

CLUB 80

Club 80 ist
der

DANDY -
GENTE -
und KOMOEK-
ANWENDER

20. AUSGABE



Lass't den Nachwuchs an die Kiste
— oder: wie der Vater so der Sohn
(Heinrich Betz Junior)

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRASSE 6 / 6209 HEIDENROD 1
POSTFACH 27 / TEL.: 06124 / 3913

Seite:
und Autor:

Clubinternes

Neues vom Vorstand	1 - 3	Hartmut Obermann
Termine / Messen	4	Redaktion
Vorstellung neuer Clubmitglieder ...	5 - 8	Reiner Stober
.. und noch ein Leserbrief.....	9 - 10	Heinrich Betz
		Klaus-Jürgen Mühlenbein

Software

MSDOS im INFO	11	
Pascal für Interessierte	12 - 14	Rüdiger Sörensen
Hardkopie auch ohne Interrupt	15 - 16	Andreas Rychlik
Mathematischer Überfluß	17 - 19	Klaus-Jürgen Mühlenbein
Ein Schoner für die Bildröhre	20 - 21	Arnulf Sopp
Futter für Genie III's CRTC	22 - 26	Ulrich Heidenreich
Callerlei	27 - 34	
.. weil Kürze so würzig ist	35 - 36	Klaus-Jürgen Mühlenbein
Kleinschrift für den EDTASM/PLUS ...	37 - 43	
R-Befehl MSDOS-Like	44 - 56	Ulrich Heidenreich
Tape-Games auf dem Model III/4/4p ..	57	
Umkehr !	58 - 60	Hartmut Obermann
Keine Angst vor Haifischen!	61	
Trennen in TSCRIPS	62	Klaus-Jürgen Mühlenbein

Die letzten Zeiten

Impressum	117
Schluß	118
Clubmitgliederadressen	am INFO-Ende
	Redaktion

Seite:
und Autor:

Hardware

Der Nachhund	63	Heinrich Betz
Auslesen des Grafikspeichers	65 - 70	Helmut Bernhardt, Jörg Seelmann-Eggebert
Tastatur- und Druckerplatine	71 - 80	Bernd Drowälder
Anschluß gesucht	81 - 89	Artikel aus c't

Börse

Wer hat was -- wer will was	91 - 92
-----------------------------------	---------

Sonstiges

Nachtrag: In VISICALC sortieren	93	Klaus-Jürgen Mühlenbein
Fehler: Extended MEMDOS	94	
Nachtrag: RESET-feste MEMDISK	94	Hartmut Obermann
Im Schatten des Dinosauriers	95 - 96	
Auf zu neuen Ufern	97 - 98	Hartmut Obermann
PC statt GENIE ?	99 - 100	Paul-Jürgen Schmitz
Mathematik für Feinschmecker	101 - 107	Artikel aus PASCAL
Bin ich strahlengefährdet	109 - 110	

Club Bibliothek

Club-88 - Bücherei	111 - 116	Hartmut Obermann
--------------------------	-----------	------------------

Neues vom Vorstand

Entschuldigung!

Durch meinen Beitrag "Schau - trau keinem?", erschienen im letzten Info Nr. 19 unter der Rubrik "Neues vom Vorstand", habe ich mich selbst mit beiden Beinen in den berühmten Eimer mit der stinkenden braunen Masse gestellt (und ich kann euch sagen, es ist keine Schokolade!!)

Mit dem besagten Beitrag wollte ich den Mitgliedern, die sich nicht trauen aktiv am Clubinfo mitzuarbeiten, weil sie fürchten, wegen orthographischer Fehler belächelt zu werden, diese Angst nehmen. Als Aufhänger dazu diente mir ein Beitrag von Kajott. Dabei habe ich in meinem Übereifer einige Regeln vergessen, die ich ab sofort zu beachten verspreche!

1. Ich habe den Beitrag direkt im Anschluß an die Lektüre der beiden Artikel von Kajott Mühlenbein und Bernd Drowälder geschrieben, statt erst mal eine Nacht darüber zu schlafen und damit gewisse Distanz zu dem Problem zu gewinnen.
2. Ich habe angenommen, daß der Beitrag von Kajott in einem direkten Zusammenhang mit dem von Bernd steht, es aber unterlassen nachzuprüfen, ob dieser Umstand wirklich zutrifft!
3. Ich habe meinen Beitrag direkt an die Redaktion geschickt, es aber unterlassen Kajott einen Abzug davon zukommen zu lassen und ihn dadurch der Möglichkeit beraubt, die Sache noch vor der Veröffentlichung aufzuklären!

Hiermit möchte ich mich in aller Form bei Kajott entschuldigen und verspreche nochmals, die drei von mir verletzten Regeln ab sofort einzuhalten!

Um das Ganze noch einmal aus der Sicht von Kajott darzustellen, ist es mir ein Bedürfnis, seinen Brief an mich im folgenden zu veröffentlichen! (siehe Seite 2)

Trotz dieses bedauerlichen Vorgangs stehe ich aber zu meiner Meinung, daß Veröffentlichungen wie "Trau - schau, wem!" und "DER - DIE - DAS" (Info 19, Seite 75/76) zwar sicher der Allgemeinbildung und vielleicht auch der Hebung des nach Kajott "stellenweise grausamen Clubniveaus" dienen, bezweifle aber, daß sich dadurch neue Artikelschreiber für's Info finden lassen! Und ich bleibe auch dabei, daß für mich die fachliche weit vor der orthographischen Richtigkeit rangiert und deshalb nicht ständig auf Tipp- und/oder Schreibfehlern herumgeritten werden sollte (egal, ob nun jemand direkt angesprochen wird oder die Belehrung "der Allgemeinheit" gilt!)

Damit soll (zumindest für diesmal) das Thema abgeschlossen sein. Was es noch so an Neuigkeiten gibt, erfahrt ihr nach den beiden Briefen von Kajott, wobei ich noch darauf hinweisen möchte, daß der Brief an mich gekürzt wiedergegeben wird!

Hartmut Obermann

Klaus-Jürgen Mühlenbein
Diplom-Physiker

Im Mönchsarten 28
6940 Weinheim
Tel 06201/55052

K.-J. MÜHLENBEIN, Im Mönchsarten 28, 6940 Weinheim

Herrn
Hartmut Obermann

Schwalbacherstr. 6

6209 Heidenrod/Kemel

Ihr Schreiben	Ihr Zeichen	Mein Schreiben	Datum
18.06.87	*	*	24.06.87

Lieber Hartmut!

Ich bin recht traurig darüber, daß Du die Atmosphäre zwischen uns vergiftet hast. Ich habe noch nie jemand persönlich öffentlich angegriffen, und meine Wertschätzung für den sachlichen Gehalt dessen, was unsere Kollegen veröffentlichen, solltest Du längst kennen. Wenn ich schon sonst nicht viel verstehe, kann man mir wenigstens die Versuche konzedieren, das "Clubniveau" zu heben, denn das ist stellenweise grausam (was keine "Einbildung" ist) - aber das stets ohne persönlichen Angriff. Bernds Artikel habe ich vorher gar nicht gekannt. Wie hätte ich mich dazu überhaupt äußern können?

Natürlich muß ich mich jetzt gegenüber Bernd rechtfertigen. Da ich nicht hinterhältig bin, füge ich Dir einen Durchschlag bei, denn es gehört sich, daß man jemand informiert, ^{vor} ~~wann~~ man über ihn spricht/schreibt. Eine gewisse Schärfe darin muß Du aus meiner großen Enttäuschung und Verärgerung heraus verstehen. Denn gegenüber allen übrigen Mitgliedern bleibt ja nun einmal ein falscher Eindruck haften...

Falls Du überhaupt an meiner Mitgliedschaft noch interessiert bist (auch wenn ich im Moment keine organisatorische Aufgabe übernehmen konnte), solltest Du im nächsten INFO eine Richtigstellung bringen - denn Irren ist ja menschlich, eine Begradigung ist keine Schande und ich bin keineswegs nachtragend.

