

CLUB INFO

29. AUSGABE

KONTAKTADRESSE : CLUB 80 / ALEXANDER SCHMID / ST. CAJETAN STR. 38 VII / 8000 MÜNCHEN 80
TEL.: 089 / 495326

- I N H A L T S U E R Z E I C H N I S -

Seite:
und Autor:

Seite:
und Autor:

Etuiinternes

Neues vom Vorstand	01	Alexander Schmid
Fischköpfe-Treffen 89	02	Hartmut Übermann, Gerald Schröder
Vorstellung	03	Stefan Nitschke
Termine	04	Redaktion
Weissagung	05 - 06	Klaus-Jürgen Mühlenbein

Software

Entdecke die Unbekannten	07 - 08	Klaus-Jürgen Mühlenbein
mittelmäßig	09 - 10	Kurt Müller
TSCRIPS retten - wenn's geht, heute	11	Arnulf Sopp
Modifikation des HRG-Treibers RB	12 - 17	Egbert Schröder
Grafikpaket für die HRG 1B-Platine	18 - 31	Oskar Drechsler
Austausch illegaler Z80-Befehle	33 - 36	Artikel aus MC

Hardware

Ein Versuch	37 - 52	
Analog mit drei IC's	53 - 55	Kurt Müller

Börse

Wer hat was -- wer will was	56 - 57
-----------------------------------	---------

Sonstiges

Mikrocomputer-Betriebssysteme -2+3+4	59 - 64	Gerald Schröder
Tips für den Computer-Kauf	65 - 66	Artikel aus ...
Der Computer	67 - 74	Artikel aus Hamburger Hochschulzeitung

Die letzten Seiten

Impressum	75	
Schluß	76	Redaktion
Mitgliederadressenliste	am INFO-Ende	Alexander Schmid
Inhaltsübersicht	am INFO-Ende	
Sonderinfo HD 64180	am INFO-Ende	

Fischköpfe-Treffen 1989 in Heikendorf

Im Gegensatz zu den Nord-Italienern im Club haben es die Nordlichter auch dieses Jahr wieder geschafft, ein Regionaltreffen auf die Beine zu stellen. Diesmal hat sich ein Veteran des Clubs (Helmut Bernhardt) die Arbeit gemacht, eine geeignete Tagungsstätte zu finden und die Organisation zu übernehmen.

Anwesend waren diesmal (das dreckige Dutzend):

Helmut Bernhardt (mit Tandon-PC)
Jörg Brans (mit Model IVp und HP DeskJet)
Harald Braun
Hans-Günther Hartmann (mit Packet-Radio-Ausrüstung)
Andreas Magnus (mit Genie IIIs im Tower-Gehäuse)
Kurt Müller (mit Mega-ST)
Gerhard Neebe (mit TRS-80 im PC-Gehäuse)
Hartmut Obermann (mit Toshiba-PC)
Frank-Michael Schober
Gerald Schröder (mit Genie IIs)
Arnulf Sopp (mit Genie IIIs)
als Gast: Erik-Jan Plog

Neues (?) vom Vorstand

Lange ist's her, daß das letzte Info erschienen ist. Eigentlich wollte ich die neuen Mitglieder immer einzeln begrüßen und willkommen heißen, aber nehmt es nicht persönlich, wenn ich jetzt nur sage: 'Herzlich willkommen, alle, die seit dem letzten Mal dazugekommen sind'. An den Vorstellungen habt Ihr ja gesehen, wer da neu zu uns gestoßen ist und wer was macht.

Es liegt mir fern, unverschämt oder böseartig zu sein, aber ich möchte sagen, daß viele wohl endlich mal gemerkt haben, was Sache ist. Die, die im Info immer um Artikel bitten und betteln werden mit einem Lächeln als Miesmacher und Störenfriede abgetan, aber jetzt jetzt hat es jeder selber gesehen: ohne neue Artikel läuft absolut garnichts. Das Info lebt nunmal von Euren Ideen und wenn nichts kommt, kommt auch nichts zurück. Wie es scheint, sind die ewigen Vielschreiber etwas kürzer getreten und jetzt ist die breite Masse gefordert. Jergendein Problem wird doch wirklich jeder schon gehabt haben, das ihm schlaflose Nächte bereitet hat. Ihr müßt ja nicht schreiben, daß Ihr eine Woche drangesessen seid, ein oder zwei Tage glaubt auch jeder und keiner lacht, daß Ihr Euch 'bei so einem lächerlichen Problem' so blöd angestellt habt. Wer sich da lustig machen kann, der soll sich selber an die Nase fassen und fragen, warum er das nicht schon längst im Info gebracht hat. Ich hoffe, daß keiner so schadenfroh ist und sich freut, daß er die Lösung hat, an der sich andere vielleicht die Zähne ausbeißen. Dann können wir uns gleich in 'Rambo-Club, Vereinigung der härtesten aller Einzelkämpfer' umbenennen.

Jetzt aber genug gelästert. Als ich das letzte Vorwort geschrieben habe, war von Weihnachten weit und breit nichts zu sehen und jetzt wird es wohl etwas zu spät sein, aber trotzdem wünsche ich Euch noch ein gutes Neues Jahr und dem Club viele fleißige Autoren.

Alexander Schmid

Der "harte Kern" reiste schon am Freitag an, während sich die Masse erst am Sonnabend von ihren Familien losreißen konnte. Neben dem offiziellen Teil, der sich aus vier Vorträgen zusammensetzte, wurde gefressen, geraucht, gesoffen und natürlich die Hardware begrabbelt. Folgende Themen kamen zum Vortrag:

Zunächst stellte Hans-Günther die Möglichkeiten des Packet-Radio (Datenübertragung per Funk) vor. Da er zur Zeit noch kein geeignetes Programm für CP/M und NEWDOS zur Verfügung hat, lief die Demonstration auf Helmut's PC (den Helmut nur zu diesem Zweck mitgebracht hatte).

Nach einer kurzen Verschnaufpause und einigen Bierchen führte Kurt seinen selbstgestrickten Analog-Digital-Wandler vor. Der Wandler läuft an jedem Rechner mit Centronics-Schnittstelle und eignet sich z.B. zur Aufnahme von Meßwerten (Temperatur, Spannung usw.). Als Beispiel führte Kurt auf seinem Atari vor, wie man aus einem Rechner ein Speicher-Oszilloskop macht.

Zu diesen beiden Vorträgen werden sich noch Veröffentlichungen im Info gesellen, damit auch das faule Volk, das dieses Jahr (wieder) nicht zu unserem Treffen erschienen ist, davon was hat (Zitat Hartmut). Die folgenden Geistes-Ergüsse bleiben dagegen dem dreckigen Dutzend vorbehalten.

Da wäre als erster Arnulf zu nennen, der in unvergleichlicher Manier vorführte, wie die moderne Technik in Form eines Diktiergeräts Souffleur und Manuskript ersetzt (fast sogar den Vortragenden selber). Inhaltlich bezog sich sein Vortrag auf die Optimierung von Assembler-Programmen, wobei er besonders einige sehr leistungsfähige, aber wenig benutzte Z80-Befehle hervorhob. Andreas ergänzte dieses Kapitel mit einem Referat über die unterschiedlichen Möglichkeiten, einen Speicherbereich mit einem Byte zu füllen.

Abschluß des Vortrags-Marathons bildete Gerald, der sich über die Kommentierung und Dokumentation von Programmen, besonders Assembler-Programmen, ausließ. Dabei wurde deutlich, daß optimierte Assembler-Programme Gefahren in sich bergen, wenn sie nicht sehr ausführlich kommentiert sind. Obwohl die beiden Vorträge eigentlich sehr unterschiedliche Aspekte der Programmierung beleuchteten, ergänzten sie sich doch.

Besonders erwähnen möchten wir noch unsere Überraschung und Freude, als plötzlich unser DDR-Vertreter Frank-Michael auftauchte. Helmut hatte Franks Anreise geheim gehalten, so daß vor allem Hartmut, der gerade mit dem Booten von Foodware beschäftigt war, das Gebiß aus dem Gesicht fiel (Zitat Gerald).

Abschließend bedanken wir uns ganz herzlich bei Helmut, der uns Quartier gab und seine Wohnung auch bis 5 Uhr morgens für die Hartgesottenen als Tagungsort zur Verfügung stellte.

Hartmut und Gerald

HEFT
29
Januar
1990

Ich möchte mich kurz vorstellen.

03

Mein Name ist Stefan Nischke und wohne in Walzbachtal 1. Ich bin 25 Jahre alt und studiere Physik an der Universität Karlsruhe.

Zu CP/M kam ich durch den günstigen Einkauf eines ECB-Bus Rechners Anfang dieses Jahres. Es stellte sich jedoch für mich als schwierig heraus, Software für CP/M-Rechner zu bekommen, da meine Bekannten Atari ST oder MS-DOS Rechner benutzen.

Das Betriebssystem meines Rechners (ZDOS) ist CP/M 2.2 kompatibel.

Interesse habe ich an fast allen (Höheren) Programmiersprachen, und Hardwareprojekten. Für Numerikprogramme habe ich hier einen Arithmetikprozessor 8231A von Intel herumliegen.

Das Einbinden einer Grafikkarte in CP/M 2.2 interessiert mich ebenfalls. Ich kenne das Angebot für ECB-Bus Systeme nicht außer Grid5 von Conitec welche in den Anzeigen die ich davon habe runde 700 DM kostet (sind allerdings schon etwas älter). Wenn einer von Euch ein preisgünstigeres Angebot kennt z.B. auch eine Selbstbau-Karte meldet Euch doch bitte.

Ich programmiere in Turbo-Pascal, Turbo-Modula, Assembler und ein wenig in Fortran.

Turbo-Pascal habe ich an den 8231A Coprozessor angepasst und bring nur für einen Z80-Rechner eine angenehme Rechenleistung. Eine entsprechende Anpassung an Modula2 habe ich noch nicht geschrieben, da ich den Compiler erst seit kurzem habe. Wenn einer von Euch schon eine Einbindung vorgenommen hat oder mir ein paar Tips geben kann schreibt mir doch bitte, oder schreibt an Jens Neueder.

Mein Rechner besteht aus:

- CPU-Karte : Z80 mit 4Mhz Takt von Janich+Klass HKN 105X
- Floppycontr. : JK 82 FDC 8/5 ebenfalls von Janich+Klass
- Speicher : 64 KBytes Eigenbau, 2 x 5.25" 40-Spur-Laufwerke
- Terminal : Infoton nicht grafikfähig, keine Attribute, maximal (nur) 9600 Baud
- Drucker : Speedy 100-80

Meine Adresse:

Stefan Nischke
Germanenstr.5
7519 Walzbachtal 1
Tel.: 07203/452

Mit Grüßen

Stefan

PS.

Ich hoffe Ihr seid mittlerweile nicht alle vom PC, AT oder ST Füßchen angesteckt

-- Termine -- Termine -- Termine --

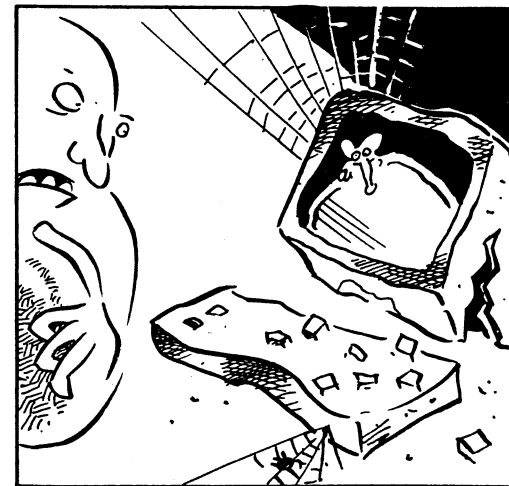
04

Nächster Redaktionsschluß Mitte / Ende Februar

Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!

-- Messen '90 --

Int. Spielwaremesse	Nürnberg	08.02. - 14.02.
mit Fachmesse Modellbau, Hobby u. Basteln		
CeBIT	Hannover	21.03. - 28.03.
Telematica	Stuttgart	11.06. - 14.06.
electronica	München	06.11. - 10.11.



*** WEISSAGUNG ***

Der Satz:

"Mit Statistik kann man alles beweisen"

muß ganz entschieden bestritten werden!

Wer das behauptet (und das tun viele) will entweder Böses aushecken oder hat keine Ahnung von den mathematischen Grundlagen wahrer Statistik (sog. "Stochastik")

Das Folgende darf daher nicht so ernst genommen werden (falls einer meiner Beiträge überhaupt je ernst genommen wurde). Es geht schlicht um eine Behauptung; vielleicht auch eine Weissagung - insoweit aufgrund einer Statistik extrapoliert werden darf.

Doch keine Furcht: Falls die "Weissagung" berechtigt ist, bedeutet sie nicht mehr und nicht weniger, als daß die Zukunft als ein "weißes Blatt" vor uns - dem "CLUBBO" - liegt, d.h. als ein leeres Blatt, und dieses mag außer seiner Weißheit vielleicht sogar einige Weisheit aufweisen... (Im Malerjargon auch "aus-weißen".)

Als ich dem "CLUBBO" beitrug, waren bereits 9 Hefte erschienen. Diese kaufte (!) ich sz. nach. Ihre Eingangsdaten kenne ich deshalb nicht. Als Neuling wurde mir jedoch mitgeteilt, daß alle zwei Monate ein Heft, jährlich also sechs Hefte erscheinen würden. Da bei mir eingehende Post einen Eingangsvermerk erhält (falls sie mir wichtig oder interessant erscheint), war es ein Leichtes, die nachstehende "Statistik" anzufertigen.

Sie soll diejenigen einmal nachdenken machen, die mit dem IBO und seinen Nachfolgern umgehen und daran interessiert sind, Erfahrungen von Gleichgesinnten (Gleichsündern) und Gleichgesonnenen (gleichen "Sonnen") zu erfahren - wenn schon nicht mit diesen auszutauschen.

Ich weiß nicht, ob es noch einen entsprechenden Club geben wird, zu dem diese in 1991 ausweichen können, falls das Tabellenende (die "Statistik") stimmt! Wahrscheinlich nicht; sondern nur noch Clubs der "Kompatiblen" - sprich: IBM-Jünger!

Also: Geniert Euch nicht, Ihr Stummen !! Packt aus, was Ihr bisher angepackt habt an Eurem stummen Diener! Nur ER ist zur Stummheit = Dummheit verdammt. Auch wenn er inzwischen noch so hochgerüstet wurde!

Packt Eure Aktivitäten aus, seien sie auch noch so bescheiden! (Ich tu's auch...) Denn wer schreibt, der denkt. Wer denkt, der kann irren (muß aber nicht), und wer irrt, ist überhaupt nicht dumm - im Gegenteil, denn er dachte; - nur wer schweigt und nie was sagt oder auch fragt - ja, der ist wahrscheinlich wirklich dumm!

Dieses statement soll die nicht treffen, die bisher keinen Beitrag geleistet haben, weil sie absolut keine Zeit hatten (??), sondern die, die in Zukunft noch ...

Na denn, schaut Euch erst mal die Tabelle an. Vielleicht geht manch einem von selbst ein Licht auf!

*** INFO-Statistik des CLUB-80 (extrapoliert) ***

(oder: "Die Rückwärtsreihe der Natürlichen Zahlen")

Heft Nr.	Eingang	Anz.	Preis DM/Heft	Heft Nr.	Eing.	Anz.	Preis DM/Heft		
10	Nov.85			22	22.1.88				
11	Jan.86	⑥	8.33	23	14.4.88	④	12.50		
12	25.3.86			24	23.7.88				
13	27.5.86			25	12.10.88				
14	26.7.86			26	10.5.89				
15	2.10.86	⑤	10.-	27	26.7.89	③	16.67		
16	Weihn.86			28	9.12.89				
17	21.1.87			29	?..?.90			②	25.-
18	12.3.87			30	?..?.90				
19	23.6.87	①	50.-	31	?..?.91	Lohnt sich das?			
20	25.8.87								
21	17.11.87								

Preisbasis: 50.-DM/a

* P.S.: „extrapoliert“ does not mean: «Extra polished», like „extra poliert“ (= geschönt) but «extended» ...

NB: Ich weiß natürlich, daß außerdem Werte vorgehalten werden! [Bücher, Programme] + Porti ...

HEFT
29
Januar
1990

07

Boucuse kocht nicht nach Rezepten.
Die Rezeptur entwickelt er im Kopf.
(Trotzdem schmeckt es vielen...)

Zwar bin ich kein Koch. Und schon gar kein *Boucuse*.
Doch reizt es auch mich nicht, meine Suppe nach fertigen Lösungsrezepten - wir nennen sie "Programme" - zu kochen. Die Lösung programmiere ich lieber selbst. Auch wenn es schon zwei bis hundert "bessere" Programme gibt.
Es soll sogar Köche geben, denen ihr eigenes Gebräu und Gebrutzel am besten schmeckt. Weil nämlich die eigene Zuwendung und Arbeit drinsteckt.
In diesem (oder dem letzten) Heft findet ihr mein Programm zur Lösung von Determinanten beliebiger Ordnung.
Ein System aus m linearen Gleichungen für n Unbekannte ist dann besonders sympathisch, wenn genau so viele Gleichungen zur Verfügung stehen wie Unbekannte zu lösen sind, d.h. wenn $n=m$.
Noch sympathischer ist es, wenn mindestens eine dieser Gleichungen von Null verschieden, also wenn sie 'inhomogen' ist. Dann kann nämlich das Lösungsverfahren nach *CRAMER* angewendet werden. Dies läuft folgendermaßen:
Zuerst werden die Unbekannten in allen Gleichungen mitsamt ihren Koeffizienten in der gleichen Reihenfolge auf die linken Seiten gebracht und die Konstanten (die sog. "absoluten Glieder") auf die rechten Seiten. Die Koeffizienten bilden dann eine Matrix. Deren Determinante wird berechnet. (Wir nennen sie die "Hauptdeterminante" oder den "Nenner").
Dann werden die Koeffizienten der ersten Unbekannten in allen Gleichungen durch die jeweiligen absoluten Glieder ersetzt. Die Determinante dieser leicht veränderten Matrix wird berechnet und durch den Nenner dividiert. Das Ergebnis ist die erste Unbekannte.
Zur Berechnung der übrigen Unbekannten wird in gleicher Weise verfahren.
Wie man sieht, ist dies ein stures Schema, also bestens für unseren Rechenknecht geeignet.
Nachdem ein Programm zur Lösung von Determinanten beliebiger Ordnung (!) vorlag (s.o.), war es naheliegend, es bei dem hier beschriebenen Verfahren nach *CRAMER* anzuwenden.
Wie das geschieht, zeigt mein nachstehendes "wohlstrukturierter" Programm. Was bei seinem Ablauf gerade geschieht, wird auf dem Bildschirm laufend mitgeteilt. Diese Mitteilungen dienen im Listing gleichzeitig als Kommentare für die Unterprogramme. (Auch in PASCAL sind Unterprogramme das Konzept der Strukturierung.)
Wenn wenig zu rechnen ist (wenige Unbekannte), laufen diese Mitteilungen so schnell ab, daß man sie kaum lesen kann - was ja auch gar nicht so wichtig ist; nur bei längeren Berechnungen möchte man sicher sein, daß der Rechner was tut.
Schließlich wird der Bildschirm gelöscht und die Ergebnisse, d.h. die Lösungen der n Unbekannten werden angezeigt.
Das Programm wurde mit mehreren Gleichungssystemen verschiedener Ordnung getestet. (Wer evtl. trotzdem noch einen Fehler findet, teile ihn mir bitte mit.) Seine Geschwindigkeit ist befriedigend. Das entsprechende Programm von *Rugg und Feldman*, das der eine oder andere von euch vielleicht kennt, läuft auch nicht schneller.

Evgnxa!

(H-eure-ka-jot)

08

Lösung eines inhomogenen linearen Gleichungssystems

```

10 'Mühlenbein, Weinheim, 27.10.1989
20 CLS:PRINT " Lösung eines inhomogenen Gleichungssystems":
PRINT:PRINT "Das Gleichungssystem muß in der 'Normalform' vo
rliegen:":PRINT "Links vom Gleichheitszeichen die Unbekannte
n mit ihren
Koeffizienten, rechts die Konstanten!":PRINT
30 DEFINTI-Z:INPUT "Anzahl Unbekannte (= Anzahl Gleichungen!
)";R:DIMA(R,R),C(R,R),B(R),D(R):PRINT:PRINT "Eingabe :":PRIN
T:FORZ=1TOR:PRINT "Zeile";Z:FOR S=1TOR:PRINT " Spalte";S;": "
;:INPUT A(Z,S):C(Z,S)=A(Z,S):NEXTS:PRINT
40 INPUT "Rechte Seite (Konstante) ";B(Z):PRINT:PRINT:NEXTZ
50 FORSS=0TOR
60 PRINT:PRINT "Rechte Seite einsetzen":GOSUB110
70 PRINT:PRINT "Berechnung der Determinante":GOSUB130
80 PRINT:PRINT "Hauptmatrix regenerieren":GOSUB120
90 NEXTSS
100 CLS:FOR S=1TOR:PRINT:PRINT " X(";MID$(STR$(S),2,3);") =
";D(S)/D(O):NEXT:END
110 FORZ=1TOR:A(Z,SS)=B(Z):NEXT:RETURN
120 FORZ=1TOR:FOR S=1TOR:A(Z,S)=C(Z,S):NEXTS,Z:RETURN
130 FORS=1TOR-1:IFA(S,S)=0THEN GOSUB150
140 FORJ=0TOR-S-1:F=A(S+J+1,S)/A(S,S):A(S+J+1,S)=0:FORI=1TO
R-S:A(S+J+1,S+I)=A(S+J+1,S+I)-F*A(S,S+I):NEXTI,J,S:D=1:FORI
=1TOR:D=D*A(I,I):NEXT:D(M)=D*(-1)^M:AX=M+1:RETURN
150 FORI=S+1TOR:IFA(I,S)<>0THEN GOTO160ELSE NEXTI
160 FORU=1TOR:CMD "F=SWAP",A(S,U),A(I,U):NEXT: X=2*(I-S)-1:RE
TURN

```

mittelma3sig...

In meinem Programm zum ANAloginterface benutze ich zur Anzeigeberuhigung eine Mittelwertbildung, die nicht erst n Werte aufsummiert und dann das Ergebnis präsentiert. Die dadurch entstehende Wartezeit erschien mir unpraktisch zumal bei sehr großem n die Wartezeiten wirklich stören.

Hiervon ausgehend, habe ich mir überlegt, wie der Vorgang der Summenbildung so umgestaltet werden kann, daß er mit jedem eingelesenen Meßwert einen Mittelwert ausgibt. Bei der "normalen" Lösung wird - nehmen wir an über 10 Werte - solange alles auf einen Haufen gepackt, bis eben jene 10 Werte eingelesen sind. Dann erfolgt die Division durch 10, womit der Mittelwert berechnet ist. Schwankt der Meßwert nun während dieser Zeit, so werden natürlich auch die einzelnen Meßwerte unterschiedlich sein. Die schwankungsbreite dividiert durch 10 ergibt dann die noch verbleibende "Unruhe" in der Anzeige. Daran ändert auch meine Lösung nichts, aber es spricht nichts dagegen, diese Mittelwertbildung bis zum Erreichen eines Endwertes anzuzeigen. Womit die Forderung pro Meßwert eine Ausgabe konform geht.

Das Problem in einen Algorithmus aufgedröselst sieht dann so aus.

alter Weg:

$$\hat{u} = \frac{\sum_{i=1}^n u_i}{n}$$

\hat{u} -> anzeigen

neuer Weg:

```

u(1..n) = 0      !Alle Elemente Null setzen
DO
  u = u - un-1  !un-1 = letzter alter Meßwert
  u = u + un    !un = neuer Meßwert
  u / 10 -> anzeigen
LOOP

```

Damit meine Methode funktioniert, ist ein eindimensionales Array mit n Felder notwendig, in das die Meßwerte eingetragen werden. Ausprogrammiert sieht das dann so aus:

```

OPTION BASE 0      !Es gibt ein "Nulltes Element"
summier_faktor%=10 !über 10 Werte Mitteln
DIM mittel%(summier_faktor%)
ARRAYFILL mittel%,0 !Alle Elemente Null setzen
mittel%(0)=1       !Feld 0 von mittel%() = Array-Pointer ins Feld !

```

```

summe%=0
adc%=0
REPEAT
  rd_adc(ska!,off!) !Hole einen Meßwert vom ADC
  mittel(adc%,summe%,mittel%()) !Bilde Mittelwert
  PRINT AT(1,1);adc% !Wert ausgeben
UNTIL INKEY$("<>")
END
,
PROCEDURE mittel(adc_wert%,VAR summe_wert%,mittel_wert%())
  IF summier_faktor%>1 THEN
    SUB summe_wert%,mittel_wert%(mittel_wert%(0))
    ADD summe_wert%,adc_wert%
    mittel_wert%(mittel_wert%(0))=adc_wert%
    INC mittel_wert%(0)
    IF mittel_wert%(0)>summier_faktor% THEN
      mittel_wert%(0)=1
    ENDIF
    adc%=summe_wert%/summier_faktor%
  ELSE
    adc%=adc_wert%
  ENDIF
RETURN

```

Für diejenigen, die sofort mit klammen Fingern auf's Hackerbrett zustürzen und meine kühnen Behauptungen erproben wollen, zuvor noch einige Zeichenerklärungen. Das Ausrufezeichen wie auch das einfache Hochkomma leiten eine Kommentar im Programmbeispiel ein. Im Unterprogramm "MITTEL()" taucht bei den Übergabeparametern die Anweisung "VAR" auf. Das bedeutet, daß die nachfolgenden Variablen selbst übergeben werden. Mit anderen Worten, das Unterprogramm kann den Inhalt selbiger verändern. Die Variablen werden jedoch auf den Namen im Prozedur-Kopf umgetauft. Die Zuordnung erfolgt in der Reihenfolge der Nennung:

```

          mittel(adc%,          summe%,          mittel%())
                |                |                |
                v                v                v
PROCEDURE mittel(adc_wert%,VAR summe_wert%,mittel_wert%())

```

Arrays werden dabei ohne Angabe der Dimension übergeben. Übrigens kann unter Benutzung des Unterprogramm MITTEL() sehr einfach die 1. Ableitung des Meßwertes gebildet werden. Um diese zu erhalten muß lediglich vom aktuelle Meßwert der durch MITTEL() berechnete Wert subtrahiert werden. Also:

```

rd_adc(ska!,off!)          !Hole einen Meßwert vom ADC
mittel(adc%,summe%,mittel%()) !Bilde Mittelwert
diff%=mittel%(mittel%(0)-1)-adc% !1. Ableitung berechnen

```

Logisch, das jetzt natürlich sofort der Regelungstechniker einhackt und sich im stillen fragt: "Wenn das so ist, wie baue ich mir dann einen PI-, PID- oder PD-Regler zusammen...?" Ich für meinen Teil verdrück mich jetzt nach unten zu Frauchen an den Weihnachtstisch und überlaß den Hackern das Feld.Tschüs

Handwritten signature

Umlaute mit NewDos/80 Vers.2.0; neuer Tastaturtreiber

11
Lese ich richtig? Da ist mir also bei irgendeinem Systemabsturz des Bierglases auf das CPU-Board ein TSCRIPS-Text verstümmelt worden. Den soll ich nun erstens bis sechstens bzw. gegf. über fünf-a-tens und DUMP und SUPERZAP ... Nö, ehrlich!

