIC lernt zeichnen 21/17



Schaltplan des Parallelrechners

CPU 680XX

Suchprogramm für C-Compiler
Große Hilfe beim Editieren 21/24

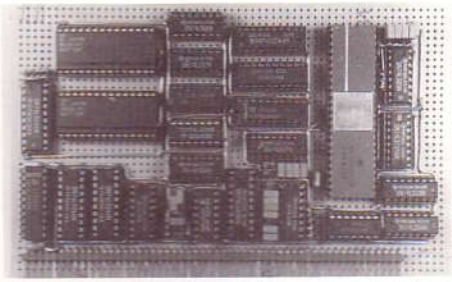
JMS-Paint
In neunmonatiger Entwicklungszeit entstand ein Grafik-Programm, daß keine Wünsche offen läßt. 21/25

Sage mir, was Du hast 21/28

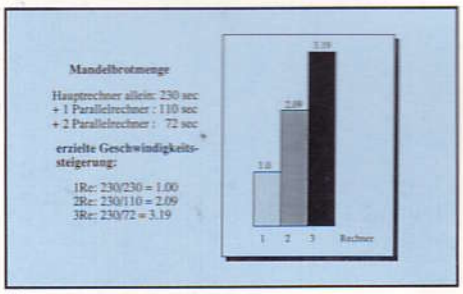
Das 68020-Grundprogramm Version 6.1
Nach der Überarbeitung gibt es die vielen neuen Möglichkeiten auch für den 68020. 21/30

OS-9
Übersicht über die Dienstprogramme 21/31

Mit JMS-Paint gemalte Bilder



Vorderseite des Parallel-Rechners

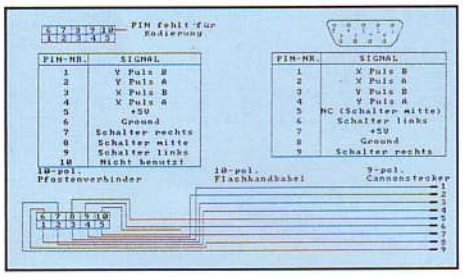


Überblick über die Rechenzeiten

mc-modular-AT

Zweimal Laufwerk B?
Der Controller wird überlistet 21/34

Die IBM-Kompatible Atarimaus
Wie man eine Maus zur Zusammenarbeit überredet 21/36



Anschlußbelegungen von Maus und Interface

Rubriken

Editorial 21/3

In eigener Sache
GES auf der CeBIT in Hannover 21/20

Loop-Titelbild

Kleinanzeigen / Ergänzungen 21/38

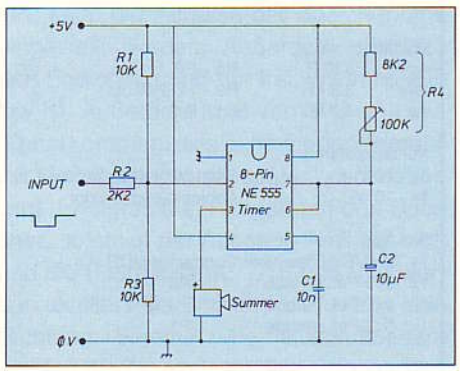
2988 - Comic-Strip 21/27

Technische Hinweise
Bootet Ihre CPU8088 nicht, oder ... 21/37

Für Sie gelesen:

Einführung und Anwendung von LISP 21/33

Von "C" zu Modula-2 und zurück 21/30



Tönender NDR-Computer mit einfacher Schaltung realisiert

Editorial

Mehr MIPS

In der Industrie zeigt sich derzeit ein eindeutiger Trend zu mehr MIPS. MIPS ist die Abkürzung für Million Instructions Per Second. Es gibt immer noch schnellere Rechner. Kaum ist der 386 da, schon folgt der 486, beim 68020 folgte kaum der 68030 und schon ist vom 68040 die Rede. Da ein PC immer auf den Prozessoren der Serie 808xx basieren muß, werden dort auch Co-Prozessoren zur Beschleunigung eingesetzt. So gibt es Transputerkarten, Arithmetik-Beschleuniger (z.B. Weitek) oder andere schnelle Prozessoren, die sonst nicht einsetzbar wären. Dieses Konzept läßt sich natürlich auch auf den NDR-Computer übertragen. So wartet schon eine Baugruppe mit einem T800 und 25MHz auf dem Labortisch auf ihren Einsatz.
Aber Transputer sind noch teuer und für viele der NDR-Benutzer wohl nicht erschwinglich (Der T800 G25 kostet nur als Chip schon mehr als 1000.- DM).

Es gibt aber auch noch andere Möglichkeiten, Rechner zu beschleunigen. Z.B. mit einem ganz normalen 68008, so wie in diesem Heft an andere Stelle beschrieben. Die Prozessoreinheit läßt sich nicht nur mit der Hauptcpu 68008 zusammen verwenden, sondern auch mit anderen Prozessoren, wie zum Beispiel der 8088-Baugruppe, so daß man nun endlich einmal zwei CPUs auf dem Bus gleichzeitig betreiben kann (die sogar gleichzeitig rechnen können). Es läßt sich sogar mehr als **ein** solcher Parallelprozessor einsetzen, so daß nun dem NDR-Rechner einer großen Erweiterung nichts mehr im Wege steht. Die Programmierung kann in Assembler erfolgen, entweder mit dem 680xx-Grundprogramm oder bei MS-DOS mit einem 68000-Cross-Assembler, der bei Bedarf als Public-Domain besorgt werden kann. Wenn Sie Interesse an dieser Baugruppe haben, so schreiben Sie uns einfach.

Wenn sich genügend viele Interessierte melden, so kann eine Leiterplatte dazu entwickelt werden. Das Gleiche gilt auch für die Transputer-Karte. Für den NDR-PC ist gerade eine EGA-Karte mit dem G2-Chip (GC201) in Vorbereitung, so daß dann der Bus-Adapter entfallen kann. Die Baugruppe unterstützt eine Auflösung von maximal 800x600 Punkten bei 16 Farben und ist voll EGA, CGA und Herkules kompatibel. Als Speicher sind 256K Byte auf der Karte vorhanden. Es ist geplant, die Baugruppe auch vom Grundprogramm 680xx zu unterstützten, so wie ggf. auch von FLOMON.

So könnte vielleicht endlich einmal die alte GDP-Karte weichen. Die EGA-Baugruppe wird allerdings Vektoren nicht mehr so schnell zeichnen können wie bei der GDP, da kein Graphik-Prozessor vorhanden ist und alle Zeichenbefehle von der Hauptcpu per Software durchgeführt werden müssen. Dafür ist aber die Textdarstellung sehr schnell (Scroll) und ein programmierbarer Zeichensatz ist in dem Chip auch vorgesehen. Für den MS-DOS-Betrieb wird diese Baugruppe auch mit Treibern für Autocad, GEM, Windows etc. geliefert.

Rolf-Dieter Klein

Tobias Thiel

Parallelrechner für die CPU68K

Wer hat nicht schon stundenlang darauf gewartet, daß der Computer ein Apfelmännchen ausrechnet? Während einer solchen Wartezeit faßte ich den Entschluß da für mich als Schüler ein Transputer nicht in Frage kommt, mit einem anderen Parallerechner des NDR-Computer zu bechleunigen. Dieser Parallelrechner sollte der CPU 68K sozusagen unter die Arme greifen und für sie verschiedene Arbeiten übernehmen können. Also zeichnete ich einen Schaltplan, verkroch mich hinter den Lötkolben und baute einen Parallelrechner...

Parallelrechner - Wie funktioniert er ?

Bild 1 gibt dazu eine grobe Übersicht: Der Parallelrechner besteht im wesentlichen aus einer CPU und einem Speicher. Das Besondere an dem Speicher ist, daß er zwischen dem NDR-Bus und der CPU des Parallelrechners umgeschaltet werden kann. Dies geschieht, wenn auf eine bestimmte Adresse, die 12 Bit umfaßt, etwas Beliebiges geschrieben wird. Schreibt nun die CPU des NDR-Computers ein Programm in diesen Speicher, so kann der Parallelrechner dieses nach dem Umschalten des Speichers abarbeiten. Damit der NDR-Rechner herausfindet, wer gerade auf den Speicher zugreifen kann oder ob der Parallelrechner bereits mit seinem Programm fertig ist, ist ein Statusregister notwendig, über das diese Informationen abgefragt werden können.

Wenn er fertig ist, wird der Speicher wieder zurückgeschaltet, und das Ergebnis vom NDR-Computer aus dem Speicher ausgelesen. Während der Parallelrechner sein Programm abarbeitet, kann der NDR-Computer sich bereits einer anderen

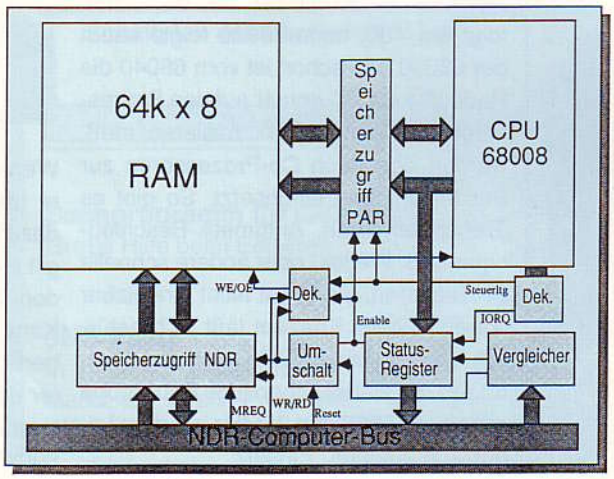


Bild 1: Blockschaltbild des Parallel-Computers

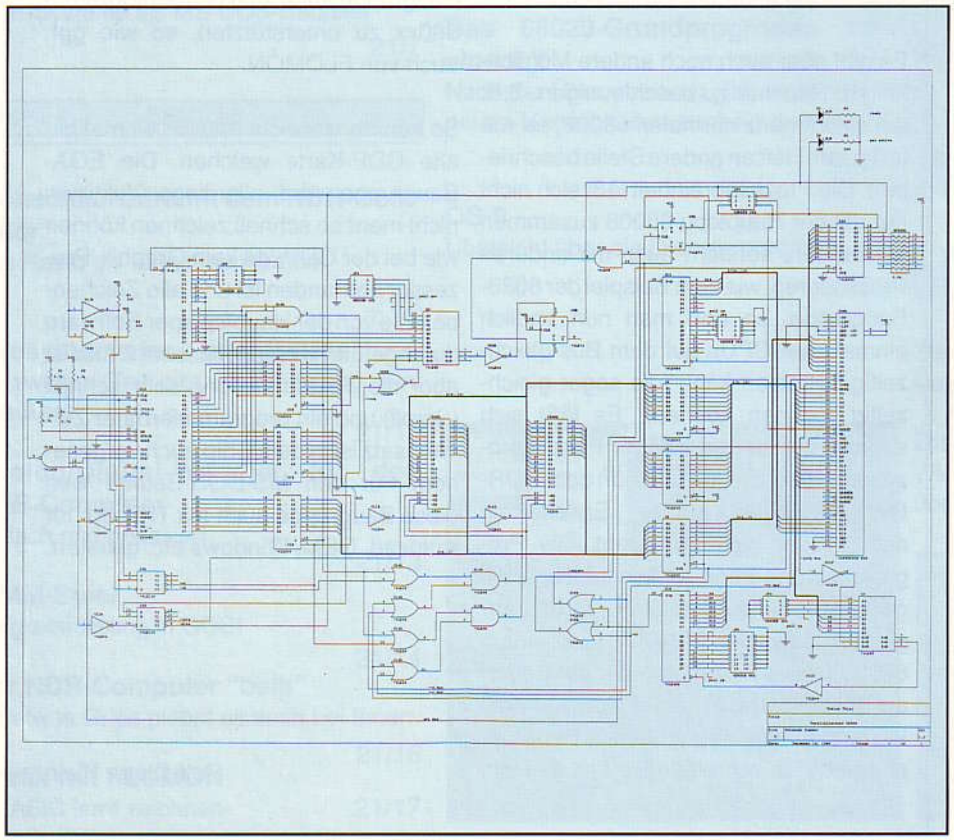


Bild 2: Schaltplan des Parallelrechners

Stückliste für den 68008 Parallelrechner

Halbleiter:

2x	HM 62256SRAM1, SRAM2	
1x	MC 68008U8	
2x	SN 74 LS 04	U7, U22
1x	SN 74 06 U11	
1x	SN 74 LS 08	U9
1x	SN 74 LS 20	U6
1x	SN 74 LS 28	U15
2x	SN 74 LS 32	U10, U13
1x	SN 74 LS 74	U4
4x	SN 74 LS 85	U3, U14, U21, U23
1x	SN 74 LS 139	U5
5x	SN 74 LS 244	U16 - U20
2x	SN 74 LS 245	U1, U2
1x	SN 74 LS 688	U12
2x	LED	D1, D2

Wiederstände:

4x	330 R	R5 - R8
24x	3.3 kR	R9 - 14, Pull up
4x	4.7 kR	R1 - R4

Kondensatoren:

5x	100 nF	Entstörkondensatoren
1x	10 uF	Abblockkondensator

Stecker:

1x	54x1 Pol. gewinkelt	Connector NDR
1x	2x3 Pol. gerade	JP3
4x	4x2 Pol. gerade	JP2, JP4 - JP6
1x	8x2 Pol. gerade	JP1

Sonstiges:

1x	Platine
9x	14 Pol. IC-Sockel
5x	16 Pol. IC-Sockel
8x	20 Pol. IC-Sockel
2x	28 Pol. IC-Sockel
1x	48 Pol. (= 2x 24 POl.) IC-Sockel

Aufgabe widmen.

Die Hardware des Parallelrechners

Am besten beschreibe ich die Hardware in der Reihenfolge, in der man beim Aufbau vorgehen sollte. Da sich der Parallel-Rechner wie eine ganz normale Speicherkarte mit 64kB RAM verhält, solange der Speicher nicht umgeschaltet wird, ist es sinnvoll, ihn zuerst als eine Speicherkarte aufzubauen, die dann erweitert wird.

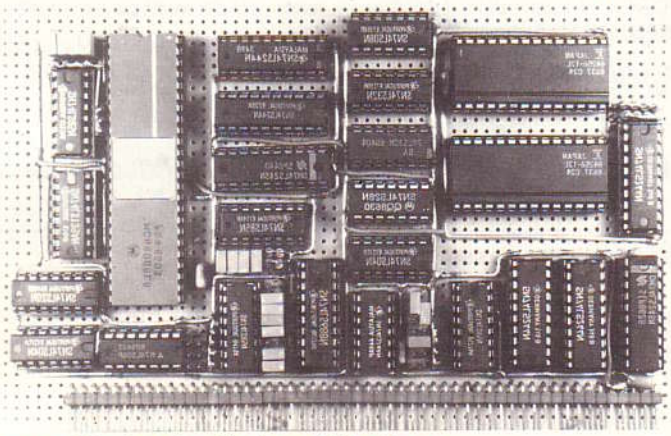


Bild 3: Vorderseite des Parallelrechners

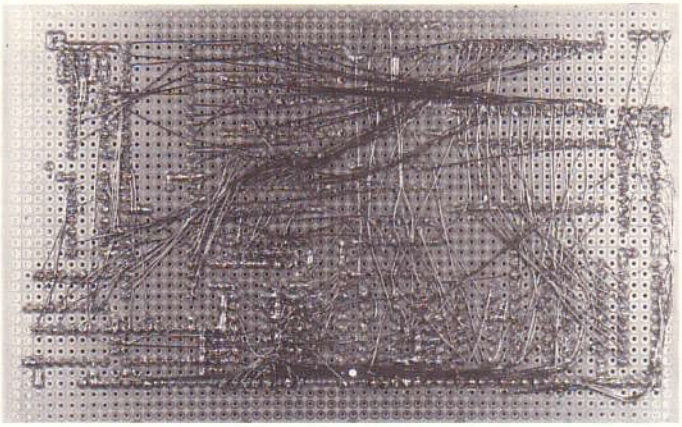


Bild 4: Rückseite des Rechners in Fädeltechnik

Bild 2 zeigt die Schaltung. Außer den Speicherbausteinen ist hier der Vergleicher U23 das Wichtigste. Über die Stiftleiste JP2 kann die Adresse der Speicherkarte eingestellt werden. Außerdem gibt U23 den Speicherzugriff nur frei, wenn MREQ low ist. Außerdem muß von JP3 ein Low-Signal kommen; dies gehört jedoch bereits zur Umschaltlogik, die später beschrieben wird. Erfolgt nun ein Zugriff auf den Speicher, so steigt der Ausgang A=B auf high und die Bustreiber U2, U16 und U17 werden aktiviert. Da U16 und U17 nur in eine Richtung Signale übermitteln können, werden die Adressleitungen nur vom Bus zu den Speicherbausteinen durchgeschaltet. Der DIR-Eingang von U6 ist mit der Read-Leitung des Busses verbunden, so daß die Daten beim Lesen von den Speicherbausteinen zum Bus, beim Schreiben vom Bus zum Speicher gelangen können. Die Signale für die Eingänge WE und OE der Rams werden von den IC`s U9 (A,B) und U10 aus den WR,RD und den Speicherzugriffsignalen beider Rechner generiert. Für die CS-Signale der Rams wird direkt A15 verwendet. Da nicht auf beide Speicherbausteine gleichzeitig zugegriffen wird, ist zwischen die beiden CS-Eingänge ein Inverter geschaltet. Wenn nun die Pins 1 und 9 des Bausteins U10 auf High und der Pin 8 des Bausteins U15 auf Low liegt, muß vom NDR-Bus auf den Speicher zugegriffen werden können.

Als nächstes sollte nun die Umschaltlogik aufgebaut werden. Auch hier spielen Vergleicher eine wichtige Rolle. Durch U3 und U12 werden die Signale von A4 bis A15 mit der an JP4 und JP1 eingestellten Adresse verglichen. Stimmen diese überein, so fällt der Ausgang P=Q an U12 auf Low. Durch U13 wird nun unterschieden, ob von der Adresse gelesen oder auf sie geschrieben werden soll. Beim Lesen wird der Inhalt des Statusregisters über U18 ausgegeben. Wird aber auf diese Adresse geschrieben, so gelangt ein kurzer Impuls an den CLK-Eingang des Flipflops U4A, das daraufhin den Q- und -Q-Ausgang invertiert. Diese beiden Signale bewirken nun, daß immer für eine CPU der Speicher gesperrt ist. Der CL-Eingang von U4A ist an RESET angeschlossen, so daß nach einem Reset von einer bestimmten CPU auf den Speicher zugegriffen werden kann. Welche das ist, kann an JP3 eingestellt werden.

Damit wäre auch die Umschaltlogik fertig, und nach einem Schreibvorgang auf die an JP1 und JP4 eingestellte Adresse müßte der Speicher aus dem Ram-Bereich verschwinden. Wird dieser Schreibvorgang wiederholt, erscheint er wieder.

Ein Teil des Statusregisters, nämlich der Vergleicher, wurde bereits beschrieben. Hinzu kommt noch der Bustreiber U18, über den der Registerinhalt ausgegeben wird, und eine 1-Bit Ausgabeeinheit, über den die Parallel-CPU eine Ready-Meldung schicken kann. Durch den Vergleicher U14 werden die Leitungen A4 bis A7 mit der an JP6 eingestellten Adresse verglichen. Wenn die Adressen übereinstimmen und zugleich sowohl IORQ und WR auf low sind, wird das Flipflop U4B gesetzt. Wird der Speicher umgeschaltet oder ein Reset ausgelöst wird es wieder gelöscht. Dieses Ready-Bit wird zusammen mit dem Speicher-Enable der Haupt-CPU über U18 ausgegeben. Zusätzlich wird der Inhalt des Statusregisters über zwei LED's für den Benutzer sichtbar gemacht.

Zum Schluß noch einige Worte zum CPU-Teil des Parallelrechners: Die Erzeugung der Signale RD, WR, IORQ und MREQ ist identisch mit der der CPU68k. Ich werde daher hier nicht näher darauf eingehen. Die Speicherzugriffslogik brauche ich auch nicht weiter beschreiben, da ihr Aufbau mit der ihr gegenüberliegenden übereinstimmt. Die Adresse des Speichers aus der Sicht des Parallelrechners muß aber $00 sein. Das Speicherfreigabesignal für den Parallelrechner wird dazu verwendet, auch die Signale RESET und HALT zu erzeugen. Solange die CPU nicht auf den Speicher zugreifen darf, wird sie lahmgelegt.

Bild 3 zeigt einen Prototyp in Fadeltechnik - alle Verbindungen mit Kupferlackdrähten (0.28mm), wie in Bild 4 zu sehen. Da ich für den Parallelrechner bis auf die CPU und die Speicherbausteine nur Standard-Bausteine benutzt habe, liegen die Materialkosten ca. bei 200 DM pro Rechner.

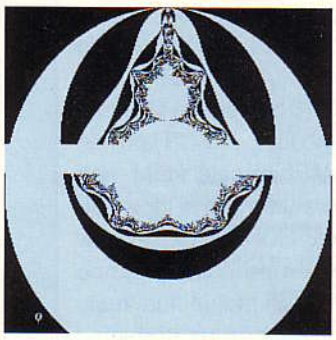


Bild 6: Programmablauf

Wie man den Parallelrechner programmiert

Zu beachten ist, daß der Speicher aus der Sicht des Parallelrechners bei $0 beginnt. Ansonsten kann der Parallelrechner in Assembler programmiert werden. Die ersten Zeilen eines solchen Parallelrechner-

programms sollten also aussehen wie auf Bild 8:

Als Programmierhilfe ist das Statusregister vorhanden, über das der Parallelrechner ein Ready-Signal ausgeben kann. Um dieses 1 Bit umfassende Signal auszugeben, muß der Parallelrechner auf die an JP6 eingestellte Adresse etwas Beliebiges schreiben.(Bild 9)

Bild 5 zeigt ein Programmbeispiel, die Berechnung des Apfelmännchens. Der Hauptrechner und der Parallelrechner arbeiten von zwei Seiten Richtung Mitte (Bild 6).

Dadurch, daß der Parallelrechner ohne Warte-Zyklen arbeitet, ist er bei Rechenaufgaben bis zu 1.5 mal so schnell wie der Hauptrechner. Mit einem Parallelrechner kann man also bis zu 2.5 mal so schnell rechnen wie ohne.

Muß der Hauptrechner noch die Ausgabe der Ergebnisse übernehmen, kann jedoch dieser Faktor geringer werden. Bild 7 zeigt dazu einige Meßergebnisse. Da man aber auch mehr als einen Parallelrechner gleichzeitig verwenden kann, sind der Geschwindigkeit nur finanzielle Grenzen gesetzt.

Falls Sie sich nun einen Parallelrechner bauen wollen, wünsche ich Ihnen viel Erfolg und viel Spaß!

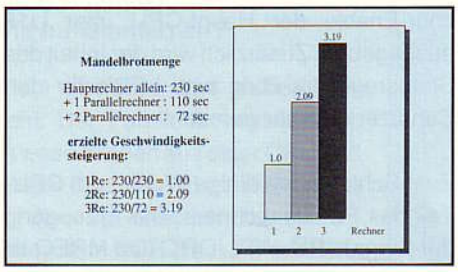


Bild 7: Rechenzeiten

```
OFFSET  $D0000    * Adresse des Speichers aus der CPU68k
ORG     $0        * Adresse aus der Sicht des Parallelrechners
DC.L    $1000     * Stack Pointer
DC.L    START     * Programmstartadresse

START:            * Programmstart
```

Bild 8: Die ersten Zeilen des Assembler-Programms

```
MOVE.B #0,$FFFFFFf0     * f ist die an JP6 eingestellte Adresse.

Den Inhalt kann man auf der Adresse lesen, auf der auch die Umschaltlogik angesprochen wird.

MOVE.B $FFFF21f0,d0     * 21 ist an JP1, f an JP4 eingestellt
                        * Der Statusregisterinhalt wird nach d0
                        * geladen
MOVE.B #0,$FFFF21f0     * Der Speicher wird umgeschaltet und das
                        * Statusregister gelöscht
```

Bild 9: Weitere Assemblerbefehle

```
*****************************************
* Apfelmännchen mit einen Parallelrechner *
*    von Tobias Thiel        20.01.1989   *
*****************************************

*--- PARALLELRECHNERPROGRAMM ---*

OFFSET $D0000           * Startadresse im Speicher des Hauptrechners
ORG $0                  * Startadresse im Speicher des Parallelrechners

dc.l $1000              * Stack Pointer
dc.l start              * Startadresse

ymin equ -2000          * Ausschnittkoordinaten * 1000
ymax equ 2020
xmin equ -2000
xmax equ 1600

start:                  * Hauptprogramm des Parallelrechners
 sub.l #1,zeile
 move.l zeile,d1
 lea ergebnis,a0
lp2:
 move.l #0,d2
lp1:
 movem.l d1-d2,-(a7)
 move.l #ymax,d0
 sub.l #ymin,d0
 muls d2,d0
 divs.l #255,d0
 ext.l d0
 add.l #ymin,d0
 move.l d0,d2           * cy
 move.l #xmax,d0
 sub.l #xmin,d0
 muls d1,d0
 divs.l #511,d0
 ext.l d0
 add.l #xmin,d0
 move.l d0,d1           * cx
 bsr mandel
 movem.l (a7)+,d1-d2
 move.l d0,(a0)+
skip:
 add.l #1,d2
 cmp.l #255,d2
 bne lp1
 move.b #0,$ffffff00    * Ready ausgeben
stop:
 bra stop               * Warten bis Reset

* Unterprogramm für den Parallelrechner

mandel:
 * d1.l= cx, d2.l = cy , d0.l = ergebn.
 * intern x=d3 y=d4 x2=d5 y2=d6
 * iter = d7
 clr.l d3
 clr.l d4
 clr.l d5
 clr.l d6
 clr.l d7
loop:
 cmp.l #120,d7
 bcc eman               * d7 > 120 dann fertig
 move.l d5,d0
 add.l d6,d0
 cmp.l #4000000,d0
 bcc eman
 move.l d5,d0
 sub.l d6,d0
 divs #1000,d0
 ext.l d0
 add.l d1,d0
 muls d3,d4
 move.l d2,d6
 muls #1000,d6          * normieren * 1000
 add.l d4,d4
 add.l d6,d4
 divs #1000,d4          * zurück normieren
 move.l d0,d3
 move.l d3,d5
 muls d3,d5
 move.l d4,d6
 muls d4,d6
 addq.l #1,d7
 bra loop
eman:
 move.l d7,d0           * Ergebnis
 rts

zeile:
 dc.l 512
ergebnis:
 dc.l 0,0,0,0,0,0,0

*--- HAUPTRECHNERPROGRAMM ---*

offset $0               * Hauptrechnerprogramm auf $30000
org $30000              * im Hauptspeicher

starte:
 move.l #512,zeile+$d0000 * Startspalte des Parallelrechners festlegen
 move.l #0,d1           * Startspalte des Hauptrechners festlegen
 move.b #0,$ff21f0      * Start des Parallelrechners
lp2e:
 move.l #0,d2
lp1e:
 movem.l d1-d2,-(a7)
 move.l #ymax,d0
 sub.l #ymin,d0
 muls d2,d0
 divs.l #255,d0
 ext.l d0
 add.l #ymin,d0
 move.l d0,d2           * cy
 move.l #xmax,d0
 sub.l #xmin,d0
 muls d1,d0
 divs.l #511,d0
 ext.l d0
 add.l #xmin,d0
 move.l d0,d1           * cx
 bsr mandele
 movem.l (a7)+,d1-d2
 and #1,d0
 beq skipe
 jsr @moveto
 move #$80,d0
 jsr @cmd
 add.l #1,d2
 move d2,d6
loop1:
 move.b $ff21f0,d0      * Parallelrechner fertig ?
 or.b $$fc,d0
 cmp.b #255,d0
 bne.s loop4            * Nein - dann weiter im Hauptprogramm
 move.b #0,$ff21f0      * Ja - dann Speicher umschalten
 lea ergebnis+$d0000,a6 * Ergebnisadresse laden
 move.l #0,d2
 move d1,d3
 move.l zeile+$d0000,d1 * berechnete Spalte laden
loop2:
 move.l (a6),d0
 and #1,d0
 beq loop3
 movem d1-d3,-(a7)
 jsr @moveto
 move #$80,d0
 jsr @cmd               * Ergebnis ausgeben
 movem (a7)+,d1-d3
loop3:
 adda.l #4,a6
 add.l #1,d2
 cmp.l #255,d2
 bne loop2
 cmp.l d1,d3
 bge endee
 move d3,d1
 move.b #0,$ff21f0      * Parallelrechner neustarten
loop4:
 move d6,d2
 cmp.l #255,d2
 bne lp1e
 add.l #1,d1
 bra lp2e

* Unterprogramm für den Hauptrechner

mandele:
 * d1.l= cx, d2.l = cy , d0.l = ergebn.
 * intern x=d3 y=d4 x2=d5 y2=d6
 * iter = d7
 clr.l d3
 clr.l d4
 clr.l d5
 clr.l d6
 clr.l d7
loope:
 cmp.l #120,d7
 bcc emane              * d7 > 120 dann fertig
 move.l d5,d0
 add.l d6,d0
 cmp.l #4000000,d0
 bcc emane
 move.l d5,d0
 sub.l d6,d0
 divs #1000,d0
 ext.l d0
 add.l d1,d0
 muls d3,d4
 move.l d2,d6
 muls #1000,d6          * normieren * 1000
 add.l d4,d4
 add.l d6,d4
 divs #1000,d4          * zurück normieren
 move.l d0,d3
 move.l d3,d5
 muls d3,d5
 move.l d4,d6
 muls d4,d6
 addq.l #1,d7
 bra loope
emane:
 move.l d7,d0           * Ergebnis
 rts

endee:
 rts
end

* ************************************ *
```

Bild 5: Programm für Apfelmännchen mit einem Parallelrechner

NDR-Computer mit CPU8088 und MS-DOS

Wir sind bereits in der letzten LOOP auf die CPU8088 ausführlich eingegangen und haben die Schaltung der CPU etwas durchleuchtet. In diesem Artikel wollen wir mehr auf das Umfeld der CPU eingehen und generell über das NDR-System als MS-DOS Rechner sprechen. Dabei gibt es, wie bereits in der LOOP19 kurz erwähnt, viele Konfigurationsmöglichkeiten.