Anders täte mir die Konsequenz gahr leid, denn ich schätze den EXUB 80 nicht nur wegen der sachlichen Hilfe, die mir schon zuteil wurde, sondern auch wegen meiner Zuneigung zu einer jungen Gemeinschaft, die von einer (unpolitischen) Sache begeistert ist, und weil sich seine Aktivität und Dynamik - angefangen von der wieder bestens gestalteten Clubzeitung über die Bibliotheken bis zu den Treffen uft. - sehen lassen können, so daß ein Ralf Folkerts mit seinem müden "Bremerhavener" (aus dem ich z.Z. gerade den Gerhard Loose für uns werbe) und dem hingequälten INFO sich nur alle 10 Finger danach lecken kann!

Denk mal drüber nach.
Ich würde mich wieder freuen.
Grüß an Deine Frau!

Kapf

HEFT
20
August
1987

02

Unsere CLUB 80 Programmbibliothek hat, dank eines günstigen Angebotes, eine erhebliche Erweiterung erfahren. Durch Vermittlung von Gerald Dreyer konnte der Club ca. 80 Disketten der CP/M UserGroup übernehmen. Die Disketten enthalten nur Public Domain-Software und kann daher ohne Bedenken weitergegeben werden.

Wer die Verwaltung dieses riesigen Softwarepools übernehmen wird, ist zur Zeit noch nicht ganz geklärt. Im Gespräch ist Andreas Rychlik; Werner Schäfer möchte ich die zusätzliche Arbeit eigentlich nicht aufbürden, er hat mit der Diskothek bestimmt schon genug zu tun. Sollte sich noch jemand in der Lage sehen, diese Aufgabe zu übernehmen (er sollte wenn möglich über 80-Spurlaufwerke verfügen und möglichst viele CP/M-Formate lesen/schreiben/formatieren können) so soll er sich bitte bei mir melden.

Ich hoffe euch schon im nächsten Info näheres mitteilen zu können!

Umzug!

Vielleicht ist es euch schon beim Durchblättern des letzten Infos aufgefallen: Ich ziehe nicht um!

Mein Dienstherr hat es sich, vierzehn Tage bevor ich an meinen "neuen" Dienstort umziehen konnte, anders überlegt und mich an den "alten" zurück versetzt. Damit bleibt also (zumindestens vorerst) alles beim alten. Erfreulich ist, daß ich nun auch wieder über die Woche zu abendlicher Stunde erreichbar bin und mich wieder mehr um den Club und die Mitglieder kümmern kann!

Tandy in Deutschland tot!

Schon im Info Nr. 18 veröffentlichten wir einen Zeitungsausschnitt, dem zufolge Tandy in Deutschland seine Pforten schließt. Im Info 19 wurde auf Seite 79 eine Liste von Vertragshändlern abgedruckt, die die Aufgaben (Verkauf, Wartung) der Tandy-Läden alter Art übernehmen. Soweit mir bekannt, wird auch die Tandy-Zentrale in Ratingen geschlossen bzw. hat schon dicht gemacht.

Durch diese Maßnahmen wird das Arbeiten mit unseren, zum Teil schon recht betagten, Maschinen sicher nicht leichter und wenn mal irgend etwas kaputt geht, steht man ganz schön im Regen! Tandy scheint jedenfalls nun auch noch den letzten Rest Interesse an den "alten" Maschinen der TRS 80-Serie und ihren Benutzern verloren zu haben und so wird diese Spezies der Computeranwender wohl immer kleiner und kleiner werden!

Dadurch wird es noch wichtiger, sich gegenseitig zu helfen wo es nur irgend geht. Dies ist auch die Chance für Zusammenschlüsse, wie den CLUB 80, noch eine relativ lange Zeit weiter zu bestehen, obwohl die betreuten Computer eigentlich schon zu den Dinosauriern der Homecomputerszene gehören und Liebhaberstücke sind.

Damit genug für diesmal. Ich wünsche euch viel Spaß bei der Lektüre des Infos und noch einen schönen Urlaub und Sommer, euer

Karl-Heinz Obermann

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß 20. September 1987

Norddeutsches Regionaltreffen ???????? 1987

A c h t u n g :

Das Regionaltreffen der "Nordlichter" verschiebt sich!
Ein neuer Termin wird nach Absprache bekanntgegeben!!!

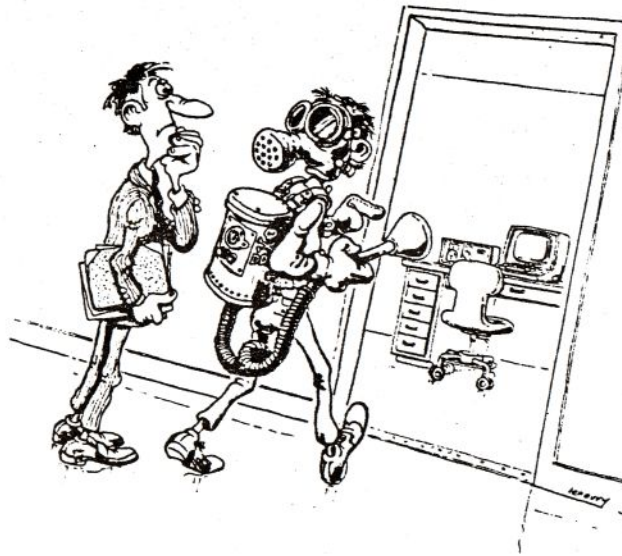
-- Messen '87 --

HOBBY ELEKTRONIK 87 Stuttgart 5. - 8. November 1987
PRODUCTRONICA München 10. - 14. November 1987

Regionaltreffen im Großraum Stuttgart

Als Termin schlagen wir vor:
Mitte Oktober bis Mitte November

Wer Interesse hat, an dem Treffen teilzunehmen,
melde sich bitte beim Klaus Hermann.



Liebe Freunde!

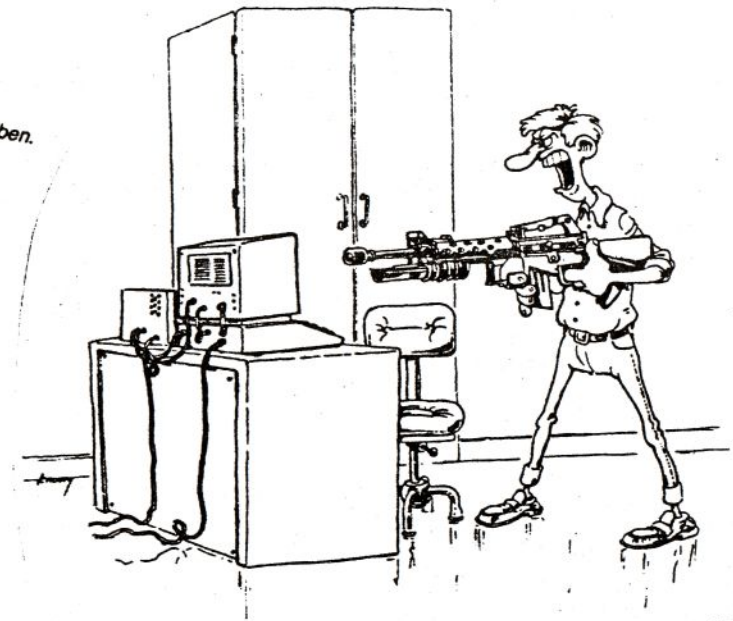
Als neues Mitglied im Club möchte ich mich kurz vorstellen.

Ich heiße Reiner Stober, geb. 15.10.56, komme aus der Gegend von Hannover und bin dort auch zur Zeit in einer Umschulung als Meß und Regelmechaniker. Aus beruflichen Gründen arbeite ich viel am Computer und glaube, das ich von Euch noch einiges dazulernen kann. Meine Anlagen sind ein Genie mit Centronicstreiber Modell I mit 16 k, sowie ein Drucker von General Elektrik, ein ZX-81 mit 16 k, und seit kurzem auch ein TRS-80 Modell I mit Controller. Leider habe ich für den TRS-80 kein Handbuch und brauche dringend die Belegungen für den Drucker. Es handelt sich um zwei Steckplätze mit je 34 Belegungen. Ferner suche ich ein 5 1/4 Zoll Laufwerk möglichst unter 100.-DM. Denn bei Völkner Elektronik bekommt man Epson Laufwerke schon für ca. 150.-DM. Als erstes suche ich einige Programme auf Cassette wie z.B.s. Assembler. Auch mit der Textverarbeitung ist nicht viel los. Es ist ein Programm aus der MC für den TRS-80.