Die Rede ist von einem Verfahren zum Restaurieren von Speicher-Datensätzen (auch ZEUS-Sources usw.), bedichtet von Kajot im Info Nr. 28. Sehen wir uns das Vorher und Nachher genauer an: Falsche Bytes, die TSCRIPS nicht alleine auf die Reihe kriegt, werden geändert ... Warum guckt ihr so ungeduldig? Damit ist es doch schon erschöpfend beschrieben, mehr ist es nicht!

Das einfachste Werkzeug für Speicheroperationen ist immer noch der Debugger. Man braucht für ihn nicht einmal zwei gesunde Hände, nur drei Finger. Die legt man - möglichst gleichzeitig - auf die Zifferntasten 1, 2 und 3 (im normalen Alpha-Block der Tastatur). Anschließend das Loslassen nicht vergessen, sonst setzt im jetzt bereits ungeduldig mit dem Cursor zappelnden Debugger die Tastenwiederholung ein! Wie jener betan wird, steht in der Anleitung zum DOS. Und davon ist auch nur das M-Kommando wichtig.

Die Verstümmelung wird man wohl während der Arbeit mit TSCRIPS bemerkt haben. Warum aussteigen? Mit 1-2-3 kann man nicht nur komfortabel vorübergehend fremdgehen - mit G, ENTER kommt man ebenso sutsche auch wieder rein, und zwar genau da, wo man die Arbeit unterbrochen hat. Nur der Bildschirm sieht jetzt wild aus. Mit Shift-Hochpfeil oder irgendeiner anderen Scrollerei ist auch das wieder eingereckt.

Aber nicht nur die Bequemlichkeit spricht für Debug. SUPERZAP (oder wie sie alle heißen) ist etliche kB groß. Der zu rettende Text wird in der Kombination TSCRIPS-SUPERZAP noch nicht überschrieben, vielleicht aber bei einer anderen Paarung von Datenschaufel und -doktor. Der Debugger aber ist ein SYS-File, belegt also kein Bit des Anwenderspeichers. Damit sind die Daten sicher.

Zwei Tips noch:

Oft will man aus TSCRIPS aussteigen, ohne den Text zu drucken. Er soll nur erst mal auf die Diskette, weil er z. B. noch nicht fertig ist. Der Drucker ist nicht eingeschaltet. In diesem Fall fragt ihn TSCRIPS hartnäckig ab und kehrt erst ins DOS zurück, wenn man endlich klein beigt und ihn anmacht. Das geht elegant auch mit 1-2-3: Jetzt sind wir im Debugger. Mit dem Kommando Q, ENTER springt dieser das DOS an.

Der Text für TSCRIPS steht keineswegs bei allen Versionen ab 9177h. Kaum ein Programm wurde von so vielen Leuten so oft modifiziert wie TSCRIPS. Es empfiehlt sich daher, zuvor nachzusehen. Mit Debug.

Arnulf Sopp

Der hier beschriebene Patch des Supertreibers ermöglicht die Eingabe von Umlauten, die Rückkehr vom BASIC zum DOS ohne die Tastenkombination 1,2,3. Außerdem erübrigt sich eine neue Definition des G-Zeichensatzes zur Darstellung von Umlauten durch einlesen des Zeichensatzes. Die Umlaute sind jederzeit aktiv. Nachteil ist das stete Geflimmer der HRG, aber hier kann eventuell ein Hardware-Fachmann helfen?

Grundidee dieses Patches war eigentlich nur ein neuer Tastaturtreiber, der es mir ermöglichen sollte jederzeit Umlaute ohne große Mühe einzugeben. Die Bedingung dazu war, daß dies mit der gleichen Tastenkombination geschehen sollte, wie bei dem von mir durchgeführten SCRIPSIT-Patch (siehe SONDERINFO 27.5). Außerdem sollte die eigentliche Funktion der Tasten nicht verändert werde. Als dieser Tastaturtreiber fertiggestellt war, stützte ich mich auf den HRG-Supertreiber V1.1 von RB-Elektronik. Es ist nämlich nicht einzusehen, daß man nach dem Laden des Treibers und Basic nun auch noch den G-Zeichensatz beispielsweise mit Umlauten neu definieren muß. Zusätzlich störte mich schon immer, daß man über DEBUG (Tasten 1,2,3) in das DOS zurückkehren mußte. Hilfsmittel für die nun folgende kriminalistische Kleinarbeit war das Monitorprogramm TASMOM.

Doch zunächst zum neuen Tastaturtreiber, der ganze 124 Bytes belegt. Min. 25 Bytes könnte man einsparen, indem man Teile des Tastaturtreibers in eine freigewordene Stelle des HRG Supertreibers installiert. Ich hielt dies nicht für sinnvoll, möchte aber auf die Möglichkeit hinweisen.

Listing 1: Tastaturtreiber:

```
*****  
;* Keyboard Driver für HRGDOS *  
*****  
;* Copyright by Egbert Schröer *  
;* Version 1.1 *  
;* October 1989 *  
*****  
; ORG 4049H ;setzt HIMEM neu  
DEFW KPDNEU-1
```



```

        ORG      0DAS4H          ;oder wo man möchte
;
;Entry Point muss dem HRGDOS angepasst werden
;
KBDNEU: LD      HL,(4016H)      ;Adresse KBDNEU in 4016H
        LD      (DRIVER+1),HL
        LD      HL,DRIVER
        LD      (4016H),HL
        JP      0EB36H          ;umschalten auf HRG
DRIVER: CALL    S-S            ;Tastaturabfrage -> Taste
;                                     gedrückt ? Wenn ja,dann
        CR      A              ;sichere den Tastaturcode
        RET     Z              ;in AKKU
        LD      (AKKU),A
;
;Wenn S gedrückt, dann warte auf nächste Taste. Ist
;es kein Vokal, wird S ausgegeben. Folgt ein
;Vokal, wird dieser in seinen entsprechenden
;Umlaut umgewandelt.
;
        LD      A,(3801H)      ;Abfrage Tastaturzeile
        BIT     0,A            ;BIT 0 gesetzt ?
        JR      Z,BACK         ;Nein, dannn ->ENDE
        CALL    0049H          ;Abfrage nächste Taste
        LD      A,(3803H)      ;und warten. Wenn <SHIFT>
        CP      1              ;dann Großbuchstabe
        JR      C,GROSS
        LD      A,(3801H)
        CP      2
        JR      Z,UMF1
        LD      A,(3802H)
        CP      108
        JR      C,UMF2
        LD      A,(3804H)
        CP      32
        JR      Z,UMF3
        LD      A,(3803H)
        CP      4
        JR      Z,UMF4
KEINV:  LD      A,40H          ;kein Vokal ->dann ALPHA
        JR      ENDE           ;in Akkumulator
GROSS:  LD      A,(3801H)
        CP      2
        JR      C,GUMF1
        LD      A,(3802H)
        CP      108
        JR      C,GUMF2
        LD      A,(3804H)
        CP      32
        JR      C,GUMF3
        JR      KEINV          ;kein Vokal
;kein S gedrückt -> zurück und weiter mit Zeichen
;in AKKU !

```

```

BACK:   LD      A,(AKKU)
        RET
ENDE:   RET
;*****
;*      UNTERROUTINEN
;*****
UMF1:   LD      A,7BH          ;A
        JR      ENDE
UMF2:   LD      A,7CH          ;B
        JR      ENDE
UMF3:   LD      A,7DH          ;C
        JR      ENDE
UMF4:   LD      A,7EH          ;D
        JR      ENDE
;
GUMF1:  LD      A,5BH          ;A
        JR      ENDE
GUMF2:  LD      A,5CH          ;B
        JR      ENDE
GUMF3:  LD      A,5DH          ;C
        JR      ENDE
;
AKKU    DEFS    1
        END     KBDNFU

```

Die Abfrage der Tasten erfolgt direkt über die Abfrage der entsprechenden Tastaturzeile. Der Entry Point DAS4H ist auf den HRG Supertreiber abgestimmt. Der Tastatortreiber kann aber auch im Speicher frei verschoben und ohne HRG benutzt werden. JP EB36H wird später erläutert.

Nun zum HRG-Treiber. Um die Umlaute stets präsent zu haben, wurden diese einmalig in den 0-Teichensatz eingelesen und anschließend das Programm von DE00B bis FF24H mit Entry Point EB00H auf Diskette gebannt.

Die Darstellung der Umlaute sollte mit der Option #PRINT(2) erfolgen, die den normalen Bildschirminhalt auf die HRG umleitet und dabei den definierten 0-Teichensatz benutzt. Anschließend sollte ein Rücksprung in den IOS erfolgen. Listing 2 zeigt das dienstverbliebene Programm von EB00H bis EB79H. Listing 3 den ausgeführten Patch von EB00H bis EB79H und den Patch ab EB05H (#Print(2) Befehl). Patch ermöglicht Rücksprung zum IOS).

Listing 2: HRG-Supertreiber V1.1

```
EB00    LD      HL,EB79H
```

```

LD      (4004H), HL
LD      HL, 0DAFFH
LD      (EB74H), HL
LD      HL, 0AAFFH
LD      (EB76H), HL
LD      HL, (401EH)
LD      (0F6B7H), HL
LD      A, (4012H)
CP      0FBH
JP      Z, EB32H
LD      HL, (EB74H)
LD      (4049H), HL      ;setzt HIMEM
LD      HL, EB51H      ;HL auf Text ab EB51H
CALL    4467H          ;Ausgabe Text RB-EL HRG usw.
LD      HL, EB4BH      ;HL auf Text ab EB4BH=EBASIC
JF      4419H          ;DOS Call-> BASIC
EB32H   LD      HL, ZBB51H
CALL    28A7H
LD      HL, (EB74H)
LD      (40B1H), HL
LD      DE, 0FFCEH
ADD     HL, DE
LD      (40A0H), HL
CALL    1B4DH
JP      1A19H
EB42H   DEFB    'BASIC'
DEFB    0DH
EB51H   DEFB    'RB-EL HRG SUPER-TREIBER V1.1'
DEFB    0DH
NOP
LD      E, L          ;Version EPS48
LD      D, B
LD      D, E
INC     (HL)
JR      C, EB75H
EB75H   NOP
EB76H   NOP
NOP
EB78H   NOP
EB79H   EX      (SP), HL
LD      A, H
CP      1DH
JR      Z, EB83H
CP      20H
JR      NZ, EB8AH
LD      A, L
CP      5BH
JR      Z, EB8AH
CP      50H
EB8AH   EX      (SP), HL
JP      NZ, 1D78H
CALL    1D78H          ;RST 10H: analysiert Programmtext
PUSH    AF            ;HL dient als Zeiger
    
```

Hier kann geändert werden!

```

CP      23H          ;-> #
JP      NZ, EC46H    ;nein, dann zurück
INC     HL          ;nächstes Zeichen;holt Z. in A
LD      A, (HL)
CP      83H
JP      Z, 0EC55H    ;#SFT
CP      82H
EBA0H   JP      Z, 0EC6DH ;#RESET
CP      0C6H
JP      Z, 0EC98H    ;#POINT
CP      0A2H
JP      Z, 0ECE0H    ;#OPEN
CP      0A6H
EBAFH   JP      Z, 0ECD8H ;#CLOSE
CP      9CH
JP      Z, 0ED09H    ;#LINE
CP      84H
JP      Z, 0ECE8H    ;#CLS
CP      0B8H
JP      Z, 0EC48     ;#CLEAR
CP      0AFH
JP      Z, 0FE76H    ;?
CP      0CBH
JP      Z, 0EEC7H    ;?
CP      0A4H
JP      Z, 0EBE9H    ;#GET
CP      0A9H
JP      Z, 0EF95H    ;#NAME
CP      0A5H
JP      Z, 0F39AH    ;#PUT
CP      0DBH
JP      Z, 0FEB9H    ;?
CP      0A0H
JP      Z, 0FE96H
CP      0F2H
JP      Z, 0F5EDH
CP      0E0H
JP      Z, 0F64BH
CP      0B2H
JP      Z0F6C5H      ;#PRINT(2)
    
```

usw.

Listing 3: Patches im HRG Treiberprogramm

```

EB00H   LD      HL, EB79H
LD      (4004H), HL
LD      HL, 0DAFFH
LD      (EB74H), HL
LD      HL, 0AAFFH
LD      (EB76H), HL
    
```

```

LD HL,(401EH)
LD (0F6E7H),HL
LD HL,(EB74H)
LD (4049H),HL
CALL 01C9H ;CLS-Befehl
CALL 04FCH ;Umschalten auf 32 cpl
LD HL,3004H ;Cursor nach 3004H
LD (4050H),HL
LD HL,EB44H ;HL auf Text ab EB44H
CALL 4467H ;Text ausgeben
LD HL,EB3DH ;HL auf DOS Command ab EB3DH
JP 4419H ;DOS Command ausführen= KBDHRG
EB36H LD A,00H
OUT (01H),A ;HRG einschalten
JP 0F6E5H ;HRG lÖschen und #PRINT(2)
EB3DH DEFB "KBDHRG"
DEFB 0DH
EB44H DEFB "HRG-DOS Version 1.6"
DEFB 0AH
DEFB 0AH
DEFB "<c> by E.Schröder '89"
DEFB 0DH

F6F2H JP 1DEH ;Steuerung Programmausföhrung
Wird ersetzt durch:
F6F2H JP 402DH ;dadurch RÜcksprung zum DOS

```

Nachdem der Tastatortreiber geladen wurde, wird ein Sprung von diesem nach EB36H ausgeföhrt und dort die HRG eingeschaltet. Dann wird der normale Bildschirmtreiber vollwertig ersetzt. Alle Ausgaben erfolgen mittels G-Zeichensatz durch die HRG.

Anstatt die Message ab EB44H auszugeben, kann man den freigewordenen Speicherplatz, wie ich eingangs erwähnte, anders nutzen oder Teile des Tastatortreibers dort unterbringen.

Das Tastatortreiberprogramm wird als KBDHRG/CMD abgespeichert, HRG/CMD habe ich in HRGDOS/CMD umbenannt.

Für Anregungen, Verbesserungen, Kritik und weitere Möglichkeiten wäre ich dankbar.

Oskar Drechsler

'Grafikpaket' für die HRG 1B-Platine

Vor langer Zeit fand ich in der 'Computer persönlich' das Grafik-Programm Graflix/Bas von Frank Debatin und war natürlich begeistert, denn es gab ja nichts vergleichbares für meine HRG-Platine. Irgendwann ging mir aber der zeitlupenhafte Kopiervorgang von Bildteilen so gewaltig auf den Wecker, daß ich auf Abhilfe sann. Da ich den einzigen Ausweg in Assembler sah, nahm ich mir erstmal das Z80-Handbuch von Rodney Zaks vor, das schon Jahre im Schrank rumlag. Über die ersten 30 Seiten kam ich nie hinaus. Aber nun hatte ich ein Ziel und biß mich durch.

Dann disassemblierte ich HRG/CMD und versuchte die Steuerung der Platine herauszufinden. Als ich das kapiert hatte, war mein Kopf plötzlich so voller Ideen, nach deren Verwirklichung das ursprüngliche Graflix/Bas nur noch als Rumpf-Programm anzusehen ist.

Bei der Beschreibung des Programmes gehe ich mal davon aus, daß das Original-Programm und die Systematik der Funktionen und der Tastenbenutzung bekannt ist.

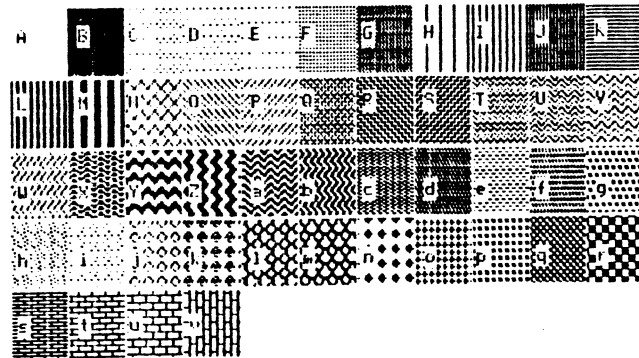
Das Menü sieht wie folgt aus:

- 1 - Neue Grafik erstellen
- 2 - Momentane Grafik ändern
- 3 - Bildschirm auf Diskette
- 4 - Diskette auf Bildschirm
- 5 - Diskette mit Bildschirm mischen
- 6 - Grafikmuster neu initialisieren
- 7 - Alle Grafikmuster zeigen
- 8 - Bildschirm aus GEAP übernehmen
- 9 - Programmende

Die Punkte 1 und 2 sind identisch mit dem Originalprogramm.

Bei Punkt 3 - 5 wird erst das Directory von Laufwerk 1 mit Extension /DIS gezeigt, und dann nach dem Filenamen gefragt, der gespeichert, geladen oder gemischt werden soll. Da beim HRG/CMD von RB-Electronic das LOAD, SAVE oder MERGE in Verbindung mit einem String als Filenamen nicht immer funktioniert, gibt es dazu noch eine Fehlerbehandlungsroutine. Mehr gibt's dazu nicht zu sagen.

Punkt 6: Im Programm sind einige Grafikmuster (Raster) fest installiert, die immer an die Cursorposition angepaßt werden und deshalb immer aneinanderpassen (Tun sie auch in 98 % aller Fälle. Warum die 2 % spinnen, habe ich noch nicht herausgefunden). Wenn der von mir vorgegebene Musteranfang nicht paßt, kann man ihn verschieben. Wenn man vor lauter Verschieberei nicht mehr durchblickt, stellt Punkt 6 den Grundzustand wieder her.



Punkt 8: Nähere Erläuterungen sind aus dem Programm-Listing ab Zeilen-Nr. 8900 ersichtlich. Wer GEAP hat, weiß sicherlich was ich meine.

Dem Basic-Grundprogramm habe ich versucht eine Struktur zu geben und durch eine geordnete Variablen-Tabelle und mir bekannte Tricks etwas Reine zu machen. Alles was trotzdem noch langweilig war (Bildteile kopieren, Flächen mit Muster ausfüllen) habe ich in 'Maschine' umgeschrieben.

Nun zu den einzelnen Funktionen und Möglichkeiten:

Folgende Funktionen habe ich nicht verändert:

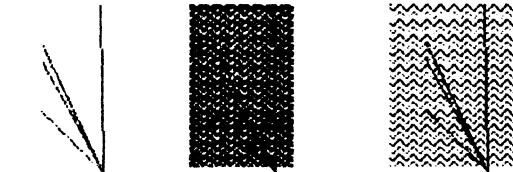
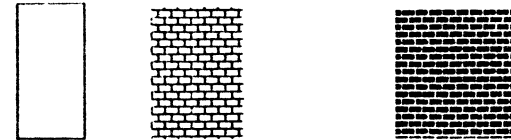
- (A)uslöschten Punkt
- (P)unkt setzen
- (M)arkiere Punkt
- (E)ingabe Cursorkoordinaten
- (G)rafikausschnitt (markiert) festlegen (etwas erweitert)
- (H)ilfe
- (I)nvertiere den gesamten Bildschirm
- (K)reis: (W)eiß (S)chwarz (A)usmalen (L)öschen (I)nvertieren
- (L)inie: (W)eiß (S)chwarz (I)nvertieren

ergänzt wurde:

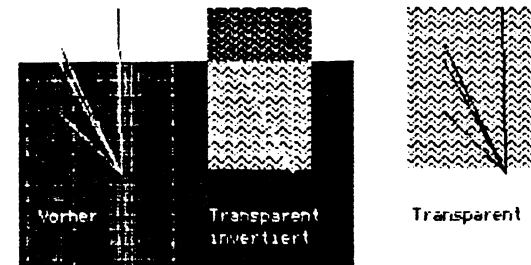
- (V)iereck: (W)eiß nur Rand
- (S)chwarz nur Rand
- (L)öschen ganzes Viereck
- (I)nvertieren ganzes Viereck
- (i)nvertiert nur den Rand
- (A)usmalen
- (T)ransparent ausmalen
- (t)ransparent invertiert ausmalen
- (a)usmalen invertiert
- wobei nach der Wahl der Ausmal-Funktion immer noch die Frage nach dem Muster kommt

Zum besseren Verständnis gebe ich auf der folgenden Seite einige Beispiele:

Viereck weiß Viereck ausmalen invertiert ausmalen



Vorher Transparent invertiert Transparent



Vorher Transparent invertiert Transparent

Kopieren von Bildteilen

Zuerst wird wie gehabt das zu kopierende Feld unten rechts markiert und der Cursor dann an die linke obere Ecke gestellt und dann 'G' gedrückt.

Dann erscheint folgendes:

Für Bildteile kopieren ohne Überlappung:

seitengleich : Cursor in linke obere Ecke, dann 0 und c
 nach links kippen : Cursor in rechte obere Ecke, dann 0 und l
 nach oben kippen : Cursor in linke untere Ecke, dann 0 und o
 nach ob/li kippen : Cursor in rechte untere Ecke, dann 0 und g

Bei transparent kopieren, anstatt 0 ein T drücken.
 Bei invers kopieren, anstatt 0 ein i drücken.
 Bei inv./trans. kopieren, anstatt 0 ein t drücken.

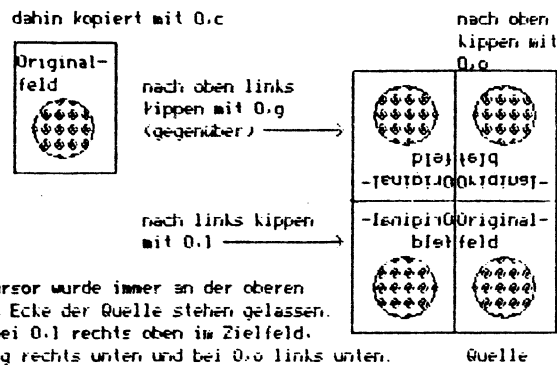
<Taste> drücken, damit das Bild wieder erscheint !

Bei der Buchstabenbenutzung ging ich von folgender Überlegung aus:

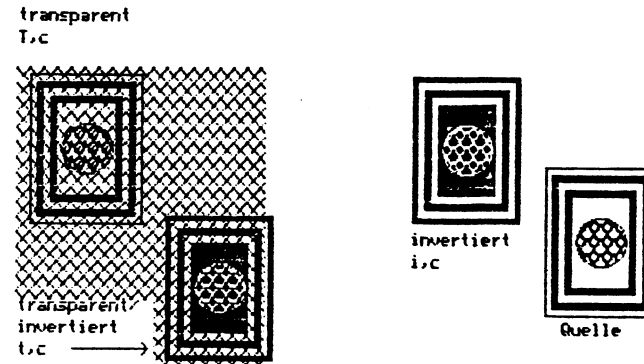
0 wie Original c = kopieren wie Originalfeld
 l = links für nach links kippen
 o = oben für nach oben kippen
 g = gegenüber für nach oben links kippen

Die Anfangsbuchstaben der anderen Kommandos sind wohl klar. Die Tasten sind nacheinander zu drücken.

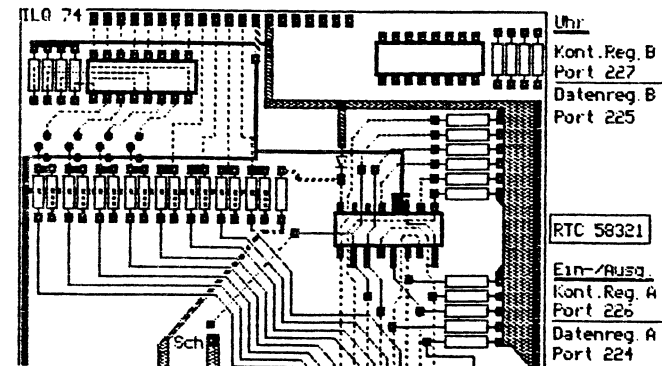
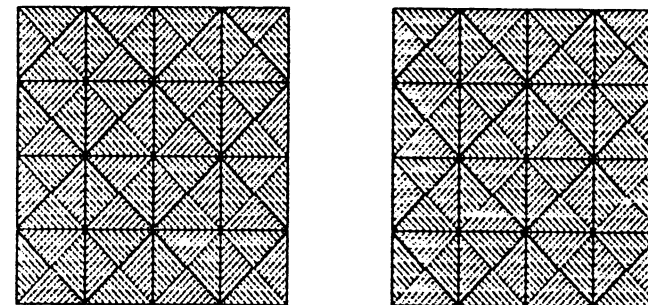
Nun wieder einige Beispiele:



Der Cursor wurde immer an der oberen linken Ecke der Quelle stehen gelassen, d.h. bei 0.l rechts oben im Zielfeld, bei 0.g rechts unten und bei 0.o links unten.