Die ursprüngliche Konfiguration ist wohl der Standard NDR-Computer mit GDP64, KEY und FLO3, sowie der CPU8088 und den NDR Speicherkarten. Mit diesen Karten ist die Funktion des NDR-PC voll gewährleistet.

Ein Nachteil dieser Konfiguration kann nicht verschwiegen werden: Die GDP64; diese NDR-Graphikkarte hat zwar eine schnelle Graphik, hat aber in der Textdarstellung vor allem beim Scroll doch Schwächen.

Auch die NDR ASCII-Tastatur ist nicht so komfortabel wie eine IBM-PC-kompatible Tastatur. Diese Schwächen können eigentlich durch geringe Investitionen behoben werden. Die Tastatur kann durch eine MF2-Tastatur ersetzt werden (wird direkt an der CPU8088 angeschlossen). Die Tastatur muß allerdings im PC-Modus betrieben werden.

Einsatz einer PC-Graphikkarte

Die Graphikkarte kann, wie aus der LOOP19 entnommen werden konnte, durch eine PC-Graphikkarte ersetzt werden. Dazu ist allerdings eine NDR-PC Buskopplungsbaugruppe erforderlich. Die BIOS Version 1.3 unterstützt bereits die gängigsten PC Graphikkarten:

- MGA (Monochrom Graphik Adapter)
- Hercules Karte
- CGA (Colour Graphik Adapter)
- EGA (Enhanced Graphik Adapter)
- VGA (Video Grapik Adapter)

Festplatte für den NDR-PC

Außer den Graphikkarten wird ein PC Festplattencontroller eingesetzt. Dabei haben wir den gängigen und auch relativ preiswerten OMTI-Controller 5520B verwendet. Dieser Controller unterstützt alle gängigen Festplatten (Seagate ST225, NEC usw.) Damit haben wir auch für den NDR-Computer eine preiswerte Festplattenlösung gefunden. Die Festplatte ist in der BIOS-Version 1.3 noch nicht implementiert (ab Version 1.4).

Dieses BIOS ist jetzt lieferbar.

Unterstütung der Smart-Watch

Um die Uhrzeit nicht ständig neu eingeben zu müssen, soll die SMART-WATCH noch unterstützt werden.

Da der PC aber seine Uhr über den Timer steuert, dient die SMART-Watch dann nur dazu, die Zeit während des ausgeschalteten Zustand mitlaufen zu lassen.

Wird der Rechner wieder eingeschaltet, wird die Zeit aus der SMART-WATCH ausgelesen und in die Betriebssystemuhr geschrieben. Hier wird sie jetzt wieder vom Timer auf Stand gehalten.

Was macht die BUSKOPP-Baugruppe ?

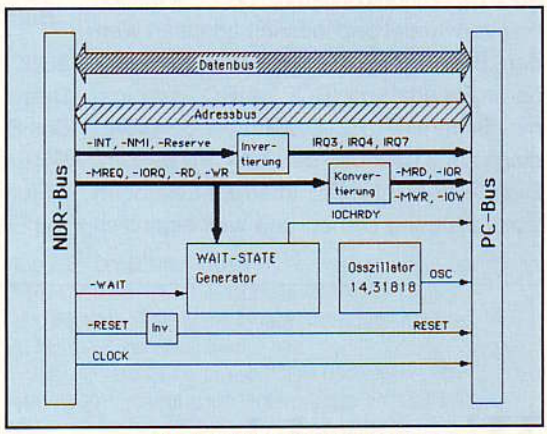


Bild 1: Blockschaltbild BUS-KOPP

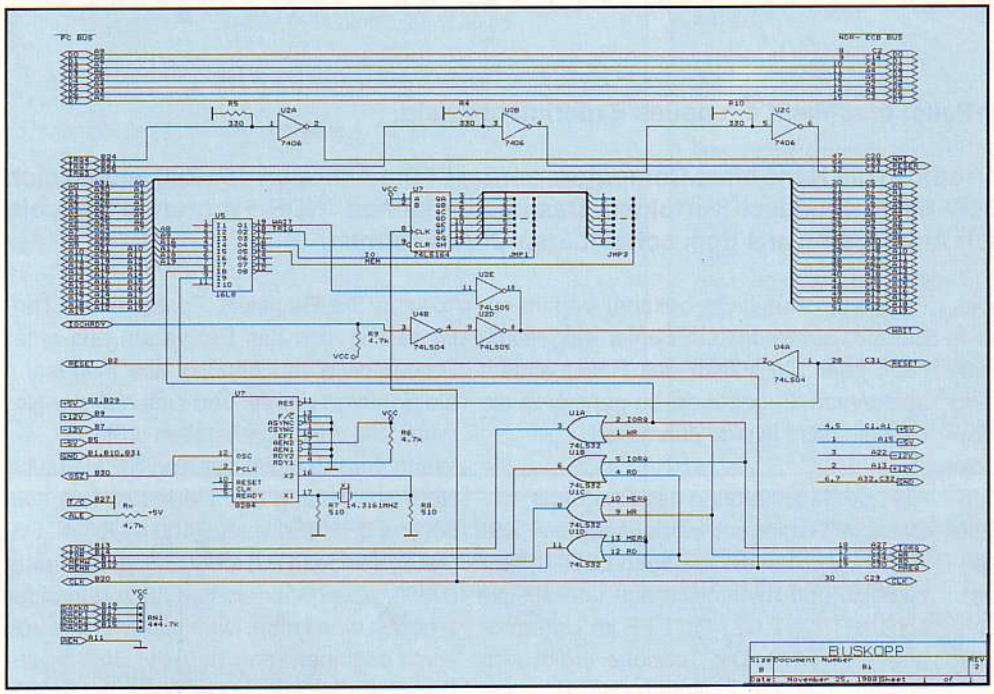


Bild 2: Schaltplan der BUSKOPP

Nun zu der oben angesprochenen Buskopplungsbaugruppe. Diese Baugruppe beinhaltet drei NDR-Steckplätze und drei PC-Steckplätze. Sie wird einfach als Buserweiterung eingesteckt (siehe Bild 3). Dadurch können drei PC-Baugruppen gesteckt werden. Mehr als drei IBM-Karten werden wohl in den seltesten Fällen verwendet. Die drei NDR-Steckplätze wurden vorgesehen, um ein NDR PC-System auf einem BUS (BUSKOPP) betreiben zu können. Hier dachten wir vor allem an die Einsteiger, die sich ihren PC selbst bauen wollen. Auf diesen drei Steckplätzen wird dann die CPU8088, eine NDR Speicherkarte (mindestens 256k) und die FLO3 Karte Platz finden.

Nun aber zu der Funktion der Buskopplungskarte: Diese Karte macht

nichts anderes als den NDR-Bus auf den PC Bus zu adaptieren.
Dabei sitzt die CPU immer nur auf dem NDR-Bus und auf dem PC Bus nur Peripherie- oder Speicherkarten. Wie das Blockschaltbild zeigt, enthält diese BUSKOPP Baugruppe noch zwei WAIT-State Generatoren für Ein/Ausgabe- und für Speicherzugriffe. Diese WAIT-State-Generatoren werden für die EGA und VGA-Karten benötigt.
Der Daten- und Adressbus werden direkt vom NDR- auf den PC Bus übertragen. Beim Steuerbus unterscheiden sich einige Signale des NDR-Busses von denen des PC-Busses. Dies sind die Signale -MREQ, -IORQ, -RD, -WR und RESET.
Beim PC-Bus heißen die entsprechenden Signale -MRD, -MWR, -IOR, -IOW, und RESET. Auch diese Anpassung übernimmt die BUSKOPP-Baugruppe.
Auch die Standard Interrupts bei PC's werden hier verwendet und müssen adaptiert werden. Beim PC-Bus sind die Interruptanforderungssignale mit IRQ0 bis IRQ7 bezeichnet. Beim NDR-PC-System werden allerdings nur 3 über den Bus zur CPU geführt. Dies, weil nicht mehr Interrupt-Leitungen zur Verfügung stehen und weil eigentlich auch nur diese drei verwendet werden.
Diese drei Signale werden auf die NDR-Bus Signale -INT, -NMI und -Reserve (Pin 54) gelegt. Damit ist übrigens die Reserve-Leitung beim NDR-Bus belegt.
Die EGA und VGA Karten benötigen außer den üblichen Steuersignalen noch einen synchronen Takt auf dem PC-Bus Signal OSC von 14,3161 MHz. Da dieses Signal auf dem NDR-Bus nicht verwendet wird, wird dieser Takt ebenfalls auf der BUSKOPP-Baugruppe erzeugt.

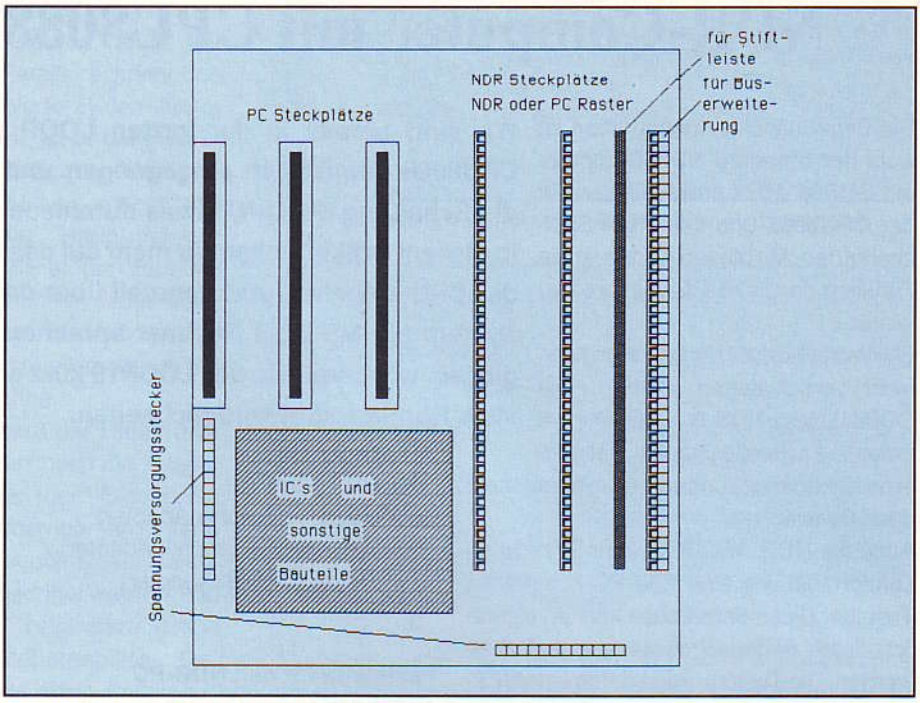


Bild 3: Steckplätze der BUSKOPP

Reinhard Brune

Wunschkonzert mit dem NDR-Computer

Das Radio neben dem Einsteiger-Paket erschließt ein neues Experimentierfeld.

Haben Sie schon einmal Ihr Kofferradio in die Nähe Ihres Computers gerückt? Die Aktionen des Z80 lassen sich dann an vielen Stellen auf der UKW-Skala akustisch verfolgen. Das ist Ihr eigenes "NDR-Programm". Mit ein wenig Maschinensprache läßt sich Anhörbares aus dem scheinbaren Chaos formen.

Sicher kennen Sie das Knacken im Radio, wenn Lichtschalter betätigt werden. Auch Ihr Computer schaltet ständig viele Stromkreise ein und aus - z.B. den Adreß- und den Datenbus. So entstehen je nach Programm schnelle Schaltfolgen, die in die nähere Umgebung abgestrahlt werden. Diese unvermeidliche schwache Störstrahlung ermöglicht es, unseren Computer mit einem Radiogerät zu belauschen!
Geben Sie doch einmal das abgedruckte Hex-Listing ein, während Sie der emsigen Computerarbeit zuhören.
Wenn Ihnen das Geräusch nach dem Programmstart nicht bekannt vorkommt, hat sich entweder ein Fehler eingeschlichen oder Sie sollten das Radio anders abstimmen. Zugegeben; so ganz paßt die Melodie nicht in jede Jahreszeit.

Hier die Beschreibung des Programms:
Damit im Radiogerät eindeutige Töne entstehen, werden die kräftigen LED-Treiber zugleich und rhythmisch ein- und ausgeschaltet (OUT 00 / OUT FF im Unterprogramm "TON"). Die Tonhöhe ergibt sich aus der Zeit, die zwischen dem Ein- und Ausschalten verstreicht (Warteschleife "Zeit" im DE-Register). Tonhöhe und Tonlänge holt sich das Programm aus einer Datentabelle (ab 8400H), die insgesamt die Tonfolge enthält und sich aus der Notentabelle zusammenstellen läßt.
Wenn zum Beispiel der Ton "a" als halbe Note am Anfang des Liedes stehen soll, beginnt die Tabelle so: 8400 47 D8 47 D8. Die Tonlänge (47H) steht vor der Tonhöhe (D8H). Zwei Viertelnoten hintereinander ergeben eine halbe. Damit die Melodie von vorne beginnen kann, muß der Umfang der Datentabelle (Anzahl der Töne) bei 8104H eingegeben werden. Mit 8301H ist die

Dr. Hans Hehl

Ökosystem eines Interpreters

Teil 2: Werkzeug zum Knacken

Wenn Sie mit dem BASIC-Programm DEZHEXBI.BAS aus Folge 1 keine Schwierigkeiten hatten, dann können Sie gleich mit dem zweiten Abschnitt dieser Folge weiter machen. Für den Anfänger soll aber die Programmzeile 240 besprochen werden, denn hier erfolgt die binäre Darstellung einer Zahl und noch ist kein Meister vom Himmel gefallen.

1) Des Rätsels Lösung

Ergänzen Sie das Programm DEZHEXBI mit folgender BASIC-Zeile: 245 STOP . Nun starten Sie das Programm und geben Sie als Bereich nur jeweils die Zahl 128 an. Ausgegeben werden folgende Werte: 128, 0080 und die Zahl 1. Nun können Sie mit PRINT K, 2^I die Variableninhalte abfragen. Sie erhalten die Zahlen 128 und wieder 128. Nun überlegen wir uns, was diese Werte ergeben, wenn wir sie mit dem AND-Befehl verknüpfen. Der AND-Befehl stellt die logische UND-Funktion einer Schaltung mit zwei Eingängen und einem Ausgang dar. Man erhält nur dann eine 1 am Ausgang, wenn der eine Eingang UND der andere Eingang den Zustand 1 besitzen (Siehe auch Rechner modular von R.-D. Klein, S. 29).
Übertragen wir diese Aussage auf den in Klammern gesetzten Befehl (K AND 2^I), so wird die binäre Zahl der Variablen K UND-verknüpft mit der Zahl 2^7, also 128. Bei der Zahl 128 ist Bit 7 gesetzt, ebenso bei dem Wert der Variablen K und wir erhalten als Ergebnis eine Eins. Der Befehl SGN gibt bei Zahlen größer Null eine 1 aus. Die Schleife von Zeile 230 - 250 ermittelt also ein Bit und gibt es aus. Mit dem Befehl CONT erhalten Sie die weiteren Bits.

> Nichts ist so leicht, wie es aussieht... und ...Alles dauert länger als man glaubt.
>
> Murphy

2) Einstieg in den Speicher

Nun wird es spannend. Zuerst wird ohne Disketten- Betriebssystem angefangen und das Grundprogramm im EPROM verwendet.

Wer die Wahl - hat die Qual

Es gibt inzwischen mehrere Möglichkeiten des Systemausbaues, die nicht alle hier aufgeführt werden können. So kann auf der SBC3 das Grundprogramm allein verwendet werden oder man besitzt schon den EFLOMON-Monitor und das Grundprogramm EGRU2000.

a) Mit EFLOMON und Grundprogramm EGRU2000

Wir starten das Grundprogramm und schauen uns mit Menue Nr. 3 den Speicher an. Bei EFLOMON 4.0 finden wir die Monitormeldung ab Adresse 1900h. Die genaue Adresse ist natürlich je nach Versionsnummer des Monitors verschieden (Vers.-Nr. 1.6 ab Adresse 1A00h, 3.2 ab 1A90h). Ab Adresse 2000h beginnt das Grundprogramm, mit dem wir gerade arbeiten und ab Adresse 3E00h finden wir die Startmeldung des Grundprogrammes.

b) Mit FLOMONCG-Monitor von R. Nahm

Wer den in LOOP 19 beschriebenen neuen Monitor verwendet, braucht das Grundprogramm nicht und kann zusätzlich auf andere Speicherkarten zugreifen. Es wird Menue 2 gewählt und z.B. mit dem Befehl D0 100 der Speicherinhalt von Adresse 0 bis 100h angezeigt. Der BASIC-Interpreter EHEBAS kann zusätzlich auf Bank E (siehe Folge 1) betrieben werden. Die Einschaltung der Bank E erfolgt mit dem Befehl B E. Geben Sie dann zum Anschauen z.B. den Befehl D 3FC0 ein und Sie finden die Startmeldung von EHEBAS.

c) So gehts auch

Wenn Sie den neuen Monitor FLOMONCG nicht haben, können Sie das Eprom Nr. 0 oder 1 von EHEBAS auch auf der SBC3 (oder Bank-Boot) auf den dritten 28-poligen Sockel (von links) stecken. Bei der SBC3 muß aber Jump 1 auf Stellung 2 gesteckt werden, sonst ist der Akku sofort leer. Nun können Sie mit dem Grundprogramm EGRU2000 ab Adresse 4000h EHEBAS analysieren.

d) Mit CP/M-Betriebssystem

Besitzen Sie die Ausbaustufe mit dem Diskettenbetriebssystem CP/M, so steht Ihrem Entdeckungsdrang nichts im Wege.

Teil 1: Aller Anfang ist schwer LOOP 20
Teil 2: Werkzeug zum Knacken LOOP 21
Teil 3: Schnitzeljagd LOOP 22
Teil 4: Geheimsprache LOOP 23
Teil 5: Ein Boarisch-BASIC LOOP 24
Teil 6: BDOS und BIOS LOOP 25
Teil 7: Innereien LOOP 26
Teil 8: Auf die Plätze, fertig, los LOOP 27
Teil 9: Patchwork mit Variablen LOOP 28
Teil 10: Hexeneinmaleins LOOP 29

Mit dem Programm DDT.COM, einem sogenannten Debugger (debugging = "entwanzen") für 8080-Programme können Programme getestet und in veränderter Form wieder abgespeichert werden. Es gibt auch verbesserte Programme im Handel, die auch die Z80 - Befehle kennen, wie z.B. DDTZ.COM. Nach dem Systemstart (CP/M meldet sich mit dem Symbol "A>") geben Sie ein: DDT HEBASNDR.COM <CR> (Programmname des Interpreters), wenn sich beide Programme auf derselben Diskette befinden.
Wir erhalten dann bei der Disk-HEBAS-Version G4 die Meldung:

```
DDT VERS 2.2
NEXT PC
4B00 0100
-
```

Damit haben wir auch gleich die Endadresse 4AFF des Interpreters. Die Eingabe von "DIOO, 2OO" ergibt einen Speicherauszug, der aus zwei Gruppen besteht, dem sedezimalen Zahlenbereich und dem Bereich in ASCII-Darstellung. Mit <CTRL-P> können wir die Speicherbereiche auch ausdrucken. Bei manchen Druckern muß übrigens vor dem Ausschalten des Drukkers mit <CTRL-P> ein <CTRL-E> eingegeben werden, damit der Druckerpuffer geleert wird.

3) Für lange Nächte

Wer schon Disk-HEBAS verwendet, kann sich einen kleinen Disassembler für Z80-Befehle selbst bauen. Dieser ist im Buch von R.D. Klein, Franzis-Verlag, Mikrocomputer Hard- und Softwarepraxis erläutert und wurde so abgeändert (Bild 1), daß auch große Z80-Programme bearbeitbar sind. Das Programm ist beim Autor auf Diskette für DM 10.- erhältlich (mit Z80- und 8080-Befehlssatz).

> Ist ein Programm unbrauchbar,
> wird es veröffentlicht.
>
> Murphy

In Bild 2 ist ein Ausschnitt eines disassemblierten Programmteiles aufgeführt. Für eigene Ergänzungen ist nach jedem Befehl eine kurze Punktzeile vorgesehen. Bild 3 enthält eine Z80-Befehlsliste, die als sequentielle Datei auf Diskette vorliegt und mittels Texteditor spaltenweise eingegeben werden muß. Es gibt über 1100 verschiedene Z80-Befehle zusammen mit den illegalen Opcodes (vom Hersteller nicht vorgesehene Möglichkeiten der CPU), die natürlich nicht alle in der Liste sein können. Auch unser BASIC-Disassembler kann keine Texte in einem Maschinensprache-programm erkennen und er stolpert über Befehlsfolgen wie "ED 5B BA 0F" (LD DE,(0FBA)).

4) Ausblick oder Lichtblick

Sie sind jetzt in der Lage, Speicherbereiche untersuchen zu können. Wer garnicht alleine weiterkommt oder Fragen hat, darf auch mal den Autor um Rat fragen, entweder schriftlich oder telefonisch (08762/3070, bitte nicht nach 21 Uhr). In der nächsten Folge "Schnitzeljagd" wird nun der Basic-Interpreter genauer unter die Lupe genommen.

```
RAM 13830 - 13845

3606  13830  3E 19     LD A,N     0019  ...............
3608  13832  CD 7037   CALL NN    3770  ...............
360B  13835  F5        PUSH AF          ...............
360C  13836  3E 0D     LD A,N     000D  ...............
360E  13838  CD7037    CALL NN    3770  ...............
3611  13841  F1        POP AF           ...............
3612  13842  4F        LD C,A           ...............
3613  13843  3E 0E     LD A,N     000E  ...............
3615  13845  C37037    JP NN            ...............
```

Bild 2: So arbeitet der Disassembler

```
NOP 1 0
LD-BC,NN 3 0
LD-(BC),A 1 0
INC-BC 1 0
INC-B 1 0
DEC-B 1 0
LD-B,N 2 0
RLCA 1 0
EX-AFAFA 1 0
ADD-HL,BC 1 0
LD-A,(BC) 1 0
DEC-BC 1 0
INC-C 1 0
DEC-C 1 0
LD-C,N 2 0
RRCA 1 0
DJNZ-DIS 2 1
LD-DE,NN 3 0
LD-(DE),A 1 0
INC-DE 1 0
INC-D 1 0
DEC-D 1 0
LD-D,N 2 0
RLA 1 0
JR-DIS 2 1
ADD-HL,DE 1 0
LD-A,(DE) 1 0
DEC-DE 1 0
INC-E 1 0
DEC-E 1 0
LD-E,N 2 0
RRA 1 0
JR-NZ,DIS 2 1
LD-HL,NN 3 0
LD-(NN),HL 3 0
INC-HL 1 0
INC-H 1 0
DEC-H 1 0
LD-H,N 2 0
DAA 1 0
JR-Z,DIS 2 1
ADD-HL,HL 1 0
LD-HL,(NN) 3 0
DEC-HL 1 0
INC-L 1 0
DEC-L 1 0
LD-L,N 2 0
CPL 1 0
JR-NC,DIS 2 1
LD-SP,NN 3 0
LD-(NN),A 3 0
INC-SP 1 0
INC-(HL) 1 0
DEC-(HL) 1 0
LD-(HL),N 2 0
SCF 1 0
JR-C,DIS 2 1
ADD-HL,SP 1 0
LD-A,(NN) 3 0
DEC-SP 1 0
INC-A 1 0
DEC-A 1 0
LD-A,N 2 0
CCF 1 0
LD-B,B 1 0
LD-B,C 1 0
LD-B,D 1 0
LD-B,E 1 0
LD-B,H 1 0
LD-B,L 1 0
LD-B,(HL) 1 0
LD-B,A 1 0
LD-C,B 1 0
LD-C,C 1 0
LD-C,D 1 0
LD-C,E 1 0
LD-C,H 1 0
LD-C,L 1 0
LD-C,(HL) 1 0
LD-C,A 1 0
LD-D,B 1 0
LD-D,C 1 0
LD-D,D 1 0
LD-D,E 1 0
LD-D,H 1 0
LD-D,L 1 0
LD-D,(HL) 1 0
LD-D,A 1 0
LD-E,B 1 0
LD-E,C 1 0
LD-E,D 1 0
LD-E,E 1 0
LD-E,H 1 0
LD-E,L 1 0
LD-E,(HL) 1 0
LD-E,A 1 0
LD-H,B 1 0
LD-H,C 1 0
LD-H,D 1 0
LD-H,E 1 0
LD-H,H 1 0
LD-H,L 1 0
LD-H,(HL) 1 0
LD-H,A 1 0
LD-L,B 1 0
LD-L,C 1 0
LD-L,D 1 0
LD-L,E 1 0
LD-L,H 1 0
LD-L,L 1 0
LD-L,(HL) 1 0
LD-L,A 1 0
LD-(HL),B 1 0
LD-(HL),C 1 0
LD-(HL),D 1 0
LD-(HL),E 1 0
LD-(HL),H 1 0
LD-(HL),L 1 0
HALT 1 0
LD-(HL),A 1 0
LD-A,B 1 0
LD-A,C 1 0
OR-A 1 0
CP-B 1 0
CP-C 1 0
CP-D 1 0
CP-E 1 0
CP-H 1 0
CP-L 1 0
CP-(HL) 1 0
CP-A 1 0
RET-NZ 1 0
POP-BC 1 0
JP-NZ,NN 3 2
JP-NN 3 2
CALL-NZ,NN 3 2
PUSH-BC 1 0
ADD-A,N 2 0
RST-0 1 0
RET-Z 1 0
RET 1 0
JP-Z,NN 3 2
CB 2 0
CALL-Z,NN 3 2
CALL-NN 3 2
ADC-A,N 2 0
RST-8 1 0
RET-NC 1 0
POP-DE 1 0
JP-NC,NN 3 2
OUT-N,A 2 0
CALL-NC,NN 3 2
PUSH-DE 1 0
SUB-N 2 0
RST-10H 1 0
RET-C 1 0
EXX 1 0
JP-C,NN 3 2
IN-A,N 2 0
CALL-C,NN 3 2
DD 3 0
SBC-A,N 2 0
RST-18H 1 0
RET-PO 1 0
POP-HL 1 0
JP-PO,NN 3 2
EX-(SP),HL 1 0
CALL-PO,NN 3 2
PUSH-HL 1 0
AND-N 2 0
RST-20H 1 0
RET-PE 1 0
JP-(HL) 1 0
JP-PE,NN 3 2
EX-DE,HL 1 0
CALL-PE,NN 3 2
ED 2 0
XOR-N 2 0
RST-28H 1 0
RET-P 1 0
POP-AF 1 0
JP-P,NN 3 2
DI 1 0
CALL-P,NN 3 2
PUSH-AF 1 0
OR-N 2 0
RST-30H 1 0
RET-M 1 0
LD-SP,HL 1 0
JP-M,NN 3 2
EI 1 0
CALL-M,NN 3 2
FD 2 0
CP-N 2 0
RST-38H 1 0
END 0 0
```