Wer mich anrufen möchte, meine Telefon Nr. 05153/1564 aber nur am Wochenende ab Freitags 18.00 Uhr.

Gruß, Reiner.

Diesmal werde ich ihm die Mucken für immer austreiben.



Was soll das heißen, Bedienerfehler!?

HEFT
20
August
1987

06

Als neues Mitglied habe ich mit Bewunderung in den Infos gelesen, in welche Tiefen der Hard- und Software des TRS-80 manche aktiven Clubfreunde eingestiegen sind. Offen gesagt, habe ich bis jetzt bei meinem 4P-Computer nur wenig unter der Oberfläche geschürft. Er dient mir als nützliches Werkzeug, um Assemblerprogramme für Single-Board-Computer zu erstellen.

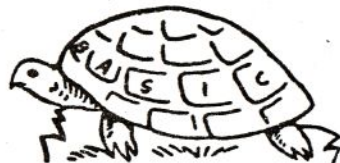
Das Werkzeug



Etwas verlegen wurde ich, als mir Hartmut Obermann empfahl, als Clubneuling einen Beitrag in einem der Infos zu veröffentlichen. Was sollte ich von meinen bescheidenen Kenntnissen an andere Clubmitglieder weitergeben? Doch im Gespräch stellte sich heraus, daß nicht nur die Anatomie des Computers interessiert, sondern auch seine Anwendungen. Da bin ich dann schon eher wieder dabei, denn seit einigen Jahren betreue ich die Meß- und Prozeßtechnik in einer mittelständischen Schokoladenfabrik.

Wie viele von Euch habe ich die erste Bekanntschaft mit Computern über die Programmiersprache BASIC gemacht. Was lag näher, als zu versuchen, kleine Prozeßsteuerungen auch in dieser Sprache zu programmieren. Vor einigen Jahren lieferte National Semiconductor den Prozessor INS 8073, der in seinem internen ROM einen Basic-Interpreter mitführte. Der Aufbau war so geschickt gemacht, daß man mit einem RS-232-Terminal und zwei externen Spannungen völlig autark Programme entwickeln und in ROMs brennen konnte. Mehrere Steckkarten mit diesem Prozessor sind heute noch an wichtigen Stellen der Schokoladenherstellung im Einsatz und haben über Jahre hinweg ihre Leistungsfähigkeit und Zuverlässigkeit bewiesen.

Zwei Nachteile haben jedoch dazu geführt, daß ich den National Prozessor durch den Z-80 ersetzte. Einmal wird der INS 8073 seit etwa drei Jahren nicht mehr geliefert, zum anderen habe ich feststellen müssen, daß BASIC zwar bequem zu programmieren, aber bei der Ausführung des Programms recht langsam ist. Bei einer Anwendung ging es darum, die von einer Waage mit 300 bd seriell abgegebenen Gewichtswerte zu empfangen und zu verarbeiten. Schon bei diesem relativ langsamen Datenstrom war der INS 8073 mit seinem Basicinterpreter völlig überlastet. Es kam zu Overrun-Errors im UART und zum Datenverlust.



Erste Versuche mit einem Single Board Computer "EMUF" der Firma Kanis in Pöcking brachten die gewünschte Steigerung der Geschwindigkeit. Erheblich mehr Aufwand war jedoch für die

Programmierung erforderlich, nachdem sich kein freundliches "READY" mehr nach dem Einschalten meldete, sondern der Z-80 ab seiner Adresse 0 mit Programmcode versorgt werden wollte. Mit den Steckkarten, die eine Z80-CPU, UART, CTC, ROM und RAM enthalten, habe ich mehrere Steuerungen aufgebaut. Je nach Anforderung an die Genauigkeit wurden verschiedene Analog/Digital-Wandler eingesetzt: CA3162, ICL 7109, ICL 7135.

"Safety First" habe ich als Überschrift gewählt und tatsächlich ist die Zuverlässigkeit oberstes Gebot im industriellen Bereich. Die Schäden, die beim Ausfall oder bei Fehlfunktion der Elektronik entstehen können, sind enorm. Als Beispiel sei der Defekt eines kleinen Kondensators genannt, der den A/D-Konverter dazu brachte, zu niedrige Werte anzugeben. Die Folge war, daß ein großer Behälter mit Kakaobutter beim Füllen überlief und eine beachtliche Menge des teuren Rohstoffs im Abfluß verschwand.

Nur erstklassige Bauelemente zu verwenden und diese möglichst noch überzudimensionieren ist eine der goldenen Regeln um Ausfällen vorzubeugen. Bei Seriengeräten, die in großen Stückzahlen hergestellt werden, muß man mit dem Pfennig rechnen, bei Sonderanfertigungen für Einsätze an wichtigen Stellen im Prozeß lohnt es sich nicht, am Material zu sparen. Ein einziger Produktionsausfall oder Ausschußproduktion kosten ein Vielfaches der ersparten Kosten beim Einkauf der Bauelemente.

Eine gefährliche Fehlerquelle sind Störungen, die über das Starkstromnetz oder über die I/O-Leitungen eingefangen werden. Über dieses Thema und die Maßnahmen, um die Einstreuung von Spikes zu verhindern, ließe sich ein eigener Bericht schreiben. Als recht störungsfest haben sich Schaltkreise in CMOS-Technik erwiesen. Alle aktiven Bauelemente auf der SBC-Platine habe ich in dieser Technik bestückt. Bei den Peripherieschaltungen (Decoder, Latches, Monostables) hat sich am besten die Metal Gate CMOS-Serie 4000 bewährt, die wegen ihrer hohen Signalpegel einen guten statischen Störabstand hat und wegen ihrer relativ langsamen Schaltgeschwindigkeit (2 MHz, bei 5 Volt) unempfindlich gegen kurze Spikes ist.

Besondere Anforderungen an die Zuverlässigkeit gelten auch bei der Software. Sämtliche nur denkbaren äußeren Bedingungen müssen so abgefangen werden, daß es nie zum Programmabsturz kommt. Zum Test muß der Bediener auf alle Tasten in beliebiger Folge und Kombination drücken können, um anschließend immer wieder in die Grundstellung zurückzufinden.

Meinen TRS-80 verwende ich, um die Programme für die Single Board Computer zu entwickeln. Per Download über die serielle Schnittstelle werden sie zunächst ins RAM des SBC geladen und dort getestet. Sind die Ergebnisse des Versuchslaufs zufriedenstellend, wird der Objectcode über die serielle Schnittstelle an einen EPROM-Programmer (Minato, Japan) übertragen und das dort erstellte EPROM dem SBC eingesetzt.

Heinrich Betz

...und noch ein Leserbrief...

an die stumme Gemeinde
=> mit Preisausschreiben! <==

Liebe Kollegen,

die ihr bisher stumm wart! Sicher habt ihr entweder-

- schon einmal ein kleines BASIC-Programm geschrieben; oder -
- euch an einem Maschinenprogramm (in Assembler) versucht oder-
- bei einem gekauften, geschenkten oder... also bei einem nicht selbst verfaßten Programm irgendeine Entdeckung gemacht oder-
- einen "Hänger" oder sonstige Schwierigkeit und somit eine Frage dazu gehabt; oder -
- irgendetwas sogar schon selbst an eurem Computer oder seiner Peripherie herumgebastelt; oder -
- zumindestens hierzu ein paar Fragen gehabt; oder -
- ein Programm, einen INFO-Beitrag oder eine Publikation in irgendeiner der zahlreichen Zeitschriften entdeckt, das/der/die euch besonders ge- oder mißfallen hat; oder -
- oder - oder - - -

Wenn nichts davon zuträfe, wäret ihr auf keinen Fall in einem Club. Und schon gar nicht in unserem! Denn dann läge euer Interesse ja irgendwo anders. Ihr seid doch bestimmt nicht daran interessiert, ständig über Sachen im INFO zu lesen, mit denen ihr nicht im geringsten zu tun habt bzw. in irgendeiner Form, auch nur auf einem Teilgebiet, auch selbst umgeht!

Also darf ich davon ausgehen, daß ihr alle irgendwie und irgendwo Praktiker seid. Ja, auch als sogenannte "Einsteiger" oder Anfänger!

Wißt ihr übrigens, wer der größte Praktiker ist?