Dieses Feld dahin kopiert dauert 25 sec



Die letzte Hardcopy zeigt das 'Teil-Layout' meiner Hardware-Uhr. Es ist für eine 2-seitige Leiterplatte gedacht. Die durchgezogenen Linien liegen auf der Bestückungsseite. Wenn man z.B. mehrere Widerstände schön in eine Reihe haben will, braucht man nur einen zu zeichnen und zu markieren und dann den Cursor nur um festgelegte Punkte weiterzuschieben und immer O,c zu drücken und fertig ist der Kram. Bei meinem Drucker paßt zufällig auch das Rastermaß 2.54 mm. Deshalb habe ich die Uhr komplett am Bildschirm entworfen. Da er nicht ganz ausreichte, habe ich alles auf 2 Bildschirme verteilt und dann nahtlos aneinandergedruckt. Vorher habe ich bei dem Layout für die Oberseite die unterbrochenen Linien gelöscht und beim Layout für die Unterseite die durchgezogenen Linien. Anschließend habe ich die Hardcopy nachgeschwärzt, alles dreifach auf Folie kopiert, diese aufeinandergelegt und die Platine belichtet. Man muß sehr genau arbeiten, aber es geht.

Nun noch ein Wort zum Kopieren, wenn sich das Quellfeld und das Zielfeld überlappt. Man kann nur mit Überlappung kopieren, wenn das Zielfeld über oder/und links vom Quellfeld stehen soll. Das liegt daran, daß das Quellfeld von oben links nach unten rechte Punkt für Punkt auf das Zielfeld übertragen wird. Außerdem ist noch zu sagen, daß beim Zielfeld kein Bildschirmrand erkannt wird. Das habe ich aus Gründen der Verarbeitungsgeschwindigkeit weggelassen. Wird das nicht beachtet, hängt das Programm sich zwar nicht auf, aber das Bild wird an einer falschen Stelle weitergemalt.

Nun weitere Funktionen. Dazu muß ich vorausschicken, daß ich den Artikel mit Genie-Text schreibe. Hier sind Doppelkreuz, Dollarzeichen, größer-als und kleiner-als-Zeichen Steuerzeichen und somit nicht darstellbar.

Größer als : kopiert Bildschirm in den 2. Speicher
 Kleiner als: kopiert den 2. Speicher in den Bildschirm

*und wandert
 Punkt 2.2.*

(m)ischen Bildschirm mit 2. Speicher
 (w)echseln Bildschirm mit 2. Speicher (vertauschen)

() ab Cursorposition waagrecht schreiben mit G-Zeichensatz
) ab Cursorposition senkrecht schreiben mit G-Zeichensatz

Doppelkreuz: wie (, nur K-Zeichensatz
 Dollarzeichen: wie), nur K-Zeichensatz

Hardcopy vom Bildschirm:

```

Club 88
1
u
b
8
8

```

Aus diesen Funktionen kommt man mit dem Klammeraffen wieder heraus. Geschrieben wird transparent. Der Cursor wandert mit der Schrift nicht mit, sondern steht nach Drücken des Klammeraffen wieder an der alten Stelle.

Nun die neu hinzugefügten Funktionen:

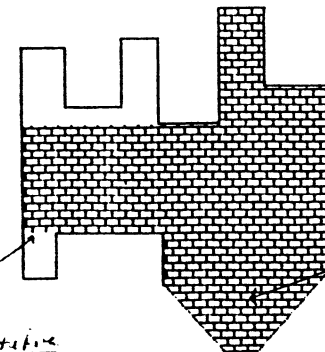
(U)ngelmäßiges Feld mit Muster ausfüllen

Hier mußte ich leider einige Einschränkungen machen. Bei der Architektur dieser Grafik-Karte ist das Ansprechen eines Punktes schon recht zeitraubend. Wenn man nun noch ein Muster bei der Cursorbewegung mitrotieren läßt, die Begrenzung eines Feldes erkannt werden muß und wenn diese vergessen wird, auch der Bildschirmrand erkannt werden soll, kommt man schnell an einen Punkt, wo auch reine 'Maschine' zu langsam wird, wenn man noch mehr will.

Ich bin folgendermaßen vorgegangen: Zuerst wird der Cursor irgendwo in das weiß begrenzte Feld gestellt. Nach drücken von 'U' und Auswahl des Musters beginnt die Suche ab Cursorposition nach oben, bis ein gesetzter Punkt gefunden wird oder der obere Bildschirmrand erreicht ist. Dann wird nach links gesucht, bis ein gesetzter Punkt oder der linke Bildschirmrand erreicht ist. Dabei wird in 'Gedanken' immer das Muster mitrotiert. Sodann wird nach rechts gezeichnet, bis ein gesetzter Punkt gefunden oder der rechte Bildschirmrand erreicht wird. Jetzt erfolgt ein Sprung zu der Stelle, an dem bei der 'Aufwärtssuche' auf 'Linkssuche' umgeschaltet wurde und untersucht, ob eine Zeile tiefer ein Punkt gesetzt oder der untere Bildschirmrand erreicht ist. Wenn nein, beginnt die Suche wieder nach links usw.

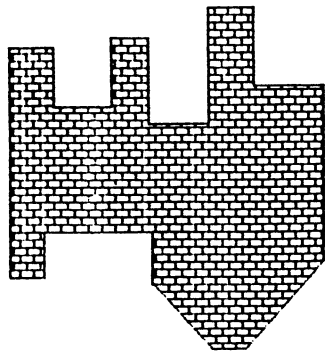
Daraus geht hervor, daß man am besten den Cursor in die Spalte setzt, die am weitesten nach oben und/oder nach unten reicht. Bei Feldern, die Taschen enthalten, die bei der Suche nicht mit erreicht werden, muß man halt nochmal ansetzen. Da das Muster mitrotiert, paßt das Muster bis auf die bereits erwähnten 2 % zusammen.

Dazu wieder einige Beispiele:

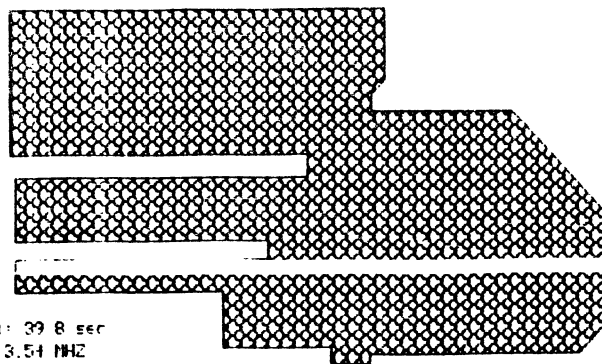
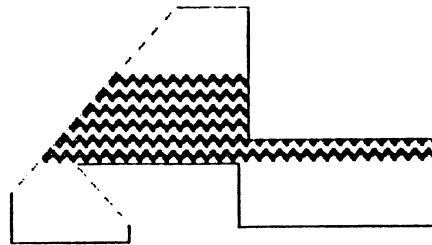
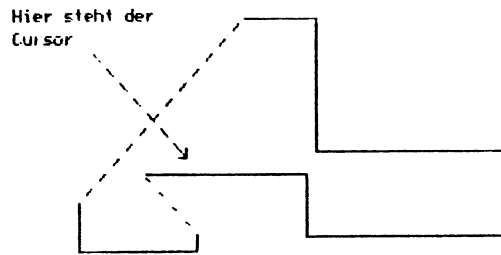


Hier stand der Cursor

*unter der haken
 stelle des cursors
 stehen, dann geht
 das haken auf
 keinen fall*



So sieht es fertig aus
Auch das Muster paßt!



Kein Programm fehlt,
sondern das Fachwissen

25

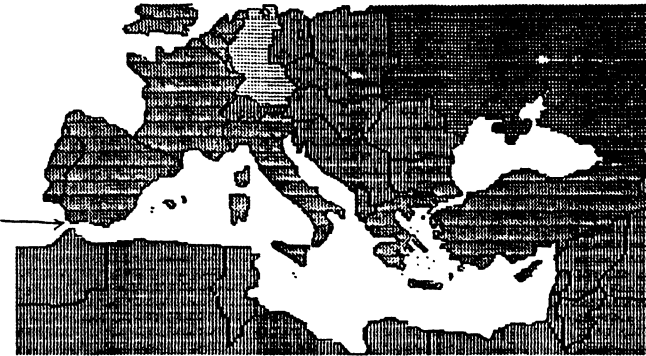
Zeit: 39,8 sec
bei 3,54 MHz

(u)regelmäßiges Feld weiß ausfüllen

Hier gibt es nur die Einschränkung, daß aus Geschwindigkeitsgründen der Bildschirmrand nicht erkannt wird. Ist das begrenzte Feld irgendwo offen, daß der Cursor den Bildschirmrand erreichen kann, kommt der Cursor irgendwo am Bildschirm wieder zum Vorschein und malt alles weiß, was ihm über den Weg kommt.

Taschen werden erkannt, gespeichert und nachträglich ausgemalt. Und das alles in akzeptabler Geschwindigkeit.

Beispiel:



Zurück in
Sibirien und
rüber muß
in Japan
gebaut werden



Die Trockenlegung des
Mittelmeeres dauert

53 SEC

HEFT
29
Januar
1996

26

Zuerst wird der Cursor 1 Punkt über die Stelle gebracht, an der das 1. Zeichen beginnen soll, dann wird & gedrückt. Anschließend wird das Directory von Laufwerk 1 mit /FON gezeigt, damit man weiß, welche Font's zur Verfügung stehen. Dann wird der Abstand zwischen den Buchstaben in Dot's abgefragt und dann nach dem auszugebenden Wort oder Satz gefragt. Ist der Satz länger als noch Platz in der Zeile am Bildschirm ist, wird er entsprechend abgeschnitten. Die Buchstaben werden transparent gezeichnet. Mehr gibt's dazu nicht zu sagen.

Bei meinen Arbeiten mit Graflix suchte ich oft mit müden Augen den winzigen Cursorpunkt. Versuche, den Cursor größer zu machen scheiterten an der Verarbeitungsgeschwindigkeit und der genauen Definition des Cursor-Mittelpunktes, der dann invers zum umgebenden Rand sein sollte. Deshalb habe ich in Anlehnung an das GEM meines AT im Büro das 'Gummi-Viereck' erfunden:

g zeichnet kurz ein Viereck um Markierung und Cursorposition

Da die Line-Funktion der HRG bei großen Abstand zwischen Markierung und Cursor zu langsam war, habe ich mich für die Line-Funktion der Block-Grafik entschieden. Hier kommt es ja nicht auf Genauigkeit an, sondern nur um den Cursor zu finden oder eine Vorstellung zu haben, wie groß ein zu zeichnendes Viereck wird. Das geht natürlich nur, wenn man im NEWDOS das Sys29/Sys hat. Auf der Diskette, die ich Oliver schicken werde, ist alles notwendige drauf.

Z zeichnet bei jeder Cursorbewegung kurz ein 'Gummi-Viereck' wie oben beschrieben. Z hebt das wieder auf

l Muster um 1 Punkt nach links rotieren

r Muster um 1 Punkt nach rechts rotieren

a Muster um 1 Punkt abwärts rotieren

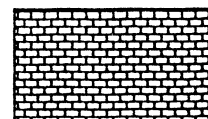
o Muster um 1 Punkt nach oben rotieren

Diese o. a. Funktionen verschieben die fest eingespeicherten Muster. Bei der Installierung der Muster wollte ich erreichen, daß beim Ausfüllen verschiedener Flächen mit gleichem Muster diese zueinanderpassen, wenn diese Flächen überlappen oder einander berühren. Das hat zur Folge, daß die Muster quasi fest am Bildschirm stehen. Da das nicht immer gut aussieht oder nicht paßt, muß man die Muster verschieben können.

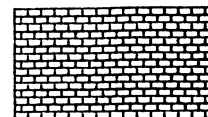
Beispiel: Will man ein Haus mit Ziegelmuster zeichnen, kann es vorkommen, daß das Muster nicht zur Hausecke paßt. Die Ziegel müssen ja im Verbund um die Hausecke gehen.

Hierzu wieder ein Beispiel:

So wirkt das nicht out



2 x nach links und 1 x nach oben und nochmal V.A.u und schon sieht das besser aus.



Dann habe ich noch die NAME-Funktion eingebaut, die mir aber nicht viel bringt. Deshalb habe ich auch keine Lust sie zu erläutern. Die 'Hilfe' erläutert das aber ausreichend. Die (H)ilfe besteht übrigens aus 4 Bildschirmen. Weiterschalten mit irgendeiner Taste oder mit Klammeraffe Abbruch und Rückkehr zum Grafikbildschirm.

Mit den Pfeiltasten bewegt man den Cursor um einen Punkt, mit SHIFT/Pfeiltasten um 10 Punkte. Will man also ganz an den Rand, darf man die letzten 10 Punkte SHIFT nicht benutzen.

Mit Klammeraffe kommt man zurück zum Hauptmenü

Als die bisher beschriebenen Funktionen liefen, war ich endlich zufrieden und auch etwas stolz auf mich. Fast ein halbes Jahr lang. Doch ganz tief in meiner Computerseele piekste mich immer noch was. Vor allem dann, wenn ich im Büro mit GEM spielte und neidisch den schönen großen still dastehenden Pfeil der Maus sah. Das hat mich fertig gemacht.

Zu allem Unglück fiel mir dann auch noch das türkische Sprichwort ein: 'Wenn Du etwas nicht erreicht hast, hast Du nicht richtig gewollt.'

Fazit: ein maschinengesteuerter Cursor mußte endlich her! Ich will nun nicht alle Irrungen wieder aufwärmen, aber das Resultat ist nach meiner Ansicht das Beste was aus dem Genie I bei 3.54 MHz herauszuholen ist. Wenn mein geplanter 'Helmut Bernhardt I' fertig ist und ich neben der Hercules-Karte auch noch Lust habe, die HRG 1B-Platine mit anzuschließen und das auch möglich (und sinnvoll) ist, bieten sich sicher wieder neue Möglichkeiten.

+ Sprung von der Basic-Ebene in die Maschinenebene.

Hier erscheint ein rechter Winkel als Cursor. Er ist weiß und blinkt nicht. Die Punkte, die der Cursor abdeckt, werden im RAM zwischengespeichert und nach einer Bewegung wieder zurückgebracht. Die Spitze des rechten Winkels ist der eigentliche Cursorpunkt und blinkt über dem überdeckten Punkt. Wenn der Cursor sich einem Bildschirmrand nähert, wechselt er seine Form, um bis an den Rand heranzukommen. Da bei einer rein weißen Fläche der Cursor nicht sichtbar wäre (außer der winzigen blinkenden Spitze), befindet sich innerhalb des rechten Winkels noch ein negativer Winkel.

Das klingt sicher etwas verwirrend, eine nachstehende Grafik verdeutlicht das aber sicherlich.

Mit SHIFT/Pfeiltasten geht alles etwas schneller. Hier wird aber nicht um 10 Punkte weitergesprungen, sondern eine Zeitschleife wird übersprungen.

Es sind auch Kombinationen der Pfeiltasten möglich, mit denen man sich also schräg vorwärtsbewegen kann. Das ist sehr nützlich bei der Zeichenfunktion, die gleich beschrieben wird. Der Bildschirmrand wird natürlich erkannt.

In dieser Ebene sind folgende Funktionen möglich:

p oder P = Punkt setzen
 a oder A = auslösch Punkt
 m oder M = markiert Punkt
 Klammeraffe = Rücksprung in Basic-Ebene

Die Markierung wird übrigens in die Basic-Ebene mit übertragen, wenn man zurückspringt.

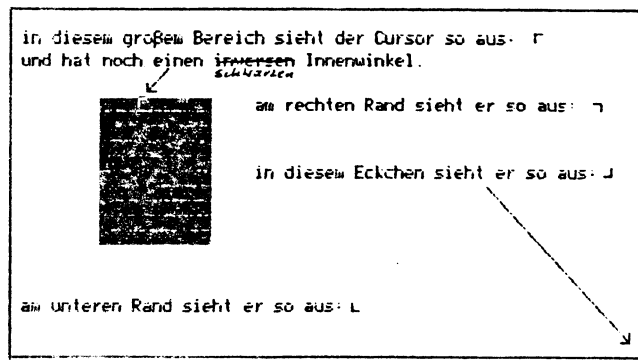
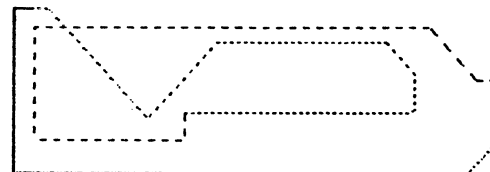
Drückt man die Enter-Taste, erscheint in der rechten oberen Ecke ein '?'. Dann drückt man die Tasten 0-7.

0 = löschen mit dem Cursor
 1 - 7 = zeichnen mit dem Cursor (Punktraster und Strich)

29 Drückt man anstatt der ENTER-Taste 'T', zeichnet der Cursor transparent.

CLEAR-Taste hebt das Zeichnen/Löschen mit dem Cursor wieder auf.

- Muster 1
- Muster 2
- Muster 3
- Muster 4
- Muster 5
- Muster 6
- Muster 7



Die Cursorbewegung ist ohne SHIFT ausreichend langsam, mit SHIFT ausreichend schnell. Geschwindigkeit von links nach rechts:

ohne SHIFT: 19 sec
 mit SHIFT: 5 sec

Wenn das zu langsam oder zu schnell ist, kann das ändern. In 60D2H steht: LD BC,1300H, das ist für die Zeitschleife in 0060H.

In 60D3H steht also 00H und in 60D4H steht 13H (19D). Entweder ändert man den Source-Code von HRGGR.ASRC und assembliert neu oder nimmt in Graflix eine Zeile neu auf und pokt so lange in 60D4H herum, bis das paßt.

Die Diskette enthält folgendes:

31

Autostart mit Graflix/JCL. Dieser Chain-File setzt den Basic-Anfang hoch und lädt HRGGra/CMD. HRGGrafl/CMD macht MEM1 und RENEW und läßt Graflix/HRG rennen.

HRGGra/CMD enthält fast alle von mir hinzugefügten Maschinenroutinen und einen Tastaturtreiber. Siehe hierzu HRGGra/SRC.

HRGGrafl/CMD ist das ursprüngliche HRG/CMD. Hier habe ich in der Lprint-Routine (für Epson-kompatible Drucker) die beiden Initialisierungsroutinen vor und nach der Druck-Routine herausgenommen, um Hardcopy's nahtlos aneinanderfügen zu können. Auch habe ich den etwas unschönen G-Zeichensatz durch die Zeichen ersetzt, die der normale Charaktergenerator erzeugt. Da HRG/CMD nur bis FF24H geht, habe ich hier noch ab FF25H die noch fehlenden Maschinen-Routinen angefügt (siehe Grafli1/SRC) und das alles HRGGrafl/CMD genannt.

Graflix/Ori ist das Original-Programm, das mit Mem1 nicht läuft.

Graflix/HRG ist das um die REM's gekürzte Basic-Programm.

GraflixA/CMD ist ein Bildschirm mit einem Vorstellbild.

GraflixD/HRG sind Data-Zeilen mit den Mustern, das gemerzt wird und nach Abarbeitung wieder gelöscht wird. (Platz ist knapp)

GraflixH/HRG sind die Hilfe-Zeilen, die bei Bedarf ebenfalls gemerzt werden.

XXX/BI ist ein mit GEAP erzeugtes Bild als Test, das mit Menüpunkt 8 geladen werden kann.

Zum Schluß ein Wort zu den fest programmierten Grafikmustern. Wer die ändern oder ergänzen will muß die Systematik kennen.

Ich will mal das Muster 'u', die großen Ziegel erklären:

Die Data-Zeile 61560 lautet: 8.255.4.4.4.255.64.64.64
8 = 8 Byte hoch (höchstens)

Darstellung der Byte's:

```
!x!x!x!x!x!x!x! 255 = 11111111 binär
! ! ! ! ! ! ! ! 4 = 00000100
! ! ! ! ! ! ! ! 4 = 00000100
! ! ! ! ! ! ! ! 4 = 00000100
!x!x!x!x!x!x!x! 255 = 11111111
!x! ! ! ! ! ! ! 64 = 01000000
!x! ! ! ! ! ! ! 64 = 01000000
!x! ! ! ! ! ! ! 64 = 01000000
```

oder die Data-Zeile 61170: 4,136,68,34,17 Diagonale
4 = 4 Byte hoch

```
!x! ! ! ! !x! ! ! ! 136 = 10001000 binär
! !x! ! ! ! !x! ! ! ! 68 = 01000100
! ! !x! ! ! ! !x! ! ! ! 34 = 00100010
! ! ! ! !x! ! ! ! !x! ! ! ! 17 = 00010001
```

Wo ein x steht, wird der Punkt gesetzt. Und aneinandergereiht sieht das alles recht gut aus.



32

Austausch illegaler Z80-Befehle

Ersatzroutinen für die Basic-Interpreter XITAN und HEBAS

In vielen Z80-Programmen werden illegale Befehle verwendet. In Z80-Emulationsprogrammen und bei der neuen CPU 64180 können diese Befehle jedoch nicht verwendet werden. Unser Autor zeigt hier am Beispiel der Basic-Interpreter XITAN und HEBAS, wie diese illegalen Z80-Befehle ersetzt werden können. Ebenso wird ein neuer Basic-Befehl USER für den mc- und den NDR-Klein-Computer vorgestellt.

alte Folger:		neue Folger:	
2BE1	4E LD C, (HL) DD 69 LD LX, C 23 INC HL 4E LD C, (HL) DD 61 LD HX, C 23 INC HL	2BE1	4E LD C, (HL) 23 INC HL 46 LD B, (HL) C5 PUSH BC DD E1 POP IX 23 INC HL 00 NOP 2BE9 4E LD C, (HL)

Bild 1. Sechs Bytes werden ab der durch das HL-Register angegebenen Speicherstelle in die Register DE, IX und BC kopiert

alt:		neu:	
25C3	DD 7C LD A, HX 1F RRA DD 67 LD HX, A DD 7D LD A, LX 1F RRA DD 6F LD LX, A 25CD CB 1A RR D	25C3	E5 PUSH HL DD E5 PUSH IX E1 POP HL CD 40 47 CALL 4740 E1 POP HL 00 00 NOP NOP 25CD CB 1A RR D

Bild 2. Diese Routine verändert Low- und High-Byte der Register BC, DE und IX durch den Befehl RRA

alt:		neu:	
26BC	DD 7C LD A, HX 1F RRA DD 67 LD HX, A DD 7D LD A, LX 1F RRA DD 6F LD LX, A 26C6 CB 1C RR H	26BC	E5 PUSH HL DD E5 PUSH IX E1 POP HL CD 40 47 CALL 4740 E1 POP HL 00 00 NOP NOP 26C6 CB 1C RR H

Bild 3. Bei der Multiplikationsroutine finden sich ähnliche illegale Opcodes wie in Bild 2. Es kann der gleiche Einsprung bei 4740H verwendet werden

Im mc 1/82 wurden 422 neue Z80-Befehle veröffentlicht, welche durch eifriges Experimentieren gefunden wurden. Diese Befehle waren jedoch bereits auf einem anderen Kontinent bekannt und wurden dort auch verwendet. Damit stehen 1116 Befehle der Z80-CPU zur Verfügung. Besondere Bedeutung erlangen die zusätzlichen Befehle für die einzelnen Hälften der Indexregister IX und IY. Diese werden so zu vollwertigen Registerpaaren wie die Register HL, DE, oder BC.

Ein Beispiel soll dies erläutern, genauere Hinweise sind in [1] zu finden. Der 8-Bit-Ladebefehl LD C, H (4C) des HL-Registers lautet für das IX-Register LD C, HX (DD 4C). HX steht für die Bits 8...15 von IX. Damit wird das High-Byte des IX-Registers nach Register C kopiert. Entsprechend gibt es zum Befehl LD C, L (4D) den illegalen Teil LD C, LX (DD 4D), wobei LX für die Bits 0...7 steht. Als Ergänzung zu dem in [1] aufgeführten Artikel seien die beiden illegalen Befehle OR HX (DD B4) und OR LX (DD B5) des IX-Registers nachgetragen.

Assembler und Emulatoren steigen aus

Ein großer Nachteil bei der Verwendung der illegalen Befehle besteht darin, daß Hilfsprogramme wie DDT, DDTZ und viele Assembler diese Befehle nicht beherrschen und meistens nur mit einem erstaunten ??? reagieren. Bei einem Debugger kann man sich mühsam durch Setzen von Breakpoints behelfen. Bei Assemblern müssen die Befehle per Byte-Definition direkt gesetzt werden.

Die günstige Preisentwicklung bei Speicherbausteinen und die Entwicklung von 16-Bit- bzw. 32-Bit-CPUs lassen das Arbeitspferd Z80 zunehmend in den Hintergrund treten.

Doch so mancher Umsteiger von CP/M 2.2 auf CP/M-68k würde seine unter CP/M 2.2 erstellten Basic-Programme gerne weiter verwenden. Dazu gibt es einen Z80-Emulator der Firma Soft-Design [2], der auf jedem CP/M-68k-System ein CP/M-80-System Vers. 2.2 nachbildet [3]. Leider werden aber die illegalen Z80-Befehle vom Emulator nicht akzeptiert. Gleiches gilt für die Nachfolger des Z80, z.B. für die CPU 64180, die 512 KByte direkt adressieren kann.

XITAN und HEBAS werden „akzeptabel“

Bei den Basic-Interpretern XITAN von Neil Colvin und dem verbesserten Nachfolger HEBAS (Franzis-Software-Service

und Firma Graf, Kempten) kann man die illegalen Opcodes durch legale ersetzen. Nun laufen auf dem 680XX-NDR-System mit Z80-Emulator beide Interpreter mit den schon vorhandenen Basic-Programmen. Auch die neue CPU-64180-Baugruppe der Fa. Graf macht keine Schwierigkeiten bei den angegebenen Basic-Interpretern.

Ein Nachteil sei nicht verschwiegen. Es können die Speicherbereiche über 64 KByte nicht angesprochen werden, da der Befehl LD (nn), HL (22 XXXX) nur FFFFH als höchste Adresse ansprechen kann. Das gilt auch für CP/M 2.2.

Für die Programmteile des Interpreters, welche die illegalen Opcodes enthalten, wird jeweils die Ersatzroutine angegeben. Da diese mehr Platz benötigen, wurde am Ende des Interpreters Platz reserviert, der von Basic-Programmen nicht überschrieben wird. Der Rücksprung aus der Erweiterung zu gleichen Programmteilen erfolgt zunächst mit Relativsprüngen, dann geht es mit RET (C9) zurück ins Hauptprogramm des Interpreters.