Bild 3: Die Z80-Befehlsdatei

```
10 ' Disassembler (C) R.-D. Klein
30 ' verändert von Dr. Hehl Hans
40 CLEAR 3000:DV=0
50 DIM M$(257),A(257),R(257):PRINT
60 INPUT"8080 = 1 / Z80-Mnemonic = 2";ZC:PRINT
70 NA$ = "Z80.LIB"
80 :   IF ZC = 1 THEN NA$ = "8080.lib"
90 INPUT"Laufwerk";LW$:PRINT:IF LW$="" THEN LW$="a"
100 ST$="          ----------------------------------------"
110 ZS$="................"
120 :
130 PRINT"Befehlscode ";NA$;" wird geladen":PRINT
140 OPEN #20,"I",LW$+":"+NA$:I=0
150 ON EOF GOTO 270
160 A$=""
170 B$ = BYTE$(#20): IF B$ = " " THEN 210
180 :   IF (B$=CHR$(13)) OR (B$=CHR$(10)) THEN 150
190 :   IF B$="-" THEN B$=" "
200 A$=A$+B$:GOTO 170
210 M$(I) = A$
220 A(I) = VAL(BYTE$(#20))
230 IF BYTE$(#20)<>" " THEN PRINT#DV, EINGABEFEHLER"
240 R(I) =VAL(BYTE$(#20))
250 OUTBYTE#0,42:I=I+1
260 GOTO 160
270 CLOSE #20:PRINT:PRINT
280 :
290 INPUT"Quelle Disk = 1  RAM = 0";Q:PRINT
300 :   IF Q = 0 THEN Q$ = "RAM":GOTO 330
310 INPUT"Quelldatei";Q$:PRINT
320 OPEN#20,"I",Q$
330 ZZ=0:INPUT"Drucker 2 oder Konsole 0 ";DV:PRINT
340 :
350 INPUT"Startadresse dezimal (oder sedezimal mit &)";P1:PRINT
360 INPUT"Programmendadresse dezimal (oder sedezimal mit &)";
    E:E=E+1:PRINT
370 INPUT"disassemblieren ab Adresse dezimal (oder sedezimal mit
    &)";P2:PRINT
380 FOR I=1 TO 2:PRINT#DV:NEXTI
390 PRINT "Programm ";HEX$(P1);"h  bis ";HEX$(E);"h von ";
    HEX$(P2);"h";
400 PRINT" an disassemblieren":PRINT
410 INPUT"richtig j/n";RJ$: IF MID$(RJ$,1,1) <> "j" THEN 290
420 PRINT#DV,TAB(7);Q$;" ";P1;" - ";E-1:PRINT#DV
430 :   IF Q=0 THEN A=P1:GOTO 460
440 :   IF P2>P1 THEN FOR I = 1 TO P2-P1:B=BYTE(#20):NEXT:P1=P2
450 :
460 B1$="":B2$="":B3$="":M$="":L$=""      :'löschen
470 AD$=HEX$(P1):PC=P1                    :'Adresse hex
480 GOSUB 950                             :'Byte holen
490 :   IF B=999 THEN 910                 :'Abbruch
500 B1$=RIGHT$(HEX$(B),2):BT=B            :'1.Byte
510 M$=M$(B)                              :'Befehlsname (Mnemonic)
520 :
530 :   IF R(B)=1 THEN 680          :'relative Adresse mit 1 Byte
540 :   IF R(B)=2 THEN 780          :'absolute Adresse mit 2 Bytes
550 :
560 ' Normaler OPCODE
570 P1=P1+A(B)                            :'Programmzähler erhöhen
580 N=B                                   :'Byte B retten
590 :   IF A(N) =1 THEN 900               :'Druck
600 GOSUB 950                             :'Byte holen
610 B2$ = RIGHT$(HEX$(B),2):L$=" " +HEX$(B):C=B
620 :   IF A(N) = 2 THEN 900              :'Druck
630 GOSUB 950                             :'Byte holen
640 :   IF B=999 THEN 910
650 B3$ = RIGHT$(HEX$(B),2):L$=" "+HEX$(C+256*B)
660 :   IF A(N) = 3 THEN 900              :'Druck
670 PRINT#DV,"Fehler":GOTO 900            :'Druck
680 :
690 ' Relative Adresse berechnen
700 P1=P1+2
710 GOSUB 950                             :'Byte holen
720 B2$=RIGHT$(HEX$(B),2)
730 :   IF B<&80 THEN B=B+P1:GOTO 760
740 :   IF B>=&80 THEN B=(B-256)+P1
750 :   IF B>&FFFF THEN 900               :'Druck
760 L$=" "+HEX$(B)
770 GOTO 900                              :'Druck
780 :
790 ' Absolute Adresse berechnen
800 P1=P1+3
810 GOSUB 950                             :'Byte holen
820 C=B                                   :'Byte retten
830 GOSUB 950                             :'Byte holen
840 B2$=RIGHT$(HEX$(C),2):B3$=RIGHT$(HEX$(B),2)
850 B=C+256*B                             :'Adresse berechnen
860 :   IF B>&FFFF THEN 900               :'Druck
870 L$=" " +HEX$(B)
880 '
890 ' disassemblierten Befehl ausgeben
900 GOSUB 1050                            :'Druck
910 IF P1>E OR A>E THEN P1=0:ZZ=0:GOTO 290
920 GOTO 460
930 END
940 :
950 IF Q = 1 THEN 1000
960 B=PEEK(A):A=A+1                       :'Byte aus RAM
970 :   IF A>E THEN B=999
980 RETURN
990 '
1000 B=BYTE(#20)                          :'Byte von Disk
1010 ON EOF GOTO 1030
1020 RETURN
1030 B = 999:GOTO 290
1040 '
1050 ' Ausdruck
1060 ZZ=ZZ+1                              :'Zeilenzähler
1070 PRINT#DV,USING 1080;AD$,PC,B1$,B2$,B3$,M$,L$,ZS$
1080 !        'LLL  ######   'L 'L 'L   'LLLLLLLLL 'LLLLLL  'E
1090 :  IF LEFT$(M$,2)="RE" OR LEFT$(M$,1)="J" THEN PRINT#DV,
     ST$:ZZ=ZZ+1
1100 :  IF ZZ>60 THEN FOR C=1 TO 11:PRINT#DV:NEXTC:ZZ=0
1110 RETURN
```

Bild 1: Disassembler in Basic

Lothar Reinholz

Einfacher A/D Wandler für den NDR

Hier stelle ich einen einfachen und preiswerten A/D Wandler vor, den jeder geübte Elektronikbastler nachbauen kann.

Ich habe den A/D-Wandler auf einer kleinen Lochrasterplatine aufgebaut und mit einer Flachleitung an die IOE Karte angeschlossen. Das IC ist eigentlich für ein dreistelliges Meßgerät gedacht und hat Multiplexausgänge. Die Auflösung beträgt 999 Digit positiv und 99 Digit negativ. Mit einem kleinen Programm - (MESS3162.ASM) - kann man jedoch das IC abfragen . Es ist eine Meßrate von 2 bis 3 Messungen/Sek zu erreichen. Das ist zwar nicht sehr schnell, reicht aber für viele Anwendungen vollkommen aus.

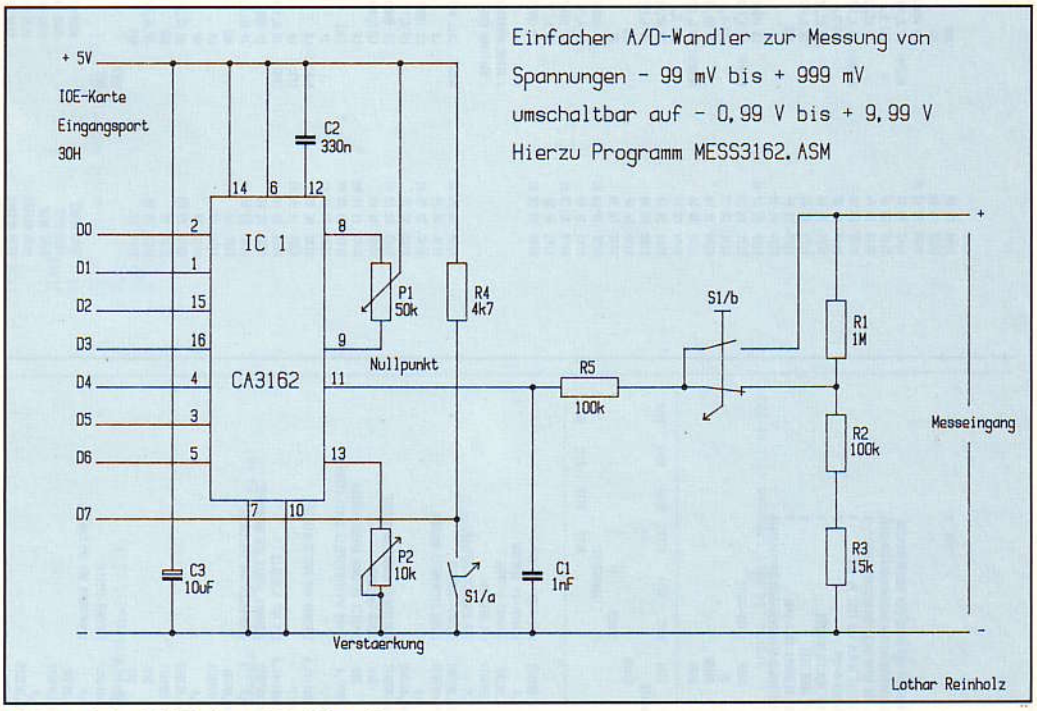


Bild 1: Schaltbild des A/D-Wandlers

Aufbau

Aus der Zeichnung ADP.CAD Kann man den genauen Aufbau auf einer Lochrasterplatine erkennen. Sie wird über eine Flachbandleitung an die IOE-Karte 30h angeschlossen oder direkt auf der IOE-Karte aufgebaut. Die horizontalen Linien entsprechen den Leiterbahnen der Platine. Die vertikalen Linien sind die Brücken welche man aus dünnem Kupferdraht biegt. Die Unterbrechungspunkte muß man, aus Gründen der Übersichtlichkeit der Zeichnung, selbst heraussuchen . Dies ist aber sehr einfach.

Falls auf einer horizontalen Leiterbahn zwei oder mehr Linien hintereinander sind so sind z.B.mit einem 4mm Bohrer die Leiterbahn dazwischen zu unterbrechen. die Punkte am Rand der Zeichnung sind das Rastermaß 2.5mm hieran kann man vorzüglich den Abstand der Teile abzählen. Bei Unklarheiten kann man in das Schaltbild AD.CAD sehen.

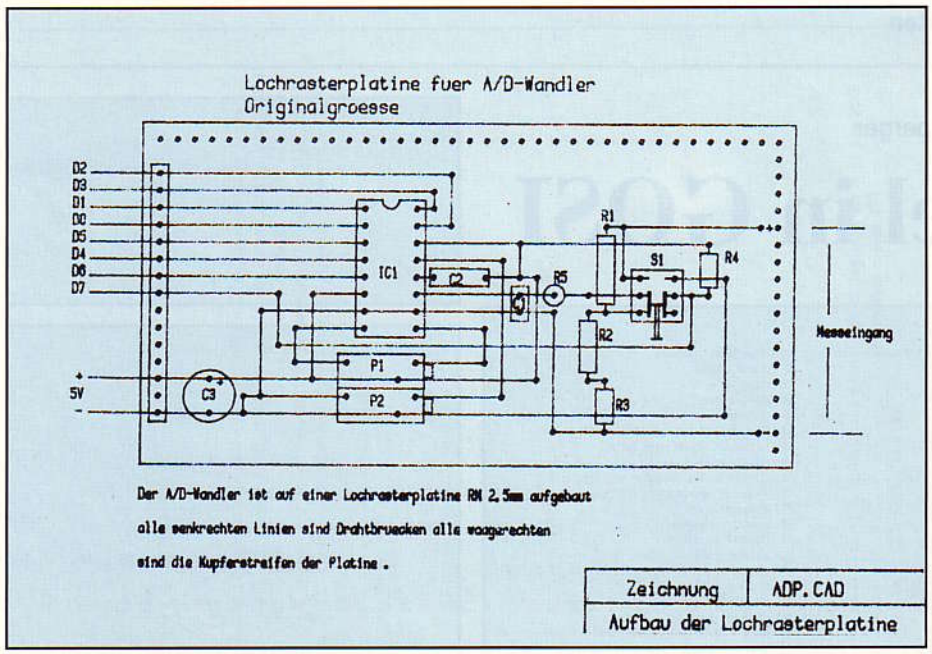


Bild 2: Aufbau der Lochrasterplatine

Teile

Das IC 1 CA3162 bekommt man für ca.15 DM in jedem Elektronikversand Für die Meßwiderstände R1,R2 und R3 sollte man 1% Metallschichtwiderstände benutzen, und 10-Gang Spindeltrimmer für P1 und P2. Der Schalter S1 ist ein 2 poliger Umschalter von Siemens , ein Kontakt schaltet die Meßspannung um ,der zweite schaltet Bit 7. Auf diese Weise kann die Schalterstellung im Meßprogramm abgefragt werden. Wenn man das nicht braucht kann S1,R1,R2 und R3 auch weggelassen werden. Die Bits 0 bis 3 sind das 4-Bit Datenwort, und die Bits 4 bis 6 geben den Stellenwert an, der aus dem Assemblerprogramm zu ersehen ist.

```
MEßPROGRAMM ZUM AUFNEHMEN VON MEßWERTEN                    PAGE  1

                    TITLE MEßPROGRAMM ZUM AUFNEHMEN VON MEßWERTEN
                    ;############################################
                    ;# PROGRAMM ZUM AUFNEHMEN VON MESSWERTEN MIT #
                    ;# DEM IC CA 3162 UEBER I/O PORT 30 VON      #
                    ;# 0-1V ODER VON 0-10V                       #
                    ;# AUSGABE AUF DEM BILDSCHIRM                #
                    ;# UNTER CP/M ODER ZEAT                      #
                    ;# (C) LOTHAR REINHOLZ 25.3.1986             #
                    ;############################################

                    MPORT   EQU     030H    ; IOE-KARTE
                    WSTART  EQU     0H
                    SYST    EQU     0005H
                    TAST    EQU     068H
                    ;----------------------------------------------------------------
                    ;MPORT  BIT7    BIT6    BIT5    BIT4    BIT3    BIT2    BIT1    BIT0 !
                    ;       SCHALTER                                                     !
                    ;               EINER                                                !
                    ;                       ZEHNER                                       !
                    ;                               HUNDERTER                            !
                    ;                                       BIT3    BIT2    BIT1    BIT0 !
                    ;----------------------------------------------------------------

0100  DB 30         MEIN:   IN A,(MPORT)    ; ABFRAGE VON BIT 7
0102  CB 7F                 BIT 7,A         ; DANN 10V BEREICH
0104  28 4C                 JR  Z,MEIN3     ; SONST 0-999mv
0106  DB 30         MEIN01: IN A,(MPORT)    ; 1.MESSWERT
0108  CB 67                 BIT 4,A         ; ? HUNDERTER STELLE
010A  20 FA                 JR NZ,MEIN01    ; SONST ZURUECK
010C  DB 30         MEIN02: IN A,(MPORT)    ; ZUR SICHERHEIT NOCHMAL
010E  CB 67                 BIT 4,A
0110  20 FA                 JR NZ,MEIN02
0112  DB 30                 IN A,(MPORT)    ; MESSWERT HUNDERTER STELLE
0114  E6 0F                 AND     0FH     ; MSB LOESCHEN
0116  F6 30                 OR      30H     ; ASCII ZEICHEN DARAUS MACHEN
0118  32 C8 01              LD (HUST),A     ; UND SPEICHERSTELLE DAMIT LADEN
011B  3E 2C                 LD A,','
011D  32 C9 01              LD (HUST+1),A   ; KOMMA AUF ZWEITE STELLE
0120  3E 56                 LD A,'V'
0122  32 CC 01              LD (HUST+4),A   ; V AUF 5.STELLE
0125  DB 30         MEIN1:  IN A,(MPORT)    ; ZEHNER STELLE WIE HUNDERTER
0127  CB 6F                 BIT 5,A         ; ABFRAGEN UND SPEICHERN
0129  20 FA                 JR NZ,MEIN1
012B  DB 30         MEIN10: IN A,(MPORT)
012D  CB 6F                 BIT 5,A
012F  20 FA                 JR NZ,MEIN10
0131  DB 30                 IN A,(MPORT)
0133  E6 0F                 AND     0FH
0135  F6 30                 OR      30H
0137  32 CA 01              LD (HUST+2),A
013A  DB 30         MEIN2:  IN A,(MPORT)    ; EINER STELLE
013C  CB 77                 BIT 6,A         ; ABFRAGEN UND SPEICHERN
013E  20 FA                 JR NZ,MEIN2
0140  DB 30         MEIN20: IN A,(MPORT)
0142  CB 77                 BIT 6,A
0144  20 FA                 JR NZ,MEIN20
0146  DB 30                 IN A,(MPORT)
0148  E6 0F                 AND     0FH
014A  F6 30                 OR      30H
014C  32 CB 01              LD (HUST+3),A
014F  C3 9B 01              JP AUSG

                    ; AB HIER DAS GANZE FÜR DIE MV AUSGABE

0152  DB 30         MEIN3:  IN A,(MPORT)    ; ALLES NOCH EINMAL FUER mV
0154  CB 67                 BIT 4,A
0156  20 FA                 JR NZ,MEIN3
0158  DB 30         MEIN30: IN A,(MPORT)
015A  CB 67                 BIT 4,A
015C  20 FA                 JR NZ,MEIN30
015E  DB 30                 IN A,(MPORT)
0160  E6 0F                 AND     0FH
0162  F6 30                 OR      30H
0164  32 C8 01              LD (HUST),A
0167  DB 30         MEIN4:  IN A,(MPORT)
0169  CB 6F                 BIT 5,A
016B  20 FA                 JR NZ,MEIN4
016D  DB 30         MEIN40: IN A,(MPORT)
016F  CB 6F                 BIT 5,A
0171  20 FA                 JR NZ,MEIN40
0173  DB 30                 IN A,(MPORT)
0175  E6 0F                 AND     0FH
0177  F6 30                 OR      30H
0179  32 C9 01              LD (HUST+1),A
017C  DB 30         MEIN5:  IN A,(MPORT)
017E  CB 77                 BIT 6,A
0180  20 FA                 JR NZ,MEIN5
0182  DB 30         MEIN50: IN A,(MPORT)
0184  CB 77                 BIT 6,A
0186  20 FA                 JR NZ,MEIN50
0188  DB 30                 IN A,(MPORT)
018A  E6 0F                 AND     0FH
018C  F6 30                 OR      30H
018E  32 CA 01              LD (HUST+2),A
0191  3E 6D                 LD A,'m'
0193  32 CB 01              LD (HUST+3),A
0196  3E 56                 LD A,'V'
0198  32 CC 01              LD (HUST+4),A
                    ;
                    ; DIESE AUSGABESCHLEIFE KANN DURCH ANDERE
                    ; ROUTINEN ERSETZT WERDEN

019B  CD B3 01      AUSG:   CALL WARTE      ; WARTESCHLEIFE
019E  3E 24                 LD      A,'$'   ; $=ENDEZEICHEN
01A0  32 CE 01              LD (HUST+6),A
01A3  3E 20                 LD A,' '
01A5  32 CD 01              LD (HUST+5),A
01A8  0E 09                 LD C,09H        ; DRUCKE STRING
01AA  11 C8 01              LD      DE,HUST ; STRING ANFANG
01AD  CD 05 00              CALL    SYST    ; SYSTEMEINSPRUNG
01B0  C3 00 01              JP      MEIN
01B3  0E 03         WARTE:  LD      C,3     ; ANZAHL #100MS
01B5  11 0E 2E      LP3:    LD      DE,11790; =100MS
01B8  DB 68         LP4:    IN A,(TAST)
01BA  FE 8D                 CP 8DH
01BC  C2 00 00              JP NZ,WSTART
01BF  1B                    DEC     DE
01C0  7A                    LD      A,D
01C1  B3                    OR      E
01C2  20 F4                 JR NZ,LP4
01C4  0D                    DEC     C
01C5  20 EE                 JR NZ,LP3
01C7  C9                    RET
                    HUST:
                    END

0030 MPORT                  0000 WSTART
0005 SYST                   0068 TAST
0100 MEIN                   0106 MEIN01
010C MEIN02                 0125 MEIN1
012B MEIN10                 013A MEIN2
0140 MEIN20                 0152 MEIN3
0158 MEIN30                 0167 MEIN4
016D MEIN40                 017C MEIN5
0182 MEIN50                 019B AUSG
01B3 WARTE                  01B5 LP3
01B8 LP4                    01C8 HUST
01C8 END

no fatal error(s)

SUM=5A96
CRC=BE37
```

Programm zum Aufnehmen von Messwerten

Leo Hepperger

Nimm-Spiel in GOSI

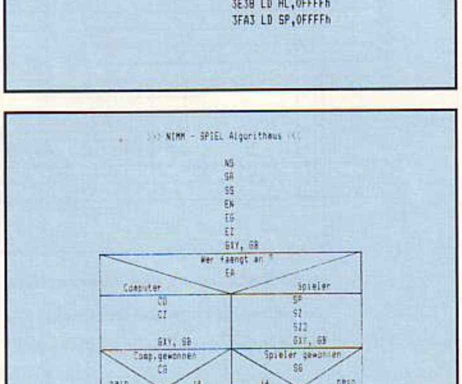
Da es in GOSI sehr wenig Programme gibt, möchte ich iermit einen Beitrag leisten.

Ich habe dieses Programm für meine Kinder, Peter 7 und Claudia 12 Jahre, geschrieben. Sie haben sehr viel Spaß damit.

Prozeduren sind großgeschrieben. Variable sind zur Unterscheidung kleingeschrieben. Prozeduren mit LERNE eingeben.

```
Erklaerung-Prozeduren: NS  ;Nimm Spiel   (Start)
                       SR  ;Spiel - Regel
                       SS  ;Speicher setzen
                       EN  ;Eingabe Name
                       EG  ;Eingaben
                       EZ  ;Eingabe Zahl (Anfangs-Stueckzahl)
                       GXY ;Grafik XY (Koordinaten setzen)
                       GB  ;Grafik Balken
                       EA  ;Eingabe Anfang (Computer oder Spieler ?)
                       CO  ;Computer (spielt !)
                       CZ  ;Computer zieht
                       CG  ;Computer gewonnen ?
                       SP  ;Spieler (spielt !)
                       SZ  ;Spieler zieht
                       SZ2 ;Spieler zieht 2 (wegen Rekursion von SZ getrennt)
                       SG  ;Spieler gewonnen ?
                       GA  ;Gewinn Anzeige
                       NES ;Noch ein Spiel ?
                       NA  ;Name Ausgabe (von RAM Adr.40000 = 9C40h)
                       VZ  ;Verzoegerung (fuer laengere Bildausgabe)
```

```
Erstellt mit : Z80 - CPU
     EGOSI2 1.1  fuer 32 KByte RAM geaendert (LOOP 6 Seite 9)
          Adresse 3E17 LD HL,0A000h
                  3E20 LD HL,0FFFFh
                  3E38 LD HL,0FFFFh
                  3FA3 LD SP,0FFFFh
```



Listing des GOSI-Programmes NIMM

```
ZEIGE NS
NS
SR SS EN EG
ENDE

ZEIGE SR
SR
LB VI DZ [                    >>> NIMM - SPIEL gegen Computer <<<]
DZ [        Spielregel : Von der vorhandenen Menge  werden  abwechselnd]
DZ [                          1, 2 oder 3 genommen .]
DZ [                   Wer zuletzt nehmen muss, hat verloren !]
DZ [                Wie heisst Du ?  (Bitte eingeben - dann CR)]
ENDE

ZEIGE SS
SS
SETZE *zna 0 SETZE *zb 0 SETZE *zcg 0 SETZE *zsg 0
SETZE *znm 0 BLINKER 35 6
ENDE

ZEIGE EN
EN
SETZE *bu :TASTE
MEM 40000+:zna :bu  WENN :bu=13 [RK]
ZEICHEN :bu
SETZE *zna :zna+1
EN
ENDE

ZEIGE EG
EG
EZ GXY EA
ENDE

ZEIGE EZ
EZ
LB BLINKER 12 0 NA DZ [, mit wieviel Stueck beginnen wir ? (6 bis 24)]
DZ [                          Bitte eingeben - dann CR]
BLINKER 38 3 SETZE *za :ZAHL
WENN :za<6 [EZ]
WENN :za>24 [EZ]
ENDE

ZEIGE GXY
GXY
VZ 20
SETZE *hz :za
LB SH AUFXY 30 260 AK 90 SA
GB
ENDE

ZEIGE GB
GB
VI VW 100 RE 90 VW 5 RE 90 VW 100 RE 90 VW 5 SH RW 15 RE 90 SA
SETZE *hz :hz-1 SETZE *zb :zb+1
WENN :hz=0 [SETZE *zb 0 RK]
WENN :zb=5 [SETZE *zb 0 SH RE 90 VW 15 LI 90 SA]
GB
ENDE

ENDE

ZEIGE EA
EA
BLINKER 18 0 DZ [Soll der Computer anfangen ?  (J o. N  druecken)]
BLINKER 38 3 SETZE *ag :TASTE WENN :ag=74 [CO]
WENN :ag=106 [CO] WENN :ag=78 [SP]
WENN :ag=110 [SP]
EA
ENDE

ZEIGE CO
CO
CZ GXY CG SP
ENDE

ZEIGE CZ
CZ
BLINKER 30 3 DZ [ Computer nimmt !]
SETZE *za :za-1 SETZE *be :za/4 WENN :za>(4*:be)+1 [CZ]
ENDE

ZEIGE CG
CG
BLINKER 30 3
WENN :za=1 [SETZE *zcg :zcg+1 DR [Computer hat gewonnen !!]VZ 40 GA]
ENDE

ZEIGE SP
SP
SZ SZ2 GXY SG CO
ENDE

ZEIGE SZ
SZ
BLINKER 18 3 NA DZ [, nimm 1, 2 oder 3 !   (dann CR druecken)]
BLINKER 38 5 SETZE *me :ZAHL
WENN :me<1 [SZ]
WENN :me>3 [SZ]
ENDE

ZEIGE SZ2
SZ2
SETZE *za :za-:me
ENDE

ZEIGE SG
SG
BLINKER 25 3
WENN :za=1 [SETZE *zsg :zsg+1 NA DZ [, bravo ! Du hast gewonnen .]VZ 40 GA]
ENDE

ZEIGE GA
GA
LS DZ [                          GEWINNANZEIGE !] DZ []
DR [        Computergewinn     ] DR :zcg
DR [              Spielergewinn     ] DZ :zsg
NES
ENDE

ZEIGE NES
NES
BLINKER 20 4 DZ [Noch ein Spiel ?  (J o. N druecken)]
BLINKER 35 6 SETZE *ne :TASTE WENN :ne=74 [EG]
WENN :ne=106 [EG] WENN :ne=110 [CALL 0]
WENN :ne=78 [CALL 0]
NES
ENDE

ZEIGE NA
NA
SETZE *nam :MEM 40000+:znm WENN :nam=13 [SETZE *znm 0 RK]
ZEICHEN :nam SETZE *znm :znm+1
NA
ENDE

ZEIGE VZ
VZ :zvz
SETZE *zvz :zvz-1 WENN :zvz=0 [RK]
VZ :zvz
ENDE
```

Jost-Reimer Hoof

Bei mir piept es

Warum piept der NDR-Computer nicht?
Lange Zeit surrte bei mir nur der Lüfter still vor sich hin, ab und zu tackerte eines der Laufwerke, und jetzt piept mein Rechner.

Viele Programme erzeugen bei Fehlermeldungen gleichzeitig ein Bell-Signal, ein CTRL-G (= 07h). In unseren Monitoren, angefangen vom FLOMON Version 1.5 bis zum FLOMONCG, wird dieses Signal ausgewertet, aber nicht ausgenutzt. Was fehlt, ist eine kleine Schaltung und ein paar Programmzeilen.

Hardware

Die Schaltung nach Bild 1 erzeugt einen Summton, wenn auf dem Input ein LOW-Pegel anliegt. Realisiert ist die Schaltung bei mir auf der Drucker-IOE-Karte, da hier reichlich Platz und Ausgabe-Bits vorhanden sind.
Die benötigten Teile sind in der Stückliste Bild 2 aufgeführt. Die Kabelverbindungen lassen sich mit Draht oder auch mit Lötzinnverbindung zwischen den einzelnen Rasterpunkten realisieren. Die Input-Verbindung muß an einen Pin von Port 41 (Out 1), z.B. an Pin 4 (Pin 0 ist bei mir durch Strope des Druckers belegt).