Der Anfänger!! * Der "probiert" nämlich herum und herum und herum - und eines Tages entscheidet es sich: Entweder er gibt auf => dann tritt er auch in keinen Computerclub ein; oder er hat die "Zitterschwelle" überwunden und ist gefesselt; dann geht er vielleicht in einen Club. Falls er das tut, läßt er sich von den Kollegen, die schon weiter sind, anregen. Und auf einmal ist er "mittendrin" im Kreis der "Hobbyisten" (auf Deutsch: "Freaks") und merkt gar nicht, daß hinter ihm Neulinge - nicht nur Jüngere! - stehen, die, wie einst er selbst, noch am Anfang sind und die ersten Fragen stellen!

Die ersten "dummen Fragen"? - Ihr wißt doch: Dumme Fragen gibt es nicht - nur dumme Antworten! Wer hier fragt, bekommt immer eine gute Antwort von irgendeinem unseres Häufleins von 55 Mann (Stand: Juni 87); jawohl, dafür verbürge ich mich! Eine Frage, ein Bericht, ein Programm o.ä. kann gar nicht "so dumm" sein, daß es nicht abgedruckt und von allen Lesern beachtet und gewürdigt würde! Diese Erfahrung habe ich an mir selbst gemacht. Es gibt nämlich garantiert immer noch eine Handvoll Leute, die zur Zeit (!) noch weniger wissen und beherrschen als man selbst! Und für die ist jeder Beitrag äußerst wertvoll, selbst der kleinste. Das ist meine persönliche Erfahrung in drei Clubjahren. Und es ist wirklich schön, hin und wieder ein Echo zu bekommen!!

Vor kritischen Lesern braucht sich niemand zu fürchten. Für etwaige Tippfehler oder Fehler in der Rechtschreibung (wer hat schon eine Sekretärin!) braucht man sich nicht zu schämen, wenn man darin nicht so sicher ist! Zualererst kommt es darauf an, was einer mitteilen will, und wenn es, wie gesagt, noch so "bescheiden" ist. Wir sind keine Politiker, die mit schönen Worten hohle Phrasen - sogenannte "Worthülsen" - von sich geben. (Auch die machen oft genug sprachliche Schnitzer!) Und schon gar keine lauten, aber nicht lauterer Werbelaute!

Und am allerwenigsten braucht sich einer zu schämen, dem eine Hilfe angeboten wird, sei es beim Programmieren, Basteln, Gehen, Fliegen, Sprechen oder Schreiben...!

Wer kennt das alte Lied der Kavallerie? Bei den Reitern und Rittersleuten des EXUR 80 lautet es:

Wohlauf, Kameraden, an die Feder, auf's Pferd!
In das INFO mit Freiheit gezogen!
Denn im INFO, da ist ja der Freak noch was wert,
da sind doch nur das Köpfchen gezogen!
Es muß nicht gleich ein edler Pegasus sein,
auch ein Pony genügt, sei's auch drollig und klein!

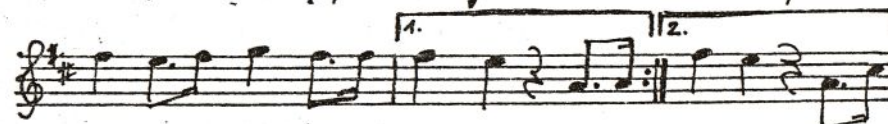
Jens und Peter sollen mal zeigen, was für einen Zustrom an Beiträgen aus eurem Kreis sie "verkräften" können - das walte Gott! - Ja?ot



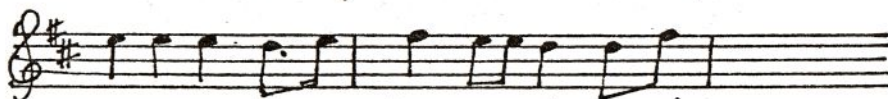
Wer die Trompete bläst? Die CPU natürlich!



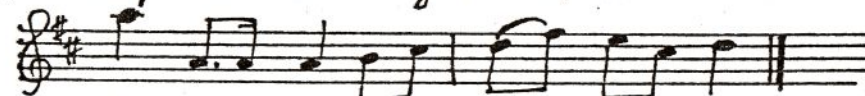
1) Wohlauf, Kameraden, an die Feder, auf's Pferd! In das
-> 2) -> Info, da ist ja der Freak noch was wert, da wird



1) Info mit Freiheit gezo-gen! Seem im-> 2)
2) doch nur des Köpfchen ge- -Wogen! Es muß



nicht gleich ein edler Pegasus sein, auch ein



Pony genügt, sei's auch drollig und klein!

[Melodie und halber Text: Altes Marschlied der preußischen Kavallerie - (aber auch bei Bayern beliebt) -; Textvarianten von T-Jolt]

=> Wer diese Melodie auf Disk (40 Tr., sonst beliebig) einsendet [Maschinensprache], ~~erhält~~
erhält von mir eine Flasche Reitersekt (Flaschengärung)

=> Marke "DERBY"! <==

JK

HEFT
20
August
1987

10

Im letzten Info wurde die Frage aufgeworfen (von wem, konnte ich nicht lesen. Verzeihung !!!), ob es sinnvoll sei, sich als Z80-Klein-Unternehmer mit der großen weiten Welt der IBM-Kompatiblen zu befassen. Dazu hätte ich auch den einen oder anderen Gedanken zu Papier zu bringen. Im Rahmen meines Studiums bringe ich viel Zeit vor der Tastatur eines IBM-AT zu, von dessen Eigenarten und Betriebssystem man auf diese Weise ja auch etwas ergattert, sofern man sich nicht nur mit Textverarbeitung beschäftigt, oder gar damit, für teures Geld erworbenen Programmen bei der Arbeit zuzusehen. Das Eigentümliche an diesem Rechner/Betriebssystem ist bekanntlich die weite Verbreitung, von der wir nur träumen können (CP/M Benutzer natürlich ausgenommen). Es hat eigentlich schon immer Leute gegeben, welche die Leistungsfähigkeit jener Maschinen mit derjenigen eines Apparates vergleichen, der mal gerade ohne Tricks 64 Kbyte adressieren kann. Deswegen habe ich keine Lust, das zum wiederholten Male zu tun.

MSDOS ist ein Betriebssystem, das wesentlich mehr kann als alles, was uns beim Computern das Leben leichter macht. Und die Prozessoren sind schneller. Und der Speicher ist mehr. Und die Festplatten sind billiger. Aber leider, MSDOS läuft nicht auf dem Z80. (Ach, übrigens, schon von Zilogs neuem Z280 gehört ?)

Also, warum eine MSDOS-Ecke ? Sollen Leute, die sich doch einer anderen Hardware verschrieben haben, noch Beiträge für den MSDOS-Blätterwald beisteuern ? Schließlich, man braucht ja nur eine der gängigen Computerzeitschriften aufzuschlagen und bekommt Tips, Programme, Literaturhinweise, Händlernachweise, Hardwareangebote, u.s.w.u.s.f. in Hülle und Fülle zu sehen (es kommen mir manchmal die Tränen). Selbst 80-Micro publiziert seit einiger Zeit mehr für das Model 1000 und Co. (Hardware Requirements : 128K, GWBasic) als für das Model 4.

Dennoch gibt es eine Reihe von Gemeinsamkeiten und Anregungen, die für uns von Nutzen sein können. Integrierte Software gehört meiner Meinung nach gerade nicht dazu. In erster Linie sind wohl Softwarelösungen zu nennen, insoweit sie in für uns zugänglichen Hochsprachen formuliert sind und nicht gerade die Neuorganisation von Harddisks behandeln. Ein gängiges Werkzeug in diesem Zusammenhang ist TURBO-Pascal, da es in nur wenig abweichenden Versionen für MSDOS und CP/M existiert. Es gibt eine ganze Reihe von glänzenden Programmen, die in dieser Sprache formuliert werden/wurden. Auch C und Fortran, COBOL und wie sie alle heißen machen Ideen portabel. Insofern lasse ich mir ein Eckchen MSDOS gefallen, wenn ich es auf meinem Model 4 gebrauchen kann. Die Moral von der Geschichte :

1) Das Bessere (?) muß nicht immer des Guten Feind sein.

2) lernt Pascal !!!

In diesem Sinne, c.u., Rüdiger.