Die Änderungen

Im Rahmen dieses Artikels kann keine genaue Erläuterung aller hier geänderter Programmteile erfolgen. Dies ist einer vollständigen Dokumentation des überarbeiteten Interpreters vorbehalten, die als Kurzfassung (ca. 90 Seiten) z. Zt. auf Diskette vom Autor bezogen [4] und als Dokumentation vom Elektronikladen (Detmold) gedruckt erworben werden kann. Bevor jedoch die aufgezeigten Änderungen (Bilder 1...16) durchgeführt werden können, muß sowohl bei XITAN als auch bei HEBAS der Inhalt der Adresse 46BAH in A0 47H umgeändert werden. In Adresse 479FH muß dann der Wert 0 enthalten sein. Die Routine in Bild 2 verändert Low- und High-Byte der Register BC, DE und IX durch den Befehl RRA. Dabei wird der Inhalt des Akkumulators nach rechts verschoben, wobei der Inhalt des Übertragsbits nach Bit 7, der Inhalt von Bit 0 ins Übertragsbit geschoben wird.

Bei der Ausgabe einer Zahl muß das interne Format in die ASCII-Darstellung umgewandelt werden (Bild 5). Dies geschieht durch wiederholten Abzug der in einer Tabelle abgelegten, elf sechszehnstelligen Konstanten (1 000 000 000 bis 1) von den in den Registern DE, IX und C enthaltenen Werten. Elf Konstanten liegen deshalb vor, da maximal eine elfstellige Zahlenausgabe möglich ist. In Register B wird die ASCII-Zahl, mit 30H beginnend, hochgezählt und dann jeweils in das ASCII-Zahlen-Flag F92H bis F99H

alt:		neu:	
2512	DD 4C LD C, HX DD 7D LD A, LX DD 67 LD HX, A DD 6A LD LX, D 251A AF XOR A	2512	E5 PUSH HL DD E5 PUSH IX E1 POP HL 2516 CD 4B 47 CALL 474B 2519 E1 POP HL

		474B	4C LD C, H 7D LD A, L 67 LD H, A 6A LD L, D 474C 1B F1 JR 4744
alt:		neu:	
2911	5F LD E, A DD 67 LD HX, A DD 6F LD LX, A 2916 B7 OR A	2911	E5 PUSH HL CD 4E 47 CALL 474E E1 POP HL 2916 B7 OR A

Bild 4. Illegale Ladebefehle werden ersetzt		474E	5F LD E, A 67 LD H, A 6F LD L, A 4751 1B F1 JR 4744

alt:		neu:	
4287	DD 7D LD A, LX 9E SBC A, (HL) DD 6F LD LX, A 23 INC HL DD 7C LD A, HX 9E SBC A, (HL) DD 67 LD HX, A 4292 23 INC HL	4287	D5 PUSH DE DD E5 PUSH IX D1 POP DE 7B LD A, E 9E SBC A, (HL) 5F LD E, A 428E CD 53 47 CALL 4753 D1 POP DE 4292 23 INC HL

		4753	23 INC HL 7A LD A, D 9E SBC A, (HL) 57 LD D, A D5 PUSH DE 4758 1B EB JR 4745

alt:		neu:	
25AC	DD 55 LD D, LX 0B EX AF, AF' DD 7C LD A, HX DD 6F LD LX, A 0B EX AF, AF' DD 61 LD HX, C 25B4 0E 00 LD C, 0	25AC	E5 PUSH HL DD E5 PUSH IX E1 POP HL 55 LD D, L 6C LD L, H 25B2 CD 5A 47 CALL 475A E1 POP HL 25B4 0E 00 LD C, 0

Bild 5. Umwandlung des internen Formats in die ASCII-Darstellung		475A	61 LD H, C 475B 1B E7 JR 4744

alt:		neu:	
2693	DD 55 LD D, LX 0B EX AF, AF' DD 7C LD A, HX DD 6F LD LX, A 0B EX AF, AF' DD 61 LD HX, C 4F LD C, A 269E C9 RET	2693	E5 PUSH HL DD E5 PUSH IX E1 POP HL 55 LD D, L 6C LD L, H CD 5A 47 CALL 475A E1 POP HL 4F LD C, A 269E C9 RET

Bild 7. Dieser Abschnitt ist fast identisch mit dem ab 25ACH

alt:	neu:
24F9 DD 9D SBC A,LX DD 6F LD LX,A 23 INC HL 7E LD A,(HL) DD 9C SBC A,HX DD 67 LD HX,A 2503 23 INC HL	24F9 D5 PUSH DE DD E5 PUSH IX D1 POP DE 9B SBC A,E 5F LD E,A CD 5D 47 CALL 475D D1 POP DE 23 INC HL

Bild 8. Von den in den Flags F8AH bis F8EH enthaltenen Werten werden die in den Registern DE, IX und C enthaltenen Werte abgezogen

475D 23 INC HL 7E LD A,(HL) 9A SBC A,D 57 LD D,A D5 PUSH DE 4762 1B E1 JR 4745

alt:	neu:
2597 DD 9D SBC A,LX DD 6F LD LX,A 7D LD A,L DD 9C SBC A,HX DD 67 LD HX,A 25A0 7D LD A,L	2597 D5 PUSH DE DD E5 PUSH IX D1 POP DE 9B SBC A,E CD 64 47 CALL 4764 D1 POP DE 25A0 7D LD A,L

Bild 9. Ersatz eines illegalen Subtraktions- und Ladebefehls

4764 5F LD E,A 7D LD A,L 9A SBC A,D 57 LD D,A D5 PUSH DE 4769 1B DA JR 4745
--

Test auf den Wert FF bei den Registern DE, IX und BC

alt:	neu:
2560 DD 2C INC LX CO RET NZ DD 24 INC HX 2565 CO RET NZ	2560 E5 PUSH HL CD 6B 47 CALL 476B E1 POP HL C0 RET NZ

Bild 10. Test auf den Wert FFH bei den Registern DE, IX und BC

476B DD E5 PUSH IX E1 POP HL 2C INC L E5 PUSH HL DD E1 POP IX CO RET NZ 24 INC H 4774 1B CE JR 4744
--

alt:	neu:
2578 DD 8D ADC A,LX DD 6F LD LX,A 23 INC HL 7E LD A,(HL) DD 8C ADC A,HX DD 67 LD HX,A 2582 23 INC HL	2578 D5 PUSH DE DD E5 PUSH IX D1 POP DE 8D ADC A,E 5F LD E,A CD 76 47 CALL 4776 D1 POP DE 2582 23 INC HL

Bild 11. Hier werden die Werte der Flags F8AH bis F8DH zu den Werten der Register DE, IX und C addiert

4776 23 INC HL 7E LD A,(HL) 8A ADC A,D 57 LD D,A D5 PUSH DE 477B 1B CB JR 4745

abgelegt. Somit kann diese Zeichenkette ausgegeben werden. In Bild 8 werden von den Werten der Flags F8AH bis F8EH die Werte der Register DE, IX und C abgezogen und die Ergebnisse wieder in den Registern DE, IX und C abgelegt. Das HL-Register dient als Zeiger auf die Flags F8AH bis F8EH. Das Register B wird gesondert behandelt. In diesem Zusammenhang sei auf das Buch von Rückath aus der Serie Computerpraxis des Franzis-Verlages [5] verwiesen, das grundsätzliche Erläuterungen der Arbeitsweise von Basic-Interpretern enthält.

Eigene Basic-Erweiterungen

Allerdings soll kurz auf die Möglichkeit eingegangen werden, wie eigene Erweiterungen in Maschinensprache im Interpreter resident abgelegt werden können. Es kann der Speicherbereich nach den Systemmeldungen des Kaltstarts Verwendung finden, der aber nicht von Basic-Programmen überschrieben werden darf. Die Anfangsadresse eines Basic-Programmes steht in Adresse 46BAH. Bei XITAN steht dort 45DBH, bei HEBAS je nach Version eine höhere Adresse, z.B. 4740H. Setzt man bei Adresse 46BAH z.B. den Wert 4900H ein, so werden ab dieser Adresse die Basic-Programme abgelegt. Natürlich verringert sich dadurch der zur Verfügung stehende Speicherplatz. Wichtig ist, daß das Byte vor dem ersten Byte des Basic-Programmes den Wert 0 besitzt. Ein Ausführungsbeispiel für einen neuen Basic-Befehl USER soll dies verdeutlichen.

USER-Befehl unter XITAN- und HEBAS-Basic

Der unter CP/M zur Verfügung stehende USER-Befehl kann mit folgender Änderung auch in Basic anstelle des selten benötigten Basic-Befehls USR als Direktbefehl eingerichtet werden. Der USER-Befehl ermöglicht die Zusammenfassung bestimmter Dateigruppen, z. B. nur Systemprogramme oder nur Basic-Programme in einem eigenen Bereich des Inhaltsverzeichnisses. Sie erscheinen dann beim DIR-Befehl nur nach Umschaltung auf den gewählten User-Bereich. Damit bei beiden Interpretern mit dem DIR-Befehl alle Dateien mit beliebigen Dateizusatz angezeigt werden, müssen ab Adresse 35B9H die drei Bytes 42H, 41H und 53H jeweils durch 3FH ersetzt werden (alte Maske ??????BAS, neue Maske ?????????? beim DIR-Kommando).

höhere Programme auf eine höhere USER-Ebene kopiert werden können, muß das Kopierprogramm PIP.COM auf der gewählten USER-Ebene vorhanden sein. Dies kann mit DDT.COM geschehen, wie in [6] beschrieben und in Bild 15 gezeigt ist. Bei Verwendung von Micro-Shell oder anderen komfortablen Dienstprogrammen entfällt der Vorgang, da nach dem zu kopierenden Programm auch auf anderen USER-Bereichen gesucht wird.

Der neue Basic-Befehl USER verwendet nur die USER-Bereiche 0 bis 9. Die Umschaltung erfolgt z.B. für den Bereich 5 mit CALL USER(5). Der aktuelle USER-Bereich kann mit CALL USER(X) abgefragt werden, es wird dann die Zahl ausgegeben. Ein Basic-Programm wird mit LOAD von der eingeschalteten USER-Ebene geladen. Dann wird mit CALL USER(5) z.B. der 5. USER-Bereich aufgerufen und mit SAVE das Programm dort abgespeichert. Bild 16 zeigt die Änderungen für beide Interpreter. Bei den 5 1/4-Zoll Laufwerken mit 80 Spuren und etwa 780 KByte Speicherkapazität wird das Directory doch recht unübersichtlich. Unter Basic mit dem USER-Befehl kann man es nun in bis zu 10 Teile aufspalten. So kann man die Systemprogramme auf dem User 0 belassen und Basic-Programme auf höhere User-Ebenen verlagern. Hier gibt es inzwischen auch in einer Version für CP/M 3.

Literatur

- [1] Häfelberg, M.: 422 neue Z80-Befehle, mc 1982, Heft 1, Seite 27.
- [2] Thies, H.: Z80-Emulator für 680XX-Systemen, Soft-Design 8 München 19, Hadwigstr. 3.
- [3] Bittner, F.: CP/M auf dem mc-68000-Computer, mc 1985, Heft 2, Seite 89.
- [4] Hehl, H.: LOOP, Zeitung für Computerbauer, G. Graf GES GmbH, Kempten.
- [5] Röckroth, L.: Microsoft-Basic, Konzepte, Algorithmen, Datenstrukturen, Franzis-Verlag.
- [6] Plate, J.: Betriebssystem CP/M, Franzis-Verlag.

Adresse	altes Byte	neues Byte
116F	46	13
1170	01	47
1C71	52	45
1C72	20	52
468A	DB	A0
468B	45	47

Bild 16. Änderungen für den neuen Basic-Befehl USER bei XITAN und HEBAS

alt:	neu:
2771 B3 OR E DD B4 OR HX DD B5 OR LX 2776 20 9F JR NZ, 2717	2771 E5 PUSH HL CD 7D 47 CALL 477D E1 POP HL 2776 20 9F JR NZ, 2717 ***** 477D B3 OR E DD E5 PUSH IX E1 POP HL E4 OR H B5 OR L 4783 1B BF JR 4744

Bild 12. Die Veroderungen des High- und des Low-Bytes des IX-Registers werden durch legale Befehle ersetzt

alt:	neu:
2875 DD 7C LD A,HX BE CP (HL) CO RET NZ 2B DEC HL DD 7D LD A,LX BE CP (HL) 287D CO RET NZ	2875 D5 PUSH DE DD E5 PUSH IX D1 POP DE CD 85 47 CALL 4785 D1 POP DE 287D C0 RET NZ ***** 4785 7A LD A,D BE CP (HL) C0 RET NZ 2B DEC HL 7B LD A,E BE CP (HL) 478B C9 RET

Bild 13. Die Werte der Register BC, IX und DE werden mit denen in den Flags F8FH bis F8AH verglichen.

alt:	neu:
292A DD 2B DEC IX DD 7C LD A,HX DD A5 AND LX 2930 3C INC A	292A E5 PUSH HL CD BC 47 CALL 478C E1 POP HL 00 NOP 2930 3C INC A ***** 478C DD 2B DEC IX DD E5 PUSH IX E1 POP HL 7C LD A,H A5 AND L 4793 C9 RET

Bild 14. Illegale Opcodes von Logikbefehlen werden ersetzt

Befehl:	Hinweise:
DDT PIP.COM	PIP.COM mit DDT.COM geladen unter NEXT-Anzeige steht 1E00 (1E00 = 1*16 + 14 = 30 pages zu je 256 Byte)
G 0 USER 3 SAVE 30 PIP.COM	DDT verlassen gewünschter User-Bereich PIP.COM auf User-Bereich ablegen

Kopieren von Programmen auf den gewählten User-Bereich:
PIP a: = a:DAT.BAS RG0VD R bzw. 0 = eckige Klammer

Programm DAT.BAS wird aus User 0 nach User 3 kopiert (gleiches Laufwerk).

Bild 15. Mit dem Debugger DDT.COM wird ein Kopierprogramm auf eine höhere USER-Ebene unter CP/M 2.2 gebracht

bei HEBAS: folgende Bytes werden ab Adresse 4713 eingegeben:

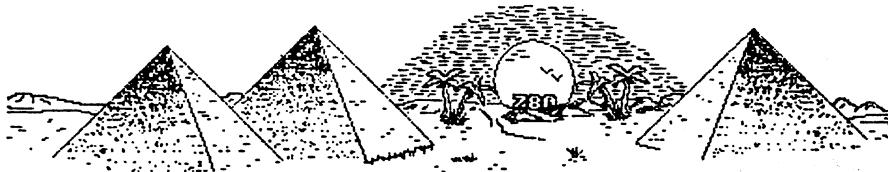
4713	21 58 11 0E 20 7E FE 28 C2 48 20 23 CD
4720	C7*44 D6 30 FE 28 2B 03 FE 0A B2 48 20 5F CD 05
4730	00 1B DD 1E FF CD 05 00 C6 30 CD 94 32 1B D1 00

bei XITAN: gleiche Eingabe, es ändern sich nur die zwei unterstrichenen Bytes:

4732: C4 und 473E: B8

also: CALL USER(X): Ausgabe des aktuellen User-Bereiches
CALL USER(9): Umschaltung z.B. in User-Bereich Nr. 9

CLUB 80 HARD - - - CLUB 80 HARD



Kurt Müller Sophie-Scholl-Ring 3B 2854 Geesthacht CLUB 80

Ein Versuch ...

Hallo Clubfreunde, das Nordlicht-Treffen ist wieder einmal vorüber. Sicher steht am Anfang des Infos der Tagungsbericht, aus dem alles zu entnehmen ist. Meine Absicht ist es, das von mir auf dem Treffen vorgestellte Analoginterface an dieser Stelle noch einmal näher zu beschreiben. Diesmal aber so, daß auch die weniger mit Lötzinn Bekleckerten es verstehen können.

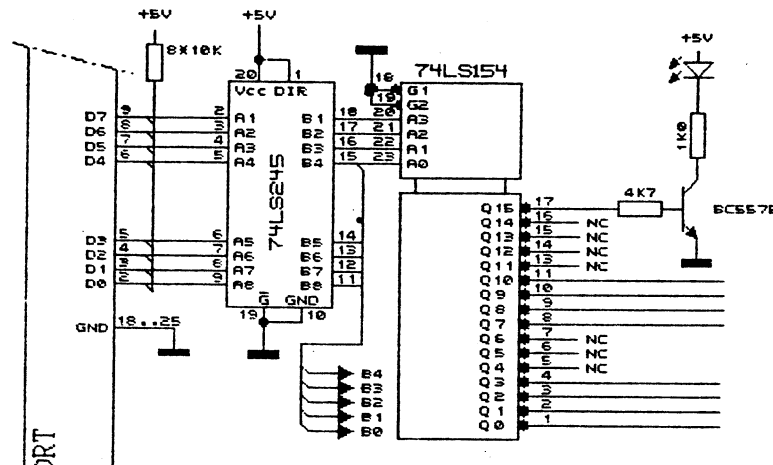
Kommen wir zum Thema. Bevor ich mich in Einzelheiten vergrabe, als erstes ein paar Worte zum Sinn der Übung. Entstanden ist die ganze Sache aus dem Frust heraus, einen tollen Composter unter den Händen zu haben, der aber zu nichts anderem benutzt wird, außer zum Schreiben von Briefen. Das erschien mir dann doch zu wenig. Um diesem Mißstand abzuwehren, habe ich mir ein Analoginterface mit zwei analogen Eingängen (12 Bit Auflösung) und einem 5-bit breiten Datenbus (nur Ausgabe, warum s. u.) zusammengebaut. Das die hier beschriebene Schaltung nur eine Auflösung von 8 Bit hat, liegt einfach am Bauteilpreis. Außerdem bereitet ein Nachbau bei 8 Bit Auflösung weniger Probleme.

Ein Einbau in den Rechner scheidet schon wegen der notwendigen Dekodierlogik und der ungewissen Frage nach dem Adress- bzw. Portbereich aus. Außerdem ist es nicht jedermanns Sache alles in seinem Compi fest zu installieren, zumal selbst das größte Gehäuse nur begrenzt Platz bietet. Von den vielfältigen Möglichkeiten, sein liebstes Stück beim Einbau in die Ewigen Jagdgründe schicken, einmal ganz zu Schweigen. Was also von Außenherb kommt, muß also eine reguläre Buchse benutzen. In den meisten Fällen gibt es an jeden Rechner einen Druckeranschluß, "Centronic" genannt. Dieser hat aber häufig deutlich weniger als 8 Eingangsleitungen. Ein Genie i z.B. vier und bei meiner Kiste sogar nur einer (!). Echt sparsam, oder? Damit ist die von mir eingeschlagene Richtung eigentlich beschrieben. Bleibt noch der Unkostenbeitrag auch "Spaßsteuer" genannt (unabhängig davon, ob die Aktion in Rauch aufgeht oder nicht), zu nennen. Deutlichen Einfluß auf die Rechnerhöhe hat hier die eigene Bastelkiste, wer aus ihr Platinen, Netztrafo, A/D-Wandler und ein Gehäuse zocken kann, spart spürbar (gute 2/3 vom Gesamtpreis). Der Kleinkram wie Sockel, Widerstände, Kondensatoren, TTL-IC's und die unvermeidlichen Buchsen und Stecker vereinen den Rest auf sich. Wer keinen Fundus hat, wird für die 8-bit Lösung ca. DM 100.-- an Materialkosten locker machen müssen. der Zeitaufwand für den Zusammenbau des Gerätes liegt bei etwa 20 Std. Die Ehrenrunden für die messtechnische Erforschung einiger Verdrahtungsfehler einmal nicht mitgerechnet. Die meiste Zeit geht in das Verdrahten der Bauteile. Ich habe mich hier einem Gemisch aus Drahtverbindungen

und Fädeltchnik bedient. Der Analogteil ist normal verdrahtet (Analogschaltungen sind zum fädeln ungeeignet), die digitalen Teile sind gefädelt, da die Leitungen anders nicht unterzubringen sind. Als Platine habe ich eine Version mit Squarepads auf der Unterseite und einer Massefläche auf der Oberseite verwendet. Diese Kupferfläche ist wichtig, garantiert sie doch eine saubere Masse (man hat sie immer genau dort wo sie auch benötigt wird!). Hersteller dieser Platinen ist die Fa. Bicc-Vero. Sie kostet mit ca. DM 16.-- gutes Geld, aber ich kann sie nur empfehlen, die Investition lohnt sich.

... mit der Elektronik

Damit ist es nun soweit, ich nähere mich der Schaltungsbeschreibung. Den Vorstoß wagt als erstes der Befehlsdekoder. Er besteht aus dem 74LS245 als Buffer und Sabotageschutz. Sollte nämlich jemand einmal einen Kurzschluß auf den Digitalleitungen zustande bringen, erwischt es nur dieses IC und nicht den Drucker-Port im Rechner. Daran anschließend folgt ein 74LS154, als Befehlsdekoder. Er macht nichts anderes, als die oberen 4 Bit (D7...D4) einer 4-zu-16 Umwandlung zu unterziehen und mit der richtigen Ausgangsleitung dann die zugehörige Aktion zu starten. Diese sind den Sample and Hold zu steuern, den A/D-Wandler zu starten, das Schieberegister mit dem Ergebnis der Wandlung zu laden und den Wert unter Anleitung der CPU seriell auszulesen. Außerdem wird noch ein Monoflop zum Erzeugen eines Schreiblocks für den 5-bit Datenbus getriggert. Die übrigen Ausgänge werden nicht benutzt.

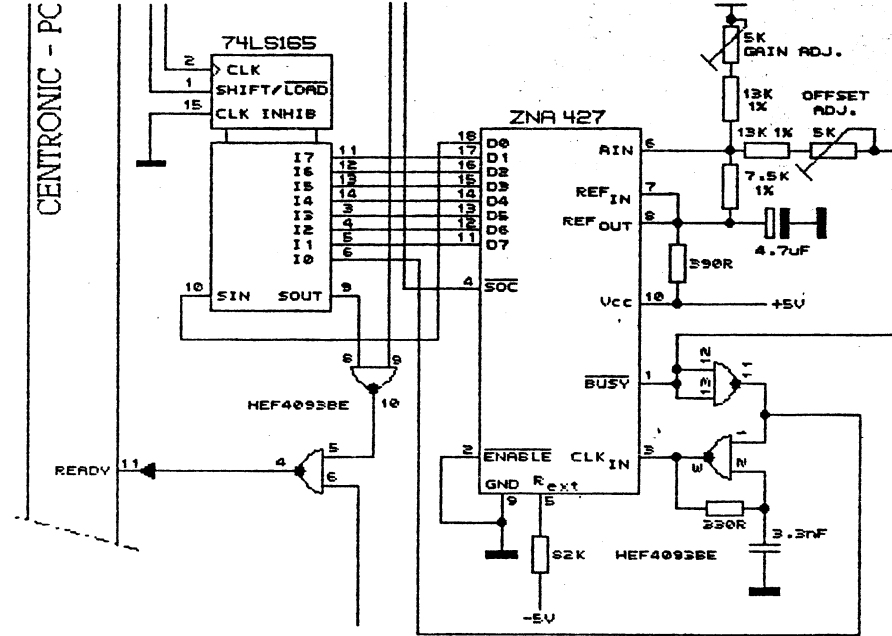


Wem es noch nicht aufgefallen ist, der sei schon hier darauf gestoßen, es wird so ziemlich alles zu Fuß, d.h. durch die CPU per Programm erledigt. Eine Ablaufsteuerung, die alles automatisch managed, ist wegen des unnötigen Aufwandes (=Geld) nicht vorgesehen. Dafür hat aber der Programmierer auf alles Einfluß und kann sich seine Ablaufsteuerung ausdenken.

Die Leitung Q15 ist die BUSY-Anzeige des Interface, d.h. solange dieser Ausgang HIGH ist (=Interface angesprochen), leuchtet die LED. Es läßt sich so leichter erfassen, wann und ob überhaupt vom Rechner auf das Interface zugegriffen wird. Eine besondere Stellung hat D4. Diese Leitung geht gleichzeitig mit D3...D0 auf den externen Datenbus. D4 ist von mir dazu ausersehen worden, als

Statusleitung zu dienen. Ist sie HIGH (=1), so liegt ein 4-bit Kommando an, mit dem die externe Hardware gesteuert werden kann. Ist es LOW (=0), so übertragen D3...D0 lediglich Daten, die verarbeitet werden sollen. Diese Vorgehensweise ist mit dem OP-Code der Z80 CPU vergleichbar, der auch aus einem Befehls- und einem Operandenteil besteht.

Der Dreh- und Angelpunkt der ganzen Sache kommt jetzt als A/D-Wandler mit Schieberegister. Wie ich schon weiter oben angedeutet habe, hat nicht jeder Centronic-Port acht Eingangsleitungen. Um diesem Problem abzuwehren, benutze ich die BUSY-Leitung zum Einlesen der Daten, denn die gibt es immer. Aus dem Grund auch das Schieberegister. Eine Baudrate gibt es nicht, die Auslesegeschwindigkeit ergibt sich unmittelbar aus der Arbeitsgeschwindigkeit der CPU bzw. aus der Schnelligkeit des Programms.

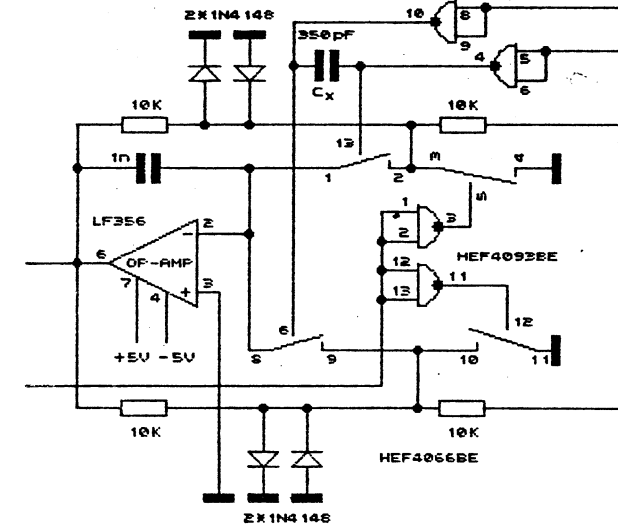


Der Pin 'SOUT' am Schieberegister wird schlauer Weise über ein UND-Gatter auf die READY-Leitung geführt. So ist es problemlos möglich, bei Bedarf von Außen auf den READY-Pin ein Signal zu schalten. Damit aber in jedem Fall das externe Signal gelesen werden kann, wird über ein zusätzliches Gatter eine eventuelle '1' am Ausgang 'SOUT' überschreibbar gehalten.