Software

Softwaremäßig ist folgendes nötig:
1. Muß die Schaltung beim Einschalten initialisiert werden, damit je nach zufälligem Zustand des Ports der Summer keinen nervtötenden Dauerton erzeugt.
2. Dann muß die Auswertung der Bell-Funktion durch einen Sprung in ein Zusatzprogramm ergänzt werden. Dieses Zusatzprogramm kann die Initialisierungsroutine sein und ist es bei mir auch!
Mein Realisierungsvorschlag ist folgender: Ich habe im FLOMON Version 3.2 die Initialisierung für die CAS-Karte geopfert, da sie bei mir nicht mehr benötigt wird. Diese Initialisierung wird gleich nach Anschalten des Rechners angesprungen. So steht bei Adresse 0239h und folgenden:

statt:		jetzt:	
3E 53	LD A, 53h	3E 00	LD A, 0h
D3 CA	OUT (0CAh),A	D3 41	OUT (41h),A
3E 50	LD A, 50h	3E 10	LD A, 10h
(*)			
D3 CA	OUT (0CAh),A	D3 41	OUT (41h), A
C9	RET	C9	RET

(*) 10h gilt nur für Anschluß von Input an Bit 4 ! (Bit 1 ==> 02, Bit 2 ==> 04, Bit 3 ==> 08, Bit 5 ==> 20h, Bit 6 ==> 40h, Bit 7 ==> 80h)

Mit dem ersten beiden Befehlen wird ein LOW an Port 41 gegeben, was ein Piepsen auslöst, dann wird wieder ein HIGH erzeugt, und alles ist still.
Jetzt muß nur noch bei der Bell-Auswertung auf Adresse 1889h statt der 3 NOPs ein Sprung in diese Initialisierungsroutine geschrieben werden:

C3 39 02
JP 0239h
Bei der FLOMON - Versionen 1.5 bzw. 4.2 sind die Adressen:
CAS-Initialisierung: bei 0233h bzw 0223h
Bell-Sprung bei 185Ch bzw 17DCh
mit Sprung nach 0233h bzw 0223h!!

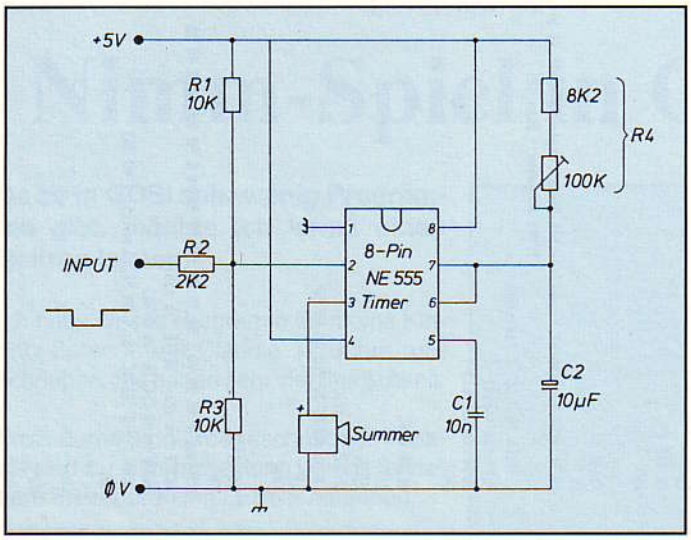


Bild 1: Schaltung für Bell-Signal

Wer schon in Besitz des FLOMONCG ist und auch die Quelldiskette erworben hat, findet dort ein nachladbares FLOMON32.COM. Dieses FLOMON wird beim Aufruf auf die BankBoot-Karte ab 4000h kopiert und angesprungen und ersetzt den zuerst gestarteten Monitor (aber nur, wenn dort RAM ist!!). Dieses Programm eignet sich besonders gut zum Testen obiger Schaltung. Die CAS-Initialisierungs-Routine liegt bei Adresse 0421h, die Bell-Auswertung mit den 3 NOPs bei 1A75h. Der Sprung muß aber nach 0239h gehen!
Auch FLOMONCG hat die Bell-Auswertung vorgesehen, aber noch nicht ausgenutzt. Da man sich das FLOMONCG aus dem Quellcode selbst generieren kann, empfehle ich folgende Arbeitsschritte:
1. Modul CHR.MAC ergänzen:

```
bell_port equ 41h          ;bell_port vereinbaren
bel-Unterprogramm mit folgender Routine ergänzen
LD      A, 0h              ; Summer aktivieren
OUT     (bell_port), A
LD      A, 10h             ; Summer auf Bit 4 aus
OUT     (bell_port), A
RET
- Exportliste - Proceduren am Anfang ergänzen, da diese Routine von der Initialisierung im Modul HW.MAC angesprungen wird
public BEL                 ; Bell-Routine
```

2. Modul CHREXTRN.MAC ergänzen:

```
extrn BEL                  ; Bell-Routine
```

3. Modul HW.MAC in dem Unterprogramm init_term folgenden Unterprogramm-Aufruf ziemlich am Anfang einbauen

```
call    bel                ; Initialisieren des
                           ;Bel-Portes
```

4. Nun muß man alle Module mit dem M80.COM assemblieren. Hier hilft GEN.SUB bzw GENLD.SUB für die nachladbare Version. Dann wird alles mit L80.COM gelinkt.
Eine genaue Anleitung fand ich auf der Source-Diskette für das WINDOW-Programm von Rüdiger Nahm.

Anwendung

Was nutzt das Ganze, wenn das Bell-Signal nicht ausgenutzt wird. Bei ZEAT wird die Bell-Funktion kräftig unterstützt, ebenso bei LADDER und der letzten Version von TAST-KOD.COM, wo folgende Routine vor jeder Fehlerausgabe eingebaut wurde bzw. ein CALL in den Bell-Aufruf:

```
bel-aufruf:
LD      E, 07h             ; <=> CTRL-G
LD      C, 2               ; Consolausgabe
CALL    0005h              ; System-Call
( RET )                    ; wenn als UP mit
                           ; CALL bel-aufruf
```

Johannes Werner Gleim

Grafik mit FLOMON

Teil 2 - Von einem, der auszog, MBASIC das Zeichnen zu lehren

Wer von den Benutzern des Microsoft-Entwicklungssystemes hat nicht schon den Verzicht auf die grafischen Möglichkeiten von RDK-Basic beklagt? Diesem Mangel soll hiermit abgeholfen werden. Als Demonstrationsobjekt wird ein komfortables Viewpage-Programm entstehen. Der Artikel zeigt auch den Umgang mit dem MBASIC-Entwicklungsprogramm und dem BASCOM-Compiler.

Programm ZEIGBLATT

Wer je die Gelegenheit hatte, eines der komfortablen Textverarbeitungsprogramme, wie z. B. "WORDSTAR 2003" auf einem PC zu testen, wird wohl etwas neidisch auf einige Features sein. Gibt es da doch beispielsweise einen Menüpunkt "Viewpage", der nach Aufruf ein verkleinertes Layout der aktuellen Seite liefert.Könnte man sich nicht auch für Wordstar 3.0 so etwas basteln? Es muß ein *.COM-Programm sein, da nur solche mit der Option "R" aus dem Wordstar-Menü aufgerufen werden können. Als Sprache kommt u.a. MBasic in Frage, hierfür steht der Compiler BASCOM zur Verfügung.

Erst Denken - dann Programmieren

Vor dem Programmieren sind einige Vorüberlegungen nötig. Zum einen benötigen wir einen Algorithmus, der eine Datei liest und je nach ASCII-Zeichen ein verkleinertes Symbol auf dem Bildschirm zeichnet.

Zum anderen müssen wir uns über die Zeichen-Größe klarwerden. Auf dem Bildschirm lassen sich in Y-Richtung 255 und in X-Richtung 511 Punkte darstellen. Das ist gerade 1 Punkt weniger, als für 64 Zeilen mit 4 Punkten Zeilenhöhe benötigt werden. Je ein Punkt muß dem Zeilenabstand geopfert werden. Bleiben also 3 Punkte für Großbuchstaben und 2 Punkte für Kleinbuchstaben. Um das Seitenverhältnis richtig darzustellen, ist der Zeichenabstand mit 4 Punkten und die Zeichenbreite mit 3 Punkten zu wählen. Für jedes darstellbare ASCII-Zeichen soll ein entsprechendes Rechteck gezeichnet werden. Das läßt sich mit den Moveto- und Drawtofunktionen aus Flomon verwirklichen. Man schreibe zunächst die Unterprogramme:

Das Programm:

```
2000   PRINT CHR$(27);CHR$(27);"GY0;P0"   :REM Grafik EIN S.0
2010   RETURN
2100   PRINT "M";X;Y :REM Moveto
2110   RETURN
2200   PRINT "D";X;Y :REM Drawto abolut
2210   RETURN
2300   PRINT "J";DX;DY        :REM Drawto relativ
2310   RETURN
2400   PRINT "A"              :REM Grafik AUS
```

Danach können die Voreinstellungen für Tabulator, Zeichengröße, Seitenlänge, rechtem und linkem Rand, Kopfzeilen und Fußzeilen definiert werden. Anschließend wird in ein Auswahlmenü verzweigt:

```
170    INPUT "Bitte wählen Sie ",X
180    IF X = 1 THEN 220    :REM Voreinstellungen ändern
190    IF X = 2 THEN 440    :REM Datei lesen und anzeigen
200    IF X = 3 THEN END
210    GOTO 170
```

Zunächst muß festgestellt werden, ob die Datei überhaupt existiert. Gegebenenfalls muß die Fehlermeldung abgefangen werden. Das geht wie folgt:

```
1 ON ERROR GOTO 3000   :REM Fehlerbehandlungsroutine
520 INPUT "Welche Datei anzeigen ";DATEI$
530 OPEN "I",#1,DATEI$ :REM Datei öffnen und schließen
540 GOTO 560
550 STOP:REM Richtige Diskette einschieben und CONT
555 REM oder Neustart mit RUN wenn falscher Name
3000 IF ERR=53 AND ERL=530 THEN PRINT "Datei existiert nicht"
3010 RESUME 550          :REM Arbeit wiederaufnehmen
3020 REM Error 53 = Datei nicht gefunden; ERL = Errorline
```

Wurde die Datei gefunden, kann der Grafikmodus eingeschaltet und der Rahmen gezeichnet werden.

```
560 GOSUB 1000  :REM Grafik einschalten
570 X = 100: Y = 255
580 GOSUB 1100  :REM Moveto links oben
590 DX = PB * PTCHR :DY = 0 :REM Seiten- * Zeichenbreite
600 GOSUB 1300 :REM Drawto rel. rechts oben
610 DX=0: DY = 1 - (PL * PTLIN) :REM Seiten- * Zeilenhöhe + 1
620 GOSUB 1300 :REM Drawto rel. rechts unten
usw. bis wieder am Ausgangspunkt.
670 X=100: Y = 255 - (MT+4) * PTLIN :REM Kopfzeilen abziehen
680 GOSUB 1100 :REM Moveto Textbeginn
```

Die Datei wird wie folgt verarbeitet:

```
720 OPEN "I",#1,DATEI$ :REM Datei wieder öffnen
730 WHILE NOT EOF(1) :REM und bis Textende
740 CH$=INPUT$(1,#1) :REM lesen
750 CH = ASC(CH$)
760 IF CH > 127 THEN CH = CH - 127 :REM Bit 7 = Null
770 IF CH = 13 THEN GOTO 880 :REM Neue Zeile
780 IF CH =  9 THEN GOSUB 1180:GOTO 820 :REM Tabulator
```

```
790 IF CH = 32 THEN GOSUB 1230:GOTO 820 :REM Leertaste
800 IF CH > 95 THEN PT=2:GOSUB 1300:GOTO 820 :REM Klein- und
810 IF CH > 32 THEN PT=3:GOSUB 1300:GOTO 820 :REM Groß-
schrift
820 WEND
830 CLOSE
840 INPUT "Ende ";X$: IF X$ = "" THEN 840 :REM Warten bis
850 GOSUB 2400 :REM Grafik AUS
860 END
```

Jetzt fehlen nur noch die Unterprogramme. Im UP "Neue Zeile" wird Kopfzeilen und Fußzeilen, nach END verzweigt. Außerdem wird der Cursor auf den Anfang der nächsten Zeile ge"moveto"t. UP "Tabulator" ruft HT mal UP "Leertaste" auf. Dieses setzt den Cursor 4 Punkte weiter nach rechts. UP Klein- und Großschrift sieht folgendermaßen aus:

```
1300 X = X + 4:GOSUB 2100:REM nächstes Zeichen
1310 Y = Y + PT-1 :GOSUB 2200:REM Kästchen malen
1320 X = X + 3 :GOSUB 2200
1340 Y = Y - PT+1 :GOSUB 2200
1350 X = X - 3 :GOSUB 2200
1360 RETURN
```

Sind alle Fehler durch Austesten beseitigt, so kann mit SAVE "ZEIGBLAT.BAS",A abgespeichert werden. Wichtig ist die Option A anzugeben, da das Programm nur als ASCII-File weiter verarbeitet werden kann.

Compilern und Linken

Mit SYSTEM wird MBasic verlassen und der Compiler BASCOM aufgerufen.

BASCOM ZEIGBLAT,ZEIGBLAT=ZEIGBLAT/E

BASCOM verwandelt ZEIGBLAT.BAS unter Verwendung der Laufzeitbibliothek BASLIB.REL in das relokative File ZEIGBLAT.REL. Der Schalter /E dient dazu, die für die Fehlermeldung notwendige Zeilenreferenztabelle zu erstellen.

Nachdem (hoffentlich) die Meldung "00000 Fatal Error(s)" erscheint, muß das Programm gelinkt werden. Das macht L80.

L80 ZEIGBLAT,ZEIGBLAT/N/E

Mit /N wird der Name der *.COM-Datei spezifiziert und die Datei auf Diskette gespeichert. /E wählt den Ausgang zum Betriebssystem; alternativ wäre /G für den anschließenden Start möglich. Mit DIR ZEIGBLAT.* kann man sich das Ergebnis ansehen:

ZEIGBLAT.BAS	**ZEIGBLAT.REL**
ZEIGBLAT.PRN	**ZEIGBLAT.COM.**

.REL ist das relokative Programm und kann ebenso wie das lauffähige.COM mit einem Debugger (SID oder DDT) untersucht werden.
*.PRN enthält den Quellcode und den übersetzten Assemblercode, kann mit TYPE oder einem Editor aufgelistet werden.

Zum Schluß muß ZEIGBLATT.COM noch mit PIP oder dem Menüpunkt O (^KO) von WordStar auf die Wordstardiskette kopiert werden. Jetzt kann das Pogramm mit R aus dem WordStar-Menü aufgerufen werden. Ein erster Testlauf bringt die Fehlermeldung "BRUN.COM not found". Dabei handelt sich um das Laufzeitmodul, welches die eingebundenen Unterprogramme von MBasic enthält. Also ist auch dieses auf die WordStar-Diskette zu transferieren. Zusätzlich muß die Datei BCLOAD verfügbar sein und auf das Laufwerk mit BRUN.COM zeigen. Leider ist das .COM-Programm mit 3 min 23 sek bei einem Testbrief kaum schneller als das .BAS-Programm mit 3 min 39 sek. Verantwortlich dafür sind die mehrfachen IF-Abfragen und die relativ langsamen Grafikroutinen. Hier wäre es ratsam, nicht jedes Zeichen für sich, sondern immer ein Wort darzustellen.

Neuer Versuch mit OBASIC

Zuvor jedoch noch ein Versuch mit OBASIC.COM, dem alten Basic der Firma Microsoft, das ebenfalls im Paket enthalten ist. Hier wurden die Basic-Unterprogramme noch direkt in den Code integriert. Das Laufzeitmodul ist dadurch entbehrlich.

BASCOM ZEIGBLAT,ZEIGBLAT=ZEIGBLAT/E/O
L80 ZEIGBLAT,ZEIGBLAT/N/E

Mit der Option /O wird die alte Laufzeitbibliothek OBSLIB.REL verwendet. Der Testlauf ist jetzt in 3 min 04 sek absolviert. Sieht man sich die Dateilängen an, so wird das verständlich:

a) mit BASLIB.RE
b) mit OBSLIB.REL
ZEIGBLAT.COM 4 kB 32 Rec
ZEIGBLAT.COM 18 kB 134 Rec
BCLOAD 2 kB 1 Rec
BRUN.COM 16 kB 121 Rec
Summe 22 kB 154 Rec Summe 18 kB 134 Rec

Zeiten werden optimiert

Analysiert man die Programmstruktur, so fällt auf, daß für jedes ASCII-Zeichen bis zu 7 IF-Abfragen und 5 Wertezuweisungen mit Flomon-Grafik-Aufrufen fällig werden. Bei häufigen IF-Abfragen reagiert eine ON X GOSUB oder GOTO-Anweisung schneller. Zwar benötigt das ON- Statement mit 3.0 ms etwa gleiche Zeit wie das IF-Statement mit 3.1 ms pro Abfrage, jede weitere Möglichkeit wird aber in nur 0.18 ms absolviert, so daß trotz zusätzlichen Rücksprungbefehles schon bei zwei Abfragen ein zeitlicher Vorteil sichtbar wird. In der-ON-CH-GOSUB-Anweisung sind demzufolge 127 Sprungziele (4 Zeichen pro Zeilennummer) anzugeben. Aber hier erlebt man seine Überraschung. MBasic akzeptiert nur 256 Zeichen in einer logischen Befehlszeile. Ein Ausweg könnte wie folgt aussehen:

```
730 WHILE ...

760   ON CH    GOSUB Adr1, ..... Adr16
765   ON CH+16 GOSUB Adr17, .... Adr31
770   ON CH+31 GOSUB Adr32, .... Adr46
und so weiter
820 WEND
```

Tatsächlich ist der Testbrief damit in 2 min 15 sek abgearbeitet. Jedoch wird aus nicht aufklärbaren Gründen nur der Rahmen

sichtbar. Das Einfügen von

```
762 IF CH < 17 THEN 820
767 IF CH < 32 THEN 820
```

etc. verkürzt die Laufzeit auf 1 min 48 sek, füllt aber nicht das leere Blatt.

Der nächste Versuch mit

```
730 WHILE ...

760   ON CH GOSUB Adr1, ... Adr61
770   ON CH GOSUB Adr62 ... Adr122
780   ON CH GOSUB Adr123 .. Adr127
820 WEND
```

bringt seltsame Linien auf dem Bildschirm, die eher an eine verdrehte "Leiter" als an einen Brief erinnern. Und das bei einer auf 5 min 16 sek verlängerten Zeitdauer, vermutlich weil nur bis ASCII- Wert 61 geprüft wird, und das gleich doppelt und dreifach. Alle übrigen Zeichen durchlaufen die Abfragen umsonst.

So hat man auch hier zusätzliche Zeilen einzufügen:

```
765 IF CH < 62 THEN 820: CH = CH - 61: REM Ist gefunden?
Wenn
775 IF CH < 62 THEN 820: CH = CH - 61: REM nicht, verringern
```

Ergebnis: Jetzt wieder 2 min 12 sek, Grafik ähnlich wie vorher, Variante "Schienen, die in die Unendlichkeit führen".

Für Rechtecke hat der GRAFIK-Modus einen besonderen Befehl anzubieten: PRINT "R";DX;DY. Fügen wir ihn wie folgt ein:

```
2600 PRINT "R";DX;DY          :REM Rechteck dx * dy
2610 RETURN
```

und verwenden wieder die anfängliche IF-THEN-Abfrage. Aufruf der neuen Funktion, z.B. Großbuchstabe:

```
1400 X = X + 4: GOSUB 1200      :REM nächste Position
1410 DX=3: DY=2: GOSUB 1600     :REM Rechteck 3 * 2
1420 RETURN
```

Das gewünschte Ergebnis wird nun in 2 min 28 sek und bei geordeneter Abfrage in 2 min 17 sek auf dem Bildschirm dargestellt.

Möglicherweise läßt sich auf diese Weise noch einiges erreichen, da aber Kurzvektoren wesentlich schneller als Flomon-Grafik sind, möchte ich lieber diesen Weg gehen. Benutzen wir weiter die ursprüngliche IF-THEN-Abfrage und lassen den Flomon-Grafik-Aufruf noch für Leerzeichen und Positionieren des Kästchens zu. Die Unterprogramme für Buchstaben sind wie folgt zu ändern:

```
80 GDP = &H70
a) Kleinbuchstaben:
1300 X = X + 4: GOSUB 2100      :REM nächste Position
1330 OUT GDP+1,11               :REM in Fenster schreiben
1340 OUT GDP,208                :REM 2 Punkte nach rechts
1350 OUT GDP,210                :REM 2 Punkte nach oben
1360 OUT GDP,214                :REM 2 Punkte nach links
1370 OUT GDP,212                :REM 2 Punkte nach unten
1380 RETURN
b) Großbuchstaben:
1400 X = X + 4: GOSUB 2100
1430 OUT GDP+1,11
1440 OUT GDP,248                :REM 3 Punkte nach rechts
1450 OUT GDP,250                :REM 3 Punkte nach oben
1460 OUT GDP,254                :REM 3 Punkte nach links
1470 OUT GDP,252                :REM 3 Punkte nach unten
1480 RETURN
```

Auch diesmal ist nur ein Rahmen zu sehen. Trotz unsystematischer Abfrage ist die Angelegenheit aber in 2 min 05 sek erledigt, und sichtlich vertragen sich Kurzvektoren nicht mit dem Grafik-Modus. Also schalten wir den Grafik-Modus nach jeder Verwendung wieder aus.

```
690 GOSUB 2400        REM Grafik Aus

2100 GOSUB 2000       :REM Grafik Ein
2110 GOSUB 2150       :REM Moveto
2120 GOSUB 2400       :REM Grafik Aus
2150 PRINT "M";X;Y    :REM Moveto
2160 RETURN
```

Damit wird zum einen die Laufzeit auf 2 min 58 sek erhöht und zum anderen der Rahmen auseinandergerissen, ohne daß nach wie vor ein Text angezeigt würde. Ersetzen wir darum auch das Leerzeichen-UP durch Kurzvektoren:

```
1230 OUT GDP+1,10     :REM nicht schreiben oder löschen
1240 OUT GDP,208:OUT GDP,208 :REM 2x2 Punkte nach rechts
1250 RETURN
```

Zum ersten Mal erscheint wieder ein Brief-Layout und schneller als je zuvor, doch nach den ersten 20 Zeilen wird nur noch in der untersten Zeile geschrieben. Benutzen wir auch für das <CR>-Unterprogramm GDP-Befehle:

```
880 ZZ = ZZ + 1   :REM Zeilenzähler
890 IF ZZ = PL-MT-MB THEN 860   :REM Seitenende
900 X=100: Y=255-PTLIN*(MT+4+ZZ) :REM Zeilenanfang positionieren
910 OUT GDP+1,10      :REM nicht schreiben/löschen
920 OUT GDP+8,0 : OUT GDP+9,100  :REM X = 100
930 OUT GDP+10,0: OUT GDP+11,Y   :REM Y = Y - 4
940 GOTO 820
```

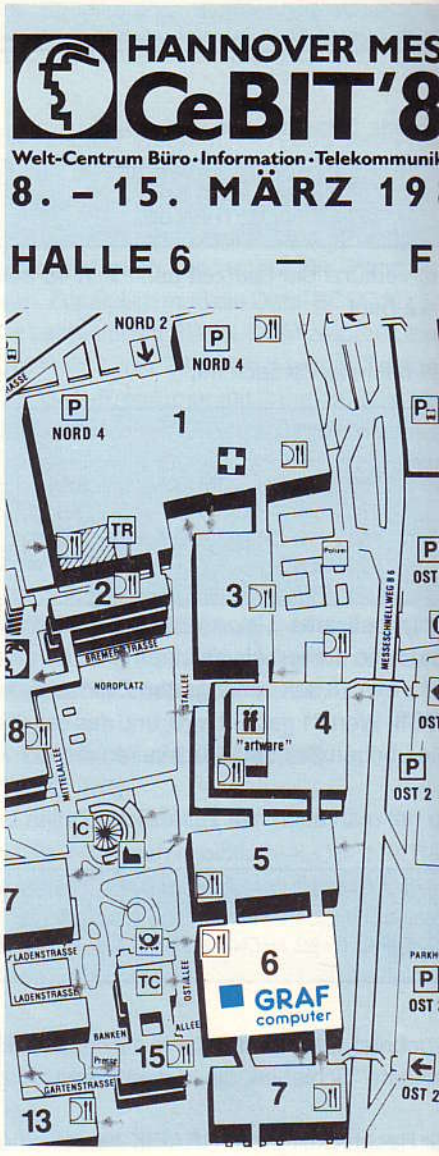
Dadurch wird das Programm noch etwas schneller und das Layout er scheint jetzt vollständig auf dem Schirm.

Feinschliff

Anschließend können noch einige Feinheiten in das Programm integriert werden; Einzelheiten sind aus dem Listing zu ersehen. Ebenfalls ist daraus zu erkennen, wie die Laufwerksabfrage und Anzeige des entsprechenden Directory gelöst wurden, sowie mögliche Fehlermeldungen abgefangen werden. Lediglich die BDOS-Error-Meldung "Laufwerk nicht bereit" konnte ich nicht abfangen.

Um mehr als nur eine Seite darzustellen, muß beim Erreichen der max. Zeilenzahl (kein Abbruch durch EOF) abgefragt werden, ob weitergemacht werden soll. Wenn ja, muß zuvor der Bildschirm

GRAF computer



BESUCHEN SIE UNS – WIR ZEIGEN:

Das L.E.I.K.-System

Wir gehen mit Ihnen neue Wege der Ausbildung: PC-Ausbildung für jedermann!

Sehen Sie sich das neue L.E.I.K.-System an.

Logik-Simulator für PC

Digitale Schaltungen am Bildschirm aufbauen und testen: kein Problem mit dem Logik-Simulator, ein preiswertes, mächtiges Programm für alle PCs.

Das Multi PC-Interface

8 Digital Ein, 8 Digital Aus, 8 Analog ein, 4 Motoren aus, V 24: Das Multi PC-Interface läßt keine Wünsche offen. Es läuft mit dem Logik-Simulator!

NDR-Computer

Das universelle System zum Selbstbau: vom Z80 bis zum 68020 und NEU: die CPU8088.

Bauen Sie sich Ihren PC selbst!

Ideal geeignet zum OEM-Einsatz auch in kleinen Stückzahlen.

OS/9 für den NDR-Computer

Das Multi-User, Multi-Tasking Betriebssystem: nun auch auf dem NDR-Computer.

Auftrag-, Lager-, Stückliste-, Produktion:

Die universelle Software für produzierende Unternehmen. Netzwerkfähig, problemlos zu installieren, Daten auf dBase-Basis: Info am Stand.

LOOP – Die Zeitschrift für Computer-Anwender

Lernen Sie LOOP kennen (Probeheft kostenlos sprechen Sie mit der Redaktion: am Stand!

AT auf einer Baugruppe

Ideal geeignet für industriellen und privaten E Stecken Sie Ihre Zentraleinheit in den passive Wir zeigen /286-CPU (12 MHz) sowie /386-C

AT-Rechner im 19"-Gehäu

Die robuste, preiswerte Industrie-Lösung: Mo ATs mit passivem Bus für die Industrie, präs von Heel Industrie Elektronik.

Kommen Sie – schauen Sie – Besuch lohnt sich!