In einem anderen Beitrag habe ich zum Erlernen der Sprache Pascal aufgefordert. Obwohl es eine Unzahl von Literatur zu diesem Thema gibt, will ich versuchen, in einigen kleinen Aufsätzen meine Erfahrungen mit dieser Sprache zu schildern. Meine erste Begegnung fand, glaube ich, so um 1982 statt, als ich über die Sprache las. Bis dahin beschränkten sich meine Programmierungsfahrungen auf FORTRAN 66, eine aus heutiger Sicht unakzeptable Tortur. Dann mußte ich im Rahmen eines Numerik-Praktikums ALGOL 60 studieren, meine erste Bekanntschaft mit sogenannten strukturierten oder auch strukturierten Sprachen. Um keine Mißverständnisse aufkommen zu lassen : Strukturiert programmieren kann man und sollte man in jeder Sprache, selbst in Assembler. Eine strukturierte Sprache ist aber ein außerordentlich nützliches Hilfsmittel. Der Unterschied zwischen Pascal und BASIC soll nachfolgend etwas verdeutlicht werden, denn man kann wohl davon ausgehen, daß BASIC verfügbar und benutzbar für jeden ist.

Pascal wurde als Hilfsmittel zum Lehren der Programmierkunst entwickelt. Darunter hat der Anwender ein wenig zu leiden. Viele Implementierungen an Großrechenanlagen sind für Naturwissenschaftler ungeeignet, weil numerische Funktionen schlicht fehlen und nur schlecht ergänzt werden können (Beispiel komplexe Zahlen) oder weil die strenge Typüberprüfung Pascals, für Lehrzwecke ideal, die Modularisierung von Programmen schwierig macht. All das gilt für Micros nicht (mehr) uneingeschränkt. Turbo-Pascal ist DIE Sprache für CP/M. Keine andere mir bekannte Hochsprachenimplementierung ist nur annähernd so leistungsfähig, jedenfalls nicht für CP/M.

Aber nun ans Eingemachte! Beschäftigen wir uns zunächst mit den Datentypen von Pascal. Was ein Typ ist, weiß ein jeder, der schon mal ein DEFINIT in BASIC gesehen hat. Pascal unterscheidet eine Reihe von vordefinierten Typen, von denen die meisten Bestandteil aller Sprachen sind.

INTEGER :

das sind ganze Zahlen im Bereich von -MAXINT bis +MAXINT, wobei diese vordefinierte Konstante von der jeweiligen Implementierung abhängt, bei 8-Bit-Rechnern ist MAXINT typisch 7FFFH.

REAL :

das sind rationale (nicht etwa reelle) Zahlen, die äußerlich durch Dezimalbrüche dargestellt werden, also zum Beispiel 3.141592. Der gültige Bereich der REAL-Zahlen variiert stark je nach System, bei TURBO ist m.E. die größte darstellbare REAL-Zahl so um 1.0E+37

CHAR :

das ist schon eine Neuerung. Eine Variable vom Typ CHAR hat den Wert eines Zeichens aus dem verfügbaren Zeichensatz, und das ist i.allg. ASCII.

Auf den Typen INTEGER und CHAR ist eine Ordnungsfunktion definiert, d.h. man kann die Werte eines Types abzählen, z.B. folgt die INTEGER-Zahl 423 auf 422 und das Zeichen "2" ist (in der Regel) der Vorgänger des Zeichens "3". Typen mit Ordnungsfunktion werden von Pascal bevorzugt behandelt, es gibt einige vordefinierte Funktionen, die auf diese Typen Anwendung finden. Der Typ REAL hat keine Ordnungsfunktion (Preisfrage : welche Zahl folgt auf 1.234 ?).

Mit einem Typ allein kann man noch kein Programm entwickeln. Wie man weiß, benötigt man dazu mehr, vor allem Kontrollstrukturen und Variablen. Auch in BASIC sind Variablen typbehaftet, man kann einer Stringvariablen A# keinen Ausdruck zuweisen, der als Ergebnis eine INTEGER-Größe hat. Der Ausdruck :

A# = 2*J

ist illegal, unabhängig vom Typ der Größe J. Im Gegensatz zu BASIC muß man in Pascal eine Variable vereinbaren, das heißt, der Übersetzer meckert, wenn ein Name im Programm auftaucht, den er nicht kennt. Wenn BASIC dasselbe passiert, nimmt es einfach an, daß es eine SINGLE-Größe ist. Also nun zur formalen Deklaration (Erklärung, Bekanntmachung) von Variablen. Im Programm muß zuerst das Schlüsselwort VAR stehen, gefolgt von einem oder mehreren Namen und deren Typen. Beispiel :

```
VAR x,y : REAL; i,j : INTEGER; c : char;
```

Durch diese Zeile werden Speicherplätze für 5 Variablen reserviert und deren Namen und Adressen dem Übersetzer bekannt gemacht. In Pascal muß ein Name mit einem Alphazeichen anfangen, dann dürfen Alphazeichen oder Ziffern folgen. Die erlaubte Gesamtlänge von Namen ist bei TURBO (glaube ich) 32 Zeichen, wobei auch der Unterstrich erlaubt ist. Zurück zu den Typen : REAL, CHAR und INTEGER sind unverträglich. Man kann (bis auf eine Ausnahme) Werte nicht zwischen Variablen dieser Typen zuweisen, die Anweisung

```
j := x;
```

mit den oben definierten Variablen ist syntaktisch falsch. Dagegen ist

```
x := j;
```

richtig (die erwähnte Ausnahme), Pascal läßt dies zu, weil keine versehentlichen Rundungsfehler auftreten können (Es gibt keine andere Möglichkeit, INTEGER-Größen in REAL umzuwandeln). Darüber hinaus gibt es eine Reihe von vordefinierten Funktionen, mit denen Typtransfer legalisiert wird :

ROUND(x) liefert den gerundeten Wert der REAL-Variablen x zurück.

TRUNC(x) liefert den abgerundeten Wert.

CHR(i) liefert das Zeichen mit der Ordnungszahl i.

ORD(c) liefert die Ordnungszahl des Zeichens c;

Den Zuweisungsoperator haben wir auch schon kennengelernt (das "Gleichheitszeichen") :

```
" := "
```

Für das allererste Pascal-Programm fehlen nur noch ein paar Kleinigkeiten : der Programmkopf und die Blockklammern. Ein Pascal-Programm sieht immer so aus :

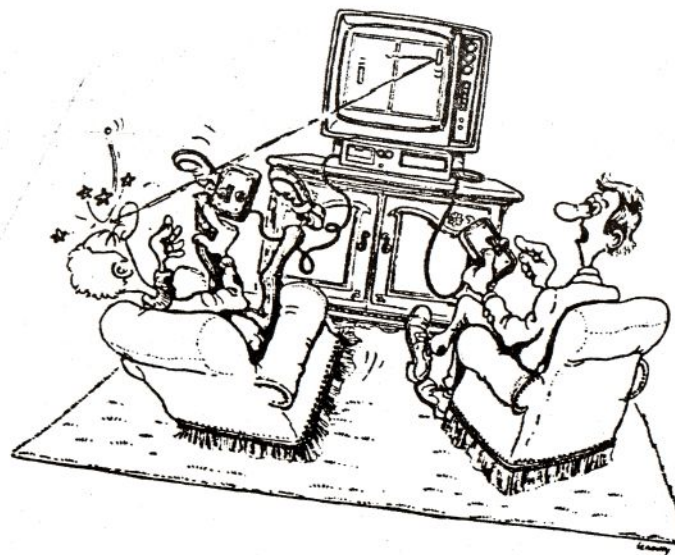
```
PROGRAM name (extfiles);
  var section;
  begin
    (* Programmrumpf *)
  end.
```

Die VAR - Sektion haben wir behandelt, den Programmrumpf basteln wir noch :

```
PROGRAM erstes (INPUT,OUTPUT);
VAR
  x,y : REAL;
  i,j : INTEGER;
  c : CHAR;
BEGIN
  x := y*(i/j);
END.
```

Natürlich ist dieses Programm, obwohl syntaktisch richtig und vollständig, völlig blödsinnig. Es würde zwar arbeiten, aber seine Ergebnisse wären zufällig und blieben in Ermangelung von Ein/Ausgabe auch noch geheim. Beim nächsten Mal gibt's dann ein Richtiges zum Ausschneiden !

Viel Spaß mit der Syntax wünscht Rüdiger.



Oh, Verzeihung!