Auffallend ist auch der Poti-Kneuel am Eingang 'AIN' des ZNA427. Diese Lösung stammt von FERRANTI und ist dem Datenblatt entnommen. Die Dimensionierung ist so gewählt, daß sich ein Eingangsspannungsbereich von +1.25V ergibt. Also ziemlich genau 10mV/Digit. Ich würde aber empfehlen, die Eingangsverstärker vor dem Sample-and-Hold in der Verstärkung so einzustellen, daß sich eine Empfindlichkeit von 1mV/Digit ergibt. Abschwächen ist einfacher bewerkstelligt, wie zu verstärken. Die Berechnung für den Fußpunktwiderstand im Gegenkopplungszweig erfolgt nach der Gleichung:

$R_1 = R_2 / (V_u - 1)$. Für 1mV/Digit ist eine Verstärkung von 10fach notwendig, R_1 also: $10K / (10 - 1) = 1.11K$. Der Eingangsspannungsbereich ist dann +0.125V. Mehr gibt es zu diesem Schaltungsteil nicht zu sagen, außer noch folgendes: Da das benutzte Schieberegister nur 8 Bit breit ist, ich aber 9 Bit verarbeiten muß, ist das Bit 0 vom Wandler auf den 'SIN'-Eingang des 74LS165 gelegt und das BUSY-Signal des Wandler auf den Eingang 10 des Schieberegisters! Durch diesen Trick ist es möglich, per Programm festzustellen, wann der ZNA427 fertig ist und muß nicht blind warten, in der Hoffnung, daß er in dieser Zeit zum Ende kommt. Damit gewinnt aber die Vorgehensweise beim Heraustakten der Bits an Bedeutung. Man muß jetzt nämlich erst Takten und dann an 'SOUT' nach dem logischen Pegel sehen. Sonst wird das BUSY-Signal unversehens zum MSB. Um das BUSY-Signal des Wandlers lesen zu können, genügt es, die SHIFT/LOAD-Leitung am Schieberegister zu takten. Der Zustand von 'SOUT' ist nach der Übernahme in das Register immer gleich '10'.

Kommen wir zum Sample-and-Hold. Die Schaltung sieht etwas verworren aus. Das liegt aber nicht etwa daran, daß ich hier den letzten Schaltungstrick ausgegraben und zu Papier gebracht habe (Stichwort 'ELEKTOR'), sondern an dem leidigen Problem, mit möglichst wenig Bauteilen ein Maximum an Funktion in die Stufe zu packen. Dazu kommt, daß die Analogschalter (HEF4066BE) mit TTL-Pegeln angesteuert werden, diese selbst aber mit Spannungspegel von maximal +15V konfrontiert werden. Bei einer Versorgungsspannung von nur +5V würde der erlaubte Bereich von -0.3V und +5.3V mehr als nur überschritten werden. Damit dies aber nicht geschieht, habe ich die Analogschalter in den virtuellen Nullpunkt des OP's gelegt und mit jeweils zwei Dioden die Spannungshöhe am Teilerausgang auf < +0.7V bei offenem Gegenkopplungspfad begrenzt. Damit liege ich zwar nicht ganz in der Spec, aber wer will kann die Silizium-Dioden ja

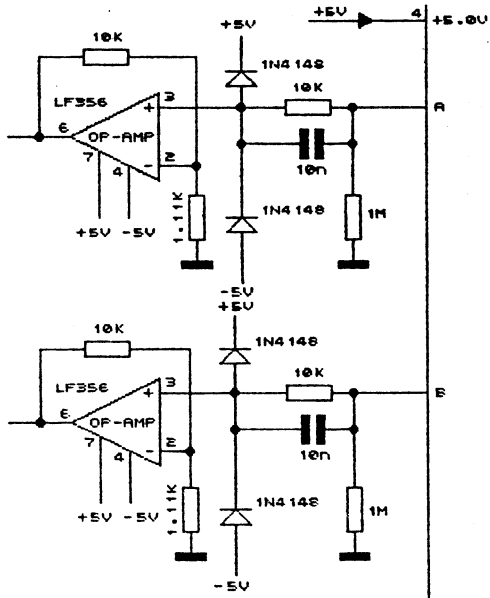


durch Germanium-Typen ersetzen. Bei mir sind diese Dinger nicht mehr allzu stark verbreitet und damit nicht zum Einsatz gekommen. Außerdem dient mir der S&H gleichzeitig als Analogmultiplexer, deshalb auch die zwei Spannungsteiler. Welcher 'Kanal' abgetastet wird, hängt davon ab, welchen Zweig ich takte. Eine Besonderheit ist der Kondensator Cx. Er verringert das Übersprechen beim Ein- und Ausschalten der Kanäle untereinander. Ohne ihn würde die Anzeige von z.B. Kanal A sich geringfügig mit dem Signal am Kanal B verändern (umgekehrt natürlich etc.). Der Wert von 350pF ist experimentel ermittelt und kann sich u. U. in

der Spezifikation abweichen. Bei mir sind diese Dinger nicht mehr allzu stark verbreitet und damit nicht zum Einsatz gekommen. Außerdem dient mir der S&H gleichzeitig als Analogmultiplexer, deshalb auch die zwei Spannungsteiler. Welcher 'Kanal' abgetastet wird, hängt davon ab, welchen Zweig ich takte. Eine Besonderheit ist der Kondensator Cx. Er verringert das Übersprechen beim Ein- und Ausschalten der Kanäle untereinander. Ohne ihn würde die Anzeige von z.B. Kanal A sich geringfügig mit dem Signal am Kanal B verändern (umgekehrt natürlich etc.). Der Wert von 350pF ist experimentel ermittelt und kann sich u. U. in

einem anderen Aufbau etwas ändern. Insbesondere, wenn IC's anderer Hersteller eingesetzt werden wie in meinem Aufbau. Die Größe des Haltekapazitors (1nF) hat ebenfalls einen Einfluß auf die gegenseitige Beeinflußung. Je größer er dimensioniert wird, umso geringer ist sie. Gleichzeitig steigt aber auch die Einschwingzeit des S&H an. Also ein echtes Kompromiss-Problem. So wie die Sache gewählt ist, benötigt der S&H etwa 70µsec. um auf 0.1% und 100µsec., um auf 0.01% genau auf den Meßwert einzuschwingen. Bei diesem Tempo ist Sound-Sampling bestenfalls mit Telefonqualität möglich. Für Geschwindigkeitsfanatiker sicher nichts, aber für mehr Tempo heißt es auch mehr DM's spendieren. Die Pinbelegung der Analogschalter sollte unverändert bleiben, da sie so gewählt ist, daß sich ein Minimum an Störungen durch die Digitalsignale ergibt.

Kommen wir zu den Eingangsverstärkern.



lauschiges Plätzchen auf der Hauptplatte suchen. Im Übrigen ist mit Stützkondensatoren (~100nF) an den Versorgungspins der einzelnen IC's nicht zu sparen. Eine saubere Versorgung ist bei den meisten Schaltungen der halbe Weg zu Sieg! Man betrachte hierzu auch den Verdrahtungsplan der Stromversorgungsanschlüsse für die IC's. Der Analogteil ist hier über 50 Ohm Widerstände und einem 47µF Elko entkoppelt.

Die Software. Ohne diese so gefragte Substanz geht auch hier nichts. Da ich z. Zt. nur ein Programm für meinen Atari habe, gebe ich dieses zum Besten (Es spielt Digitalvoltmeter). Allerdings ist es mit dem Komfort nicht weit her. Es war und ist auch nur als Demo bzw. Testkrücke für die Inbetriebnahme meines Analoginterfaces auf die Welt gekommen. Hier also nun erst einmal das Programmchen (im Anschluß gehts weiter !):

Von diesen gibt es wegen der zwei Kanäle entsprechend zwei. Die Beschaltung birgt keine besonderen Geheimnisse. Damit nicht gleich beim ersten Meßausflug der große Manitou Rauchsignale empfängt, sind die Eingänge durch eine Schutzschaltung gegen Überspannung geschützt. Koppelkondensatoren zum Abtrennen von Gleichspannungsanteilen sind nicht eingezeichnet, da die sicherlich jeder nach eigenem Gusto einbauen will, speziell direkt am zugehörigen Umschalter.

Das Thema Netzteil ist besonders schnell abgehandelt. Es besteht nur aus zwei 5V-Reglern. Die 2.5V-Referenz ist dort nicht so gut aufgehoben und sollte sich ein

' Allgemeine Adress- und Datenvereinbarungen

```

OPTION BASE 0
DIM mittel_a%(90),mittel_b%(90)
ARRAYFILL mittel_a%(0),0 !Erledige einige Vorbereitungen ...
ARRAYFILL mittel_b%(0),0 !Feld 0 von mittel_wert%() = Array-Pointer ...
mittel_a%(0)=1 !ins Feld !!!
mittel_b%(0)=1
summen_wert%=2
summe_a%=0
summe_b%=0
offset_a%=0
offset_b%=0
ctrl_register:=15 !Steuer-Register fürs Programm:
! Bit-No.: 7 6 5 4 3 2 1 0
! : : : : : : : :
! : : : : : : : :
! : : : : : : : : Kanal A (1=aktiv)
! : : : : : : : : Kanal B (1=aktiv)
! : : : : : : : : Mittelwert-Bildung Cha. A (1=aktiv)
! : : : : : : : : Mittelwert-Bildung Cha. B (1=aktiv)
! : : : : : : : :
! : : : : : : : : Frei
! : : : : : : : : Frei
! : : : : : : : : Frei
! : : : : : : : : Frei
! : : : : : : : : Frei
! : : : : : : : : Frei
! : : : : : : : : Frei

port_b:=&H8F !Port-B = Centronic-Port (Atari-Spezifisch)
mfpx:=&HFFFA01 !Adresse des BUSY-Signal (Atari-Spezifisch)
busy:=1 !Bit-Maske fürs BUSY-Bit (Atari-Spezifisch)
adc%=0 !Empfangsregister für ADC-Wert
k=0.128/128 !128mV/128 Digit = 1mV/Digit

' Die Interface-Kommandos
-----
nzd:=&H0 !... nicht definiert
clk:=&H10 !Schiebe Register ein Bit nach links
lds:=&H20 !Lade Schieberegister
ads:=&H30 !A/D-Wandlung starten
x01:=&H40 !Verstärkung auf x1 stellen
x10:=&H50 !Verstärkung auf x10 stellen
skb:=&H60 !Kanal B sampeln
ska:=&H70 !Kanal A sampeln
nzd:=&H80 !... nicht definiert
dah:=&H90 !Schreibe Kommando-Nibble ins Interface
dal:=&HA0 !Schreibe Daten-Nibble ins Interface
nzd:=&HB0 !... nicht definiert
nzd:=&HC0 !... nicht definiert
nzd:=&HD0 !... nicht definiert
sum:=&HE0 !Sample Kanal A u. B als Summe
off:=&HF0 !Letztes Kommando löschen und Access-LED aus
-----

' Hauptprogramm
-----
CLS
rahmen !Zeichne einige Krinkel ...
PRINT AT(10,10);"Offset-Korrektur : OFF";
PRINT AT(10,11);"Mittelwertbildung: ON ";
PRINT AT(10,12);"Mittelwertfaktor : ";summen_wert%;
DO
taste$=INKEY$
IF taste$(">") THEN
IF taste$="a" THEN

```

```

ctrl_register:=BCLR(ctrl_register!,0)
PRINT AT(33,5);" ###.###";
ELSE IF taste$="A"
ctrl_register:=BSET(ctrl_register!,0)
ELSE IF taste$="b"
ctrl_register:=BCLR(ctrl_register!,1)
PRINT AT(33,7);" ###.###";
ELSE IF taste$="B"
ctrl_register:=BSET(ctrl_register!,1)
ELSE IF taste$="0"
PRINT AT(10,10);"Offset-Korrektur :   ON ";
set_zero
ELSE IF taste$="o"
PRINT AT(10,10);"Offset-Korrektur :   OFF";
offset_a%=0
offset_b%=0
ELSE IF taste$>="0" AND taste$<="9"
IF taste$="0" THEN
ADD summen_wert%,10
IF summen_wert%>90 THEN
summen_wert%=90
ENDIF
ELSE
summen_wert%=VAL(taste$)
ENDIF
PRINT AT(10,12);"Mittelwertfaktor :   ";summen_wert%;" ";
ARRAYFILL mittel_a%(0),0
ARRAYFILL mittel_b%(0),0
mittel_a%(0)=1
mittel_b%(0)=1
summe_a%=0
summe_b%=0
IF summen_wert%=1 THEN
PRINT AT(10,11);"Mittelwertbildung:   OFF";
ELSE
PRINT AT(10,11);"Mittelwertbildung:   ON ";
ENDIF
ENDIF
ENDIF
IF BTST(ctrl_register!,0)=TRUE THEN
rd_adc(ska!,off!) !Wecke den ADC aus seinem Schlaf
IF BTST(ctrl_register!,2)=TRUE THEN
mittel_adc%+offset_a%,summe_a%,mittel_a%()
ENDIF
PRINT AT(25,5);USING "Kanal A: ###.###V",CFLOAT(adc%)*k
ENDIF
IF BTST(ctrl_register!,1)=TRUE THEN
rd_adc(skb!,off!) !dto. für Kanal B
IF BTST(ctrl_register!,3)=TRUE THEN
mittel_adc%+offset_b%,summe_b%,mittel_b%()
ENDIF
PRINT AT(25,7);USING "Kanal B: ###.###V",CFLOAT(adc%)*k;
ENDIF
EXIT IF taste$="x" !Prüfe, ob's reicht ...
LOOP
EDIT
' -----
' Die Unterprogramm-Bibliothek
' -----
PROCEDURE wr_intf(cmd!) !Schicke ein Kommando ans Interface
~XBIOS(28,W:cmd!,W:port_b!)

```

```

RETURN
PROCEDURE rd_intf !Schieberegister auslesen
LOCAL I%
adc%=0
FOR I%=1 TO 8
~XBIOS(28,W:clk!,W:port_b!) !Einmal schieben wegen EOC-Bit
~XBIOS(28,W:off!,W:port_b!)
adc%=SHL(adc%,1) OR (PEEK(mfp%) AND busy!) !Bit abholen
NEXT I%
RETURN
PROCEDURE rd_adc(cmd!,off_mode!) !Kanal A/B abtasten u. auslesen
LOCAL I%
adc%=0
~XBIOS(28,W:cmd!,W:port_b!) !Sample Kanal A/B
~XBIOS(28,W:ads!,W:port_b!) !ADC starten
REPEAT
~XBIOS(28,W:lds!,W:port_b!) !ADC Status holen
UNTIL (PEEK(mfp%) AND busy!)=0 !ADC fertig ?
~XBIOS(28,W:lds!,W:port_b!) !ADC-Wert laden
FOR I%=1 TO 16
adc%=SHL(adc%,1) OR (PEEK(mfp%) AND busy!)
~XBIOS(28,W:clk!,W:port_b!)
~XBIOS(28,W:off_mode!,W:port_b!)
NEXT I%
adc%=(adc%&H80)
RETURN
PROCEDURE set_zero !Den Offset-Abgleich per Programm
rd_adc(ska!,off!) !Bestimme Offset von Kanal A
offset_a%=-adc%
rd_adc(skb!,off!) !dto. für Kanal B
offset_b%=-adc%
adc%=0
RETURN
PROCEDURE rahmen !Zeichne einen Fensterrahmen
BOX 259,60,322,83 !Box um Kanal A
BOX 185,60,259,83 !Box um "Kanal A:"
LINE 187,87,320,87 !Trennlinie zwischen A & B
LINE 187,88,320,88 !dto.
BOX 259,92,322,115 !Box um Kanal B
BOX 185,92,259,115 !Box um "Kanal B:"
BOX 180,55,327,120 !Box um Alles
RETURN
PROCEDURE mittel(adc_wert%,VAR summe%,mittel_wert%())
IF summen_wert%>1 THEN
SUB summe%,mittel_wert%(mittel_wert%())
ADD summe%,adc_wert%
mittel_wert%(mittel_wert%())=adc_wert%
INC mittel_wert%()
IF mittel_wert%()>summen_wert% THEN
mittel_wert%()=1
ENDIF
adc%=summe%/summen_wert%
ELSE
adc%=adc_wert%
ENDIF
RETURN

```


Da das Programm nicht unbedingt wie 'gewöhnliches' BASIC aussieht, einige erklärende Worte dazu. Zuerst zu dem was es so macht. Nach dem Start wird per Grafik ein Rahmen gezeichnet, in den die ins Dezimale umgerechneten Analogwerte geschrieben werden (siehe Bild). Soll ein Kanal nicht angezeigt werden, so ist über die Tastatur ein kleines "a" bzw. "b" einzugeben. In der entsprechende Anzeige erscheinen dann Nummernzeichen. Unterhalb des Rahmen wird ausgewiesen, ob ein Offsetabgleich durchgeführt wurde (Großes "O" für Abgleich, kleines "o" für Abgleich

Kanal A:	-1.921V
Kanal B:	0.024V

Offset-Korrektur : OFF
 Mittelwertbildung: ON
 Mittelwertfaktor : 2

löschen). In den beiden Zeilen darunter ist nachzulesen, ob und wenn ja, über wie viele Messwerte gemittelt wird. Bei einer "1" wird die Mittelwertbildung abgeschaltet. Es können Werte in Einerschritten 1,2,3,...9,11,12,13.. bis 90 eingestellt werden. Für die Zehnerschritte dient die "0", deshalb sind

Mittelwertfaktoren von 10,20,30... nicht möglich. Aber wen stört das? Verantwortlich für die Mittelwertbildung ist die Procedure "mittel". Das Einlesen der Messwerte erfolgt über die Procedure "rd_adc" im Hauptprogramm. Ein Wort noch zu den Zeilen "XBIOS(...)". Hier wird eine Routine im Betriebssystem aufgerufen, mit der ich den Centronic-Port beschreibe. Hier kann/muß abhängig vom Rechner auch ein 'OUT port_b%,cmd%' stehen. Die Wirkung ist Gleich (hoffentlich !). Beim Tandy-Basic würde die Routine 'rd_adc' in etwa so aussehen:

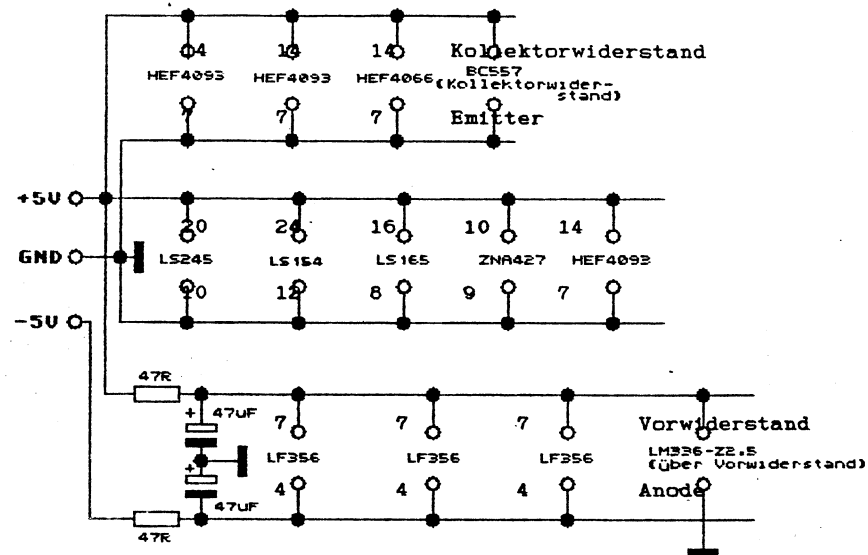
```

1 PROCEDURE rd_adc(cmd%, off%)           !Kanal A/B abtasten
2   adc%=0
3   OUT port_b%, cmd%                   !Sample Kanal A/B
4   OUT port_b%, ads%                   !ADC starten
5   OUT port_b%, lds%                   !ADC Status holen
6   IF NOT((INP(mfp%) AND busy%)=0) GOTO 6 !ADC fertig ?
7   OUT port_b%, lds%                   !ADC-Wert laden
8   FOR i%=1 TO 8
9     OUT port_b%, clk%                 !Schieberegister auslesen
10    OUT port_b%, off%                 !Erst schieben...
11  END FOR
12  adc%=(adc%+adc%) OR (INP(mfp%) AND busy%) !...dann lesen
13 NEXT i%
14 adc%=(adc%+adc%)/2                   !ADC-Wert korregieren
15 RETURN
  
```

Zum Schluß noch einen Tip zur Mittelwertbildung. Diese Routine summiert nicht erst X_n -Werte auf und gibt das Ergebnis dann aus, sondern hier wird ein gleitender Mittelwert gebildet. Ich meine damit, daß bei Anlegen einer Spannung die Anzeige mit jedem Abtastschritt eine Ausgabe liefert und sich mit fortschreitender Anzahl an Abtastungen dem Endwert annähert (Die zu messende Spannung ändert sich währenddessen nicht !!!). Schwankt die zu messende Größe, so erfolgt je nach Höhe des eingestellten Mittelwertfaktors, eine mehr oder weniger starke Integration der Schwankungen auf einen mittleren Wert (ähnlich verhält sich ein

Tiefpaß)

Als letztes noch der Anschlußplan der einzelnen IC's an die 5V bzw. gefilterten 5V:

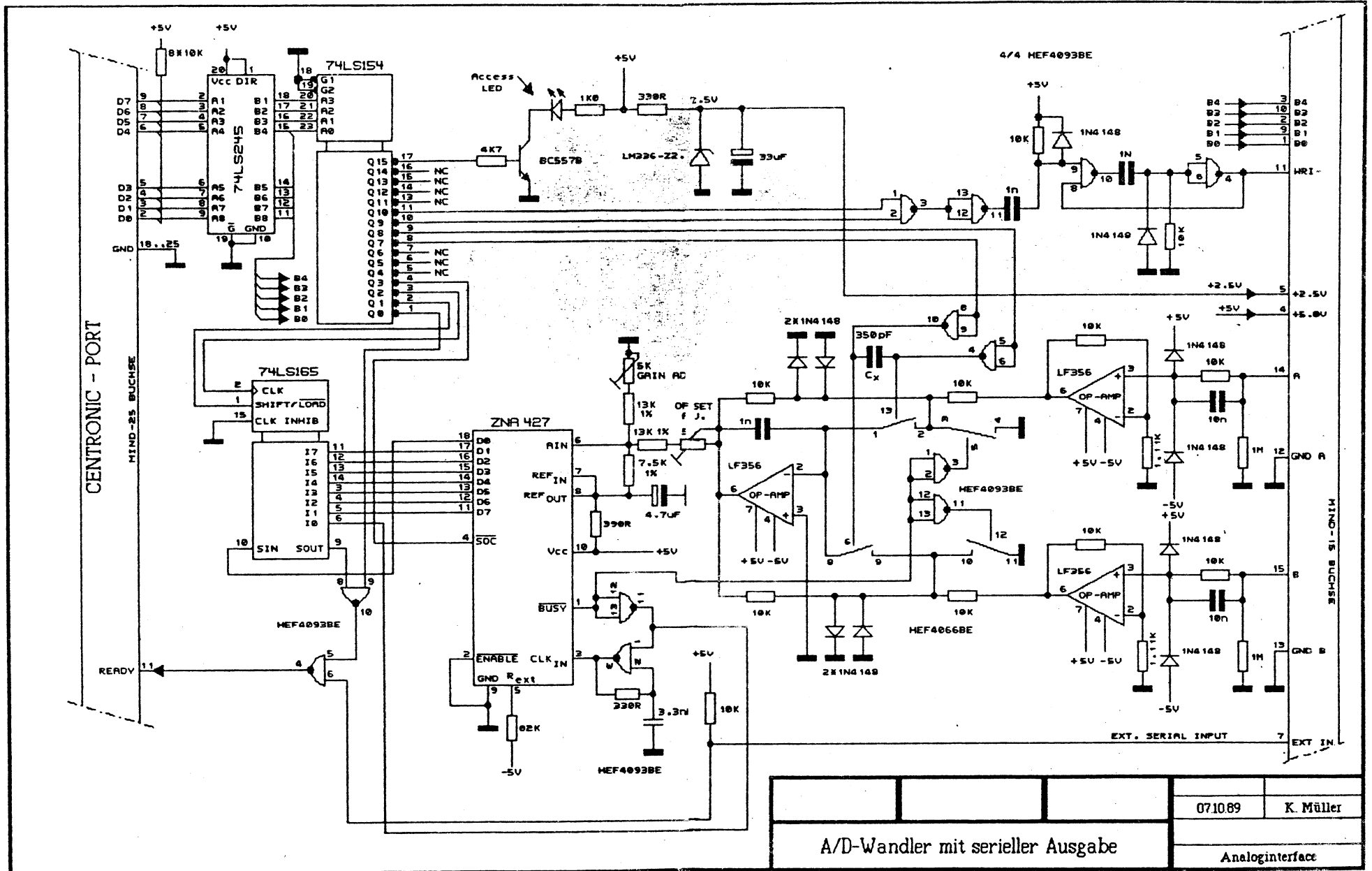


HEFT
 29
 Januar
 1990

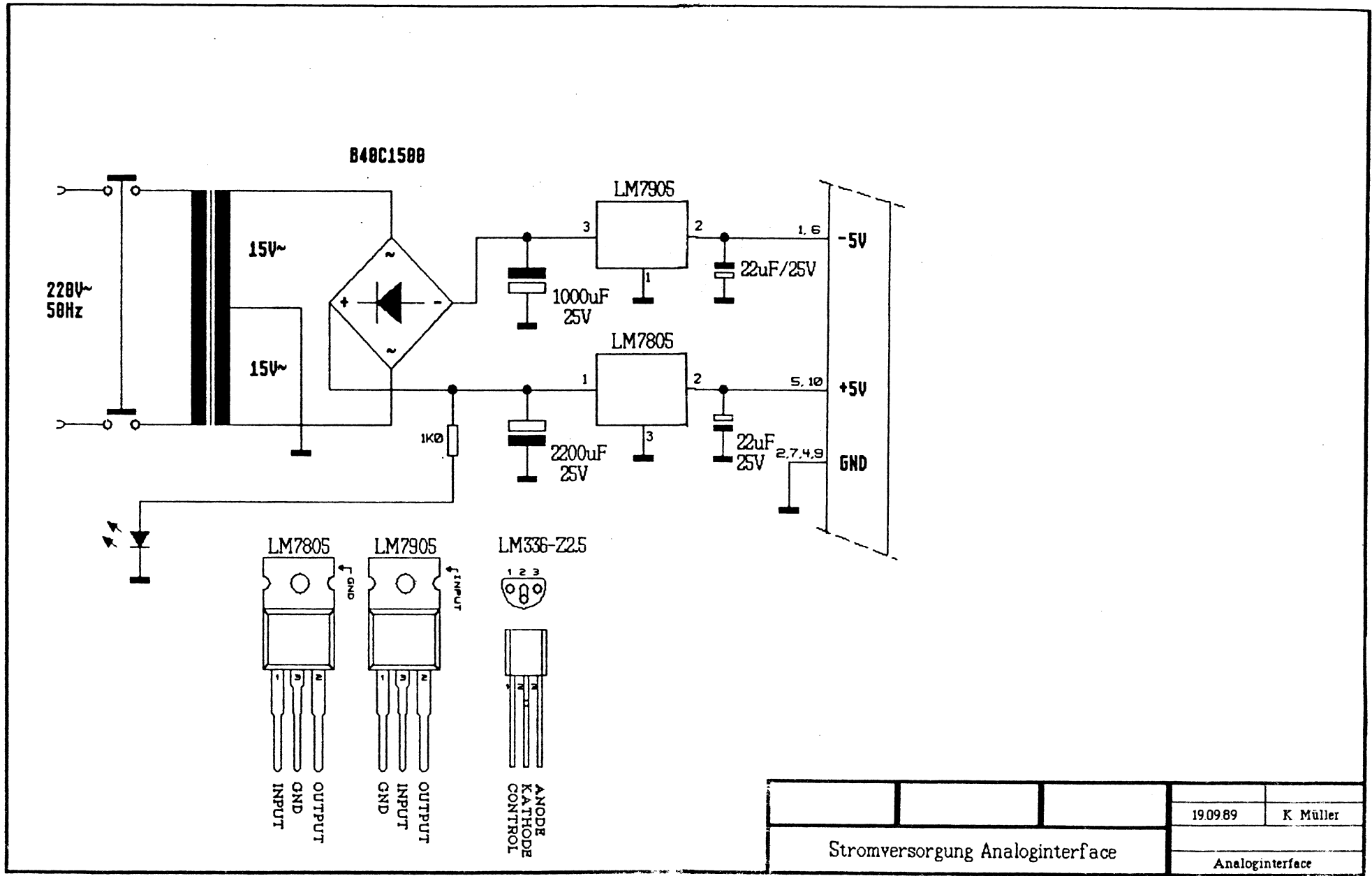
Ich glaube, nun ist es genug.