GRAF ELEKTRONIK SYSTEME GMBH

Magnusstraße 13 · Postfach 1610 · 8960 Kempten (Allgäu) · Telefon: (08 31) 62 11
Telefax: (08 31) 61 086 · Teletex: 831804 = GRAF · Telex: 17 831804 = GRAF
Datentelefon: (08 31) 6 93 30

Filiale Hamburg:
Ehrenbergstraße 56
2000 Hamburg 50
Telefon (0 40) 38 81 51

Filiale München:
Georgenstraße 61
8000 München 4
Telefon (0 89) 27

Händler:
Jörg Korb · 1000 Berlin 33 · Telefon (0 30) 8 21 19 47
GMCP mbH · 2800 Bremen 33 · Telefon (0 421) 2
A: Stecher Elektronik · A-6020 Innsbruck · Tel. (0 52 22)

GRAF ELEKTRONIK SYSTEME, Magnusstraße 13, 8960 Kempten, Tel.: 0831 / 6211

GRAF computer

Preisliste März `89

GRAF computer

Artikel	Bezeichnung	DM
Bücher:		
70234	68008 Aufbau Listing V4.3	10,-
70233	68008 Grundprogramm V4.3 Listing	10,-
70244	68000 Grundlagen V4.3	50,-
70239	Einführung in LOGO	40,-
70245	M68000-Familie, Teil 2	50,-
70240	Der Einstieg in Pascal	35,-
70238	Einstieg in C	65,-
70246	Assemblersprache d. 68000	40,-
70236	Rodney Zaks, CP/M 2.2	40,-
70237	Planen u. Kalk. m. Multiplan	49,-
70241	C-Programmierung	49,-
NDR:		
70253	BANKBOOT Fertiggerät	49,-
70214	BANKSEL Bausatz	30,-
70215	BANKSEL Platine	10,-
70230	C64ADAPT Bausatz	40,-
70229	C64ADAPT Fertiggerät	60,-
70232	C64ADAPT Handbuch	5,-
70231	C64ADAPT Platine	25,-
70243	Handbuch CP/M68K,dt.	49,-
70235	CP/M68K Kompakt - Plate dt.	10,-
70252	CPU64180 Fertiggerät r2	169,-
70251	CPUZ80 Fertiggerät r3	79,-
70220	CRT1 Fertiggerät (SMP - BUS)	150,-
70221	Eprom-Floppy Bausatz	99,-
70223	Eprom-Floppy Handbuch	10,-
70222	Eprom-Floppy Platine r2	20,-
70247	FLO2 Fertiggerät mit Kabel	198,-
70254	GDP64K Fertiggerät	198,-
70257	GEH2 + BUS4 + Netzteil - komplett verdrahtet und aufgebaut.	390,-
70255	KEY Fertiggerät r3	69,-
70212	div. BAS Monitore	57,-
70225	POW26/22 Bausatz V1	39,-
70228	POW26/22 Bausatz V2	55,-
70224	POW26/22 Fertiggerät V1	68,-
70227	POW26/22 Fertiggerät V2	97,-
70226	POW26/22 Platine	5,-
70217	RAM16 Bausatz	20,-
70216	RAM16 Fertiggerät	30,-
70218	RAM16 Handbuch	5,-
70219	RAM16 Platine	10,-
70248	RAM256 Fertiggerät	198,-
70250	ROA64 Fertiggerät	49,-
70249	SER Fertiggerät	97,-
IBM:		
70174	TURBO Sockel für PC (80286)	570,-
70170	IBM 5250 Emulationsadapter	2468,-
70176	IBM Speicherkarte AT 128 KB	300,-
70165	IBM Monochrom Graphikkarte	97,-
70208	IBM CGA Farbbildschirm. 14" TTL	400,-
70185	IBM Monochrom Bildschirm	250,-
70175	IBM Hercules plus	250,-
70180	IBM Traktor für PCNT	147,-
70187	IBM Thermodrucker 5202-001	2200,-
70178	Interface seriell LQ1500	348,-
70186	IBM Graphikdrucker II	1398,-
70188	30MB Festplatte AT03	900,-
70189	HD-LW 20MB für XT	400,-
70190	80MB Festplatte für IBM	1400,-
70161	360KB Laufwerk für PC/XT	160,-
70145	IBM PC Finanzbuchhaltung	698,-
70139	XENIX-Betriebssystem engl.	498,-
70137	DOS2.0 Handbuch dt.	40,-
70138	DOS2.1 Handbuch dt.	40,-
70169	5250 Emulationsprogramm dt.	890,-
70143	Writing Assistant dt.	298,-
70144	Graphic Assistant dt.	298,-
70140	FILING Assistant dt.	298,-
70141	Reporting Assistant dt.	298,-
70142	Planning Assistant dt.	298,-
70206	Decathlon PC-DOS IBM PC	50,-
70201	Open Access, IBM PC/XT	2050,-
70204	Wordlord Data-Manager	130,-
70205	Wordlord Font-Editor	130,-
70203	Wordlord Mailmerge	130,-
70202	Wordlord Textverarbeitung	130,-
70167	IBM Portabel Tastatur	498,-
70146	IBM PC Emulation LQ-1500	97,-
70147	IBM PC Emulation FX100+	49,-
70135	IBM PC XT Bedienerhandbuch	40,-
70136	IBM PC AT03 Bedienerhandbuch	40,-
70133	IBM PC Bedienerhandbuch	40,-
70131	Microsoft-Chart IBM PC DOS	350,-
70134	Lotus 1-2-3 deutsch	800,-
70162	UNICAD	350,-
70182	TURBO Pascal Toolbox dt.	120,-
70163	TURBO Pascal 3.0	145,-
70148	WORDSTAR 2000 deutsch	900,-
70210	BMC-Drucker BM-80	148,-
70152	STARDIVISION Star-Manager	99,-
APPLE:		
70156	Interface Text für AppleII	50,-
70158	Interface Graphik AppleII	50,-
70197	Applesoft Programmier Handbuch	50,-
70199	Arbeitsbuch Macintosh	20,-
770154	Kabel für Applekarte 8131	43,-
70153	Kabel für Apple Graphik 8132	73,-
EPSON:		
70213	Epson Drucker FX100+	800,-
70159	Epson Kabel TRS80	59,-

gelöscht und der Rahmen neu aufgebaut werden. Deshalb wurde dieser Programmteil in ein Unterprogramm verwandelt. Für die Umwandlung von Kleinschrift in Großschrift bei Eingaben wurde ein weiteres Unterprogramm fällig.

Das anschließend mit BASCOM
ZEIGBLET.TER,ZEIGBLET=ZEIGBLET/E/O
compilierte und mit
L80 ZEIGBLET,ZEIGBLET/N/E
gelinkte Programm überrascht mit der sagenhaften Geschwindigkeit von nur 16 Sekunden (EF9366 + Z80A, 4 MHz). Vergleich: Originalgeschwindigkeit 10 Sekunden auf einem 10-MHz-PC (8/16-Bit Prozessor).

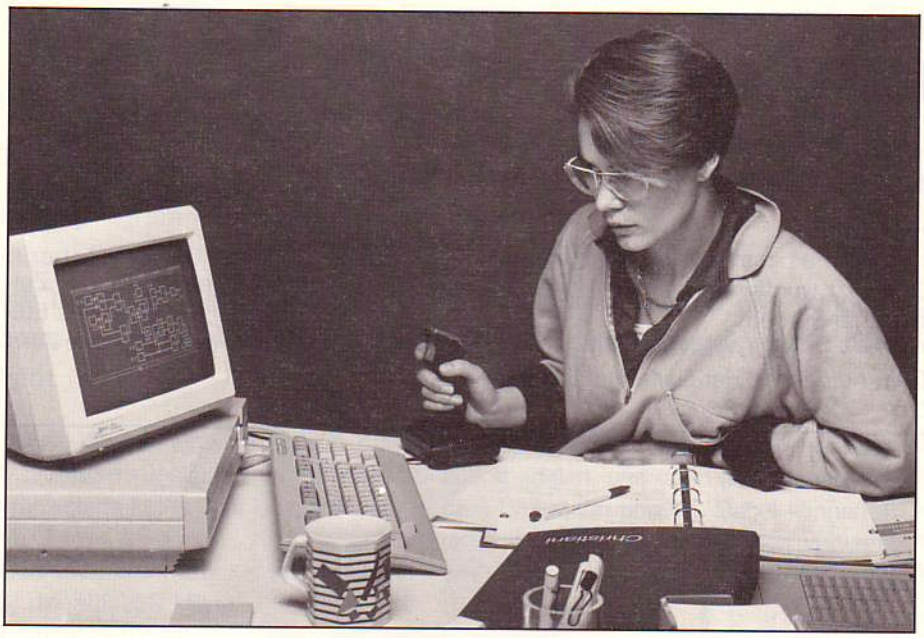
Weitere, noch nicht verwirklichte Möglichkeiten sind:
- Abfrage ob Breitschrift oder Schmalschrift. Dazu müssten die sprechenden Steuerzeichen, (bei meiner WordStar-Patch-Version ^A, ^N und ^R) abgefragt werden, die entweder die Zeichen-UP mehrfach aufrufen oder eigene UP's anspringen.
- Abfrage des aktuellen Laufwerkes (Adr. 0004, LSB) und der bearbeiteten Datei (File Control Block, Adr. 005C ff.)
- volle Integration in WordStar. Dazu verweise ich auf Wordstar Tuning von Werner Borsbach, S. 191 ff. In NOFTAB ist ein neuer Befehl (z.B. ^Z 00 59 03 ab Adr. 0478h) einzutragen. In Adr. 0478 ist ein Sprung zu ZEIGBLET.COM zu notieren. ZEIGBLET.TER muß mit RET statt mit END enden und ZEIGBLET.REL ist dafür auf eine feste Adresse (/P:Adr) zu linken. Ganz Versierte könnten schließlich noch den neuen Befehl im Wordstar-Hauptmenü eintragen (WSMSG.OVR, WSOVLY1.OVR).

Laufzeit-Vergleich:
Version

	.BAS un (geordnet)	.COM un (geordnet)
FLOMON-GRAFIK, IF-Abfragen	3'40" 3'16"	3'23" 3'04"
dgl. ON CH+X GOSUB Y	2'20"	; nur Rahmen
zusätzlich IF CH < X+1 THEN 520	1'48"	; sichtbar
altern. ON CH GOSUB Y	5'16"	; Linien-
dgl. CH verringern	2'12"	; gewirr
Rechteck mit print "R";DX;DY	2'28" 2'17"	
KURZVEKTOREN für Buchstaben	2'05"	; nur Rahmen
dgl. print 1B;1B;"G";"M";"A"	2'58"	; Rahm. zerstört
Leerzeichen mit Kurzvektoren	1'36" 1'23"	; 1. Hälfte i.O.
dgl. <CR> mit GDP-Befehlen	1'20"	0'16"

```
1 ON ERROR GOTO 3000
5 PRINT CHR$(27);"z1"          :REM Deutscher Zeichensatz
10 PRINT CHR$(26)              :REM Clearscreen
20 PRINT "ZEIGBLET.TER  Programm zur Darstellung der gesamten Seite
30 PRINT "   (c) Johannes Werner Gleim        25.10.1988"
34 PRINT "    Rev.1: Zeile 842: '(N)eue Datei'
35 PRINT "           Zeile 665: LW$ + Datei$  21.12.1988"
40 PRINT: PRINT "Rechteck und Leerzeichen mit Kurzvektoren"
50 PRINT "RETURN mit X- und Y-Register"
60 PRINT "Rahmen mit Moveto- und Drawto relativ-Befehlen"
70 FOR I=0 TO 4000: NEXT
80 REM ------------- Voreinstellungen und Menü
90 GDP = &H70: SEITE = 1        :REM GDP-Port / erste Seite
100 PTLIN = 4: PTCHR = 4        :REM Linien pro Zeile/Zeichen
110 PRINT: PRINT
115 REM linker/rechter Rand, Tabulator, oberer/unterer Rand
116 REM Bildschirmfenster, Zeichenzähler, Seitenlänge- und breite
120 LR=12: RR=73: HT=0: MT=1: MB=4: PLLIN=256: PBPT=220: ZZ=1
130 PL = INT (PLLIN / PTLIN): PB = INT (PBPT / PTCHR)
140 PRINT CHR$(26)
150 PRINT "(1) Voreinstellungen zeigen und/oder ändern"
160 PRINT "(2) Datei lesen und anzeigen"
170 PRINT "(3) EXIT"
180 PRINT: PRINT
190 INPUT "Wählen Sie bitte ",I
200 IF I = 1 THEN 240
210 IF I = 2 THEN 600
220 IF I = 3 THEN END
230 GOTO 190
240 PRINT: PRINT: PRINT "Voreinstellungen sind": PRINT
250 PRINT TAB(30) "Oberer Rand";MT
260 PRINT TAB(10) "Linker Rand";LR;TAB(50) "Rechter Rand";RR
270 PRINT TAB(30) "Unterer Rand";MB
280 PRINT PL-MT-MB;"(";PL;") Zeilen pro Seite";
290 PRINT TAB(43) RR-LR;"(";PB;") Zeichen pro Zeile"
300 PRINT TAB(8) PTLIN;" Linien pro Zeile";TAB(50) PTCHR;" Punkte pro Zeichen"
310 PRINT TAB(20) "TAB's alle";HT;"Zeichen gesetzt"
320 PRINT: PRINT: INPUT "Soll geändert werden ";I$
330 IF I$ = "j" OR I$ = "J" THEN 340 ELSE 140
340 PRINT
350 PRINT "(1) Linker Rand": PRINT "(2) Rechter Rand": PRINT "(3) Oberer Rand"
360 PRINT "(4) Unterer Rand": PRINT "(5) Seitenlänge": PRINT "(6) Zeilenhöhe"
370 PRINT "(7) Blattbreite in Zeichen": PRINT "(8) Zeichenbreite"
380 PRINT "(9) Tabulatoren": PRINT "(0) Menü"
390 INPUT;I: ON I+1 GOTO 140,410,420,430,440,450,460,480,490,470
400 GOTO 390
410 INPUT " Linker Rand ";LR: GOTO 390
420 INPUT " Rechter Rand ";RR: GOTO 390
430 INPUT " Oberer Rand ";MT: GOTO 390
440 INPUT " Unterer Rand ";MB: GOTO 390
450 INPUT " Seitenlänge ";PL: PLLIN=PL*PTLIN: GOTO 390
460 INPUT " Linien / Zeile ";PTLIN: PL=INT(PLLIN/PTLIN): GOTO 390
470 INPUT " Tabulatoren alle x Zeichen ";HT: GOTO 390
480 INPUT " Blattbreite in Zeichen ";PB: PBPT=PB*PTCHR: GOTO 390
490 INPUT " Punkte / Zeichen ";PTCHR: PB=INT(PBPT/PTCHR): GOTO 390
590 LW$ = ""
                                :REM damit Dateiname unverfälscht
600 PRINT: INPUT "Directory (j/n) ";I$
610 IF I$ = "n" OR I$ = "N" THEN 650
620     INPUT "Welches Laufwerk ";LW$
625     Y$ = LEFT$(LW$,1): GOSUB 2500: LW$=Z$   :REM in Großbuchstaben wandeln
626     IF LW$ <> "A" AND LW$ <> "B" THEN PRINT "ungültiges Laufwerk": GOTO 600
630     DIR$ = LW$ + ":*.*"     :REM alle Files des betreffenden LW
640     FILES DIR$
645     PRINT: PRINT "Laufwerkspezifikation ";LW$;": angeben"
646     GOTO 660
650 PRINT: PRINT "Laufwerkspezifikation A:/B: angeben"
660 PRINT: PRINT: INPUT "Welche Datei anzeigen ";Y$
661 GOSUB 2500: DAT$ = Z$                   :REM in Großbuchstaben wandeln
662 IF MID$(DAT$,2,1) = ":" THEN 720
665 DAT$ = LW$ + DAT$                       :REM vollständ. Dateibezeichnung
670 GOTO 720
680 PRINT "Eingabe wiederholen (j/n) ? ": INPUT WDH$
690 IF WDH$ = "j" OR WDH$ = "J" THEN 590
700 GOTO 840                                :REM da Datei nicht gefunden
720 OPEN "I",#1, DAT$                       :REM Datei öffnen
725 GOSUB 1100                              :REM Rahmen zeichnen
730 WHILE NOT EOF(1)                        :REM bis Textende
740     CHAR$ = INPUT$ (1,#1)               :REM jeweils ein Zeichen lesen
750     CHAR = ASC(CHAR$)
760     IF CHAR >127 THEN CHAR = CHAR - 128     :REM Bit 7 Null setzen
770     IF CHAR > 95 THEN GOSUB 1500: GOTO 820  :REM Kleinbuchstabe
780     IF CHAR = 32 THEN GOSUB 1400: GOTO 820  :REM Leerzeichen einfügen
790     IF CHAR > 32 THEN GOSUB 1600: GOTO 820  :REM Großbuchstabe
800     IF CHAR = 13 THEN GOTO 900              :REM (CR)
810     IF CHAR =  9 THEN GOSUB 1300            :REM  Tabulator expandieren
820 WEND
830 CLOSE
840 PRINT "(E)nde oder"
842 INPUT "(N)eue Datei";I$
845 IF I$ = "N" OR I$ = "n" THEN CLOSE: GOTO 90
850 PRINT CHR$(26)                          :REM Clearscreen
860 END
890 REM -------------------- Unterprogramm für (CR)
900 ZZ = ZZ + 1                             :REM Zeilenzähler erhöhen
910 IF ZZ < PL - MT - MB THEN 960           :REM Seitenende?
920     INPUT "Weiter (j/n";I$
930     IF X$ = "j" OR X$ = "J" THEN 940 ELSE 840
940     SEITE = SEITE + 1: GOSUB 1100       :REM neuer Rahmen
950     ZZ = 0
960 Y = 255 - PTLIN * (MT+4 + ZZ)           :REM (CR)
970 OUT GDP+1,10                :REM nichtschreiben oder löschen
980 OUT GDP+8,0 : OUT GDP+9,100             :REM X = 100
990 OUT GDP+10,0: OUT GDP+11,Y              :REM Y = Y - 4
1000 GOTO 820
1080 REM ---------------------- Unterprogramm Rahmen
1100 PRINT CHR$(26)                         :REM Clearscreen
1110 PRINT "Seite";SEITE
1120 X = 100: Y = 255
1130 GOSUB 1700                             :REM Grafik EIN
1140 GOSUB 1800                             :REM Moveto linke obere Ecke
1150 DX = PB * PTCHR: DY = 0
1160 GOSUB 2000                             :REM Drawto rechte obere Ecke
1170 DX = 0: DY = 1 - (PL * PTLIN)
1180 GOSUB 2000                             :REM Drawto rechte untere Ecke
1190 DX = - PB * PTCHR: DY = 0
1200 GOSUB 2000                             :REM Drawto linke untere Ecke
1210 DX = 0: DY = (PL * PTLIN) - 1
1220 GOSUB 2000                             :REM Drawto linke obere Ecke
1230 X = 100: Y = 255 - (MT+4)*PTLIN
1240 GOSUB 1800                             :REM Moveto Textbeginn
1250 GOSUB 2100                             :REM Grafik AUS
1260 RETURN
1290 REM ---------------------------- Unterprogramm Tabulator
1300 FOR I = 1 TO HT
1310     GOSUB 1400                         :REM Moveto nächste Position
1320 NEXT
1330 RETURN
1390 REM ---------------------- Unterprogramm 1 Leerzeichen
1400 OUT GDP+1,10                           :REM nicht schreiben oder löschen
1410 OUT GDP,208                            :REM 2 Punkte nach rechts
1420 OUT GDP,208                            :REM nochmal 2 pt
1430 RETURN
1480 REM ---------------------------- KURZVEKTOREN
1490 REM                                        Kleinbuchstaben
1500 OUT GDP+1,10                           :REM nicht schreiben oder löschen
1510 OUT GDP,209                            :REM 2 Punkte nach rechts
1520 OUT GDP,209                            :REM nochmal 2 pt
1530 OUT GDP+1,11                           :REM im Fenster schreiben
1540 OUT GDP,248                            :REM 2 pt nach rechts
1550 OUT GDP,250                            :REM 2 pt nach oben
1560 OUT GDP,214                            :REM 2 pt nach links
1570 OUT GDP,212                            :REM 2 pt nach unten
1580 RETURN
1590 REM ---------------------------- GROSSBUCHSTABEN
1600 OUT GDP+1,10                           :REM nicht schreiben oder löschen
1610 OUT GDP,208                            :REM 2 Punkte nach rechts
1620 OUT GDP,208                            :REM nochmal 2 pt
1630 OUT GDP+1,11                           :REM im Fenster schreiben
1640 OUT GDP,248                            :REM 3 pt nach rechts
1650 OUT GDP,250                            :REM 3 pt nach oben
1660 OUT GDP,254                            :REM 3 pt nach links
1670 OUT GDP,252                            :REM 3 pt nach unten
1680 RETURN
1690 REM ---------------------------- Grafikroutinen
1700 PRINT CHR$(27);CHR$(27);"GV0;P0"       :REM Grafik EIN
1710 RETURN
1800 PRINT "M";X;Y                          :REM Moveto
1810 RETURN
1900 PRINT "D";X;Y                          :REM Drawto (x,y)
1910 RETURN
2000 PRINT "D";DX;DY                        :REM Drawto relativ
2010 RETURN
2100 PRINT "A"                              :REM Grafik aus
2110 RETURN
2490 REM ---------------------------- Großbuchstabenwandlung
2500 YLEN = LEN(Y$): Z$ = ""
2510 FOR I = 1 TO YLEN
2520     CH = ASC(MID$(Y$,I,1))             :REM Einzelbuchstaben testen
2530     IF CH > 96 THEN CH = CH - 32
2535     CH$ = CHR$(CH)
2540     Z$ = Z$ + CH$                      :REM String wieder zusammensetzen
2550 NEXT
2560 RETURN
3000 IF ERR=53 AND ERL=720 THEN PRINT "Datei existiert nicht": CLOSE:RESUME 680
3010 IF ERR=64 AND ERL=720 THEN PRINT DAT$;" falsches Format": CLOSE:RESUME 680
3020 ON ERROR GOTO 0
```

Bild 1: Vollständiges Listing des Programms ZEIGBLET.TER



Kompakt-Kurs

Digital-Computer-Labor

NEU

200 Seiten reichbebildertes Lehrmaterial mit Sammelordner, Register und Logik-Simulator für den Commodore C64 / C128 oder IBM- und kompatible Computer mit MS-DOS. Gesamtpreis DM 448,–

Lernen Sie die Bauelemente und Grundschaltungen der Digitaltechnik mit einem ausgezeichneten Logik-Simulator kennen.

Der Lehrgangsaufbau

Der Lehrstoff ist in zwei Fachgebiete aufgeteilt; die Digitaltechnik und die Steuerungstechnik. In der Digitaltechnik machen Sie zunächst Bekanntschaft mit den Grundlagen. Theorie und Praxis gehen Hand in Hand. Sie lernen einfache Schaltungen mit dem Logik-Simulator auf dem Bildschirm darzustellen und auszutesten.

Eine Zeichnung auf dem Bildschirm, die man ausprobieren kann – ein einmaliges Erlebnis sinnvoller Computeranwendung. Nahezu alle Schaltungen, die im Lehrmaterial beschrieben sind, können mit dem Logik-Simulator ausgetestet werden. Das Aufbauen von Schaltungen hat damit ein Ende. Sie werden begeistert sein.

Digitaltechnik

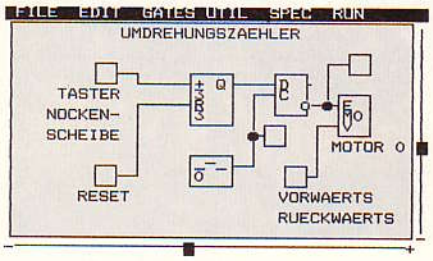
Die Digitaltechnik behandelt alle wichtigen Bauelemente. Wir beginnen mit den logischen Verknüpfungen und gehen über Speicher und Flip-Flops, Schaltalgebra, Zahlensysteme, Codierschaltungen, bis hin zur Mikroprozessortechnik, um nur einige der interessanten Themen zu nennen.

Steuerungstechnik

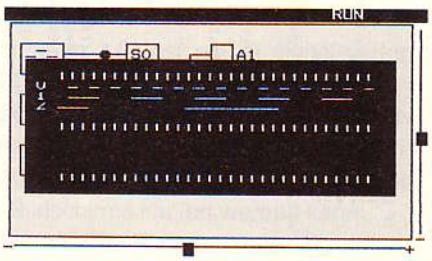
Die Steuerungstechnik hingegen zeigt Ihnen die Anwendung der digitalen Bauelemente. Ablaufsteuerungen, Schaltverstärker, Schrittmotorsteuerungen und vieles mehr wird innerhalb des Kurses ausführlich und leichtverständlich behandelt.

Der Logik-Simulator

Sicher wird Sie der neue leistungsstarke Logik-Simulator begeistern. Über POP-UP-Menüs können die verschiedensten Funktionen angewählt werden. Digitale Verknüpfungen, Schalter, Leuchtdioden, Motor, Lautsprecher, Flip-Flops, Zeitgeber und Zähler sind abrufbare Elemente.



Damit dynamische Prozesse auch leicht verfolgt werden können, kann der Logik-Simulator sogar ein Oszilloskop darstellen. Einfach das Symbol des Oszilloskops anwählen, und schon kann man bis zu 8 Signale verfolgen.



Für wen ist der Kompakt-Kurs interessant?

Für alle, die sich in das Gebiet der Digitaltechnik einarbeiten wollen. Für alle, die den Computer als Werkzeug bei der Schaltungsentwicklung kennenlernen und ausnutzen wollen. Und natürlich für jeden, der sich mit Computeranwendungen und -steuerungen auskennen will.

Was Sie brauchen

Sie brauchen als Vorkenntnisse lediglich Elektronik-Grundlagen und natürlich einen Commodore C64 oder C128 mit Diskettenlaufwerk und Joystick oder einen IBM- oder kompatiblen Computer mit Mouse. Die Software (Logik-Simulator) ist in der Kursgebühr enthalten.

Fangen Sie jetzt an! Die Technik der Computeranwendungen wartet nicht. Bald wird es selbstverständlich sein, Schaltungen aller Art mit dem Computer zu entwickeln. Mit dem Wissen aus diesem Kurs liegen Sie dann ganz vorn.

Christiani
Fortbildung

Postfach 35 000 · 7750 Konstanz
Telefon (0 75 31) 58 01-0

Jürgen Becker

Suchprogramm für C-Compiler

Wer unter CP/M 68K mit dem C-Compiler arbeitet und dafür den Editor des Grundprogrammes benutzt, soll hier angesprochen werden.

Da der Editor EDITRDK ohne Zeilennummern arbeitet, wird man es schwer haben in größeren Programmen Fehler zu finden, welche der Compiler aber mit Zeilennummer meldet. Das beigefügte C-Programm, welches ich SUCH.C nenne, soll da eine Hilfe sein.

Nachdem es zum SUCH.68K Programm übersetzt wurde, kann es z. B. wie folgt gestartet werden:

SUCH SUCH.C

Dem Aufruf muß der Name der Datei folgen, in der der Fehler gesucht wird (in diesem Beispiel der Datei SUCH.C). Dann wird man aufgefordert die gesuchte Zeilennummer einzugeben.

Diese wird dann mit der Zeile vor und nach der gesuchten auf der Console ausgegeben.

Man braucht nur noch eine markante Zeichenfolge über die Suchfunktion des Editors eingeben und hat die fehlerhafte Zeile gefunden.

```
/*****************************************************************************
*  -  Programm SUCH.C  -                 - Quelltext - Zeilensuchprogramm - *
*  1986      Juergen  Becker,        4500 Osnabrueck      861026     V1.0 *
*****************************************************************************/

#include <stdio.h>
#define  EOL 10

main(argc, argv)
int argc;
char **argv;

{                          /* BEGINN HAUPTPROGRAMM */

char getc();
char buff[20];
int zeich,nummer,vergleich,drei,f;
FILE *fopen();
FILE *fp;

if (argc != 2)
  {
  fprintf(stderr,"Bitte Dateinamen mit angeben ");
  exit(1);
  }

do {                 /* noch einmal von vorne suchen */
  vergleich = 1;

  if (( fp = fopen(argv[1],"r")) == NULL )
    {
      printf("\n\tFehler :\n . . %s kann nicht ge|ffnet werden !", argv[1]);
      exit(2);
    }

  do {                 /* weiter suchen */

    do {
      printf(" Gib die Zeilen Nummer ein ! :");
      gets(buff); sscanf(buff,"%d", &nummer);
      f = nummer - 3;
      } while (nummer < 3 || f < vergleich);

    --nummer;        /* Damit auch die Reihe vorher ausgegeben wird */

  do {
    do {
      zeich = getc(fp);
      } while (zeich != EOL && zeich != EOF);
        /* Wenn Zeilenende kommt, aus Schleife raus */
    ++vergleich;
    }  while (vergleich != nummer && zeich != EOF);
        /* Wenn Zeilenummer gefunden aus Schleife raus */

    if (zeich != EOF)
    {
    drei = 0;

    while (drei != 3)
      printf("\nZeile Nr. %d :\n", vergleich);
      zeilout(fp);
      ++drei;
      ++vergleich;                }

    }
    else
    {
    printf("\n\nF e h l e r !  Falsche Eingabe !\n Zeilen ");
    printf("Nummern }ber  > %d < gibt es nicht ! !\n", vergleich);
    }

    printf("\n Noch einmal von vorne suchen  =  2   weiter suchen  =");
    printf("  1   Ende  =  0\n\n");
    gets(buff); sscanf(buff,"%d", &nummer);

    } while (nummer == 1);      /* Ende weiter suchen */

  fclose(fp);

  } while (nummer == 2);    /* Ende noch mal von vorne suchen */

  printf(" \n\nEnde \n");

}     /* Ende main */

zeilout(fp)
FILE *fp;
{
int zei;

while ((zei = getc(fp)) != EOF && zei != EOL) putchar(zei);

}
```

Listing des Suchprogramm

Jürgen Meyer

JMS - PAINT

Das Malprogramm für den NDR - Computer.
Ausbau: CPU 68O xx, COL256/256K
Nach neunmonatiger Entwicklungszeit kann nun endlich das Malprogramm vorgestellt werden, das alle bisherigen in den Schatten stellt: JMS - Paint.

90 kByte Maschinencode sorgen dafür, daß Sie Ihren künstlerischen Fähigkeiten am Bildschirm freien Lauf lassen können.