Liebe Clubkameraden,

ich wollte schon immer Hardkopien von Spielen mit Blockgrafik machen, doch bei den meisten Spielen werden die Interrupts ausgeschaltet und somit ist keine Hardkopie möglich.

Auch ist es nicht möglich die Grafik solcher Spiele in die HRG 1B zu kopieren.

Ferner sind bei CP/M keine Bildschirm-Hardkopien möglich und der Befehl Ctrl-P, der die Bildschirmausgabe auf den Drucker umlenkt, klappt nicht immer.

Doch ist mir da eine Idee gekommen, die mir schon oft geholfen hat:

Beim booten einer Single-Density Diskette wird der erste Sektor in den Speicher zwischen 4200H und 42FFH geladen und anschließend springt der Rechner nach 4200H. Also habe ich ein Programm auf den ersten Sektor einer Diskette geschrieben, welches den Bildschirmspeicher von 3C00H bis 3FFFH in die oberen Bereiche des RAMs verschiebt. (In meinem Beispiel zwischen F000H und F3FFH.) Nachdem ich nun diese Diskette boote, hängt sich zwar der Rechner auf, aber der "alte" Bildschirminhalt ist im Bereich von F000H bis F3FFH verschoben worden. Wenn ich nun die System-Diskette wieder boote, wird der "alte" Bildschirmspeicher nicht zerstört, da ja das NEWDOS nur bis 5200H geht. Auch werden die Interrupts durch das booten wieder eingeschaltet. Anschließend wird der "alte" Bildschirminhalt durch ein kleines Programm wieder in den aktuellen Bildschirmspeicher verschoben und von dort aus kann er mit JKL als Hardkopie ausgedruckt werden.

Ich weiß, daß diese Lösung zwar recht umständlich und nicht optimal ist, doch ist mir keine bessere Möglichkeit eingefallen.

Das Programm MOVEUP verschiebt den Bildschirmspeicher nach F000H:

```
4200 21003C LD HL,3C00H ;von Speicherbereich
4203 1100F0 LD DE,F000H ;nach Speicherbereich
4206 010004 LD BC,0400H ;Anzahl der Bytes
4209 EDB0 LDIR ;verschiebe
420B C30000 JP 0000H ;springe nach MEM SIZE
```

Man formatiert nun eine Diskette in Single-Density und schreibt das Hexlisting mit Superzap am Anfang des ersten Sektors. Die Befehlsfolge lautet z.B.:

```
PDRIVE 0,1 TI-A,TD-A,TC-40,SPT-10,GPL-2,DDSL-17,DDGA-2
FORMAT 1
SUPERZAP
DD 1,0
mit MOD00 hackt man das Hexlisting ein: 21003C1100F0010004EDB0C30000
nach ENTER und Y wird die Disk modifiziert.
```

Wenn man nun eine Hardcopy von einem Bild machen will, legt man die modifizierte Disk in das Laufwerk 0 und bootet. Der Rechner meldet sich dann zwar mit MEM SIZE, aber er hat vorher das Bild verschoben.

Die modifizierte Disk ist weiterhin noch als Datendiskette geeignet.

Programm MOVEDOWN

Nach booten der Systemdiskette ruft man folgendes Programm auf, welches man schnell mit einem Monitor einhacken kann:

```
F400 2100F0 LD HL,F000H ;von Speicherbereich
F403 11003C LD DE,3C00H ;nach Speicherbereich
F406 010004 LD BC,0400H ;Anzahl der Bytes
F409 EDB0 LDIR ;Verschiebe
F40B 214038 LD HL,3840H ;HL als Zeiger auf Tastatur
F40E 7E LD A,(HL) ;hole Tastaturzeile
F40F CB3F SRL A ;Bit 0 ins Carry Flag schieben
F411 30FB JR NC,F40EH ;springe, wenn ENTER nicht gedrückt
F413 C32D40 JP 402DH ;springe ins DOS
```

Dieses Programm holt den "alten" Bildschirm wieder zurück und man kann mit JKL eine Hardkopie mit Blockgrafik machen. Mit ENTER kommt man wieder ins DOS.

Auch kann man den Bildschirm mit DUMP,NAMEN/BIL,F000H,F3FFH,402DH abspeichern.

Andreas Rychlik

Eckehard KUHN stellte mir eine mathematische Aufgabe:

"Schreibe ein Programm, das den SINUS (und andere trigonometrische Funktionen) sowie die quadratische Wurzel auf beliebig viele Stellen genau berechnet. Die Rechengeschwindigkeit spiele dabei keine Rolle!"

Ich dachte: Nichts leichter als das!
Trotzdem habe ich mich sehr darüber gefreut. Denn auch "leichte" Aufgaben können interessant sein. * Und diese hatte es in sich!

Die trigonometrischen Funktionen SINUS, COSINUS und TANGENS (sowie deren Umkehrungen) und die Wurzel sind Bestandteile unseres BASIC-Interpreters (oder ROM's - ich bitte um Nachsicht, wenn ich mich falsch artikuliere...)
Wozu also noch extra ein Programm dafür?

Ganz klar: Wenn ich z.B. Sternpositionen berechnen will, brauche ich viel mehr Dezimalstellen, als diese Maschine bietet. Wenn ich SIN(X) aufrufe, bekomme ich bestenfalls 8 Stellen (davon 7 genaue). Auch wenn ich Variable mit "doppelter Genauigkeit" wähle (X#), bringt das nicht mehr; es erscheinen dann zwar 16 Stellen, aber mindestens die letzten 9 Ziffern sind falsch!! (Probiert's und vergleicht mit eurem Taschenrechner!) Der 8-BITter schafft einfach nicht mehr.

Es sei hier dahingestellt, warum Eckehard mehr als 8 Stellen braucht (ich habe ihn absichtlich nicht gefragt, um mich bei dieser Aufgabenstellung nicht beeinflussen zu lassen). Es ist auch ganz egal; denn was aus dieser Bemühung herauskam, war für mich eine neue Erkenntnis (!): Mehr als 16 Stellen können gar nicht dargestellt werden! Auch wenn man zur "wissenschaftlichen" Schreibweise mit Gleitkommata, also zur Exponentendarstellung übergeht, gelingt es nicht, am Ende ein Ergebnis mit mehr als 16 Stellen niederzuschreiben! (Von deren "Gültigkeit" ganz zu schweigen!)

Immerhin: 16 richtige Stellen wären doppelt soviel wie 8. Mein Programm bringt 16 Stellen, davon mindestens 14 richtige. ("6 Richtige" am Samstag wären mir allerdings etwas lieber!)

Es bedient sich hierzu der bekannten Näherungsberechnungen durch Reihendarstellung der Funktionen. Die quadratische Wurzel wird mithilfe der binomischen Reihe in Taylor'scher Darstellung berechnet. Sie konvergiert zwar schlechter und ist daher etwas ungenauer. Aber, was soll's: 14 genaue Stellen genügen, um z.B. den Erdumfang auf einen Tausendstel Millimeter genau aus dem Erdradius zu berechnen! Die transzendente Zahl PI wurde von einem Großcomputer in viereinhalb Stunden mit 100265 Dezimalstellen berechnet, das ergab 22 Druckseiten... nochmal: Was soll's? *

Die Reihen werden abgebrochen, sobald zwei aufeinander folgende Glieder 16 gleiche Ziffern haben. Wie es dahinter ausschaut, "geht niemand was an"...

Demjenigen, dem es gelingt, mein Programm zu überbieten, möchte ich eigentlich den NOBEL-Preis des Monats August 1987 für binäre Mathematik verpassen.