Tschuß

Handwritten signature



		07.10.89	K. Müller
A/D-Wandler mit serieller Ausgabe			Analoginterface



HEFT
29
Januar
1990

Analog-Input:

- 1 - D0
- 2 - D2
- 3 - D4 (Command)
- 4 - +5V
- 5 - +2.5V
- 6 - NICHT BELEGT
- 7 - Ext. Serial Input
- 8 - existiert nicht
- 9 - D1
- 10 - D3
- 11 - WRI-
- 12 - GND A
- 13 - GND B
- 14 - A+
- 15 - B+
- 16 - existiert nicht

Centronic-Input:

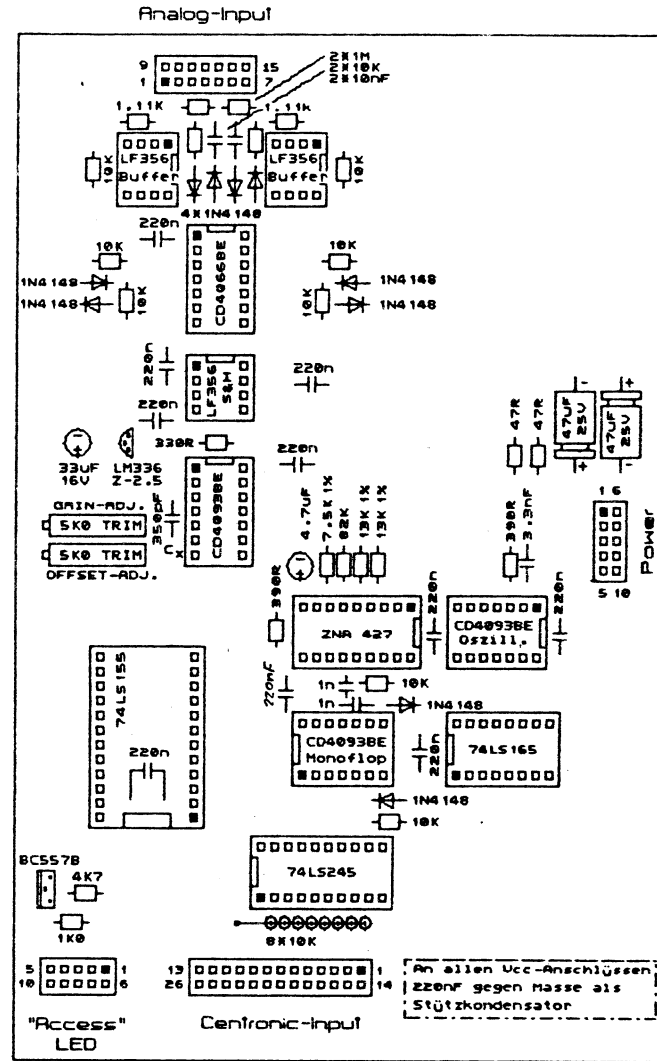
- 1 - Strobe (nicht benutzt)
- 2 - D0
- 3 - D1
- 4 - D2
- 5 - D3
- 6 - D4
- 7 - D5
- 8 - D6
- 9 - D7
- 10 - NICHT BELEGT
- 11 - READY
- 12-17 - NICHT BELEGT
- 18-25 - GND

Access-LED:

- 1,6 - GND
- 2,3,7,8 - nicht belegt
- 5,10 - Anode
- 4,9 - Kathode

Power:

- 1,6 - -5V
- 2,7 - GND
- 3,8 - NICHT BELEGT
- 4,9 - GND
- 5,10 - +5V

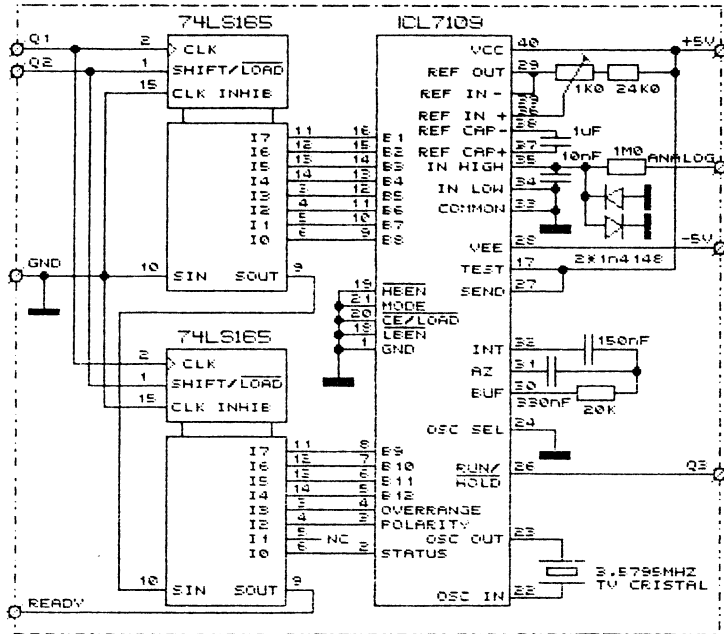


			19.09.89	K. Müller
Steckerbelegung A/D-Wandler			Analoginterface	

			19.09.89	K. Müller
Bestückungsplan A/D-Wandler			Analoginterface	

Analog mit drei IC's

Nachdem ich bereits meine "große" Version des Analog-Interface an Jens abgeschickt hatte, erhielt ich vom Helmut einen Brief, in dem er mich fragte, wie statt des von mir eingesetzten ADC's sein bevorzugter Wandler (ICL7109) zu benutzen sei. Nach einigen Hin- und Her mit dem Datenblatt (dieser Maikäfer hatte einfach ein paar Anschlüsse zu viel) kam ich dann auf die vorliegende Lösung. Mit dem Messen muß man sich jedoch etwas gedulden, schafft das gute Stück doch nur 7.5 Wandlungen pro Sekunde. Die Ansteuerung erfolgt wie gehabt über den Centronic-Port, ausgelesen wird das Ergebnis über die BUSY-Leitung. Da alle sonstigen Dinge vom ICL7109 selbst erledigt werden - nachdem ihm über die RUN/NHOLD-Leitung der Startschuß gegeben wurde - beschränkt sich die Hauptaktivität der CPU auf trivialitäten: Starten... auslesen... vermarkten... starten...



Die Belegung der Leitungen am Centronic-Port mit den Anschlüssen der Schaltung (mit Ausnahme eben von READY oder auch BUSY) kann sich im Grunde jeder selbst zurechtlegen. Mit welchen Ausgabewerten dann welche Funktion ausgelöst wird, darf sich in dem Fall dann auch selbst ausgedacht werden. Zum Erhalt der Einheit (nicht der deutschen, soweit sind wir noch nicht !) mache ich aber einen Vorschlag:

Centronic:	Wandler:
D0 (2)	Q1 (CLK)
D1 (3)	Q2 (SHIFT/NLOAD)
D2 (4)	Q3 (RUN/NHOLD)
D3 (5)	n.b.
D4 (6)	n.b.
D5 (7)	n.b.
D6 (8)	n.b.
D7 (9)	n.b.
BUSY (11)	READY
GND (19..29)	GND

Die Stromversorgung von +5V ist separat, das Analogsignal zwischen den Punkten "ANALOG" und "GND" bzw. Pin "COMMON" am ICL7109 anzuschließen. Die Empfindlichkeit für Vollausschlag beträgt +0.2V. Der genaue Anzeigewert kann mittels des 1K Trimpoti eingestellt werden. Hierzu sind natürlich 200mV zuvor anzulegen !

Zum Schluß noch ein paar Worte zur programmtechnischen Seite. Damit die Sache in Ruhestellung gehen kann, ist der Hex-Wert 03 auszugeben: CLK=1, SHIFT/NLOAD=1 u. RUN/NHOLD=0. Von diesem Wert gehen im Weiteren dann alle Aktionen aus. Die Befehlssequenz zum Auslesen des Schieberegisters hat folgendes Aussehen:

02H	CLK = LOW setzen
03H	CLK = HIGH und 1 Bit nach links schieben
	BUSY-Zustand abfragen
02H	CLK = LOW setzen
03H	CLK = HIGH und 1 Bit nach links schieben
	BUSY-Zustand abfragen

Und das insgesamt 16 mal oder innerhalb einer Schleife. Die letzte "Anweisung" ist immer 03H, es sei denn es folgen andere Kommandos. Wenn man's nicht macht, ist das übrigens auch nicht tragisch, aber so hat die Sache immer einen definierten Endzustand. Zum Starten der Wandlung gilt die Sequenz:

07H	RUN/NHOLD = HIGH, Umsetzung starten
03H	Wandler Stop, sobald Umsetzung beendet

Wie nun das Schieberegister geladen wird, ist klar. Der Vollständigkeit wegen, aber auch das noch:

01H	SHIFT/NLOAD = 0, Register laden
03H	Ruhezustand

oder ...

01H	SHIFT/NLOAD = 0, Register laden
02H	CLK = LOW setzen, SHIFT = TRUE
03H	CLK = HIGH und 1 Bit nach links schieben
	BUSY-Zustand abfragen

Die Bedeutung der Bit's im ausgelesenen Registerinhalt ist wie folgt:

55

Bit 0..11 12bit Analogwert, binär kodiert
Bit 12 Überlauf (1 = Überlauf)
Bit 13 Polarität (1 = positiv, 0 = negativ)
Bit 14 nicht benutzt
Bit 15 Status (1 = ADC busy)

Laut Datenblatt genügt es, RUN/NHOLD für eine halbe Taktperiode auf HIGH zu halten, damit der Wandler korrekt losläuft. Die Integration des Meßsignals beginnt dann 7 Taktperioden später. Zeitlich bedeutet das $7 * (1/3.5795\text{MHz}) \approx 2\text{usec}$. Erst jetzt wechselt das Status-Signal auf HIGH. Man sollte deshalb besser auf diesen Wechsel warten und dann erst RUN/NHOLD auf LOW-Pegel zurücksetzen. BASIC-Freaks werden jetzt zwar abwinken - der Interpreter schafft es wohl eher mit knapper Not mit den 7.5 Wandlungen mitzuhalten - aber in Assembler und mit entsprechender Taktfrequenz könnte man durchaus schneller sein. Die Daten sind gültig, sobald der STATUS-Ausgang auf LOW wechselt. Soll kontinuierlich gewandelt werden, genügt es, RUN/NHOLD ständig auf HIGH zu belassen. Gültige Daten können jedoch nur gelesen werden, solange STATUS auf LOW steht !

Eins sei an dieser Stelle und zum Schluß noch gesagt: Ich habe die Schaltung selbst noch nicht aufbauen können. Bis auf die Schieberegister ist alles, was um das ICL7109 herum an passiven Bauteilen zu sehen ist, dem Datenblatt entnommen. Die Bauteildimensionierung ist aber mit Sicherheit korrekt, da das IC so für die spezifizierete Funktion beschaltet werden muß. Mir erschien die Einfachheit der Lösung aber so gut, daß ich ausnahmsweise einmal ohne einen vorherigen Schaltungsaufbau die Sache kundtue. Sollte sich nichts tun, erst einmal weglegen und mindestens eine (w)einstündige Kaffeepause einlegen, bis sich die Nerven beruhigt haben, dann alles nochmals in Ruhe kontrollieren. Wenn dann immer noch nichts geht, langsam anfangen zu fluchen. Ich bin mir aber sicher, daß es so arg nicht kommen wird.

Tschüß

Reif

56

BÜRSE -- BÜRSE -- BÜRSE

zu verkaufen

- TRS-80 M I
- Expansion - Board
- 80-Zeichen-Karte
- HZG II
- 3 Diskettenlaufwerke
- reparaturbedürftiger Epson RX 80
- viel Software / 14 Ordner Literatur

Werner Förster
Christoph-Krebs-Str. 9
8720 Schweinfurt
☎ 09721/21841

Ein ehemaliges Mitglied des Bremerhavener Computerclubs betreibt in Oldenburg eine PC-Firma. Hartmut und ich haben den Waldemar Grundmann am 07. Oktober in Oldenburg besucht. Was dabei herauskam, könnt ihr hier lesen.

1. Eine Verkaufsanzeige der Fa. Computer-Service

2. Als kleines Dankeschön für die Veröffentlichung der Anzeige habe ich einige Bauteile kostenlos bekommen, die noch aus alten Tagen bei der Firma herumlagen. Diese Teile biete ich den Clubmitgliedern zum Abruf kostenlos an. Schon alleine der Bausatz HRG 1B hat seinerzeit mehr als 200,-DM gekostet. Die übrigen Bauteile kommen zusammen nochmals auf ca. 120,-DM.

1. IC - Sockel
(Doppel-Federkontakte)

- 24 x 14 pins
- 58 x 16 pins
- 23 x 24 pins
- 23 x 28 pins

2. IC's

- 2 x 74 LS 00
- 2 x 74 LS 04
- 1 x 74 LS 05
- 1 x 74 LS 20
- 1 x 74 LS 32
- 2 x 74 LS 74
- 1 x 74 LS 123
- 1 x 74 LS 155
- 2 x 74 LS 166
- 1 x 74 LS 174
- 2 x 74 LS 244
- 1 x 74 LS 245
- 1 x 74 LS 367
- 1 x 74 LS 368
- 1 x 74 LS 373

7. Für mich nicht definierbar:

- 3 IC's mit der Aufschrift:
F 21LO22PC
7735

13 x 8041016

3. Pfostenstecker

- 9 Stück 26polig
(ohne Zugentlastung)

4. 1 Bausatz HRG 1B

- (TRS80 Model 1 und VG I/II)
- Platine u. fast alle Bauteile
(incl. RAMs) und Bauanleitung

5. 1 Double Density Controller
(Model 1/Video Genie) zum
Einbau in Expansion-Interface
mit Bauanleitung

6. RAM - IC's

- 2 x 41256
- 4 x 4164
- 1 x 4116

Ab hier kostet es Geld:

Derselbe Waldemar Grundmann (Telefonnummer für Verkaufsgespräche siehe Anzeige) verkauft noch:

1 Tandy TRS 80 Model III für 100,- DM

1 Shuffle-Board, angeblich Speicheraufrüstung auf 64k, CP/M 2.2, incl. Software und Handbücher; Preis ?

Für VG IIs: 1 Speichererweiterungskarte (kenne ich nicht, es sind jedenfalls 24 Stück IC's vom 4864 drauf); Preis 50,-DM

Immer noch für VG IIs: 1 Schnittstellenkarte mit 2fach Z80-SIO/0 und 2fach Z80-PIO; Preis 50,-DM

Also auf Jungs! Ran ans Telefon! Wer zuerst kommt, malt zuerst! Da die Telefonnummer jetzt richtig im Info steht, verweise ich auf die Mitgliederliste. Wer sucht, der findet.

Hans-Günther, Dezember 1989
2876 Berne 2

Computer Service

Moslestr. 74 (Victoria-Haus) • 2900 Oldenburg • Tel.: 0441 / 25072
Im 1Z • Am Wollager 8 • 2870 Detmenhorst • Tel.: 04221 16303
Maiglockchenweg 5 • 2910 Westerstede • Tel.: 04488 1629

Jetzt sind sie da: CS Computer

<p style="text-align: center; font-weight: bold; margin: 0;">CS AT 286</p> <ul style="list-style-type: none"> - 12 MHz - 640 KB Ram - Festplatte 20 MB - 14" Monitor - 102 Tasten Tastatur <p style="text-align: center; font-weight: bold; margin: 0;">2425.-</p> <ul style="list-style-type: none"> - mit 16 MHz NEAT Board <p style="text-align: center; font-weight: bold; margin: 0;">2755.-</p> <ul style="list-style-type: none"> - 12 MHz mit 40 MB Festplatte <p style="text-align: center; font-weight: bold; margin: 0;">2775.-</p> <p style="font-size: x-small; margin: 0;">VGA Karte 800 x 600</p> <p style="text-align: center; font-weight: bold; margin: 0;">375.-</p>	<p style="text-align: center; font-weight: bold; margin: 0;">CS AT 386</p> <ul style="list-style-type: none"> - 20 MHz Takt - 2 MB Ram - Festplatte 40 MB - 14" Monitor - 102 Tasten Tastatur <p style="text-align: center; font-weight: bold; margin: 0;">4685.-</p> <p style="text-align: center; font-weight: bold; margin: 0;">CS AT 386 SX</p> <ul style="list-style-type: none"> - 20 MB Platte - sonst wie AT 386 <p style="text-align: center; font-weight: bold; margin: 0;">3525.-</p> <p style="font-size: x-small; margin: 0;">KHK Private Line 79.50 PD - Software je 7.50</p>
--	--

Selbstverständlich sind: 12 Monate Garantie,
FTZ-Nr. und Deutsches Handbuch

59

Ich habe Euch im ersten Teil zwei sehr unterschiedliche Computersysteme vorgestellt, auf denen aber (hoffentlich) dasselbe Programm laufen kann - nur einmal druckt es eben auf einem Laserdrucker und einmal auf einem Fernschreiber usw. Aber das ist doch nicht so selbstverständlich, oder?! Beim Super-Turbo-System sagt das Programm zum Betriebssystem: "Druck' mal ein 'a'!" und das BS druckt es auch wirklich auf dem Laserdrucker - aber was ist bei dem Fernschreiber-System?

Würde die Aufforderung lauten: "Drucke das Zeichen auf dem Laserdrucker!", hätte das Fernschreiber-System große Schwierigkeiten. Aber dem ist ja nicht so! Das Programm sagt eben nur: "Drucke das Zeichen!", ohne dabei den Drucker näher zu bezeichnen. Und somit ist die Art des Druckers egal. Aber woher weiß das BS, daß ein Fernschreiber "fast" das gleiche wie ein Laserdrucker ist - also daß beide Drucker sind?! Warum "druckt" das BS hier das Zeichen nicht auf eine Lochkarte? Das wäre doch peinlich, gelt?!

Da kommt eine Eigenschaft des Betriebssystems ins Spiel, die wir näher beleuchten müssen: die Aufteilung aller Geräte in Geräte-"Klassen". Alle Geräte einer Klasse leisten prinzipiell das gleiche, allerdings sind sie meistens unterschiedlich ausgestattet. Beispiel: Die Klasse der Drucker besteht aus Fernschreibern, Typenrad-, Matrix-, Tintenstrahl-, Laserdruckern usw. Alle drucken (in erster Linie) Zeichen auf Papier. Manchmal können Klassen auch noch weiter aufgeteilt werden, wie Ihr gleich sehen werdet. Ich zähle mal auf, was mir an Klassen so einfällt - ohne Anspruch auf Vollständigkeit.

Einfach ist noch die grobe Aufteilung in Massenspeicher und Ein-/Ausgabe-Geräte. Zwar ist diese Aufteilung nicht ganz eindeutig, denn schließlich geht's beim Speichern auch um das Ein- und Ausgeben, aber da gibt es einige Unterschiede: a) Auf Massenspeicher wird etwas geschrieben, das wir auch wieder vom gleichen Gerät lesen wollen. Das trifft auf andere Ein-/Ausgabe-Geräte normalerweise nicht zu. b) Bei Massenspeichern geben wir uns nicht mit einzelnen Zeichen ab, sondern erledigen die Ein-/Ausgabe blockweise (im CP/M "Record" genannt und 128 Bytes lang, im Newdos dagegen ein "Sektor" mit 256 Bytes).

Bei den Massenspeichern wird selten weiter unterschieden, jedenfalls von Disketten aufwärts (Kassetten sind "out", auch wenn IBM noch gerne eine Kassettenschnittstelle einbaut, was die Fortschrittlichkeit dieser Geräte unterstreicht). Einziges Unterscheidungsmerkmal bei Massenspeichern: wech-
selbare gegenüber festen Medien; typische Vertreter der beiden Gattungen: wechselbare Disketten und feste Festplatten. Daß wiederum Disketten-/Festplatten- und sonstige Laufwerke äußerst unterschiedlich groß sein können (nicht nur äußerlich, sondern auch von der Kapazität her), dürfte klar sein. Da helfen uns die Klassen nicht weiter.

Aber die Ein-/Ausgabe-Geräte lassen sich sehr schön weiter unterscheiden. Eine klassische Klasse ist die Konsole (engl. "console"), die in Urzeiten aus einer Schreibmaschine bestand, auf der ein Computer auch was drucken durfte. Getipptes ging außer auf's Papier auch an den Rechner. Heutzutage nennt sich sowas eher "Terminal" und besteht aus einer umfangreichen (gegenüber der Schreibmaschine) Tastatur und einem Bildschirm, auf dem sich manchmal Zeichen blicken lassen. Dies ist die Geräte-Klasse, über die jedes Programm normalerweise Kontakt zum User aufnimmt, von der es Eingaben erwartet und auf das es Ausgaben macht. Aber es ist Sache des Betriebssystems, was wirklich hinter der Konsolen-Ein-/Ausgabe steckt (Schreibmaschine, Terminal, Spracherkennung und -ausgabe ...).

Auch sehr klassisch ist die Klasse der Drucker. Drucker sind Teile, die Papier zeilen-, zeilen- und seitenweise verschmutzen, wobei das Papier wie eine Klorolle gegen endlos strebt, aber im falschen Moment zuende geht. Probleme bekommen wir aber bei grafikfähigen Druckern - dazu unten mehr bei den "Grafischen Ausgabe-Geräten".

60

Etabliert hat sich noch die Klasse der Kommunikations-Geräte, wie Modem und Akustikkoppler, über die Rechner Verbindung zur Außenwelt aufnehmen, wobei die Außenwelt oft auch nur aus anderen Rechnern besteht. Interessant wäre hier auch das Stichwort "Netzwerke", wo mehrere Rechner sich gegenseitig Daten zuschauen. Leider muß ich mangels Erfahrung da passen. Aber eins ist sicher: An sowas haben weder die Newdos- noch die CP/M-Entwickler gedacht.