Dieses Programm arbeitet auf der COL-256, die auf 256 kByte ausgebaut sein muß.
Es wird als Maschinenprogramm geladen und stellt sich selbst auf die jeweilige CPU ein.
Um dieses Programm zu bedienen, wird die Maus benötigt, mit der auch alle Einstellungen vorgenommen werden können. Um Einstellungen und Funktionen zu ändern, ist am oberen Bildschirmrand eine Statuszeile vorhanden, die in 4 Menüpunkte eingeteilt ist.
Mit dem Menüpunkt FUNKTION lassen sich alle 16 Malfunktionen aufrufen.
Die "normalen" Funktionen, wie LINIE, RECHTECK, BLOCK, KREIS oder SCHEIBE sind als Gummilinienfunktionen ausgeführt. Sie lassen sich jederzeit unterbrechen und sogar nach der Ausführung wieder rückgängig machen (falls nötig).
Mit MALE kann man auf dem Bildschirm freihändig malen, während mit RADIERE das Gemalte wieder gelöscht werden kann.
Eine Besonderheit stellt die Funktion PINSEL dar. Mit ihr kann man auf dem Bildschirm malen, wobei der Malstift einem Pinsel gleicht, der verschiedene Dicken und Formen besitzt. In einen speziellen Menüpunkt kann aus 30 verschiedenen Pinseln der geeignete gewählt werden.
Mit SCHREIBEN können Texte in das Bild integriert werden. Diese Texte werden zuerst editiert, und dann punktgenau auf dem Bildschirm plaziert. Dabei können 4 unterschiedliche Zeichengrößen verwendet werden. Auch besteht die Möglichkeit,den Zeichensatz aus 14 verschiedenen auszuwählen.

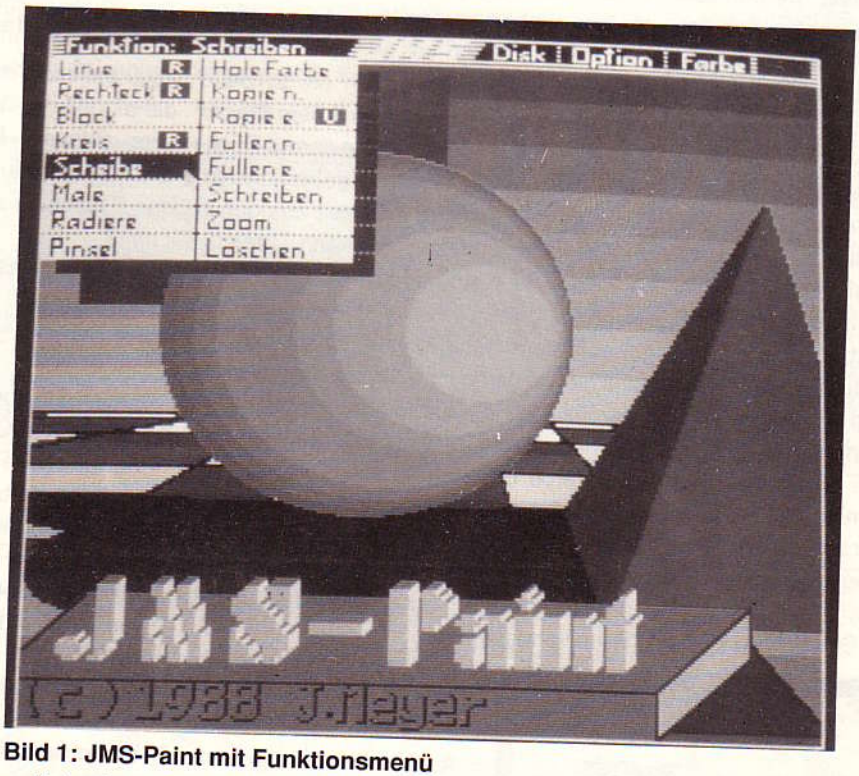


Bild 1: JMS-Paint mit Funktionsmenü

Eine ZOOM - Funktion ist auch enthalten. Sie gibt das Bild in 4 - facher Vergrößerung wieder. Dabei können einzelne Bildpunkte geändert werden, wobei noch die Möglichkeit besteht, Farben aus dem Bild als Zeichenfarbe zu übernehmen.
So können z.B. bei einem digitalisierten Bild Farbunreinheiten überdeckt werden.
Einzelne Bildteile lassen sich mit KOPIE N kopieren. Dabei wird ein Rechteck definiert, dessen Inhalt an jede beliebige Stelle des Bildschirms kopiert werden kann.
Neben KOPIE N (N wie normal) gibt es aber auch noch KOPIE E (E wie extra). Diese Funktion verhält sich im Großen und Ganzen wie der normale Kopierbefehl, nur daß hier eine (vorher definierbare) Farbe nicht mitkopiert wird. Dadurch kann man komplexe Figuren aus ihrem Hintergrund ausschneiden und sie in andere Bilder einsetzen. Auch besteht hier die Möglichkeit, Bilder oder Teile eines Bildes in eine 2. Bildschirmseite abzulegen. Dadurch lassen sich zwei unterschiedliche Bilder gleichzeitig im Speicher bearbeiten oder miteinander kombinieren.

Dies wären nun die wichtigsten Funktionen, die mit dem ersten Fenster aufgerufen werden können.
Mit DISK, der zweiten Option in der Kommandozeile, lassen sich Bilder laden und abspeichern. Hierbei besteht die Möglichkeit, das gemalte Bild normal oder in komprimierter Form auf Diskette zu bannen. Die Berechnung des Bildes erfolgt automatisch nach Anklikken des jeweiligen Menüpunktes, wobei die max. Zeit zum Berechnen des komprimierten Bildes nicht mehr als 2 Sek. beträgt. Nach dem Berechnen des Bildes wird der benötigte Speicherbedarf angezeigt und nach den Namen des Bildes gefragt. Hierbei kann ein 8-stelliger Name eingegeben werden, wobei die Bildkennung automatisch integriert wird.
Beim Laden eines Bildes öffnet sich ein zweites Fenster, in dem alle vorhandenen Bilder auf der Diskette aufgelistet werden. In diesem Fenster kann (wie in einem Buch) geblättert werden, da es bei komprimierten Bildern möglich ist, bis zu 100 Bildern auf einer Diskette zu speichern. Das gewünschte Bild muß nur noch mit der Maus angewählt werden, es wird geladen und sofort am Bildschirm angezeigt. Komprimierte Bilder werden erkannt und beim Anzeigen umgerechnet. Dadurch wurde es möglich, die Umrechnungszeit so kurz wie möglich zu halten (bei der 68000-CPU in max. 1 Sek.) Durch das normale Abspeichern des Bildes ist es möglich, dieses in eigenen Programmen zu verarbeiten.

Auch befindet sich der Algorithmus zum Umrechnen der komprimierten Bilder als Quelltext auf der Programmdiskette. Somit können Bilder dieses Programmes jederzeit in eigene Programme integriert werden.

Mit dem Optionenfeld in der Kommandozeile werden alle Einstellungen des Prgrammes vorgenmmen. Es erscheint ein Fenster, mit dessen Hilfe weitere Fenster geöffnet werden, in denen die einzelnen Funktionen geändert werden. Beim Fenster AUSGABE läßt sich z.B. die Bildschirmausabe beeinflussen. Es werden hier die Ausgabegröße der Zeichen bei der Funktion SCHREIBE und die Verknüpfungsart der Farben eingestellt. Die Verknüpfungsart kann auf ABSOLUT, AND, OR und EXOR eingestellt werden Sie beeinflusst die Farbenausgabe bei den meisten Malfunktionen. Beim Fenster-CURSOR kann das Fadenkreuz manipuliert werden. Es besteht die Möglichkeit, den Cursor wahlweise als Fadenkreuz oder als Pfeil darzustellen, wobei es beim Pfeil auch noch zwei verschieden Arten gibt. Auch kann zwischen S/W-Ausgabe oder automatischen Farbwechsel bei der Cursordarstellung gewählt werden.

Die linke Maustaste hat bei diesem Programm eine "Turbofunktion". Hiermit kann der Cursor beschleunigt werden. Der Beschleunigungsfaktor kann in diesem Fenster zwischen 1 (keine Beschleunigung) und 3 (max. Beschleuinigung) eingestellt werden. Es besteht auch die Möglichkeit, die Funktion der beiden Maustasten zu vertauschen.

Mit dem Fenster FONTS läßt sich der Zeichensatz einstellen, der bei der Funktion SCHREIBE wirksam wird. Es gibt 14 dieser Zeichensätze, sie haben eine Zeichengröße von 9*16 Punkten und deutsche Umlaute. Mit dem Fenster PINSEL kann ein Pinsel aus 30 verschiedenen gewählt werden. Das Fenster BILD 2 FUNKT läßt Operationen mit der Hilfseite zu. Es kann die Hilfsseite in die Malseite oder die Malseite in die Hilfsseite kopiert werden. Auch können die beiden Seiten ausgetauscht werden, ohne den jewiligen Inhalt zu zerstören. Dadurch kann vor einer Malfunktion das Bild in die 2. Seite kopiert werden. Nun kann mit dem Malbid experimentiert werden, ohne darauf Rücksicht zu nehmen, daß es dabei zerstört wird, da das Original jederzeit aus der 2. Seite zurückkopiert werden kann. Das letzte Feld in der Kommandozeile ist das "Farbenfeld". Hier ist immer die aktuelle Zeichenfarbe sichtbar. Durch Anklicken dieses Feldes gelangt man in die Farbenseite. Hier kann eine neue Malfarbe ausgewählt werden. Dabei ist immer die aktuelle Zeichenfarbe und deren hex. Wert sichtbar. Durch das erste Anklicken einer Farbe wird diese als Zeichenfarbe übernommen und am unteren Bildschirmrand angezeigt. Mit dem zweiten Anklicken wird dann diese Seite verlassen.

Bild 2: Verschiedene Schriftarten

Dies zur Bedienung des Malprogrammes. Alle Funktionen und Möglichkeiten dieses Programms zu beschreiben, würde zuviel Platz in Anspruch nehmen. Deshalb wurde hier nur auf die wichtigsten Punkte eingegangen. Erwähnenswert ist noch, daß alle Funktionen und Einstellungen auf der GDP sichtbar sind. Man hat also ständig einen Überblick über die Einstellungen und deren Auswirkungen auf das gemalte Bild.

Es wird auch eine Schriftprobe des eingestellten Zeichensatzes dargestellt. Auch kann man die aktuelle X - und Y - Position des Cursors ablesen. Dadurch ist es möglich, Zeichnungen aus Vorlagen punktgenau in den Bildschirm zu übertragen.

Bild 3: Kombination mit digitalisierten Bildern

Zusammenfassung der wichtigsten Eigenschaften :

* Programmlänge 90 kByte (Maschinen-Code)
* benötigter Speicher 240 kByte
* ausgebaute COL - 256 wird benötigt
* automatische Kennung der CPU
* Programm über Maus bedienbar
* Alle Funktionen über Fenster einstellbar
* Malen mit 256 Farben
* Farben verknüpfbar (UND, ODER, EXOR)
* 16 verschiedene Malfunktionen
* 2 verschiedene Kopierfunktionen mit verschiedenen Effekten (Glasscheibe, Blue-Box)
* Schreiben mit 4 verschiedenen Zeichengrößen
* 14 Zeichensätze zur Auswahl
* Malen mit Pinsel, (30 unterschiedliche Pinselformen)
* Umschaltung zwischen Fadenkreuz und Pfeil
* Darstellung des Pfeiles (oder Faden kreuz) in S/W oder mit Farbenwechsel
* Beschleunigung des Fadenkreuzes mit "Turbotaste" auf der Maus
* Turbotaste wahlweise links oder rechts
* Beschleunigung einstellbar
* Hilfsseite zum Zwischenspeichern

von Bildern oder Bildteilen
* Laden der Bilder über Maus möglich
* Abspeichern des Bildes wahlweise mit oder ohne Komprimierung möglich
* Komprimierung in 2 Sek. (68000)
* Automatische Erkennung von komprimierten Bildern beim Laden und automatische "Entkomprimierung" beim Bildaufbau (1 Sek.)
* Alle Einstellungen, Pinselform, Schriftprobe des angewählten Zeichensatzes und X/Y - Position des Fadenkreuzes auf der GDP sichtbar

Der Lieferumfang besteht aus 3 (drei) Disketten.

Die Programmdiskette
Sie beinhaltet das Hauptprogramm.

Die Demodiskette
Sie enthält eine Bedienungsanleitung, ein Demoprogramm, mit der man die Funktionen auf der GDP erklärt bekommt, wobei die Ergebnisse gleichzeitig auf der COL-256 sichtbar werden (mit über 60 Bildern).und das "Entkomprimierprogramm", mit dem es möglich ist, Bilder in eigene Prgramme zu übernehmen.

Die Bilderdiskette
Sie enthält über 60 Demobilder (gemalt, digitalisiert und vom Amiga) und JMS - Show. Mit diesem Programm ist es möglich, die Bilder wie bei einer DIA - Show darzustellen, wobei man zwischen verschieden Überblendeffekten wählen kann.

VORANKÜNDIGUNG:
Es folgen in absehbarer Zeit :

JMS - Digi
Ein Digitalisierer, mit dem es möglich sein wird, Bilder vom Fernsehgerät einzulesen. Dabei werden die Bilder in Echtzeit behandelt. Dadurch können die Bilder vom laufenden Programm eingelesen werden. Bei Verwendung eines RGB - Signales (Scart) können 256 Farben gleichzeitig sichtbar werden. Bei Anschluss eines FBAS - Signal erhält man immerhin noch 16 Graustufen.
Die Bilder werden im Paint - Format abgelegt, so daß diese Bilder mit dem Malprogramm nachbehandelt werden können.

JMS - Druck
Mit diesem Programm wird es möglich sein, Bilder in 16 Graustufen auf jeden Drucker auszugeben. Dabei kann man die Graustufen selbst definieren, Bilder dehnen, stauchen, invertieren oder spiegeln.
Alle Grafikprogramme, die noch kommen werden, sind kompatibel mit diesen Programmen, d.h. jedes Programm kann Bilder eines anderen Prgramms übernehmen.
Falls noch einige Fragen bestehen sollten, bin ich gerne bereit, diese zu beantworten.

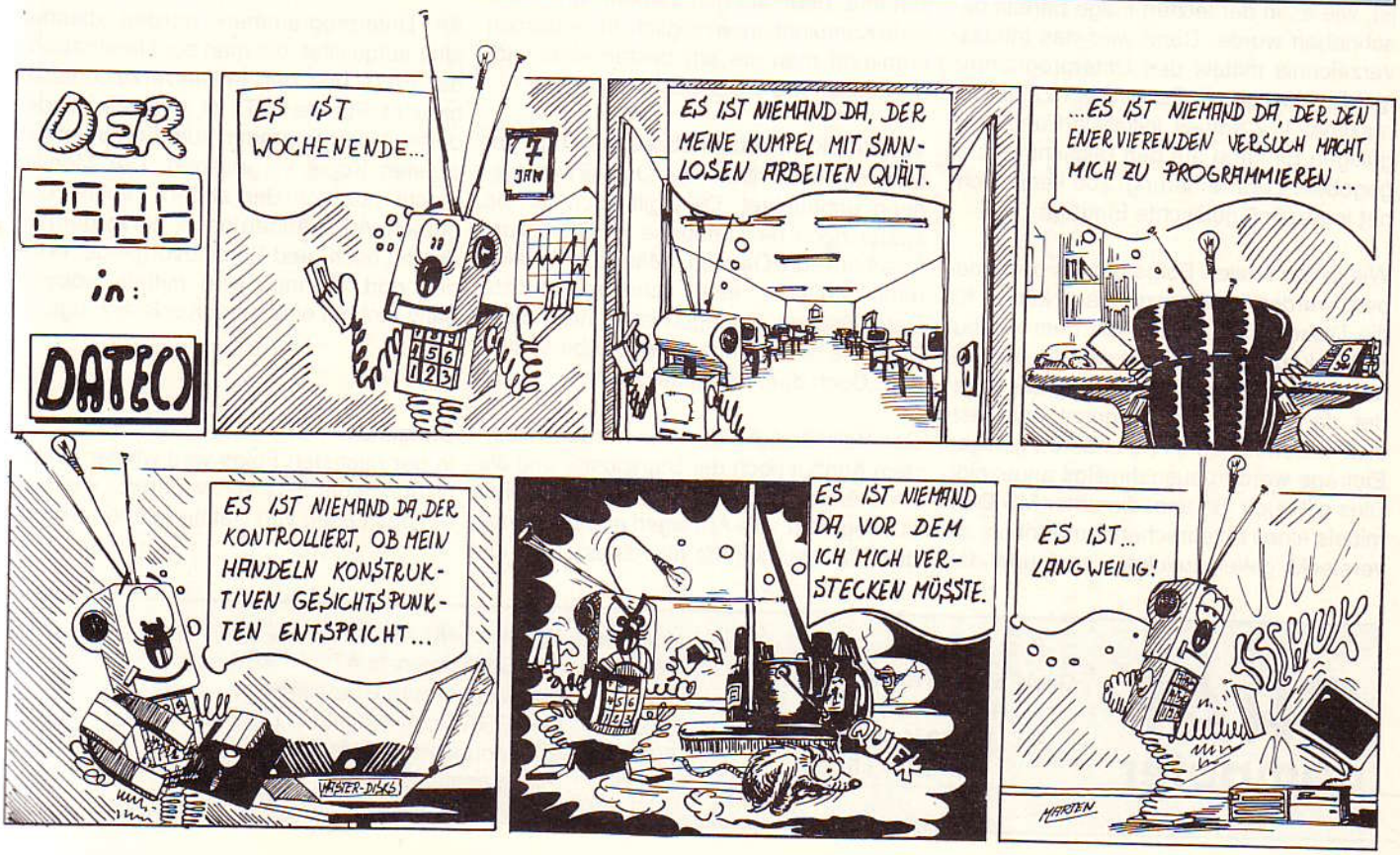
Jürgen Meyer
Rudolf Schiestl Str. 22
8510 Fürth
Tel. 0911/797807

GRAF computer

GRAF ELEKTRONIK SYSTEME
Magnusstraße 13
8960 Kempten
Tel.: 0831 / 6211

NDR-Computer / Zubehör:

70253	BANKBOOT Fertiggerät	49,-
70214	BANKSEL Bausatz	30,-
70215	BANKSEL Platine	10,-
70230	C64ADAPT Bausatz	40,-
70229	C64ADAPT Fertiggerät	60,-
70232	C64ADAPT Handbuch	5,-
70231	C64ADAPT Platine	25,-
70243	Handbuch CP/M68K,dt.	49,-
70235	CP/M68K Kompakt - Plate dt.	10,-
70252	CPU64180 Fertiggerät r2	169,-
70251	CPUZ80 Fertiggerät r3	79,-
70220	CRT1 Fertiggerät (SMP - BUS)	150,-



Teil1: Der verrückte Bootsektor
Loop 17

Teil 2: Eine gefährliche Operation
Loop 19

Teil 3: Log.-physikalisches Verwirrspiel
Loop 20

Teil 4: Sage mir, was Du hast
Loop 21

Teil 5: Jetzt wird gelesen
Loop 22

Teil 6: Ärger mit Ä
Loop 23

Günter Renner

Sage mir,was Du hast

Fürs erste ist nun genug in den Tiefen des Systems gegründelt. Es ist Zeit, sich wieder einmal auf die Oberfläche zu begeben - auf diejenige des Benutzers. Als solcher will man es heute möglichst bequem haben und nicht immer an Cluster und an Dateieinträge denken müssen.

Moderne Datenträger sind zumindest äußerlich ziemlich nichtssagend. Und nur sehr ordentliche Menschen schreiben brav aufs Etikett, was auf der schwarzen Scheibe drauf ist. Gelassenere Zeitgenossen praktizieren hier ein anderes Vorgehen, das man treffend als Diskettenpoker bezeichnet. Sie stecken einfach eine nach der anderen in den Schlitz und sehen sich das Inhaltsverzeichnis an.

Diese Funktion steht natürlich auch im Pflichtenheft unseres Filemonitors. Sie wird mit dem Programm 'inhalt' verwirklicht, das diesmal im Listing enthalten ist. Um Mißverständnissen vorzubeugen, wird zuerst einmal geprüft, ob alles verträglich ist, wie es in der letzten Folge bereits beschrieben wurde. Dann wird das Inhaltsverzeichnis mittels des Unterprogramms 'getdir' eingelesen. Es wird ein Zähler für maximal 112 entries initialisiert und alle gültigen Einträge auf den Bildschirm ausgegeben. Zur Erinnerung: $00 kennzeichnet leere, $e5 gelöschte Einträge.

Wie in der letzten Folge bereits beschrieben, werden Kennsätze, also Namen für die Diskette als solche, mit dem Attribut $08 gekennzeichnet. Sie werden von 'inhalt' einfach übergangen, was nicht schadet, da man sich den Kennsatz mit 'testdisk' ansehen kann. Alle anderen gültigen Einträge werden ausnahmslos angezeigt. Dies gilt auch für jene, die unter MS-DOS mittels eines entsprechenden Attributs als versteckt gekennzeichnet sind und dort nicht im Inhaltsverzeichnis erscheinen würden. Ebenfalls ausgegeben werden Unterverzeichnisse, obgleich der Monitor diese nicht bearbeiten kann.

Neben den Namen und ggf. den Namenserweiterungen sollte man nun noch etwas mehr über die Diskinhalte erfahren. Diesem Ziel wird mit recht einfachen Mitteln nachgegangen. Zunächst einmal wird in der dritten Spalte schlicht das Attribut der Datei ausgegeben. Bei gewöhnlichen Dateien erscheint unabhängig vom Inhalt (Text, Code usw.) der Hexwert 20. Unterverzeichnisse sind an dem Wert 10 zu erkennen. Da es sich bei dem Attribut um ein Byte handelt, dessen einzelne Bits bestimmte Bedeutungen haben, sind sehr viele Kombinationen möglich. Im einzelnen entnimmt man sie am besten einer einschlägigen MS-DOS-Dokumentation.

Von den Kennsätzen abgesehen läßt der Monitor die Attribute der Dateien weitgehend unbeachtet. Dies gilt auch für die später noch beschriebene Funktion zum Löschen von Dateien. Man könnte also ohne weiteres eine schreibgeschützte Datei löschen. Dagegen ist es nicht möglich, eine bestehende Datei zu überschreiben. Doch darüber später mehr.

Der Vollständigkeit halber werden neben dem Attribut noch der Startcluster und die Dateilänge in Bytes auf den Bildschirm ausgegeben. Alle Anzeigen erfolgen hexadezimal. Man könnte allerdings durch die Wahl anderer PRINT-Funktionen auch eine dezimale Darstellung erreichen. Wie bereits erwähnt, enthalten die Einträge noch Datum und Uhrzeit. Diese werden vom Monitor weder benutzt noch angezeigt.

Trotz ihrer Einfachheit ist die Funktion 'inhalt' in einem Punkt recht komfortabel: die Bildschirmanzeige rast einem nicht davon. Man muß nicht wie ein Heftmacher aufpassen und 'Ctrl-S' drücken, sobald eine Seite voll ist. Denn die bekannte MS-DOS-Option /p bzw. MORE ist gleich eingebaut. Der Bildschirm bleibt nach 23 Zeilen gemächlich stehen und wartet auf einen Tastendruck...

An Unterprogrammen werden diesmal jene aufgelistet, die man zur Handhabung der FATs und des Inhaltsverzeichnisses braucht. Für die FATs ist noch ein gesondertes Unterprogramm zum Erzeugen der zweiten Kopie vorgesehen. Der Monitor macht nur von der ersten Tabelle Gebrauch. Auch Änderungen, hervorgerufen durch Lösch- und Schreibvorgänge, erfolgen dort. Danach wird mittels 'copyfat' ganz einfach eine neue Kopie erzeugt.

Übrigens :
In der nächsten Folge wird wieder einmal nichts an der FAT vorbeiführen - wenn es um das Lesen von Dateien gehen wird.

GRAF computer

GRAF ELEKTRONIK SYSTEME
Magnusstraße 13
8960 Kempten
Tel.: 0831 / 6211

Angebote IBM-Karten:

70176	IBM Speicherkarte AT 128 KB	300,-
70165	IBM Monochrom Graphikkarte	97,-
70208	IBM CGA Farbbildschirm. 14" TTL	400,-
70185	IBM Monochrom Bildschirm	250,-
70175	IBM Hercules plus	250,-

```
 *s34 Patchwork-Listing Teil 4
(C) Guenter Renner

****Menuefunktionen

inhalt:
bsr crlfkon
lea fhflag(pc),a4          *fehlerfrei
clr.b (a4)                 *annehmen
bsr testdis
bmi inhal7

bsr getdir                 *Inhalt einlesen
lea dirbuf(pc),a4          *und ausgeben
move #112-1,d6             *112 Eintraege

inhal1:
moveq #23-1,d5             *23 Zeilen

inhal2:
tst.b (a4)                 *ist freier
beq inhal6
cmp.b #$e5,(a4)            *ist geloeschter
beq inhal6                 *Eintrag
cmp.b #8,11(a4)            *ist Kennsatz
beq inhal6
movea.l a4,a3              *a4 schonen
moveq #11-1,d4

inhal4:
move.b (a3)+,d0            *Name ausgeben
move #!co2,d7
bsr traprts
cmp #3,d4                  *vor der Er-
bne inhal5                 *weiterung
moveq #' ',d0              *Leerstelle dazu
move #!co2,d7
bsr traprts

inhal5:
dbra d4,inhal4             *11 Zeichen
moveq #' ',d0              *Leerstelle dazu
move #!co2,d7
bsr traprts

move.b 11(a4),d0           *ist Attribut
lea einbuf(pc),a0
move #!print2x,d7          *hexadezimal
bsr traprts
move.b #' ',(a0)+          *mit Leerstelle
clr.b (a0)                 *und Ende!
lea einbuf(pc),a0          *ausgeben
bsr meldung

move $1a(a4),d0            *Startcluster
rol #8,d0
lea einbuf(pc),a0
move #!print4x,d7
bsr traprts
move.b #' ',(a0)+
clr.b (a0)
lea einbuf(pc),a0
bsr meldung

move $1e(a4),d0            *Dateigroesse
rol #8,d0
swap d0
move $1c(a4),d0            *wegen msb/lsb
rol #8,d0
lea einbuf(pc),a0
move #!print8x,d7          *hexadezimal
bsr traprts
lea einbuf(pc),a0          *ausgeben
bsr meldung
bsr crlfkon                *neue Zeile

dbra d5,inhal6             *max. 23 Zeilen
move #!ci,d7               *dann erst Taste
bsr traprts                *naechste Seite
moveq #23-1,d5             *mit 23 Zeilen

inhal6:
adda.l #$20,a4             *naechster Eintrag
dbra d6,inhal2             *bis 112, sonst

move #!ci,d7               *Ende von allem
bsr traprts

inhal7:
bsr testflg
bra menue
```

Das 68020-Grundprogramm Vers 6.1

Endlich ist es soweit !!!

Nachdem das Grundprogramm für den 68008 und 68000 gründlich überarbeitet wurde, gibt es die vielen neuen Möglichkeiten auch für den 68020.