Aber leider bin ich weder NOBEL noch nobel.
Sondern - sapperlot
nur der Jalot

Berechnung trigonometrischer Grundfunktionen

```
1 CLS:PRINT"          TRIGONOMETRISCHE BERECHNUNGEN
  (C) K.-J.MÜHLENBEIN, WEINHEIM 1987":PRINT:PRINT
2 DEFINITI,K:DEFDBLC,6,P,S,W,X:DIMG(35):PI=3.141592653589793:INPU
T"WINKEL <IN GRAD> ";W:X=W*PI/180:G(1)=X:SI=G(1):II=3:K=1:PRINT:
PRINT"          SINUS-BERECHNUNG":PRINT
3 G(II)=X:FORI=2TOII:PRINT$408+I+II,"*":G(II)=G(II)*X/I:NEXT:K=-
K:SI=SI+K*G(II)
4 IFSZ=SI,PRINT"SIN("W")="SI:PRINT:GOTO10
5 SZ=SI:II=II+2:GOTO3
10 PRINT"          COSINUS-BERECHNUNG":PRINT
11 G(0)=1:CO=G(0):K=1:II=2
12 G(II)=X:FORI=2TOII:PRINT$601+I+II,"*":G(II)=G(II)*X/I:NEXT:K=
-K:CO=CO+K*G(II)
13 IFCZ=CO,THENPRINT"COS("W")="CO:GOTO15
14 CZ=CO:II=II+2:GOTO12
15 PRINT:PRINT"          TANGENS-BERECHNUNG":PRINT"TAN("W")="SI/
CO:END
```

HARDCOPY eines Ausführungsbeispiels:

```
          TRIGONOMETRISCHE BERECHNUNGEN
(C) K.-J.MÜHLENBEIN, WEINHEIM 1987

WINKEL <IN GRAD> ? 23.7

          SINUS-BERECHNUNG *****
SIN( 23.7 )= .4019477766559601

          COSINUS-BERECHNUNG * *****
COS( 23.7 )= .9156625933395612

          TANGENS-BERECHNUNG
TAN( 23.7 )= .4389693098524373
READY
>
```

(Anm.: Die Sterne bewahren den User vor Warte-frust...***)

HEFT
20
August
1987

18


```

5 CLS
PRINT "          WURZELBERECHNUNG"
PRINT
*10*
*** <C> K.J.MÜHLENBEIN, WEINHEIM 1987 ***
10 DEFINT I,N
DEFDBL G,W,X,Y,Z
20 INPUT "RADIKAND ";Y
N = LEN ( STR$ ( INT (Y))) - 1
IF N / 2 < > INT (N / 2), N = N + 1
25 IF 10^N < 32768,
THEN DEFINT F
30 F = 10^N
X = Y / F - 1
G = X / 2
W = 1 + G
I = 1
K = 1
40 GN = G * X * I / (I + 3)
K = - K
W = W + K * GN
IF W = WZ,
THEN GOTO 60
50 WZ = W
I = I + 2
G = GN
PRINT "**";
GOTO 40
60 Z = W
FOR I = 1 TO N / 2
Z = Z * 10
LPRINT CHR$ (07)
NEXT
PRINT
PRINT "WURZEL AUS "Y"="Z
END

```

In letzter Zeit wurden verschiedentlich Programme veröffentlicht, die nach einer Weile den Bildschirm abschalten sollen, wenn keine Taste gedrückt wurde. Kocht sich der User nämlich gerade einen Kaffee, braucht die Beschichtung der Bildröhre ja nicht unbedingt mit Elektronen bombardiert zu werden. Eine nützliche Sache also.

Bei den in unserem Club vertretenen Computern geht das ohne weiteres auch. Sinnvoll ist es aber nur bei solchen, die sofort den Bildschirm abschalten können, ohne ihn zu löschen. Die TRS-80-ähnlichen Maschinen müßten dafür einen Aufwand treiben, der das Unternehmen wieder fragwürdig erscheinen läßt: Bildschirminhalt in einen Puffer retten, Bildschirm löschen, bei gedrückter Taste Bildschirminhalt wieder restaurieren. Und das alles müßte im eh' schon arg gebeutelten Himem passieren.

Hier wird vorgestellt, wie es beim Genie IIs gemacht werden kann. Als Zeitmaß dienen die Systeminterrupts, die alle 25 ms auftreten. Die hier benutzte Methode ist die Überprüfung des Bits 12 eines 16-Bit-Zählers (Bit 4 seines MSB). Wenn der Zähler 1000h erreicht hat, sind ca. 102 Sekunden verstrichen. Jetzt unterstellt das Programm, daß der User im Moment länger nachdenken muß oder überhaupt muß; der Bildschirm wird dunkel, ohne daß sein Inhalt verändert würde. Sobald eine Taste gedrückt wird, erscheint der alte Bildschirm wieder. In der Zeit dazwischen sah man lediglich den Cursor blinken, denn der wird selbständig vom Video-Controller erzeugt. Ihn auch noch auszuschalten, erschien mir vom Aufwand her übertrieben. Es kann auch nicht schaden, auf diese Weise zu bemerken, daß der Computer überhaupt noch läuft.

Programme, die die Interrupts maskieren, machen dieser Routine natürlich einen Strich durch die Rechnung. In einem zweiten Fall ist die Routine sinnlos, sogar fatal: Programme, die die Tastatur nicht über den DCB abfragen (z. B. alle SCIPSIT-Enkel), müssen nach 102 Sekunden ohne Monitor bedient werden. Aber da gibt es einen Ausweg, der lediglich ein bißchen umständlich ist: Wenn es dunkel wird, kann man mit <DFG> ins Mini-DOS oder mit <123> in den Debugger gehen und sofort anschließend mit <;> bzw. <G> und <ENTER> dieser Unbill begegnen. Das ist Geschmacksache. Ich ziehe es vor, in diesen Fällen einfach auf den Bildschirmknipser zu verzichten.

Das Programm erklärt sich wohl hinreichend aus den Kommentaren. Modifikationen für andere Genies dürften nach ihrer Lektüre kein Problem mehr sein. Eine Besonderheit muß jedoch erwähnt werden: Die Routine läuft im Adreßbereich des Druckertreibers. Der wurde von allem überflüssigen Quatsch befreit. Als Nebeneffekt sprang dabei heraus, daß alle Codes von 0-255 mit LPRINT CHR\$(x) ausgegeben werden können. Im Memory-Dump ist der geänderte Druckertreiber unterstrichen. Die Nullen dazwischen enthalten später das Programm.

Ebenfalls erwähnenswert ist ein Schönheitsfehler: Bei Betätigung einer der F-Tasten wird der Bildschirm auch ausgeschaltet. Warum das so ist, weiß ich nicht. Aber damit kann man gut leben; beim nächsten Tastendruck ist alles wieder beim alten.

```

0580 CD4F 3613 0B78 B1C8 CB78 C339 3579 1824 06 x x 95y $
0590 0000 0000 0000 0000 0000 0000 0000 0000
05A0 0000 0000 0000 0000 0000 0000 0000 0000
05B0 0000 0000 F5CD D105 20FB F132 E837 C900 2 7
05C0 0000 0000 0000 0000 0000 0000 0000 0000
05D0 003A E837 E6F0 FE30 C9E5 3E0E CD33 0048 : 7 0 > 3 H

```



```

00001 ;*****
00002 ; Bildschirman- und -ausknipper zur Schonung der Bildröhre
00003 ; Version für das Genie IIIs
00004 ; Arnulf Sopp
00005 ;*****
00006
4200 00007 ORG 4200h ;Platz genug im Sektorpuffer
4200 2A1640 00008 start LD HL,(4016h) ;Treiberadresse im Tastatur-DCB
4203 22A605 00009 LD {kbdvr},HL ;in das Programm patchen
4206 21A505 00010 LD HL,restvid ;Adresse der Umleitung
4209 221640 00011 LD (4016h),HL ;in den DCB patchen
420C C9 00012 RET ;zurück ins DOS oder zu BASIC
00013
4601 00014 ORG 4601h ;Adr.-Operand eines CALLs in d. INT-Rout.
4601 9005 00015 DW cntvid ;stattdessen die Umleitung CALLen
00016
0590 00017 ORG 0590h ;Platz in der gekürzten Druckeroutine
0590 21BE45 00018 cntvid LD HL,45beh ;Fortsetzung der INT-Routine
0593 E5 00019 PUSH HL ;als RET-Adresse auf den Stack
0594 2AC305 00020 LD HL,(counter) ;bisherigen Zählerstand laden
0597 23 00021 INC HL ;Zähler erhöhen
0598 CB64 00022 BIT 4,H ;schon Zählerstand 1000h erreicht?
059A CBA4 00023 RES 4,H ;nur bis 0FFFh zulassen
059C F5 00024 PUSH AF ;retten
059D DBFA 00025 IN A,(0fah) ;Systembyte 1 einlesen
059F 280E 00026 JR Z,exit1 ;weitermachen, falls noch nicht 1000h
05A1 E67F 00027 AND 7fh ;Port FAh, Bit 7 löschen (Bildschirm aus)
05A3 180A 00028 JR exit1 ;dort weiter
00029
05A5 C00000 00030 restvid CALL 0000h ;Tastatortreiber aufrufen (Dummy-Adresse)
05A6 00031 kbdvr EQU 9-2 ;Adr.-Operand
05A8 B7 00032 OR A ;wurde eine Taste gedrückt?
05A9 C8 00033 RET Z ;zurück, falls nein
05AA F5 00034 PUSH AF ;ASCII-Code retten
05AB DBFA 00035 IN A,(0fah) ;Systembyte 1 einlesen
05AD F680 00036 OR 80h ;Bit 7 setzen (Bildschirm an)
05AF 22C305 00037 exit1 LD {counter},HL ;Zähler neu laden
05B2 180B 00038 JR exit2 ;Teil des Druckertreibers überspringen
00039
05BF 00040 ORG 05Bfh ;ab hier wieder Platz in der Druckerout.
05BF 03FA 00041 exit2 OUT (0fah),A ;Systembyte 1 neu schreiben
05C1 F1 00042 POP AF ;restaurieren
05C2 C9 00043 RET
00044
05C3 0000 00045 counter DW 0000h ;Platz für den Zähler
00046
4200 00047 END start