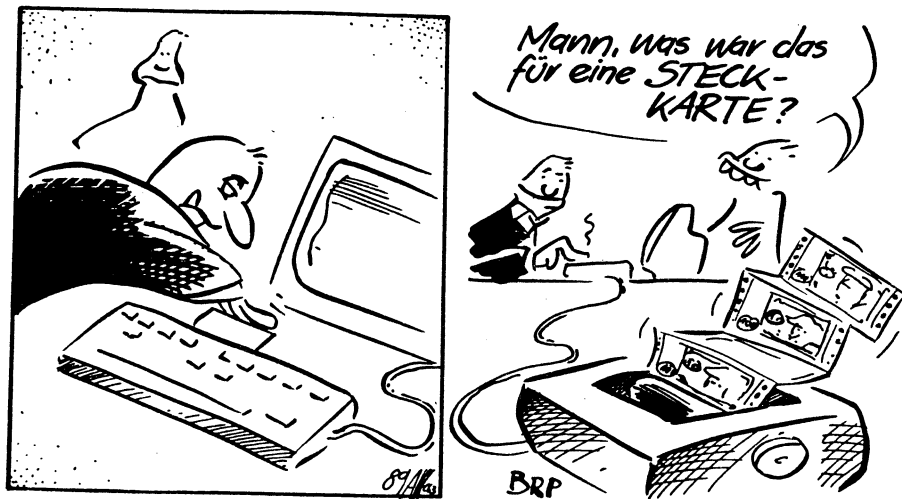
Neuerdings kommen noch zwei Geräte-Klassen hinzu: Die grafischen Ein- und Ausgabe-Geräte, als da auf der einen Seite wären: Grafik-Karten nebst Plottern und grafikfähigen Druckern, während die andere Seite von Mäusen, Trackballs, Grafik-Tablets, "Didjeteisern" und Lightpens belebt wird. Bei den Ausgabe-Geräten muß natürlich unterschieden werden zwischen "flüchtigen" (Bildschirm-) und "nicht-flüchtigen" (Drucker-) Ausgaben. Bei der Eingabe wiederum gibt es "Lokalisierer", mit denen sehr genau eine Koordinate angewählt werden kann (z.B. Digitalisier-Tisch), und "Auswähler", mit denen nur grob auf etwas (auf dem Bildschirm) gezeigt werden kann (z.B. Maus). Hier hat selbst MS-DOS noch Nachholbedarf, denn außer der (scheinbar nicht sehr erfolgreichen) Unterstützung einiger Grafikkarten wird der Rest hoheitsvoll ignoriert.

Natürlich kann ich hier etwas vergessen haben, aber die hauptsächlichsten Geräte-Klassen für Mikrocomputer sollten dabei sein. Bei größeren Rechnern kommen vielleicht noch welche dazu, bei kleineren ganz bestimmt. Warum gerade bei kleineren? Nun, da geht es oft auch um Steuerungsaufgaben, für Waschmaschinen, Walzstraßen oder Ampeln (diese Assoziationskette ist hoffentlich einleuchtend). Diese Steuerungsrechner haben also andere Probleme als unsere Mikrocomputer; dafür fehlt ihnen aber auch oft das Betriebssystem.

Erwähnen möchte ich aber noch die Schnittstellen zu den Geräten. Schnittstellen sind einerseits diese niedlichen Kleinen Buchsen und Stecker an der Rückseite des Rechner-Gehäuses, andererseits aber auch die Belegung der Kabel und die Art der elektrischen Signale. (Daneben werden auch die Verbindung vom Anwenderprogramm zum BS, die Verbindung von Programmen untereinander und viele andere Verbindungen "Schnittstelle" oder Neu-Deutsch "Interface" genannt, was zur Verwirrung beiträgt. Demnächst werden auch die schlagenden Studenten-Verbindungen in "Schnittstellen" umbenannt.)

Es gehört sich für ein Anwenderprogramm nicht, dem BS zu sagen, mit welcher Schnittstelle es arbeiten soll. Warum? Weil es schließlich egal ist, an welcher Schnittstelle welches Gerät hängt. Zwar ist die Schnittstelle zum Drucker meist parallel nach der Centronics-Norm ausgeführt und die zum Kommunikations-Gerät meist seriell nach der Norm RS-232 bzw. V.24. Aber auch andersrum oder nur mit einer Sorte geht's. Es ist Aufgabe des Betriebssystems, sich zu merken, an welcher Schnittstelle welches Gerät hängt. Das Anwenderprogramm sagt nur, welches Gerät es benutzen möchte.

Was haben wir nun von den Geräte-Klassen? Für jede Klasse von Geräten sieht das Betriebssystem besondere Aufrufe/Anfragen vor (wie schon gesehen: "Programm an Betriebssystem: Gib mal auf der Konsole (bzw. einem Gerät der Klasse "Konsole") ein 'a' aus!"). Das Programm sieht dann nur noch diese Klassen (bzw. die entsprechenden Aufrufe des Betriebssystems) und das Betriebssystem kümmert sich darum, das richtige Gerät für diese Klasse zu finden. Und das auf jedem Rechner!



Hier nochmal eine Zusammenfassung der Klassen mit typischen Vertretern (ohne Anspruch auf Vollständigkeit):

Massenspeicher	
wechselbare	(Diskette, CD-ROM)
feste	(Festplatte, RAM-Disk)
Ein-Ausgabe-Geräte	
Konsolen-Eingabe	(Tastatur)
Konsolen-Ausgabe	(Bildschirm, Drucker)
Drucker	(Laserdrucker, Fernschreiber)
Kommunikation	(Modem, Akustikkoppler)
Graphische Eingabe	
Lokalisieren	(Grafik-Tablett, Digitalisier-Tisch)
Identifizieren	(Maus, Trackball)
Graphische Ausgabe	
Flüchtige	(Grafikkarte)
Nicht-flüchtige	(Plotter, grafikfähiger Drucker)

Einführung in Mikrocomputer-Betriebssysteme
 3. Teil: Was treibt ein Treiber?
 4. Teil: Was treibt das "Basic" im Betriebssystem?
 von Gerald Schröder

Zur Erinnerung: Im letzten Teil haben wir gesehen, was das Betriebssystem für das Anwenderprogramm an Geräte-Klassen zur Verfügung stellt. Allerdings leistet ein Betriebssystem mehr, andere wieder weniger - also versteift Euch nicht darauf. Zum Beispiel werdet Ihr weder im Newdos noch im CP/M eine Routine zum Zeichnen einer Linie oder zur Abfrage einer Maus finden. Das war eben damals (bei der Entwicklung dieser Betriebssysteme) noch nicht aktuell.

Es gibt dauernd neue Geräte, die beachtet sein wollen. Gestern war noch das 8-Zoll-Diskettenlaufwerk "in" (Größe Aktenkoffer bis Klein-Kühl-schrank), heute ist es ein Laufwerk in der Größe einer Zigarettenschachtel. Früher waren 500 Kilobyte auf einer Diskette nicht schlecht, heute stößt eine löschbare Opto-Platte in den Gigabyte-Bereich vor (1000x1000 Kilo-byte). Also muß ein Betriebssystem flexibel sein.

Aber "flexibel sein" ist (wie so vieles) relativ: TRS-80/Newdos war mit der Unterstützung von 4 (in Worten: vier) Laufwerken ziemlich flexibel, denn: wer konnte sich damals (1870-1979) schon 4 Laufwerke leisten! CP/M war mit 16 Drives etwas flexibler (fragt mal Helmut, ob das reicht). MS-DOS konnte (bis vor kurzem) nur Teile von Festplatten mit einer Größe von 32 Megabyte verwalten; bei den neuen Festplatten von 500 MB haben wir dann locker 13 Partitionen (=Teile, entspricht logisch einem Diskettenlaufwerk). Und bei allen Betriebssystemen gilt: Wehe, ein File wird mal zu lang (wie hoch die Grenze auch sein mag).

Machen wir uns noch einmal klar, was das BS leisten soll und wo das Problem sitzt. Es soll leisten: Für jedes Programm die Ressourcen des Rechners verwalten und verfügbar machen. Ressourcen sind alle "Möglichkeiten" des Rechners, also die angeschlossenen Peripheriegeräte, aber oft auch Co-Prozessoren (für Arithmetik o.ä.) und der verfügbare Speicher.

Das Problem: Bei der Entwicklung des Betriebssystems müssen alle denkbaren Ressourcen beachtet werden; dies ist besonders bei Peripheriegeräten schwer, denn oft gibt es diese gerade erst auf dem Papier. Aber trotzdem möchte jedermann/-frau diese neuen Geräte nutzen können, wenn sie auf den Markt kommen, und das möglichst mit den alten Programmen!

Aber wie geht das? Wie kann ein MS-DOS von anno dazumal mit optischen Platten umgehen, die erst viel später die geheimen Entwicklungslabors verließen? Und warum nagt heute trotz aller Abwehrversuche eine Maus an den meisten IBM-Rechnern, obwohl MS-DOS doch (streng konservativ erzogen) eine Maus gar nicht kennt?

Einen Teil der Lösung kennt Ihr bereits: Die Geräte-Klassen. Wenn ein neues Gerät auf dem Markt kommt, ordnen wir es in eine der Klassen ein. Da unser Anwender-Programm nur diese Klassen sieht (und nicht die "echten" Geräte), bemerkt es den Unterschied nicht und kann sofort mit dem neuen Gerät arbeiten. Das neue Gerät ist einfach ein weiterer Vertreter aus einer der altbekannteren Klassen. Ärgerlich wird es dann, wenn für das neue Gerät keine Klasse paßt. Z.B. gibt es weder im Newdos noch im CP/M Geräte-Klassen für Grafik; keine Grafikkarte, kein Plotter, keine Maus läßt sich dort in eine Klasse einordnen.

Also ist die Einführung eines neuen Geräts für das Anwenderprogramm kein Problem, sehr wohl aber für das Betriebssystem! Denn dieses hat die Aufgabe, mit dem Gerät umzugehen, es zu steuern. Wie bringen wir dem BS bei, wie es das neue Gerät zu bedienen hat? Schließlich ist ein Unterschied, ob wir als Vertreter der Klasse "Drucker" einen Fernschreiber, einen Matrix-Drucker oder einen Laserprinter anschließen.

Das Zauberwort heißt: Treiber. Das sind nicht diese härtigen Jungs mit Panflöte, Hirtenstock und Schäferhund, sondern niedliche, kleine Programmteile, die sich nachträglich irgendwo im oder am BS einnisten und die Arbeit übernehmen, die das BS nicht alleine machen kann. Für jedes Gerät gibt es einen Treiber. Ein Treiber kann aus mehreren Unterprogrammen bestehen, z.B. bei Treibern für Geräte der Klasse "Massenspeicher": ein Unterprogramm zum Auswählen eines Sektors, eines zum Lesen eines Sektors, eines zum Schreiben usw.

Wir können jetzt festlegen:

1. Für jedes Gerät muß es einen Treiber geben.
2. Die Treiber für alle Geräte einer Klasse müssen Teile (Unterprogramme) enthalten, die das Gleiche leisten. Z.B. müssen sowohl die Treiber für Diskettenlaufwerke als auch die Treiber für Festplattenlaufwerke ein Unterprogramm zum Lesen eines Sektors beinhalten, da beide zur Klasse "Massenspeicher" gehören.
3. Das BS muß es ermöglichen, daß neue Geräte eingeführt werden, indem neue Treiber in das BS "eingebunden" werden. Nach dem "Einbinden" kennt das BS diese neuen Treiber und benutzt sie, um die neuen Geräte anzusteuern.

Gerade die letzte Forderung ist nicht zu verachten. Was nützt uns die Klasse "Massenspeicher" im CP/M, wenn wir es nicht schaffen, den Treiber für unsere neue 20-MB-Festplatte in das CP/M einzubinden? Dann kann das CP/M nie was mit der Festplatte anfangen, weil der Treiber ihm eben nicht bekannt ist. Und: Wat de Bur nich kinnt, dat frett he nich. (Für unsere Südländer: Woas i ned woap, macht mi ned hoap.)

Nun könnten wir ja denken, daß ein BS nur aus Treibern besteht. Dem ist aber nicht so! Treiber dienen nur der grundlegenden Anpassung an die speziellen Geräte, mit denen sich unser Betriebssystem rumschlagen muß. In einem Treiber wird es kein Unterprogramm geben, das ein Inhaltsverzeichnis (einer Diskette, Festplatte o.ä.) verwaltet. Das erledigen andere Teile des Betriebssystems, und zwar für alle Geräte einer Klasse gleich. Außerdem ist dieser Teil des Betriebssystems auf allen Rechnern gleich, also können sich alle Anwenderprogramme auf ihn verlassen.

Damit haben wir zwei Teile eines Betriebssystems: Einen festen Teil, der sich nie ändert, egal auf was für einem System er läuft und welche Geräte daranhängen, und einen variablen Teil, der sich je nach Rechner und Umgebung ändert. Der variable Teil sieht für jedes Gerät einen eigenen Treiber vor, während der feste Teil alle Geräte einer Klasse gleich behandelt und dabei auf die Treiber aus dem variablen Teil zurückgreift. Also ist es für den festen Teil egal, ob er es mit Disketten und/oder Festplatten zu tun hat, weil die Treiber für ihn absolut identisch wirken.

Die CP/M- und MS-DOS-Freaks haben es schon gemerkt: Der variable Teil ist das BIOS, das "Basic Input/Output System" (die Grundlage (engl.: basic) aller Ein-/Ausgaben in einem Rechner), der feste Teil dagegen wird BDOS genannt, "Basic Disk Operating System", wobei dieser Name etwas irreführend ist, denn schließlich geht es dem Betriebssystem nicht nur um Disketten. Das "Basic" hat übrigens nichts mit dem Löffel-Zwo- oder sonstwie-BASIC zu tun, mit dem immer noch die meisten Freaks ihre Gehirngänge verstopfen (sorry); ~~hier~~ ist BASIC übrigens eine Abkürzung für "Beginners All Purpose Instruction Code" bzw. "Mehrzweck-Programmiersprache für Anfänger".

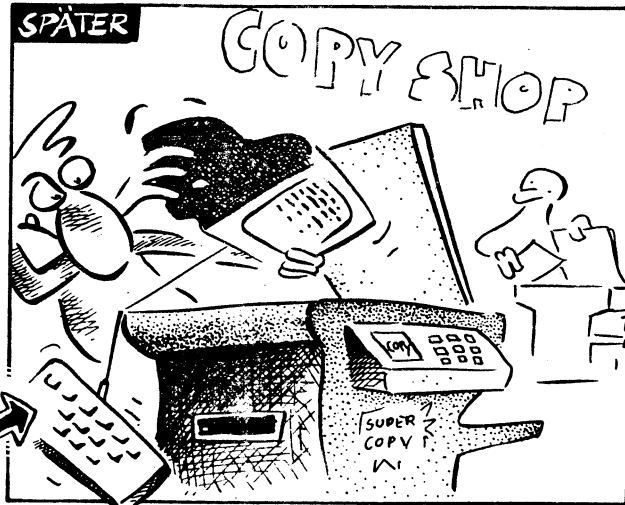
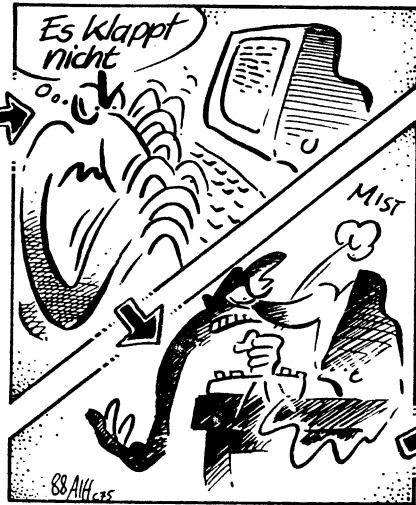
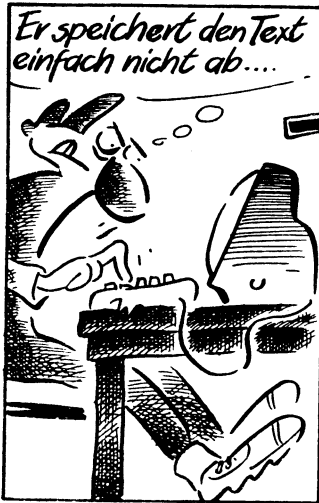
Wenn wir uns jetzt von dem Weihnachtsgeld ein neues Gerät (Grafikkarte, Laserdrucker, Festplatte, ...) kaufen und es an unseren Computer anschließen (bzw. einbauen), muß das BIOS geändert werden, aber nicht das BDOS. CP/Mler haben es hier schwer: Normalerweise muß man sich den Treiber selber schreiben und irgendwie in das BIOS bringen, wobei das "irgendwie" darauf hinausläuft, daß nur "echte" Programmierer (die Assembler mit der Muttermilch aufgesaugt haben) so ein neues Device genießen können. Sowie so kann CP/M mit vielen der neuen Geräte nicht ordentlich umgehen, wenn es auch gegenüber Newdos einige Klassen besser* ist (aber das Alter!).

Bei MS-DOS sieht die Sache anders aus: Hier gibt es nicht ein BIOS, sondern die meisten Geräte bringen ihr eigenes auf einer Steckkarte mit (was auch nicht problemlos ist); oder es müssen nur irgendwo einige Tabellen geändert werden; oder bei jedem Booten des Rechners zieht der sich von Festplatte/Diskette die neuen Treiber rein. Das Betriebssystem wächst also dynamisch (oft im Wildwuchs). Das BDOS ändert sich trotzdem nicht.

Soweit, so schlecht. Beim nächsten Mal soll es dann noch um virtuelle Geräte gehen; das sind Geräte, die es nicht gibt (genial, gelt?!).

* bzw. flexibler





Tips für den Computer-Kauf

Beste Gelegenheit, sich über das aktuelle Angebot an Personal-Computern zu informieren, bieten spezielle Messen. Möchte man sich lieber daheim mit dem Thema Computer beschäftigen, sind Computer-Magazine das geeignete Mittel, um zu einer Kaufentscheidung zu kommen. Testberichte vermitteln nicht nur einen Eindruck von Preis und Leistung des einzelnen Systems, sondern setzen sie auch ins Verhältnis zu Konkurrenzprodukten.

Anzeigen in Zeitschriften, Zeitungen und auch örtliche Händler bieten die komplette Hardware-Palette an. Die wichtigste Regel lautet, daß das billigste Angebot nicht auch das preiswerteste

ist. Zählt man nicht zu den Profis, erhält die Beratung durch den Händler besonderes Gewicht. Grundsätzlich sollte das Angebot mehrerer Händler geprüft werden. Von Interesse ist auch die Frage, welches Gerät der Händler benutzt, er hat sich bestimmt nicht das schlechteste genommen. Gute und preiswerte PC schickt auch der Versandhandel ins Haus. Computer-Versender bieten inzwischen einiges an Leistung und Service.

Manche Versender bieten einen qualifizierten Beratungs-Service per Telefon an. Damit nicht schon die Telefonkosten den Geldbeutel plündern, können Interessenten dort zum Ortstarif anrufen.

Besonders sicher kann sich der Käufer fühlen, wenn er für seinen PC 30 Tage Rückgaberecht hat. Wenn der Rechner nicht den Vorstellungen entspricht, wird er einfach zurückgeschickt. Eine Garantiezeit von zwölf Monaten und ein kostenloser Reparatur-Service vor Ort wirken sehr beruhigend, vor allem wenn der PC beruflich genutzt wird. Außerdem kann eine Firma sich einen solchen Service wohl nur leisten, wenn die Produkte von hoher Qualität sind.

Und wenn dann noch einige Standardprogramme wie Textverarbeitung, Datenbank und anderes im Lieferumfang enthalten sind, kann der PC-Spaß so richtig losgehen.



Folgender Artikel stammt aus einer Zeitung der "Marxistischen Gruppe" der Uni Hamburg. Halt, laßt nicht gleich die Scheuklappen runter! Bis auf einige klassenkämpferische Anmerkungen ist der Artikel meiner Meinung nach faktisch einwandfrei und bringt das Thema "Technischer Aufbau und soziale Wirkungen von Computern" (relativ) kurz und präzise rüber. Zwar sind die Formulierungen teilweise etwas hochgestochen (ein Hoch auf die Fremdworte!), aber es sind eine Menge Anregungen zum Nachdenken und Diskutieren enthalten. Die Hervorhebungen stammen übrigens vom Original-Autor (wer immer das auch gewesen sein mag - wahrscheinlich Marx persönlich).

Gerald Schröder

I.

Wozu ein Computer auch verwendet werden mag: Er ist eine Rechenmaschine. Im Unterschied zum volkstümlichen Taschenrechner und dessen mechanischen Vorgängern, wie sie noch vor wenigen Jahren die Büros bevölkerten, ist ein Computer ein Automat. Er wurde erfunden, um komplizierte und langwierige Berechnungen ohne menschliche Intervention zwischen den einzelnen Schritten durchzuführen. Das heißt, er kann einen vorweg festgelegten Plan der Rechnung auf die jeweils gegebenen Werte anwenden und dabei sein tatsächliches Verhalten nach den anfallenden Zwischenergebnissen richten, also etwa Fälle unterscheiden oder in Fehlersituationen anhalten. Diesem Zweck des Computers entspricht seine funktionelle Gliederung, die, wie auch immer variiert und verfeinert, bis heute grundlegend ist. An die Stelle des menschlichen Rechners, der sich mit einer herkömmlichen Tischrechenmaschine und einem Notizzettel an die Arbeit macht, tritt ein "Leitwerk", das ein "Rechenwerk" sukzessive mit der Ausführung elementarer Operationen betraut und Zahlenwerte von einem "Speicherwerk" festhalten bzw. liefern läßt. Hinzu kommen noch "Ein-/Ausgabewerke".

Dabei ist ein Computer kein Spezialist. Während ein Zigarettenautomat kein Bier verkaufen kann und ein Maschinensystem, das beispielsweise Autos fabriziert, bei jeder Variation des Produkts größere Umrüstungsarbeiten erfordert, ist die technische Gestalt des Computers, seine "Hardware", unabhängig von der besonderen Aufgabe fixiert, aber für schlechterdings jede tauglich. Entscheidend ist dafür die Konstruktion des Leitwerks. Es muß den jeweils bezweckten Gang der Dinge steuern und dabei selber nach einem invarianten Schema verfahren. Dies tut es, indem es die einzelnen auszuführenden Operationen erst aus Daten ableitet, die ihm genauso wie die Zahlen, mit denen gerechnet werden soll, im Speicher übergeben werden. Die Spezifik der Aktivität fällt somit ganz auf die Seite dieser "Software"; insofern ein Computer nicht ein besonderes Rechenverfahren verkörpert, sondern so eingerichtet ist, daß er geeignete Darstellungen solcher Verfahren - "Programme" - umsetzt und damit ausführt, ist er eine universelle Rechenmaschine.

Die primitiven Fähigkeiten des Computers, wie komplex auch immer ihr letztendlicher Gebrauch, sind sehr bescheiden. Wenn ein Mensch schriftlich addiert oder multipliziert, benutzt er Zahlenzeichen, Ziffern in Stellschreibweise, und das Rechnen selbst ist ein Manipulieren des äußerlichen Materials, Ersetzen zweier Ziffern durch eine dritte, Übertragen in eine anderen Position, wobei Art und Abfolge dieser Operationen durch Regeln - einen Algorithmus - festgelegt sind. Genauso verfährt ein Computer. Sein Geschäft ist mechanisch und läßt sich deshalb z.B. mit Hilfe von Zahnrädern, deren Stellung und Ineinandergreifen, realisieren. Tatsächlich werden elektronische Bauteile verwendet, insbesondere wegen der damit erreichbaren

Geschwindigkeit, und wenn sich dann aus technischen Rücksichten die binäre Form der Darstellung (Schalter auf, Schalter zu) aufdrängt, so belegt das augenfällig die abstrakte, ganz inhaltslose Bestimmtheit als Prinzip der Zahl und des Rechnens. Diese Tätigkeit ist nun allerdings zu unterscheiden von dem Zweck, ein numerisches Resultat hervorzubringen, und in diesem Sinne wird der Computer oft als Zeichen verarbeitende Maschine definiert. Man kann mit Hilfe seiner Bits (binary digits, Binärzeichen) auch ganz andere Dinge codieren, z.B. Buchstaben oder Merkmale vorgestellter Dinge, und dann durch geeignete Manipulationen an dieser Darstellung z.B. Schriftstücke aufbereiten oder Inventarlisten führen; die Geschichte der Verbreitung des Computers ist auch eine der Entdeckungen, was sich alles auf diese Weise bewerkstelligen läßt. Dabei ist der Umgang des Computers mit Zeichen, laut DIN 44 300 "Elemente einer vereinbarten endlichen Menge" und sonst nichts, nicht zu verwechseln mit dem Gebrauch, den die Intelligenz von Zeichen macht. Für ihn sind sie nicht die Manifestationen von Gedanken und Vorstellungen, sondern in ihrem physikalischen Dasein Material einer absolut gedankenlosen Aktivität. Ein Computer transformiert Bitketten in andere Bitketten, wie etwa ein Setzer ein Manuskript ohne Rücksicht auf den Inhalt typographisch umsetzt. Und er tut dies, indem er auf andere Bitketten, die seinem festen Satz von Befehlen angehören, reagiert, so wie eine Telefonvermittlung auf eine gewählte Nummer mit der Herstellung des zugehörigen Anschlusses antwortet.

Dem Prinzip, daß ein Computer ein universeller Rechenautomat ist, gab es in seiner nunmehr 45jährigen Entwicklungsgeschichte natürlich nichts hinzuzufügen; Leistungsunterschiede zwischen modernen und historischen, aber auch großen und kleinen Exemplaren sind quantitativer Natur, betreffen also Merkmale wie Geschwindigkeit, Speicherkapazität, Ausfallrate usw. und haben nur insofern Bedeutung für den Kreis der Anwendungen. Typische Geräte schaffen heute einige Millionen Instruktionen pro Sekunde und ähneln, ganz im Gegensatz zu beliebten Bebilderungen des machtvollen Elektronengehirns, in Größe und äußerer Gestalt üblichen Büromöbeln. Insbesondere wegen der Fortschritte der Halbleitertechnik sind sie in einem Preisverfall begriffen, dessen Ausmaß und Dauer in der Geschichte des Kapitalismus ohne Beispiel dastehen sollen, jedenfalls von einem stinknormalen Geldausgeber noch nie beobachtet werden konnten. Sie erlauben eine Vielzahl von Ein- und Ausgabegeräten, insbesondere die aus Bildschirm und Tastatur bestehenden Arbeitsplätze, und eine eindrucksvolle Fülle gleichzeitiger Arbeiten: Der eine Benutzer entwickelt neue Software, der zweite tippt Kundendaten ein, der dritte spielt Mondlandung, und zu Nutz und Frommen der Geschäftsleitung läuft außerdem noch ein Programm, das über die Aktivitäten der lieben Mitarbeiter Buch führt.