Der 68020 ist bekanntlich das Glanzstück des NDR-Computers, der mit seiner großen Rechenleistung schon fast an professionelle Systeme heranreicht.
Leider konnte er bisher nur mit "halber Kraft" laufen, da nicht alle neuen Möglichkeiten mit dem Grundprogramm und speziell mit dem Assembler ausgeschöpft werden konnten. Beim Assembler gab es zwar die FPU-Befehle und ein paar kleine Erweiterungen im Befehlssatz, aber die meisten neuen Befehle und alle neuen Adressierungsarten standen nicht zur Verfügung.
Das hat sich jetzt geändert. Die Version 6.1 beherrscht alle !!! neuen Befehle und Adressierungsarten, wodurch solche "exotischen" Befehle möglich sind:

```
move.l (Marke1, zpc, a5*4.l ,Marke2.w), (Marke3,a4,d0*2.l,Marke 4.l)
oder
fsincos.d (d0*2.l),fp0:fp1
oder
cas2 d0:d1,d2:d3,(a0):(a1)
```

Weiterhin kann der Assembler jetzt auch endlich MACRO-Befehle verarbeiten, was sich viele Programmierer schon immer gewünscht oder mit Tricks selbst eingebaut haben. Dabei können Macros verschachtelt werden, so lange der Speicher ausreicht. Natürlich können diese mit (bis zu 10) Parametern aufgerufen werden.
Jedes Macro hat einen eigenen Zähler, so daß im Macro auch Schleifen aufgebaut werden können.
Zusätzlich gibt es noch Befehle, um Variablen leichter indirekt zu einem Adressregister zu definieren.
Beispiel für die Anwendung von Macros (Linie von 0,0 bis 100,100 - Bild 1)

```
MACRO grund                     * Grundprogrammaufruf mit Paramterübergabe
        movem.l d1/d2/d7,-(a7)  * Es werden keine Register zerstört
        move 1,d1               * Parameter 1 nach d1
        move 2,d2               * Parameter 2 nach d2
        move #!0,d7             * Parameter 0 nach d7
        trap #1                 * Programm aufrufen
        movem.l (a7)+,d1/d2/d7
ENDMACRO                        * Ende

linie:                          * Linie von 0,0 nach 100,100 zeichnen
        .grund moveto,#0,#0     * Befehl MOVETO mit Position
        .grund drawto,#100,#100 * Befehl DRAWTO mit Position
rts
```

Bild 1: Beispiel für die Anwendung eines Makro

Durch diese vielen Neuerungen ist der Assembler nicht mehr nur ein einfaches Werkzeug, sondern zusammen mit dem verbesserten Editor ein fast professionelles System zur Software-Entwicklung.
Der Einzelschrittmodus ist wesentlich verbessert worden, so daß nun auch alle zusätzlichen Register des 68020 und die Register des 68881 mit angezeigt werden.
Die beiden Trace-Modi des 68020 sind verfügbar und der DIS-Assembler enthält natürlich wie der Assembler alle Erweiterungen des Befehlssatzes.
Das Grundprogramm selber nutzt auch schon alle neuen Möglichkeiten aus. Zum Beispiel wird es immer mit Cache betrieben, wodurch der Assembler einen ca. 760 Kbyte Quelltext in etwa 40 Sekunden übersetzt.
Die Anzahl der über Trap verfügbaren Programme ist auf 145 gewachsen, wobei auch Programme dabei sind, die die FPU besser unterstützen. Eine eigene WERT-Routine für Floating-Point-Zahlen ermöglicht das Ausrechnen von Ausdrücken wie:

5*3+log(x)-sin(3)^5-tan(y)....

Floating-Point-Zahlen können durch mehrere Routinen in ASCII-Darstellung oder ASCII-Zahlen in FP-Darstellung gewandelt werden.
Die Bildschirmausgabe wurde verbessert, so daß die neue JADOS-Version 3.0 sowie CP/M 68K mit der GDPHS immer mit Hardscroll betrieben werden können, womit auch die GDP-Probleme in den Hintergrund treten.
Der Quelltext ist übrigens ungefähr 760 Kbyte lang und die 8 Eproms sind bis auf 1 Kbyte voll ausgenutzt. Auch das 68020-Grundprogramm wird es als Quelle geben, wobei allerdings zwei Disketten nötig sind.
Da die 68020-CPU-Karte sehr preiswert geworden ist und wirklich alle Möglichkeiten nutzbar sind, wird vermutlich der Kreis der "20iger" weiter wachsen.
Für die glücklichen Besitzer eines 68020 ist das Grundprogramm 6.1 ein Muß.

Für Sie gelesen:
Von "C" zu Modula-2 und zurück

Claude A. Wiatrowski, Richard S. Wiener:
Von "C" zu Modula-2 und zurück
McGraw-Hill Book Company, Hamburg,1988,
ISBN 3-89028-140-0, 378 Seiten,
DM 48,-

Mit dem neuen Buch setzt die McGraw-Hill Book Company die Reihe ihrer Bücher über die Programmiersprachen "C" und Modula-2 fort.
Zu Beginn gibt das Autorenteam Wiatrowski und Wiener dem Leser einen Überblick über die Programmentwicklung in "C" und Modula-2, indem geeignete Einsatzmöglichkeiten und deren physikalische Struktur dargestellt und die Softwareerstellung in den beiden Sprachen erläutert und verglichen werden.
Anschließend gehen die Verfasser dann über zu einer kurzen Zusammenfassung der Syntax und Semantik beider Sprachen.
Teil 3 des Leitfadens beinhaltet eine Reihe von Fallstudien, die den Vergleich von Problemlösungen in "C" und Modula-2 erlauben.
Um konkrete Parallelen ziehen zu können, wurden einige Beispiele in beiden Sprachen erstellt.
Die Scwerpunkte der Anwendungen liegen in der direkten Bildschirmausgabe, der Tastatureingabe und der Datenübertragung in den Speichern sowie der Datei- und Verzeichnisverwaltung und Roboterkontrolle.

Volker Wiegand

OS-9/68000

Übersicht über die Dienstprogramme

OS-9/68000 wird mit einem Satz leistungsfähiger Dienstprogramme zur Benutzer-, Speicher- und Diskettenverwaltung ausgeliefert. Diese Dienstprogramme sind eng an entsprechende Programme des UNIX Betriebssystems angelehnt und können sehr vielseitig genutzt werden.

Teil 1: Einführung in OS-9

Teil 2: Das Modulkonzept

Teil 3: Übersicht der Dienstprogramme

Teil 4: Wie schreibt man einen Treiber ?

Wir wollen uns in dieser Folge unserer Serie über OS-9 mit den standardmäßig vorhandenen Dienstprogrammen auseinandersetzen. Dazu geben wir einen kurzen alphabetischen Überblick, welche Programme mit OS-9/68000 ausgeliefert werden.

attr
Mit diesem Kommando können Attribute von Dateien geändert werden. Dabei handelt es sich hauptsächlich um Flags, welche zum Schutz von Dateien vor fremdem Zugriff dienen, aber auch die Information, ob eine Datei ausführbar ist oder nicht, kann hiermit festgestellt werden.

backup
Dieser Befehl dient der Herstellung von Sicherungskopien von Disketten. Allerdings ist dieses Programm nur bedingt brauchbar, weil Quell- und Zieldiskette dasselbe Format haben müssen, und die Kopie nur mit kompletten Disketten durchgeführt werden kann. Ein komfortables Utility für Datensicherung wird später mit dsave vorgestellt.

binex/exbin
Diese beiden Kommandos dienen der Umwandlung von binären Dateien in übertragbare ASCII-Dateien. Dabei wird das sogenannte S-Record Format von Motorola unterstützt. Mit diesen Kommandos können also jegliche Dateien über serielle Schnittstellen ausgetauscht werden.

break
Dieses Kommando dient der Unterbrechung sämtlicher laufender Prozesse und bringt den Benutzer zum Debugger im Eprom. Dadurch können alle Funktionen des Debuggers wie Breakpoints o.ä. benutzt werden. Während der Arbeit im Debugger ist jedoch das Timesharing abgeschaltet.

build
Build ist eine einfache Eingabemöglichkeit für ASCII Zeichen in eine Datei. Zeilen werden so lange in die Datei geschrieben, bis eine Leerzeile eingegeben wird.

cfp
Zur Programmierung von immer wiederkehrenden Eingaben, die sich auf verschiedene Dateien beziehen, bietet der "command file processor" die Möglichkeit, solche Eingaben auf beliebige Dateien anzuwenden.

chd/chx
Da OS-9 ein hierarchisches Dateisystem besitzt, braucht man Befehle, um sich innerhalb dieses Systems zu bewegen. Dazu dienen die in die Shell (Benutzeroberfläche) eingebauten Befehle "change data directory" und "change execution directory".

cmp
Zum Vergleichen von Programmen kann man das cmp-Utility benutzen, wobei mir persönlich die Tatsache gefällt, daß dieses Programm nicht wie von MS-DOS her gewohnt den Vergleich unterschiedlich großer Dateien ganz ablehnt und ansonsten nach 10 Unterschieden abbricht.

code
Zur Untersuchung, welche Codes von einem Terminal oder anderen Eingabegerät an den Rechner geschickt werden, hat man hiermit eine enorme Hilfe. Dieses Programm zeigt alle eingegebenen Zeichen als Hex-Wert an.

copy
Über dieses Programm viel zu sagen ist unnötig, denn ein Kopierprogramm hat wohl jedes Betriebssystem in irgendeiner Form. Natürlich können Platzhalter wie * und ? benutzt werden.

count
Dieses oft als Lehrbeispiel für C-Programmierung verwendete Utility zum Zählen von Zeichen, Wörtern und Zeilen darf natürlich auch unter OS-9 nicht fehlen.

date
Viele OS-9 Kommandos zeigen die augenblickliche Zeit irgendwo auf dem Bildschirm an. Mit date kann man Datum und Zeit explizit abfragen.

dcheck
Auch OS-9 hat Möglichkeiten, die Unversehrtheit von Disketten oder Platten festzustellen. Mit dcheck wird die Integrität geprüft, und darüber hinaus kann auch bedingt der Datenträger repariert werden.

iniz/deiniz
Es ist ein wesentlicher Vorzug von OS-9, daß Gerätetreiber während des Betriebes eingebunden und auch freigegeben werden können. Diese beiden Kommandos sind für die Verwaltung von Treibern verantwortlich.

del/deldir
Auch Kommandos zum Löschen dürfen in OS-9 nicht fehlen. Deldir ist insofern etwas besonderes, als damit ganze Unterverzeichnisbäume gelöscht werden können. Zur Sicherheit kann das Verzeichnis vor dem Löschen angezeigt werden.

dir/mdir
Einen Überblick über die Dateien auf einer Diskette oder Platte sowie über die Module im Speicher kann man sich jederzeit leicht verschaffen, wobei auch erweiterte Informationen verfügbar sind.

dsave
Wie oben schon angesprochen, gibt es ein komfortables Programm zur Sicherung

von Dateien auf Disketten. Hiermit ist es sogar möglich, Dateien auszuwählen, die seit der letzten Sicherung oder seit einem bestimmten Datum verändert wurden oder neu hinzukamen.

dump
Zur schnellen Anzeige einer unbekannten Datei kann dump verwendet werden, weil alle Zeichen als Hex-Werte dargestellt werden.

echo
Mit echo kann man Meldungen an andere Terminals oder in Dateien versenden. Dieses Programm ist auch nützlich bei der Programmierung von Stapeldateien.

edt
Ein wenn auch primitiver Zeileneditor gehört mit zum Lieferumfang des OS-9. Dieser Editor ist allerdings nur geeignet, kleinere Änderungen an überschaubaren Texten vorzunehmen.

fixmod
Wenn Module verändert wurden, so wissen wir, daß sie von OS-9 zurückgewiesen werden, weil ihre Prüfsumme nicht mehr stimmt. Mit fixmod kann man die Prüfsumme eines Moduls berichtigen. Auch die Anzeige, ob ein Modul intakt ist, ist verfügbar.

format
Zum Formatieren neuer Disketten und (hoffentlich einmalig) Festplatten kann man format verwenden. Dieses Programm erlaubt eine sehr flexible Handhabung verschiedener Formatierungsparameter.

free/mfree
Um sich einen schnellen Überblick über den verfügbaren Platz sowohl auf einem Massenspeicher als auch im Hauptspeicher zu verschaffen, kann man diese beiden Utilities benutzen.

help
Wer des Englischen mächtig ist, kann für die meisten Programme eine Hilfsfunktion über den Aufruf und die möglichen Parameter mit help <programmname> in Anspruch nehmen.

ident
Dieses Programm gibt ausführliche Informationen über andere OS-9 Module. Dazu gehören im Prinzip die Informationen, die wir in unserer letzten Folge besprochen haben.

kill
Wenn man, was ja unter OS-9 sehr viel Spaß macht, Prozesse kreiert hat, die man nun nicht mehr braucht und die im Hintergrund vor sich hin laufen, sollte man sich dieses Zauberwort merken, damit es einem nicht wie Goethe's Zauberlehrling geht...

link/load/unlink
Wie wir bereits aus der letzten Folge wissen, unterscheiden sich Module im Speicher nicht von denen auf Disketten. Um ein Modul, meist ein Programm, in den Hauptspeicher zu laden, benutzt man load. Ein Modulzähler wird dabei automatisch angelegt, und solange der nicht Null ist, bleibt das Modul im Speicher. Jeder Benutzer, der gleichzeitig mit dem Programm arbeitet, erhöht diesen Zähler. Dieser kann aber künstlich mit link und unlink beeinflußt und das Modul so wieder aus dem Speicher entfernt werden.

list
Zum Auflisten von ASCII-Dateien wird list benutzt. Falls man sich nicht sicher ist, ob eine Datei auch nicht druckbare Zeichen enthält, sollte man besser dump verwenden.

login/tsmon
Diese Programme sind das Herz des Mehrbenutzerbetriebes für OS-9. Dabei wird tsmon eingesetzt, um ein Terminal zu überwachen. Sobald auf diesem Terminal Aktivität festgestellt wird, wird login aktiviert, um den dortigen Benutzer höflich aber bestimmt nach seinem Namen und Kennwort zu fragen.

makdir
Genauso, wie man Unterverzeichnisse löschen kann, sollte man sie auch anlegen können.

merge
Wenn man mehrere Dateien miteinander verketten will, kann man mit merge beliebig viele Einzeldateien zu einer neuen Datei zusammenfügen.

os9gen
Um eine neue Diskette bootfähig zu machen, d.h., sie in die Lage zu versetzen, das Betriebssystem hochfahren zu können, muß diese mit dem Urlader versehen werden. Dieser Urlader ist eine Sammlung von Modulen, welche vor allem den Kernel sowie die anfänglich aktiven Treiber enthält. Die Vorbereitung einer Diskette übernimmt os9gen.

pd
Sollte man nicht mehr wissen, in welchem Inhaltsverzeichnis man sich gerade befindet, kann man mit pd seinem Gedächtnis auf die Sprünge helfen.

pr
Auch ein leistungsfähiges Druckprogramm mit gewissen Umbruchfähigkeiten fehlt in OS-9 nicht. Dabei kann von einer Zeilenverschiebung oder -begrenzung bis hin zu Kopf- und Fußzeilen sehr viel eingestellt werden.

printenv/setenv/unsetenv
OS-9 ist ein sehr modernes Betriebssystem. Moderne Betriebssysteme haben selbstverständlich die Möglichkeit, Umgebungsvariablen zu definieren und zu bearbeiten. OS-9 macht hier keine Ausnahme. Diese Umgebungsvariablen werden jedem Programm zur Verfügung gestellt, wobei einige auch ohne Zutun des Benutzers vordefiniert sind.

procs
Eine Anzeige der laufenden Prozesse mit ausführlichen Informationen über deren Zustand, Standardpfade, Speicherbelegung sowie andere Parameter ist mit Hilfe von procs jederzeit möglich.

qsort
Ein Sortierprogramm nach dem Quicksort Algorithmus ist mit qsort verfügbar. Dieses Programm kann auch für eigene Anwendungen benutzt werden. Programme wie dir oder mdir haben jedoch diese Sortierung bereits eingebaut.

rename
Dieses Kommando gehört in die Kategorie der Datei-Utilities. Mit ihm ist es möglich, Dateien umzubenennen. OS-9 ist in der Wahl des Namens sehr freizügig. Es können zwar nur Buchstaben, Zahlen, die Unterstreichung, der Dollar sowie der Punkt im Namen verwendet werden, dafür ist jedoch die Länge eines Namens auf 29 Zeichen begrenzt und es dürfen beliebig viele Punkte darin vorkommen.

save
Module, die im Arbeitsspeicher liegen, können mit save als Umkehrung von load

auf den Massenspeicher ausgelagert oder gesichert werden.

setime
Da das System eine Echtzeituhr enthält, die unter Umständen auch batteriegepuffert ist, ist es nötig, selbige auch setzen zu können. Ohne Batteriepufferung wird dadurch nach dem Einschalten die Zeit gesetzt, mit Batteriepufferung wird die gepufferte Uhr ausgelesen oder auch neu gesetzt.

setpr
Wir hatten uns bereits über Prioritäten von Prozessen unterhalten. Mit diesem Kommando ist es nun möglich, diese Prioritäten nach Belieben zu verändern, natürlich nur, wenn man selbst der Supervisor ist, also der, der die Verfügungsgewalt über das Gesamtsystem hat.

shell
Die Shell, also der Befehlsinterpreter, verdient eigentlich einen eigenen Artikel innerhalb unserer Serie. Wir wollen hier nur soweit auf sie eingehen, wie es in einer Besprechung der Dienstprogramme sinnvoll erscheint. Neben der Aktivierung von Prozessen ist es hiermit nämlich möglich, diesen Prozessen Eigenschaften wie Priorität, zusätzlichen Speicher oder Ein- und Ausgangspfade beliebig zuzuweisen. Auch sind einige Befehle wie chd oder kill keine eigenen Kommandos, sondern sind einfach eingebaute Shell-Befehle.

sleep
Es ist möglich, die weitere Ausführung eines Prozesses oder vor allem einer Stapeldatei für eine gewisse Zeit auszusetzen. Dabei kann die Wartezeit in Sekunden oder "Ticks", also meist 1/20 Sekunden frei gewählt werden.

tee
Mit diesem Utility ist es möglich, die Ausgabe eines Prozesses auf mehrere andere Prozesse umzulenken. Es ist also z.B. leicht möglich, einen Text anzuzeigen und ihn gleichzeitig auszudrucken.

tmode/xmode
Bei der Arbeit mit verschiedenen Ein- und Ausgabegeräten ist es für den Computer wichtig, auf die unterschiedlichen Bedürfnisse dieser Geräte einzugehen. Dazu dienen die Programme tmode und xmode. Tmode stellt die Parameter für das eigene Terminal ein (auch die GDP und KEY sind für OS-9 ein Terminal), während xmode das für andere Kanäle erledigt. Dabei können Dinge wie Übertragungsrate oder Parität, aber auch das Zeichen für Programmabbruch oder Bildschirmpause eingestellt werden. Ob ein Drucker etwa Auto-Linefeed unterstützt oder nicht, kann ebensogut an dieser Stelle ausgeglichen werden.

touch
Jedes Programm ist mit einem Stempel versehen, der das Datum der letzten Veränderung anzeigt. Dieser Stempel kann mit touch auf den augenblicklichen Zeitpunkt gesetzt werden. Dies wird vor allen bei der Herstellung größerer Programmpakete verwendet.

Damit wollen wir zunächst für heute schließen. Wie man leicht sehen kann, sind Werkzeuge für die verschiedensten Zwecke vorhanden. Es ist natürlich auch aufgefallen, daß andere elementare Programme fehlen. Auch OS-9 gibt sich hier nicht anders als etwa Betriebssysteme wie MS-DOS, die nur mit den notwendigsten Programmen für die Verwaltung der Systemressourcen ausgestattet werden. Hier wollen wir in Zukunft jedoch einen etwas anderen Weg einschlagen.
Auf unserer letzten Softwarepartner-Tagung in Kempten im Allgäu haben wir nämlich die Weichen für eine Entwicklung gestellt, die es ermöglichen soll, für den NDR-Computer mit OS-9 eine ganze Reihe weiterer Programme verfügbar zu machen. Leider ist das aufgrund der Preispolitik des amerikanischen Herstellers Microware kaum mit original OS-9 Produkten zu realisieren. Außerdem gibt es dort auch nur einige wenige Sprachcompiler zu kaufen. Nein - wir wollen den gesamten Pool der verfügbaren NDR-Programme auch unter OS-9 benutzen. Um den Einstieg zu erleichtern, wird zum OS-9 in der Grundversion bereits ein Assembler mitgeliefert, der den von Microware weit an Geschwindigkeit übertrifft und einen Editor eingebaut hat, der schon bei der Eingabe die Syntax von Zeilen überprüft. Außerdem wird der Linker, also das Programm, das mehrere Assembler Dateien verbindet, schon im Assembler integriert sein und trotz wesentlich höherer Geschwindigkeit den vollen Umfang des von Microware gelieferten erreichen. Weiterhin gehört zur Grundausstattung ein Pascal-Compiler, der den vollen Sprachumfang von Jensen und Wirth unterstützt. Ein Hochleistungs - C-Compiler rundet schließlich das Paket ab, und da dieser ein Public Domain Programm aus der UNIX Welt ist, bekommt man den Quellcode noch dazu, und es kostet nichts als die reinen Diskettenkosten...

Für Sie gelesen:
Einführung und Anwendung von LISP

Michael Hußmann, Peter Schefe, Andreas Fittschen:
Das LISP-Buch
McGraw-Hill Book Company, Hamburg, 1988
ISBN 3-89028-098-6, 249 Seiten, DM 45,-

Konzeptionell einheitlich und klar wie kaum eine andere, erfreut sich die Programmiersprache LISP zunehmender Beliebtheit. Als eine der ältesten höheren Programmiersprachen ist LISP vor allem im Bereich der künstlichen Intelligenz von Bedeutung. Mit dem soeben erschienenen "LISP-Buch" trägt das Autorenteam Hußmann, Schefe und Fittschen dieser Entwicklung durch einschlägige Anwendungsbeispiele Rechnung. In erster Linie wollen die Verfasser den Leser aber in die grundlegenden Konzepte der Sprache einführen.
Um ein wirkliches Verständnis zu erreichen, wird das Konzept der LISP-Programmierung aus sich heraus und ohne oberflächliche Vergleiche mit anderen Sprachen erklärt.
Das Konzept der Funktionsobjekte nimmt in der vorliegenden Einführung eine zentrale Position ein, obwohl es sich erst spät herausgebildet hat und in vielen Dialekten immer noch nicht realisiert ist.
Neben den grundlegenden Aspekten bietet das "LISP-Buch" ein breites Spektrum an Programmiertechniken: Makroprogrammierung sowie datengesteuerte und objekt-orientierte Techniken. Bei allen Beispielen haben die Autoren den Dialekt Nils'LISP zugrunde gelegt. Zusätzlich findet der Anwender im Anhang weitere wichtige Informationen über verfügbare LISP-Dialekte auf Personal Computern.

F.J. Ehrensperger

Zweimal Laufwerk B?

Immer wieder besteht der Wunsch oder auch die Notwendigkeit, an einen IBM- oder kompatiblen Rechner mehrere unterschiedliche Laufwerke anzuschließen.

Beim MC-MODULAR-AT sind zwar in Verbindung mit dem UDC+ drei unterschiedliche Diskettenlaufwerke ansteuerbar, will man jetzt noch einen Streamer (IRWIN) an den Controller anschließen, so stößt man auch hier an die Grenzen des Controllers, zumal das externe Laufwerk nicht als HD-Lauf werk (1,2MB und 1,44MB) zu betreiben ist. Schade! Wie man nun mit einem kleinen Trick trotzdem seinen Streamer und zusätzlich die entsprechnenden HD-Diskettenlaufwerke betreiben kann, sollen die folgenden Zeilen und Skizzen beschreiben.

Ein MS-DOS-Rechner mit UDC+ könnte zum Beispiel folgende Diskettenlaufwerke ansteuern: Laufwerk A: 1,2MB 51/4", Laufwerk B: 1,44MB 31/2" (damit können auch 720KB Disketten bearbeitet werden) und als externes Laufwerk 360KB 51/4" (damit auch (alte) XT's die Disketten problemlos lesen können). Wollte man nun einen Streamer einbauen, so müßte man entweder auf ein Diskettenlaufwerk verzichten, oder einen Streamer verwenden, der mit einem speziellen Controller arbeitet. Ein Streamer mit Controller scheidet oft aus, weil er sowohl einen Bussteckplatz als auch einen Interrupt belegt und zudem noch Strom verbraucht auch in der Zeit in der er nicht gebraucht wird.

Ein Streamer, der mit dem Disketten-Controller betrieben wird, verhindert den Gebrauch eines Diskettenlaufwerkes (Laufwerk B), nun ist guter Rat teuer, oder?

Überlegen wir uns, wann der Streamer gebraucht wird. Ein Bandlaufwerk benutzt man in der Regel nur zur schnellen Datensicherung, um Zeit und Disketten zu sparen, also am Ende der Arbeit. Die Streamer-Software greift direkt auf den Disketten-Controller zu und erkennt, ob ein Streamer angeschlossen ist, ohne sich für den Eintrag im SETUP des Rechners zu interessieren.

Das Problem wäre gelöst, wenn Disketten- und Bandlaufwerk durch einen Schalter mit dem Controller verbunden werden könnten. Also bauen wir uns einen Umschalter ein, aber 34 Anschlüsse löten, Masse muß man nicht schalten, sind immer noch 17 Pin's, auf jeden Fall zu viele und zudem sind die Verbindungen auch noch mit Flachbandkabel zu machen. Das scheint fast nicht möglich zu sein, oder geht's vielleicht einfacher? Sehen wir uns einmal die Schaltung an. Alle Signale gehen parallel an alle Laufwerke, also auch an den Streamer. Die Laufwerke werden durch den Drive-Select aktiviert. Also machen wir das Select-Signal umschaltbar, einmal können wir somit den Select an das Laufwerk und zur Datensicherung an den Streamer legen. Aber einen Schalter in ein Flachbandkabel einzubauen ist, auch kein Vergnügen, also suchen wir nach einer besseren Lösung. Der entsprechende Drive-Selcet wird auf dem Laufwerk ausgewählt und zur Laufwerkssteuerung weitergeleitet. Benutzt man diese Brücken zur Umschaltung, dann geht das Ganze vielleicht noch einfacher.

Nachdem nun die theoretischen Überlegungen abgeschlossen sind, wollen wir uns an die praktische Arbeit machen. Zuerst muß ein Schalter montiert werden. Es reicht ein einpoliger Umschalter, entweder ein Kipp- oder ein Schiebeschalter. Für den Schalter findet sich im Streamer noch ausreichend Platz. Dazu muß die Frontblende abgenommen werden. Um die Leitungen kurz zu halten, wird der Schalter auf die Seite montiert, auf der sich auch die Steckbrücke befindet. Benutzt man einen Kippschalter, so genügt ein entsprechendes Loch in der Frontplatte. Für einen Schiebeschalter muß ein passender Ausschnitt in die Frontplatte geschnitten werden. Das ist leider etwas mehr Arbeit. Ich habe mich für den Kippschalter entschieden. Bevor der Schalter in der Frontplatte montiert wird, lötet man bereits die drei Drähte an, das erleichert uns später die Arbeit. Ist nun der Schalter endlich in der Frontplatte, so kann diese wieder montiert werden. Jetzt müssen die Laufwerke im Gehäuse angeordnet werden. Das kann aber zu Problemen führen, da nicht alle Gehäuse genug Platz für die Laufwerke bieten. Wer über die entsprechenden Möglichkeiten verfügt, kann eventuell ein kleines Gehäuse modifizieren. Ansonsten ist für mehrere Laufwerke unbedingt ein Towergehäuse zu empfehlen. Hat man für

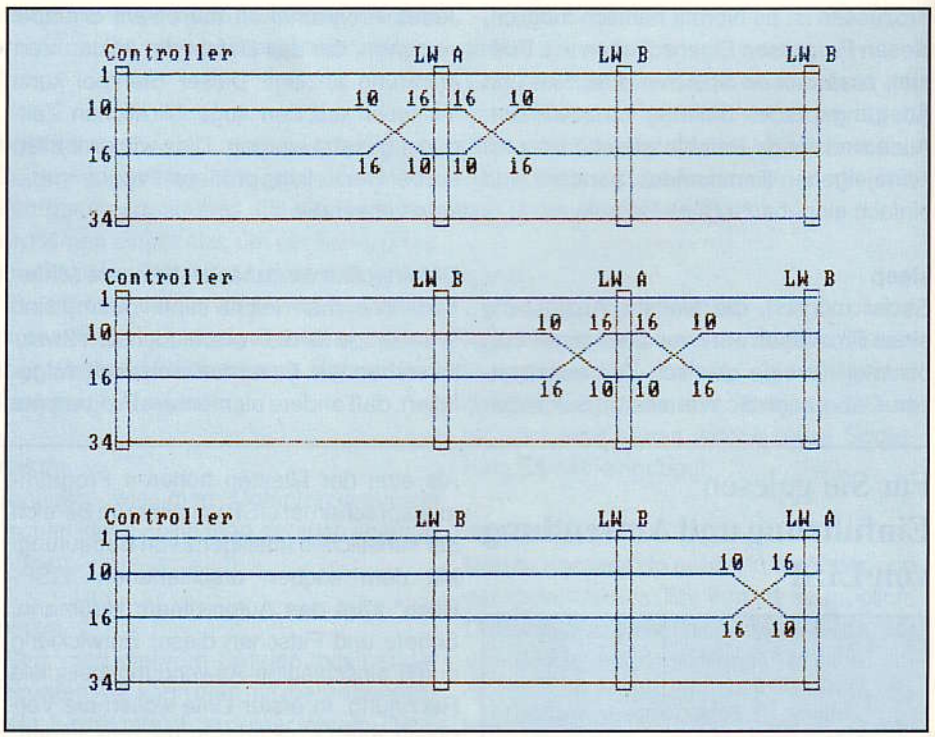


Bild 1: Verschiedene Flachbandkabel

seine Laufwerke einen passenden Platz gefunden oder geschaffen, so benötigt man nun eine entsprechende Kabelverbindung. Leider passen für eine solche Konfiguration die normalen Kabelverbindungen nicht. Wir werden also diese Verbindung selbst fertigen müssen.
Dazu benötigt werden: 34-poliges Flachbandkabel mit entsprechender Länge (lieber etwas länger machen),
2 34-pol. Pfostenverbinder und
2 34-pol. Platinenstecker.
Jetzt kommt eine nicht ganz einfache Prozedur: das Anpressen der Stecker an das Kabel.

Bei mir hat sich die Schraubstock-Methode am besten bewährt. Das Kabel wird in die Führung des Steckers gelegt, dann hält man Kabel und Stecker fest und bringt den Stecker zwischen die Spannbacken des Schraubstockes. Nun wird die Spindel etwas zugedreht, bis der Stecker nicht mehr herausrutscht. Bevor endgültig der Stecker angepreßt wird, kontrolliert man, ob Ober- und Unterteil des Steckers gerade sitzen und das Kabel auch gerade im Stecker ist. Jetzt kann die Spindel zugedreht werden, bis Ober- und Unterteil des Steckers einrasten. Zum Schluß kontrolliert man nochmals den Stecker, um sicher zu sein, daß kein Anschluß des Steckers verbogen ist.

Auf diese (oder eine andere) Methode wird als erstes der Pfostenverbinder für die Controller-Seite angepreßt. Die Kennzeichnung des Kabels sollte auf Pin 1 zu liegen kommen. Dies erleichtert uns, die Laufwerke richtig miteinander zu verbinden. Nun wird das Kabel mit dem Controller verbunden und die Stellen gekennzeichnet, an die anschließend die Stecker für die Lauf werke angepreßt werden. Es ist immer darauf zu achten, daß die Kennzeichnung auf Pin 1 zu liegen kommt. Dabei muß auch berücksichtigt werden, daß man das Kabel eventuell drehen muß. Die Abstände der Stecker sollten nicht zu knapp gewählt werden, so hat man etwas Spielraum beim Anpressen. Bei der Anfertigung des Anschlußkabels muß beachtet werden, daß zur Ansteuerung von Laufwerk A die Anschlüße Nr. 10 bis Nr. 16 gedreht werden müssen.
In der Regel ist der Stecker für Laufwerk A am Ende des Kabels. Oft würde aber die Kabelführung erleichtert, wenn der Stekker an einer anderen Stelle angebracht wäre. Die Anschlüße des Kabels werden vor dem Stecker gedreht. Dreht man nun die Anschlüße hinter dem Stecker nochmals, so kommen sie wieder in die ursprüngliche Lage zurück und es kann hinter diesem Stecker Laufwerk B angeschlossen werden. Wie nun ein Kabel für unsere Konfiguration aussehen kann, soll Bild 1 zeigen.

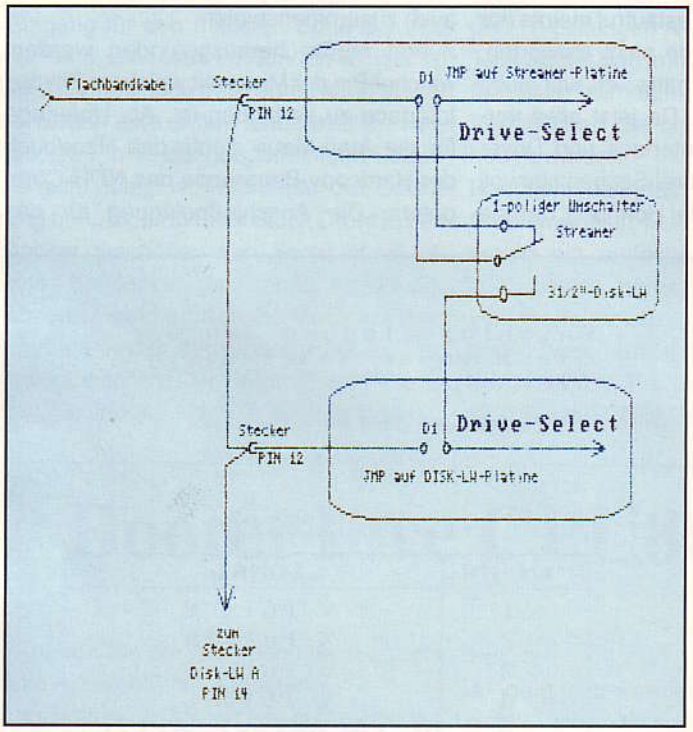


Bild 2: Drive-Verbindungen

Nachdem wir nun ein entsprechendes Kabel nach einer der obigen Zeichnungen ausgesucht und angefertigt haben, müssen wir noch den Umschalter anschließen. Dazu verbinden wir nun Streamer und 31/2"-Laufwerk mit dem Datenkabel. Nun entfernt man die Jumper für den Drive-Select auf beiden Geräten. Den Jumper vom Diskettenlaufwerk gut aufbewahren, denn er wird noch gebraucht.
Um nun festzustellen, an welchem Pin das Selectsignal vom Controller anliegt, prüft man mit dem Ohmmeter, welcher PIN des Streamers mit welchem PIN des Disketten-Laufwerkes über das Kabel verbunden ist. Da nur die beiden PIN's in Frage kommen, auf denen die Jumper waren, ist dies schnell herausgefunden. Haben wir das Select-Signal ausfindig gemacht, so wird dieser PIN des Streamers mit dem mittleren Anschluß des Schalters verbunden; auf dem Diskettenlaufwerk bleibt der entsprechende PIN offen. Die PIN's, die zum Select des Laufwerkes führen, werden mit den äußeren Anschlüssen des Schalters verbunden. Als Verbindung kann auf die Stiftreihe des Streamers ein 2-poliger Stecker mit 2,54 mm Raster verwendet werden. Ein entsprechender Stecker ist für ein 31/2"-Laufwerk leider nicht zu finden gewesen. Man kann dazu den 2-poligen Jumper mißbrauchen. Es wird der Draht an den Jumper gelötet und der Jumper so auf die Stiftreihe gesteckt, daß nur ein Pin mit der Steckbrücke verbunden wird. Wie nun die Verbindung aussehen soll, zeigt Bild 2.

Hat man diese Arbeiten hinter sich, so kann man endlich alle Laufwerke in den Rechner einbauen. Nach dem Einbau sollten nochmals alle Verbindungen überprüft werden, dann kann ein Versuch gestartet werden. Der Rechner wird eingeschaltet, nach dem Booten wird in Laufwerk B eine Diskette eingelegt und mit DIR B: das Inhaltsverzeichnis ausgegeben. Es sollte nun die LED des Laufwerks aufleuchten und das Inhaltsverzeichnis angezeigt werden.
Hat dieses funktioniert, so kann der Schalter umgelegt und das Sicherungsprogramm aufgerufen werden, um den Streamer zu testen. Läuft das auch richtig, so kennzeichnet man die Schalterstellung und ist damit fertig.

Natürlich muß nicht alles auf Anhieb funktionieren; man kann sich hierbei auch Fehler einhandeln, vor allem, wenn unsauber gearbeitet wird.
Hier ein paar mögliche Fehlerquellen: Kein Laufwerk funktioniert mehr: Datenkabel nicht oder nicht richtig am Controller angeschlossen, Fehler im Kabel selbst oder Stromversorgungsstecker an den Laufwerken vergessen. Zwei Laufwerke werden gleichzeitig selektiert: Auf einem oder mehreren Laufwerken falschen Drive-Select verwendet, Kurzschluß im Datenkabel, im Datenkabel die falschen PIN's gedreht oder Fehler bei der Verdrahtung des Schalters. Alle Laufwerke werden immer

selektiert. Auf einem oder mehreren Laufwerken den Datenstecker falsch eingesteckt oder Kurzschluß im Datenkabel.
Die Fehlerliste ließe sich wahrscheinlich über mehrere Seiten fortsetzen, wenn man jedoch vorsichtig und überlegt arbeitet, wird man die meisten Fehlerquellen von vornherein ausschließen können. Falls sich wirklich ein Fehler einschleicht, so kann er meist durch gezielte Beobachtung und Tests leicht ausgemacht werden. Auf diese Art lassen sich natürlich auch noch andere Laufwerke an den Rechner anschließen, sofern diese nicht gleichzeitig gebraucht werden. Eventuell muß bei unterschiedlichen Diskettenlaufwerken der SETUP des Rechners geändert oder der entsprechende Driver geladen werden.

F.J. Ehrensperger

Ist die Atarimaus IBM-Kompatibel?

Vor kurzem habe ich mir einen MC-Modular-AT zugelegt. Beim genauen Durchlesen der Anleitung zu meiner Bildschirm- und Druckerkarte habe ich festgestellt, daß diese auch noch ein BUS-MOUSE-Interface enthält. Es handelt sich um die JUKO-MiniG7-Karte. Da auch ein NDR-Computer mit Atarimaus bei mir seinen Dienst tut, kam ich auf die Idee, diese Maus auch am AT zu verwenden. Wie man dabei vorgeht, sollen die folgenden Zeilen und Bilder beschreiben.

Zuerst braucht man einen Driver, wenn ein AT mit Maus ausgerüstet werden soll. Nach längerem Suchen wurden auch zwei Disketten mit der Beschriftung "MOUSE DRIVER" ausgegraben. Diese stammten noch von einer Maus, welche wegen eines erheblichen Defektes in den Mülleimer wanderte. Bei einem Testaufruf meinte der Driver, die Maus würde nicht antworten, womit er noch recht hatte, es war auch keine angeschlossen. Da jetzt alles vorhanden war, Maus, Interface und Driver mußten nur noch die drei Sachen sinnvoll kombiniert werden in der Hoffnung, daß sie auch zusammenspielen.
Zuerst mußte herausgefunden werden, welcher Pin der Maus mit welchem Pin des Interface zu verbinden ist. Als Unterlage für die Atari-Maus diente das Handbuch der Hardcopy-Baugruppe des NDR-Computers. Die Anschlußbelegung für das

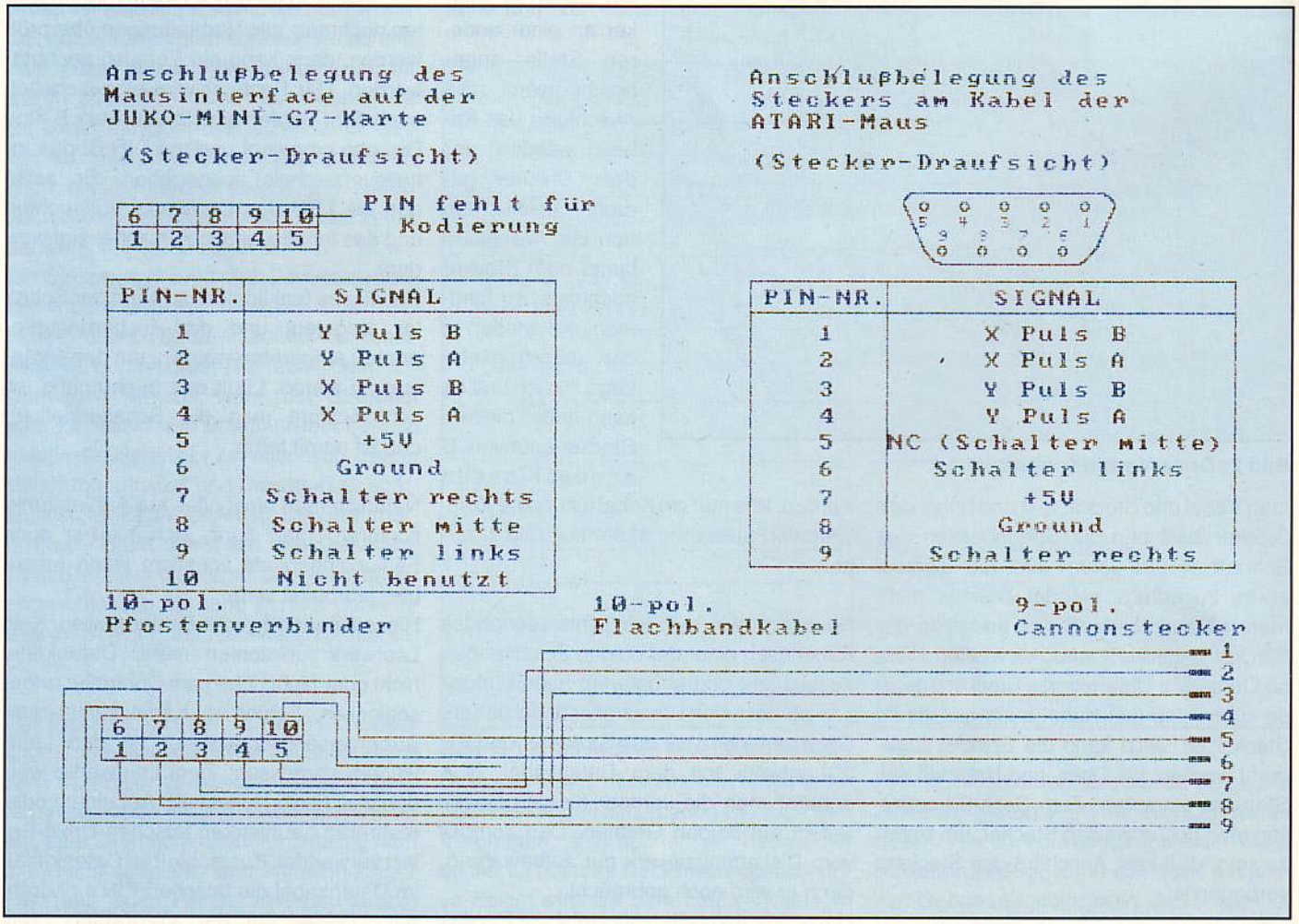


Bild 1: Anschlußbelegungen von Maus und Interface

BUS-MAUS-Interface war im Handbuch der Graphik-Karte zu finden. Nun stellte sich aber heraus, daß die Verbindung von Maus und Interface doch nicht so einfach war. Die Maus besitzt eine 9-polige Canonbuchse, das Interface eine 2*5-polige Stiftreihe. Einen passenden Canonstecker fand ich in meiner Bastelkiste, aber ein 10-poliger Pfostenverbinder war nicht zu finden. Da ich die Karte nicht beschädigen wollte, wurden alle weiteren Aktionen auf den nächsten Tag verschoben.

Am anderen Tag, bewaffnet mit zwei Pfostenverbindern (falls einer das Anpressen nicht überleben sollte), ging's neu an's Werk. Einer der Pfostenverbinder wurde an ein Stück Flachbandkabel angepreßt, das andere Ende des Kabels entsprechend der gefundenen Belegung an den Canonstecker gelötet. PIN 8 vom Interface (Eingang für den mittleren Schalter) wird mit Pin 5 der Maus verbunden (dieser ist bei der Maus noch frei). Somit kann später bei Bedarf eine dritte Taste installiert werden. Da Pin 10 der doppelreihigen Stiftleiste nicht belegt ist, wird dieser zur Codierung benutzt, indem ein Stück Draht in den Stecker geschoben wird. Somit ist ein falsches Einstecken des Kabels nicht möglich. Jetzt kann endlich die Maus mit dem Interface verbunden und ein neuer Test gewagt werden. Der Rechner wird eingeschaltet und der Maus-Treiber geladen.

Natürlich war erst mal meine Enttäuschung groß, als wieder die Maus nicht erkannt wurde. Sollte ich was falsch gemacht haben, oder funktioniert's doch nicht so? Also nochmals Anleitung studieren. Dabei fällt ein Schalter auf, der das Maus-Interface abschaltet. Also Schalter ein, Steckbrükken auch überprüfen und dann ein neuer Versuch.

Die Freude war groß als nach der Copyright-Meldung die Meldung "Mouse Driver installed" erschien. Ein Maus-testprogramm wurde geladen und anschließend die Mausfunktion überprüft. Da kam auch schon der nächste Hammer, die Tasten funktionierten, aber der Cursor bewegte sich in ganz andere Richtungen als die Maus bewegt wurde. Statt nach oben bewegte sich der Cursor nach links, statt rechts nach unten. Also wurden erst die X- und Y-Leitungen vertauscht. Ein neuer Test zeigte, daß immer noch oben und unten nicht stimmten. Dies konnte durch Vertauschen der beiden Y-Leitungen behoben werden.

Nachdem nun das Testprogramm ohne Probleme allen Mausbewegungen folgte, wurde ein einfaches Zeichenprogramm zum Testen benutzt. Auch dieser Test verlief erfolgreich.

In der Zwischenzeit hatte ich Gelegenheit, noch mehrere Programme mit dem BUS-MOUSE-INTERFACE zu testen. Alle arbeiteten bis jetzt zur vollen Zufriedenheit. Da ich natürlich nicht alle Programme testen konnte, könnte es vielleicht trotzdem Probleme geben. Vor allem bei Programmen, die die Maus direkt abfragen oder die Maus auf COM 1 erwarten.

Noch ein Hinweis: Wenn COM1 installiert ist, darf Interrupt 4 von der Maus nicht benutzt werden, ist COM2 installiert, so darf die Maus Interrupt 3 nicht benutzen. Ebenso müssen eventuell andere Interrupts überprüft werden, bevor man sie mit der BUS-MOUSE benutzt. Welche Interrupts einstellbar sind, ist dem Handbuch der Karte zu entnehmen. Die Standardeinstellung ist Interrupt 3, es kann also COM2 nicht benutzt werden. Die folgenden Zeichnungen sollen helfen, die Verbindungen leichter herzustellen.

Die Anschußbelegungen der Maus sind so bezeichnet, daß sie mit den Bezeichnungen des Interface übereinstimmen. Ob dies richtig ist, habe ich bis heute noch nicht herausgefunden.

Den Stecker des Verbindungskabels befestigt man am besten in einer entsprechenden Aussparung des Gehäuses. Ist eine solche nicht vorhanden, so kann man eine Aussparung in einen leeren Haltewinkel einbauen. Dies erfordert allerdings etwas handwerkliches Geschick und entsprechendes Werkzeug.

Bootet Ihre CPU8088 nicht, oder.....

-1- haben Sie die SER noch nicht auf die neue Adresse 0f8h umgestellt?
Zum Testen, ob das System bootet, ziehen Sie die SER einfach mal raus

-2- hängt sich die neue CPU nach dem Reset auf, so löten Sie einen 100nF Kondensator über die Stromversorgung des IC12. Dieses IC ist für den Reset -und die Taktlogik zuständig (Störspitzen an GND). Hilft dies immer noch nichts, ändern Sie den C2 von 10uF in 4,7uF.

-3- lesen Sie bis zur letzten Seite des Handbuches, Änderung auf der FLO2, FLO3, SER

-4- IBM-DOS bootet nicht! - Startfehler -

-5- Jumperstellung des JMP1 ist für das Eprom 2764, aber auch für das von uns jetzt eingesetzte Eprom 27128 (16k)! Also setzen Sie bitte den JMP1 wie im Handbuch :: = Standardeinstellung beschrieben.

-6- geht das Formatieren von Disketten nicht? - wie im Handbuch S.96/3 beschrieben - so bieten wir im Übergang (im neuen BIOS solls ja geändert werden) einen kleinen Trick an.

Stecken Sie nochmals Ihr altes System zusammen und formatieren mit dem NDR-Computer - egal ob mit Z80 oder mit 68xxx System - die Disketten mal vor.

Im Z80 System mit uform, com eingeben:

Minilaufwerke 51/4"	=1
40 Spuren	=2
Doppelseitig und SSO	=3
Doppelte Schreibdichte	=2
512k (9Track)	=2
jetzt FLO1, 2/3	=
LW	=

Jetzt muß im Menü stehen:

Anzahl der Spuren	=40
Anzahl der Sektoren	=9
Bytes/pro Sekt	=512

beide Seiten werden formatiert
Seite 1 mit SSO, Seite 0 normal
Speicherkapazität: 360k Bytes

Achtung: Diskette wird formatiert
Start = J

Wenn Sie so die Diskette auf dem NDR-Z80-System formatiert haben, und jetzt im neuen System (MS-DOS) den Befehl - format b: aufrufen, klappt es. (beachten Sie aber die Änderung der Jumper auf der FLO)

Kleine Einschränkung: Es werden nur 36O K-Disketten formatiert.

Aber: Das "richtige" Formatierprogramm kommt bald!

Kleinanzeigen

VERKAUFE: NKC im PC-Gehäuse, 2*5 l/4" Floppy, CPU 68008 GDP 64K, 2*DRAM 256, 3* ROA 64, ROA 256, HC/Maus, Atari-Maus, Prommer, ELTEC-Tast, Monitor, org. CPM68k u. CPM 2.2, Wordstar, dBase u.v.m., nur DM l.95O,—
Harald Krause, Streitfeldstraße 48, 8OOO München 8O,Tel. O89/43llOO7 oder O8O3l/l7l98

Verkaufe wegen Systemwechsel: getestet: CPU68OOO l2Mhz DM lOO,— 2x ROA64 mit EPascal, EGrund, Handbuch, 64k RAM DM lOO,— desweiteren: Monitor Zenith grün DM 8O,—, Rekorder Waltham MK3OOO für CAS DM 5O,—; evtl. mit Schäden zu DM 2O,—; CAS,KEY,IOE,SBC, Tastatur. Literatur (Franzis,mc,LOOP).
Enno Arenholz, Dolomitenstr.l8 a, 84OO Regensburg,Tel. O94l/63678, l8-2O.OO Uhr

---

Verkaufe: NDR-Klein Computer kompl. DM 48O,—
Albert Fischer, Weiherstr.7, 8722 Rötschenhausen,Tel. O9726/l2O8

Verkaufe: 2x ROA64 à DM lOO,— 2x Dyn. RAM256 à DM l5O,— (alle voll bestückt), SBC 2 mit Grundprogramm und EBASIC DM 3O,—, CAS DM 3O,—, IOE DM 25,— Dyn RAM's 4l64 à DM l,2O
Hans Spahr, Pommernstraße 8, 236O Bad Segeberg,Tel. O455l/83268

Verkaufe: NDR-Z8O Kpl. oder in Teilen und 2 Laufw., Geh., Bankb., Flo 2, Zimmermann, abends Tel. O89/27ll5ll

---

Verkaufe für NDR-Computer 25 Karten u. viel Zubehör. Einzeln oder kompl. sehr günstig abzugeben.
Horst Neuwirth, Fronackerstr. 75, 7O5O Waiblingen,Tel. O7l5l/5322O

Verkaufe 2x Bus, SBC2, CAS, GOP64K BankBoot, RAM 64/256, CPUZ8O KEY Cherry, Z8O Vollausbau bis Zeat vorhanden. Christiani Kurs Z8O Assem.Programmierung
R.Gaißer, Hegelstr., 74OO Tübingen,Tel. O7O7l/74l25

Verkaufe zum halben Preis Christiani-Kurs Digital-Computer-Labor mit Software LogSim (für IBM komp.) sowie SPS-Steuerung Band l-4.
Ulrich Höfer, Peter-Rosegger-Str.3, 8269 Burgkirchen, Tel. O8679/3l83

Verkaufe wegen Nachwuchs: SBC3 8 MHz DM l5O,— DRAM l28 K; KEY DM 5O,—; GDP DM l5O,—; Tastatur DM l5O,—; BUS4 DM lOO,—; Nicht ausgetestet: FDC DM l5O,—; SER DM 5O,—; CAS DM lO,—. Literatur: Loop 2-l9; Rechner modular DM 4O,—; mc Schritt für Schritt DM l5,—;
Marco Mohrmann, Liefenroth 38, 527O Gummersabach,Tel. O226l/72378

Verkaufe NDR-Computer 68OO8, 384 kByte RAM, (256dyn RAM, 3 ROA64), KEY, CAS, GDP64K, FDC, SER, IOE, IOE-CENT, Bankboot, Promer + POW26/22, l Floppy 3,5" TEAC, im GEH1-19", gr. Tast, Monitor 12" bernstein, Software: Grundprogramm, Pascal, CP/M-68K, Z8O -Emulator, Wordst. DM l.4OO,—
Harald Schallnus, Altmühlweg 4O, 2OOO Hamburg 65,Tel. O4O/64O6839 (abends).

Suche Commodore-Pet mit Gummi-Tastatur
Tel.: 08374/9552

Korrekturen / Ergänzungen

von Jost-Reimer Hoof

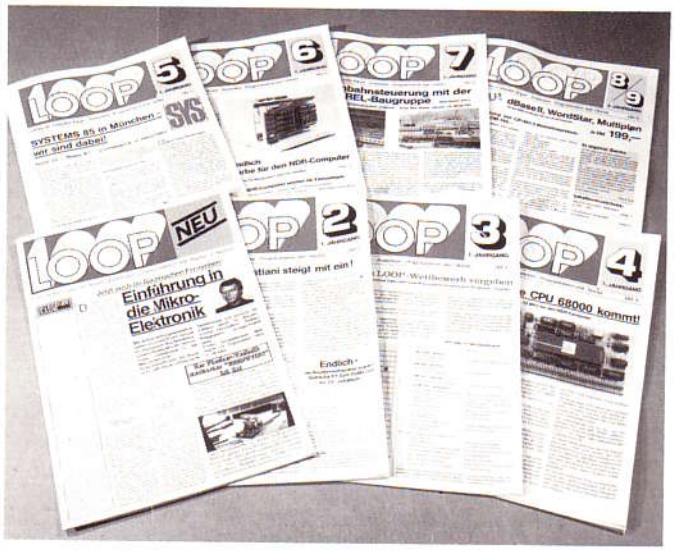
Beim Programm TAST-KOD.COM, das ich in der LOOP Nr. 20 vorstellte, waren einige Änderungen notwendig, damit es problemlos auch unter FLOMONCG arbeiten kann. Als wichtigste Änderung ist zu nennen, daß ein Teil als relokatives Programm geschrieben ist und - den Vorschlag von Rüdiger Nahm aufgreifend - den Belegungszeiger FREEMEM des Monitors auswertet und aktualisiert.
Weiterhin mußte der Sprung aus dem BIOS, in der LOOP Nr. 20 noch mit EA81h nach F003h angegeben geändert werden, da dieser Absprung in jedem BIOS unterschiedlich sein kann. Da ist die neue Absprungadresse F003h eindeutig.
Das Programm baut die bei F003h vorgefundene Sprungadresse bei sich ein, um etwaige andere Tastaturumlenkungen nicht zu behindern.
Mit dem relokativen Programmteil war auch die Deaktivierungsroutine neu zu durchdenken.
Deshalb kann es vorkommen, daß das Programm den über dem Monitor notwendigen Programmteil für die Deaktivierung nicht findet. Dann gilt die Aktivierung bis zum nächsten Reset.
Die Zeichenlänge pro Taste wurde auf 14 Zeichen verringert, um mit dem RAM-Bereich über dem Monitor ab etwa F770h sparsam umzugehen.
Es ist aber nicht schwer, im Quellprogramm die Stringlänge beliebig aufzubohren.
Diese Änderungen wurden vor Auslieferung des ersten Programmes durchgeführt.
Folgende Verbesserungen erfolgten nach dem 14.01.1989:
Der verfügbare Speicher wurde von 0FFFFh auf 0FBFFh zurückgenommen, um das freie Gebiet für Monitorbefehle und Ein-/Ausgabedaten nicht zu verletzen:
Im Source-Code ist bei Adresse 0DB3 diese Änderung durchzuführen.
==> LD DE, 0FBFFh - Speicherende + Umlenkung

Zum Schluß wurde noch die Bell-Funktion eingebaut, nachdem es "bei meinem Computer piept." (Siehe Bericht in dieser LOOP)
Bei Fehlermeldungen und wenn das Laden der Datei TAST.KOD.DAT beendet ist, wird ein CTRL-G - Signal erzeugt, das ausgewertet werden kann.

Auch nach dem Kauf:

Die Computerzeitschrift *LOOP* ist die Brücke zum Kunden – Programme, Infos, Tips + Tricks!

Jahres-Abo DM 25,–, Probeexemplar kostenlos!



Umfassend informiert Sie unser Katalog.

Schutzgebühr: DM 10,– incl. MWSt.

Bitte bestellen Sie mit anhängender Postkarte!

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	25,-

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	25,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift
Bei Minderjährigen die des gesetzl. Vertreters

Neue Produkte – Neue Preise!

Best.-Nr.	Bezeichnung	Preis DM
Buskoppelung		
11302	BUSKOPP Bausatz	198,–
11303	BUSKOPP Fertiggerät	268,–
11301	BUSKOPP Platine	30,–
Paint-Maleprogramm		
11336	PAINTJA 38	149,–
11337	PAINTJA 58	149,–

Best.-Nr.	Bezeichnung	Preis DM
68020 Grundprogramm		
11335	EGRUND20 KD	199,–
11333	EGRUND20 QUELLE 38.	148,–
11332	EGRUND20 QUELLE 58.	148,–
0S-9 für die 68000er		
11350	0S-9/68008	599,–
11351	0S-9/68000	599,–

Bitte Porto nicht vergessen

ANTWORT

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

ANTWORT

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 _____