Eine derart "komfortable Benutzeroberfläche" kommt dem Computer nicht von Haus aus zu - im Gegenteil. Um ihn zweckmäßig verwenden zu können, sind erhebliche Kenntnisse seiner technischen Details, mathematische Bildung und Erfindungsgabe sowie peinlichste Sorgfalt bei der Bitfummelerei erforderlich - mit anderen Worten akademische Grade und die Qualitäten von Schwachsinnigen: Wollte man das Ding über den Kreis seiner Erbauer hinaus verbreiten und zu einem Geschäftserfolg machen, war es nötig, einen Teil der Arbeiten, die aus dem Gebrauch eines Computers entspringen, wieder an diesen selbst zu übertragen. Statt den Anwender mit der nackten Apparatur zu konfrontieren, wird ihm mit Hilfe der "Systemsoftware" eine wesentlich leichter zu bedienende (aber immer noch universelle) Maschine dargeboten; das heißt, seine Programme werden nicht unmittelbar, sondern durch die Wirkung anderer Programme ausgeführt.

Das "Betriebssystem" übernimmt die Funktionen eines Operateurs und geht dabei weit über das hinaus, was ein Mensch zu leisten vermöchte. Es läßt, startet, überwacht Programme, regelt den Gebrauch von Betriebsmitteln wie Speicher oder Drucker und erlaubt erst, die Geschwindigkeit der Maschine durch Mehrprogrammbetrieb (scheinbar gleichzeitig verzahnte Ausführung) auszunutzen. "Programmiersprachen" und dazugehörige Übersetzungsprogramme steigern die Produktivität des Programmierers, indem sie ihn von der schier prohibitiven Mühsal und Fehleranfälligkeit der binären Codierung befreien. An die Stelle der eigentlichen Maschinenbefehle treten zunächst "aussagefähige Namen", die sich ein Mensch besser merken kann als die Bitkombinationen selbst, und des weiteren "höhere Sprachkonstrukte", die aus der Formelsprache der Mathematiker entlehnt oder aus den Erfahrungen und Bedürfnissen des Programmierens heraus erfunden wurden und vom Übersetzerprogramm in komplizierte Gebilde aus Maschinenbefehlen transformiert wurden.

Das an die Verwendungen von Computern gern geknüpfte Fehlurteil, es handle sich um eine intelligente Maschine, findet an seinem durch die Systemsoftware vermittelten Erscheinungsbild eine Stütze. Namentlich bei den Programmiersprachen kann man sich einbilden, daß der Computer verstehe, was man ihm in den eigenen Worten sage. Dabei handelt es sich um ein Werkzeug des Programmierens, so "soft" es auch sein mag, da heißt, um ein Mittel, dieses Geschäft für das Subjekt zu erleichtern. Wenn für Programmiersprachen Elemente verwendet werden, die auch in richtigen Sprachen ihre Bedeutung haben, so darf man sich doch nicht gerade diese denken, und die Bedeutung für - oder besser Wirkung auf - den Computer, die der Programmierer lernen muß, ist von derselben Qualität wie die des Knopfdrucks, mit dem man den Strom abschaltet. Deshalb, gibt es bei Programmierneulingen stets die Enttäuschung, daß der Kasten, der so schön aufs Wort hört, nicht tut, was man gemeint hat, und noch die ältesten Hasen müssen sich über die Folgen banaler Schreibfehler ärgern, die ein Mensch auch beim dritten Hinsehen einfach überliest.

Als potentieller Alleskönner ist der Computer an sich zu nichts nütze; er benötigt Programme. Das Programmieren besteht darin, zu dem Zweck, dem ein Computer dienen soll, einen Algorithmus anzugeben und in einer Programmiersprache zu fixieren, das heißt, die jeweilige Vorgabe, was die Maschine leisten soll, muß übersetzt werden in ein Verfahren, wie sie es tun kann und soll. Die berüchtigte Schwierigkeit dieses Geschäfts, bei dem auch nicht laufend das Pulver erfunden wird, besteht darin, sich überaus komplexe und verschlungene Abläufe vorzustellen und ein Urteil über deren Effekt zu bewahren bzw. zu bilden. Mit dem Computer kommt so eine neue Sorte geistiger Arbeit in die Welt und des weiteren die "Softwarekrise", die nach allgemeiner Meinung noch bis ins nächste Jahrtausend den Engpaß der schönen Technik bildet. Das Problem ist weniger der Mangel an Personal als dessen Neigung, einen eigenen Kopf zu haben, wenn es sich denselben schon so "kreativ" zerbricht. Die Antwort auf das halbkünstlerische Gebaren dieser Leute heißt "egoless programming" und beinhaltet neben rigider Führung, Arbeitsteilung und Kontrolle die womöglich computergestützte Variation und Kombination früher erarbeiteter Programmstücke.

II.

Das typische Urteil über den Computer lautet, daß er die geistigen Fähigkeiten des Menschen erweitere, genauso wie die stoffumwandelnde Maschinerie sein körperliches Vermögen ausdehnte. Dieses Urteil täuscht sich über die Natur sowohl dieses Arbeitsmittels wie der Arbeitsprozesse, denen es dient.

Allgemein gilt für die Beziehung des Computers zum Geist, daß seine Domäne die ganz gedankenlosen Tätigkeiten sind, die aber auf seiten eines Menschen, der sie ausüben soll, allerhand Einsatz seines Hirns verlangen, Aufmerksamkeit, Gedächtnis, Kenntnis und Beachtung von Regeln. Auch diese Tätigkeiten sind beim Menschen, als Verrichtungen seiner Vorstellungskraft und seines Verstandes, anderer Natur als die Prozesse in einem Rechner. Sie lassen sich aber durch Schaltungen darstellen, ohne das Ergebnis zu verfälschen. Die Leistungsfähigkeit von Computern liegt also nicht daran, daß sie das Denken gelernt hätten, sondern hat ihren Grund in einer Eigentümlichkeit der theoretischen Leistungen, die bei der Rechenmaschine so gut aufgehoben sind. Das Gedächtnis ist zwar kein Speicher, sondern abstraktes Vorstellen von Sachverhalten, die man hinsichtlich ihnen wesentlicher Merkmale und Bestimmungen einmal zur Kenntnis genommen, also eben "gedacht" hat. Fürs Gedächtnis sind die gehalten Gedanken aber ein verfügbarer "Stoff", die getätigten Abstraktionen verwendbare Vorstellungen, nicht jeweils von neuem zu denken, sondern für allerlei theoretische Tätigkeiten "abrufbar". Die Mathematik ist zwar keine "Mechanik des Geistes", sondern die Erkenntnisleistung, die Abstraktion der Quantität in den ihr eigenen Bestimmungen zu entwickeln. Jede erkannte mathematische Regel - daher auch jedes mathematisch darstellbare Naturgesetz und jede daraus abgeleitete Technologie - läßt sich aber im Gedächtnis "aufbewahren" als vorstellungsmäßige Anweisung fürs Kombinieren quantitativer Bestimmungen. Das Rechnen lernt jedes Kind an seinen Fingern; und deren abstraktes Vorstellungsbild ist immerhin so vollständig zum Medium für die Anwendung einfacher mathematischer Regeln im Abendland geworden, daß mancher gebildete Mensch sich das Rechnen überhaupt nur als Dezimalsystem vorstellen kann. Umgekehrt erlaubt die Deckungsgleichheit von mathematischem Wissen und abstraktem Vorstellen etwas in jeder anderen Wissenschaft so Unmögliches wie mathematische Genies.

Das "Elektronengehirn" mit seinen binären Schaltkreisen macht ernst mit der von jedem Kind und jedem Rechengenie geübten Bequemlichkeit, die Mathematik in der Gestalt begriffsloser Kombinationsregeln zur Anwendung zu bringen. Und diese Bequemlichkeit hat ganz außerordentlich das Bemühen beflügelt, die verschiedenartigsten Gedächtnisinhalte - z.B. Vokabeln verschiedener Sprachen - in eindeutigen Zahlenwerten darzustellen und ihre jeweils zweckmäßige Anwendung in Kombinationsregeln für diese Zahlenwerte zu "übersetzen".

Das Ergebnis heißt zu Recht "Datenverarbeitung": Es geht um das Aufbewahren, Zurverfügungstellen und Verknüpfen eigentlich numerischer und ähnlich abstrakt aufgefaßter Information. Sie operiert nicht mit Schlüssen, sondern Prüfungen auf Gleichheit und Ungleichheit; ihr Element ist nicht das Urteil, sondern der "Datensatz", d.h. eine Kombination von Angaben wie Beruf, Gehalt, Steuerklasse. Brauchbar sind solche Fähigkeiten des Computers in den verschiedensten Umständen; als epochemachend gilt er aber in einer Produktionsweise, in der immaterielle Produktion weitgehend identisch ist mit solcher Datenverarbeitung. Der Zweck der kapitalistischen Produktion, der abstrakte Reichtum, der Wert, hat nicht nur die elementaren geistigen Potenzen als Maschinensteuerung und -kontrolle von der körperlichen Arbeit getrennt, sondern auch die unmittelbare Produktion mit einem Drumherum von gigantischen Ausmaßen versehen. Das betriebliche Rechnungswesen verkörpert in seiner Selbständigkeit wie in seinen Inhalten den Standpunkt der Profitmacherei gegenüber der eigentlichen Fertigung; es wird gekauft und verkauft und überhaupt furchtbar viel Geld furchtbar viel gezählt mit all den Feinheiten des wann und wo, für was und für wen, die erst über einen gelungenen Überschub entscheiden.

HEFT
29
Januar
1996

70

Der Computer kann es einem Menschen *ersparen*, seinen Kopf auf ganz mechanische Weise anzustrengen; wo das Nachdenken eher stört und zu Fehlern führt, und die Maschine übertrifft hier die Intelligenz lässig im Umfang der bewältigten Aufgaben. Daraus folgt nicht, daß *tatsächlich* irgendeinem Menschen irgendetwas erspart bleibt. Dank seiner Universalität legt ein Computer weit weniger als andere Maschinerie schon technisch fest, wie an und mit ihm gearbeitet wird. Namentlich im wissenschaftlichen Bereich findet man, daß der Rechenautomat als *Organ* und *Werkzeug* einzelner oder kooperierender *Subjekte* fungiert. Sie bedienen sich seiner souverän gemäß ihren wechselnden Zwecken; schreiben Programme selber oder wählen sie aus vorhandenen Bibliotheken; lassen rechnen oder benutzen die Elektronik als Karteikasten oder Vehikel der Kommunikation mit Kollegen, und sie ziehen es oft vor, ihren Publikationen gleich selbst mit einem Textsystem statt einer Sekretärin die endgültige Form zu geben.

Das glatte Gegenteil trifft man in den Niederungen der Wirtschaftswirklichkeit; wenn sich das Kapital des Computers bedient, so folgt, daß *viele Leute den Computer bedienen*. Das Grundprinzip aller kapitalistischer Produktion, daß nämlich "nicht der Arbeiter die Arbeitsbedingung, sondern umgekehrt die Arbeitsbedingung den Arbeiter anwendet" (Kal Marx), ist hier so handgreiflich wie in der industriellen Fertigung. Der ordinäre Bildschirmarbeiter, ob er nun Buchhalter oder Lagerverwalter war oder einen solchen ersetzt, sieht sich mit einem laufenden Programm konfrontiert, das ihm den Tag hindurch *vorschreibt*, was er zu machen hat. Nämlich den vom Programm benötigten Input, Daten oder gelegentlich Kommandos eines "Menues" in die Tastatur zu geben. Die Maschine subsumiert den Arbeiter als ihren *Handlanger*. Sie befreit ihn nicht von der Arbeit, sondern nimmt seiner Arbeit den Inhalt; erspart ihm Routinetätigkeiten, um ihm eine noch viel einförmigere und damit umso quälendere Routine aufzuzwingen; räumt Hindernisse und Friktionen aus dem Weg und eliminiert damit jede Abwechslung und Pause. Statt einer Erweiterung seiner geistigen Fähigkeiten erlebt er seine endgültige *Verblödung* und trägt noch allerlei Augen-, Rücken- etc. Beschwerden davon.

Erfunden wurde der Computer für den Staat, insbesondere für seine militärische Abteilung, und die ist noch heute, wo jede Klitsche auf Computer umgestellt, der größte Sponsor und Motor der technischen Entwicklung. Zum Geschäftsmittel wurde der Computer durch die simple Entdeckung, daß sich seine automatische Arbeitsweise außer für die langen Berechnungen, wie sie bei der Fabrikation von Atombomben und Artillerietabellen anfielen, auch genausogut für eine Reihe kleinerer Berechnungen, wie sie zum Beispiel in der Lohnbuchhaltung vorkommen, verwenden läßt. Die *kommerzielle Datenverarbeitung* beginnt damit, die überkommenen *isolierten* Tätigkeitsfelder des Rechnungswesens an den Computer zu übertragen; es gibt Programme für Gehaltsabrechnung, Kundenkonten etc. und die entsprechende Datenhaltung. *Integration* heißt das Ziel fortan. Die verschiedenen Bereiche arbeiten mit denselben Daten, und an die Stelle von Dateien, die wie klassische Akten unter einem Verarbeitungsgesichtspunkt organisiert sind, treten *Datenbanken*, die eine Vielzahl solcher Sichtweisen erlauben. Die überkommene Arbeitsteilung wird obsolet; der typische "Vorgang", der von einem Schreibtisch zum nächsten wandert, kann ohne große Ansprüche an seinen Bearbeiter in *einem* Schritt erledigt werden, und dank der dem Computer einverleibten Fähigkeiten kann der Systemanalytiker die Informationsströme und Verarbeitungsflüsse eines Unternehmens ganz *ohne* Rücksicht auf die in einem Spektrum von Berufen festgeschriebenen Fähigkeiten *entflechten* und *neuordnen*. Das *Papier*, charakteristisches Mittel der Büroarbeit und -zugleich Inbegriff ihrer Diskontinuität, wird in doppeltem Sinne zur *Randerscheinung*; und was

noch in Briefform nach außen fließen muß, erledigt die moderne Schreibkraft mit ihrem Textsystem: kaum Verzögerungen mehr durch Fehler, Revisionen und Kopien fürs Archiv, und statt dessen jede Menge fertiger Bausteine, deren Auswahl der Sachbearbeiter womöglich schon selber besorgt.

Die bloß *ideelle Einheit* manufakturmäßig geteilter Büroarbeit (ähnliches gilt für den Bereich Konstruktion) erhält *leibhaftige Existenz* in einem Computer, der mit seiner zentralen Datenhaltung und einem System von Programmen eine Fülle von Detailtätigkeiten an Sichtgeräten oder auch Kleincomputern treibt und zusammenfaßt.

Die Analogie zur Dampfmaschine als zentraler Bewegter der alten Fabrik oder auch zum modernen Fließband ist deutlich, und eben weil die Unterwerfung des Arbeiters unter die Maschine und die Überflüssigmachung seiner Geschicklichkeit in der *materiellen* Produktion schon vollendet ist, findet der Computer hier zögernder Verwendung und wirkt *nicht mehr revolutionär*. Eigentliche Roboter, wiewohl Lieblingsikonen der Ideologie von der Abschaffung der Arbeit, sind relativ seltene Erscheinungen, beschränkt auf Fälle, die sich weniger durch die Tätigkeit selbst als deren für einen Menschen widrige Umstände empfehlen. Ein Roboter äfft den Arbeiter nach, der, indem er eine Maschine bedient, auf seine natürlichen Potenzen reduziert ist, auf Wahrnehmung, Muskelkraft und Koordination seiner Gliedmaßen, der also ganz einfache, unausgebildete Arbeit leistet. Aber für einen Computer gilt die Hierarchie der Berufe nicht: Es ist unendlich schwer, ihm zum Beispiel beizubringen, Schrauben aus einer Kiste zu greifen. Weit größere Bedeutung hat er - oft in spezialisierter Form - für die *Vervollkommnung konventioneller Maschinen oder Produktionssysteme*, die nicht auf der *Nachahmung* menschlicher Tätigkeit, sondern ganz *objektiver Analyse* der bezweckten Stoffumwandlung beruhen. Als Organ der *Prozeßsteuerung*, ob in Erdölraffinerien, Papierfabriken oder Flugzeugkanzeln (und in allem möglichen militärischen Gerät), leitet er aus Meßwerten die nötigen Korrekturen ab und bringt sie selbsttätig auf den Weg. Und auf Werkzeugmaschinen *verpflanzt*, vererbt er ihnen seine programmierte *Flexibilität* und erübrigt die fachmännische Einstellung nach Zeichnungen und Maßen.

Wie im Büro, so fungiert der Computer auch hier, wo er bereits vorhandene Automatisierung perfektioniert, als *verselbständigt Hirn* der jeweiligen Aktivität, und deshalb bemüht man sich fleißig, die fundamentale Scheidung von Büro und Fabrikation zwar nicht aufzuheben, aber doch einem ganz neuen Maß von Integration zu unterwerfen: Die ideelle Lagerhaltung kooperiert automatisch mit der leibhaftigen Bewegung von Beständen; Konstruktion und Kapazitätsplanung gehen direkt über in Maschinensteuerung.

III.

Mit dem Titel eines "Jobkillers" wird dem Computer und seinen diversen Einkleidungen zugeschrieben, in beispielloser Weise Arbeit überflüssig zu machen und de facto abzuschaffen. Damit bestände an und für sich Grund zum Jubel - denn wo Arbeit *nicht nötig* ist, muß *Reichtum* herrschen. Aber dieser Schluß existiert heutzutage noch nicht einmal als naiver Ingenieuroptimismus: Die Technik dient so hohen Zwecken wie der nationalen Konkurrenzfähigkeit und läßt für die Menschheit nichts zurück als *Probleme*. Insbesondere das Problem, daß die Mehrheit der Leute zu nichts mehr nütze ist, aber leider nicht in persona abgeschafft werden könne, sondern Betreuung brauche.

Der harte Kern der Jobkillerei spielt bei solchen Menschheitsfragen keine Rolle, die Tatsache nämlich, daß mit dem Arbeitsplatz seinem Inhaber der *Lebensunterhalt* genommen wird. So wenig diese Konsequenz die *Wirkung* eines technischen Gerätes sein kann, so wenig ist es überhaupt wahr, daß der Computer *per se* Einfluß auf die Arbeitswelt hätte: Dies bleibt seinem Herrn und Meister vorbehalten. Die Anwendung des Computers durch das Kapital soll Kosten senken, d.h. nicht Arbeit *überhaupt*, sondern *bezahlte* Arbeit einsparen, Arbeit als Anspruch auf Lohn. Dieser Rationalisierung genannte Zweck hat zwei Konsequenzen. Erstens werden Arbeiter *freigesetzt*, ob sie nun massenhaft auf die Straße fliegen oder "bloß" entdecken müssen, daß das Wachstum ganz ohne Nachfrage nach Personal blendend vorankommt. Die Beweisführung, daß, was vom Computer an Arbeit verdrängt wird, auch durch ihn wieder Arbeit erhalte, ist eine apologetische Milchmädchenrechnung: Es handelt sich bei solcher Nachfrage nicht nur um ganz andere Berufe, sondern es werden auch nicht an der einen Stelle Lohnkosten gespart, um sie als Teil der Maschinenkosten an anderer Stelle zu zahlen.

Und eben weil zweitens der Angriffspunkt der Rationalisierung der Lohn ist, heißt ihr Resultat *profitablere Ausnutzung der Arbeit*, Tätigkeiten, die zuvor nicht jeder ausüben konnte, werden jetzt von billigen, ungebildeten Kräften erledigt. Wenn die Maschine die Arbeit erleichtert, so erlaubt sie auch deren *intensivere* Verausgabung, ja dient als Mittel, diese zu erzwingen. Mit der alten Arbeitsteilung verschwindet manche *technische Notwendigkeit der Kooperation*, und dem Kapital eröffnen sich neue *Freiheiten*, Zeit und Ort der Arbeit nach seinen eigenen Bedürfnissen zu gestalten.

Die ganz andere Sorge aufgeklärter Zeitgenossen gilt ganz direkt der Fähigkeit des Computers, große *Datenmengen* nutzbar zu machen. Bei betrieblichen Informationssystemen bei Banken, Versicherungen und vor allem beim Staat befürchtet man, daß der "*verdatete Bürger*" Objekt von Maßnahmen wird, die ihm nicht passen können. Dabei wird die Legitimität solcher Institutionen und Interessen *nicht* bestritten und Kritik an ihnen kommt gar nicht erst auf. Der absurde Gedanke, mit dem ganz wohlmeinend Bedenken angemeldet werden, heißt Aushöhlung bestehender Rechte durch technischen Wandel oder auch Machtverschiebung, die eigentl. nicht so gemeint war. Beispiele dafür lassen sich ernsthaft nicht geben - es soll ja nicht bös' gemeint sein -, und deshalb erfolgt regelmäßig der Übergang zu den *Möglichkeiten*, die ein Hitler heute hätte.

Der Staat gebraucht Computer natürlich nicht zur Machtverschiebung - er hat schon die Macht -, sondern um die *Wahrnehmung seiner Aufgaben*, die ihm niemand bestreiten will, zu *verbessern*. Ein schul-, wehr-, steuer- und sonstwie -pflichtiger Bürger ist schon immer in mannigfacher Weise verdatet gewesen. Und was für die guten Bürger gilt, gilt für die schlechten erst recht, und für Verbrecher und Kommunisten waren jede Menge Akten und Karteien nötig. Was neu ist, ist die Leichtigkeit, solches Material verfügbar zu machen und zu verwalten. Datenbanken und Computernetze sparen nicht nur Zeit und Kosten, sondern vergrößern auch den Erfolg mancher Anfrage; und Informationen, die - weil räumlich und organisatorisch getrennt - nie aufeinander bezogen worden wären, finden jetzt ganz automatisch zueinander.

Das Unbehagen am Computer argumentiert mit einem *Quidproquo*: Negative Wirkungen, die die *Anwendung* des Gerätes hat, werden als die *seinen* aufgefaßt, bloß weil sie *ohne* es nicht existierten. Der Fehler dieses Arguments ist der der Maschinenstürmer zu Beginn der Industrialisierung - bloß *bekämpften* diese in der Maschine - also falsch - *das Kapital*, während heute

von solcher Gegnerschaft nichts mehr zu spüren ist. Die Sorge um die Zukunft ohne Arbeit, aber mit allwissendem Großen Bruder ist eine Manier, die Gegenwart für sehr *gemütlich* zu befinden. Deshalb fällt es diesen Mahnern nicht im Traum ein, die Zwecke zu kritisieren, die sich mit dem Computer realisieren, und sie gehen konsequent dazu über, *an ihm selbst* bedenkliche Züge zu entdecken. Er ist keine "*menschengerechte*" Technik. Zu kompliziert und voller Eigenleben, kann er nicht wirklich verstanden und beherrscht werden. Und an der Möglichkeit des "Kriegs aus Versehen" soll man ermessen, was es heißt, sich auf diese Dinge einzulassen. Als ob diese Möglichkeit nicht zuallererst eine kriegerische Situation *voraussetzte!*

Genauso schlimm wie der "Computer, der durchdreht", ist der "Computer, der nur funktioniert". Bloße Realität und Logik werden ihm attestiert, als ob diese Attribute, die ihm übrigens nicht zukommen, zu einem Vorwurf taugen. Was ihm - wie "vorprogrammiert" - abgehen soll, sind Kreativität, Problembewußtsein und innere Werte. So hat das Ding immerhin das Gute, die Arbeitsplätze von Pfaffen, Professoren und anderen Fernsehstars zu garantieren. Wir meinen allerdings: Auch daran ist nicht der Computer schuld.

aus: Hamburger Hochschulzeitung, Marxistische Gruppe, Sonderausgabe Informatik, November 1989

HEFT
29
Januar
1990

75

- 1. Vorsitzender Alexander SCHMID
St.Cajetan Str. 38 VII
8000 München 80
☎ 089 /495326
- 2. Vorsitzender Horst-Dieter SCHROERS
Breslauer Str. 9
8016 Feldkirchen
☎ 089 /9032615
- Hardwarekoordinator Andreas MAGNUS
Pommernstr. 4
4650 Geisenkirchen
☎ 0209 /870230
- NEWS-Diskotheke Oliver VOLZ
Waldburgstr. 73
7000 Stuttgart 80
☎ 0711 /7353817
- CP/M -Diskotheke Rüdiger SÖRENSEN
Thomas-Mann-Str. 3a
6500 Mainz 1
☎ 06131 /32860
- Club-Bücherei Christian MENK
Ollsener Str. 52
2116 Hanstedt
☎ 04189 /78251
- Redaktion Jens NEUEDER
Gschlachtenbretzingen
Rudolf-Then-Straße 32
7178 Michelbach /Bilz
☎ 0791 /42877
- Autoren Die Redaktion bedankt sich bei
den im INHALTSVERZEICHNIS genannten
Autoren für die Mitarbeit an der
Club-INFO.
- Bankverbindung des CLUB 80
Postgirokonto
Sonderkonto CLUB 80
Übermann, H. 6209 Heidenrod
Konto-Nummer 496 071 - 605
Postgiroamt Frankfurt
BLZ 500 100 60

Hallo Club-80er,

nachdem nun der Jahreswechsel erfolgreich -das hoffe ich für jeden von Euch- hinter uns gebracht ist, erscheint nun wie versprochen unser nächstes CLUB-INFO.

Für die geleistete Mitarbeit an unsern CLUB-INFO's möchte ich mich bei allen Mitwirkenden recht herzlich bedanken.

Ich hoffe und wünsche uns, daß es im Neuen Jahr noch besser weiter geht und ich Euch regelmäßig ein "sattes" INFO erstellen kann.

Als nächsten Redaktionsschluß möchte ich Mitte/Ende Februar ansetzen, sodaß ich noch vor unserer Jahreshauptversammlung das nächste, das 30., Club-INFO zusenden kann.

Als Beilage zu diesem INFO findet Ihr einmal das Sonderinfo
HD 64180
sowie eine aktuelle Inhaltsübersicht

Ich wünsche Euch zum Abschluß Spaß am neuen INFO,
"happy computing 90"
und einen "Guten Rutsch" gehabt zu haben.

Es grüßt Euch Euer

Jens

Eine Zensur oder Kontrolle
der Infobeiträge erfolgt nicht.
Die Redaktion

Bitte beachtet
den Überweisungsträger,
für den CLUB-Beitrag 1990,
der diesem INFO beiliegt.