```

Futter für Genie III's CRTC

Im letzten Info beschrieb Arnulf Sopp eine Möglichkeit, den #-Befehl des GDOS für die weiteren Bildformate A bis G aufzubohren. Darüber hinaus hoffte er auf ein Leser-Echo, wie dies wohl bei anderen TRS-kompatiblen als seinem Genie IIIs möglich wäre.

Hier ist ein Echo: Wer nur ein Genie III an Stelle des 'S' besitzt und auch nur ein GDOS 2.1 sein Eigen nennt, wird mit Arnulfs 3S-Patches wenig anfangen können, da hier die Sache ein wenig anders gelagert ist.

So residiert im GDOS 2.1 der #-Befehl zusammen mit dem LWT (Laufwerktest) in SYS23/SYS; SYS26/SYS dagegen enthält die Treiberrountinen für das 64x16-Bildformat zur Restaurierung desselben mit dem Befehl "64". Um auch hier die zusätzlichen Formate zu implementieren, werden folgende Patches in SYS23/SYS notwendig:

```

;...
M5085: LD BC,16 ; Länge eines Parametersatzes
LD HL,M50B0 ; Adresse des ersten Parametersatzes
CP 'X' ; ##,X ?
JR Z,M50A7 ; Ja!
ADD HL,BC ; Nein, Zeiger auf nächstes Format
CP 'V' ; ##,V ?
JR Z,M50A7 ; Ja!
ADD HL,BC ; Nein, Zeiger auf nächstes Format
CP 'H' ; ##,H ?
JR Z,M50A7 ; Ja!
ADD HL,BC ; Nein, Zeiger auf nächstes Format
CP 'S' ; ##,S ?
JR Z,M50A7 ; Ja!
CP 'T' ; ##,T ?
JR Z,M50F0 ; Ja, Bildseiten tauschen!

```

;Alt:

```

LD A,34H ; Nein, mit Fehlercode "Schlechter Parameter"
RET ; zurück

```

;Neu:

```

JP M513D ; Ansprung zur Erweiterung

```

21

00000 Fehler
cntvid 0590 counter 05C3 exit1 05AF exit2 05BF kbdvr 05A6 restvid 05A5
start 4200

HEFT
20
August
1987

22


```

M50A7: LD DE,M37F0      ; Ziel: CRTC-Kontrollblock
        LDIR             ; Parametersatz transferieren
        XOR A
        JP M37C0H        ; Daten an CRTC senden.

        ORG 513DH

M513D:  LD HL,NEWTAB     ; HL: Neue Parametertabelle
        SUB 'A'          ; aus 'A' wird 0 ... aus 'G' wird 6
        JR C,PARERR      ; kleiner als 'A' => Fehler
        CP 7             ; größer als - korrigiertes! - 'G'?
        JR C,PAROK       ; Nein, Parameter O.K.

PARERR: LD A,34H         ; Ursprüngliche Fehler-Ausgangs-
        OR A             ; Routine
        RET

PAROK:  OR A             ; Format "A", Zeiger stimmt bereits =>
PARLOP: JP Z,M50A7       ; Daten an CRTC senden.
        DEC A            ; Formatbezeichnung weiter-"scannen"
        ADD HL,BC         ; Formatzeiger weiterstellen.
        JR PARLOP        ; Irgendwann zwischen A und G funkticklappts!

        DM 'XXXXXXXX'     ; Füller-Bytes für Beginn der Parameter-
                        ; Tabellen auf einer 16-Byte-Grenze

NEWTAB EQU $

```

Leider ist SYS23/SYS derart ungünstig aufgebaut, daß ein Patch via Superzap, DDE oder ähnlichen Hilfsprogrammen nicht so einfach durchführbar ist: SYS23/SYS in der Version vom 1.12.1985 scheint nämlich häufiger mit einem Monitorprogramm überarbeitet worden sein, da hier und dort etliche NOPs eingefügt sind. So zeigt der Sektor-Dump (Superzap-Format) des SYS23/SYS neben anderen solcher ausge-NOP-ter Bereiche einen Freibereich von 101 Bytes ab 511BH gefolgt von einer Unter-routine für den Laufwerkstest, dem weitere 58 freie Bytes folgen. Um nun einen - für den Patch nötigen - zusammenhängenden Freibereich zu erzeugen, erscheint es am günstigsten, SYS23/SYS erneut zu assemblieren. Da Arnulf Sopp und ich es einmal planten, ein "GDOS decoded and other mysteries" zu erstellen, hatte ich es als glücklicher Besitzer einer reassemblierfähigen SYS23/SYS-Quellversion recht einfach, diesen Patch durchzuführen, wie das gezeigte Assembler-Listing demonstriert. Nichtsdestotrotz möchte ich hier versuchen, die dieser reassemblierten Version zugrunde liegenden Patches im Superzap-Format zu präsentieren:

Ursprüngliche Version des SYS23/SYS im GDOS 2.1:

```

DRV 00 0102 004D FEF9 2030 0D28 710D CA63 500D ...M...0.(q..cP.
0 10 3EFD 2006 EF3E 5818 0800 0D20 1B3D EF3E >....)X.....=.)
0H 20 48CD 8550 CDC9 01AF C900 0000 0000 0000 H..P.....
30 0000 0000 0000 0000 3E2A C900 0000 0000 .....)*.....
DRS 40 0000 0000 3E01 D3F6 3E40 D3F7 3E00 D3F5 .....>....)$.>...
137050 C900 0000 3E01 D3F6 3E50 D3F7 AFC9 0000 ....>....)P.....
55AH60 0000 0000 0000 0000 0000 0000 0000 .....
70 0000 0000 ED7B 6E4D AFC3 C037 CD40 4D00 .....änM...7.5M.
80 ED73 6E4D F316 0031 5F50 D5CD C901 210D .snM...1_P....!.
90 4E11 803E 0180 D0ED B021 BD4E 1140 3C01 N..>.....!N.$<.
A0 8001 EDB0 D121 EC37 3A40 38FE 0420 04CD .....!7:58.....
B0 C901 FB18 BFCD E94D 0100 00CD E94D E52A .....M.....M.* CALL des
FRS C0 0D50 3E20 7721 586F AFED 42CD 8051 3E3F .P>.w!Xo..B..Q)? LWT-UPs
0 D0 BC38 0E20 07CB 3DCB 3D00 1808 2100 3F18 .B....=.=...!?.
0H E0 0321 3F3F 3E2A 7722 0D50 E118 BB3A 1038 .!??)*w".P....8
F0 B720 1503 CB4E 28F5 3A10 38B7 200A 03CB ....N(:.8.....

DRV 00 B0B0 B0B0 B0B0 B0B0 B0B0 B0B0 0102 0050 .....P
0 10 B0B0 B0B0 B0B0 B0B0 B0B0 B0B0 BFC0 3E00 .....>.
0H 20 0000 0000 0000 0000 0000 0000 0000 .....
30 0000 0000 0000 0000 0000 0000 0000 .....
DRS 40 0000 0000 0000 0000 0000 0000 0000 .....
137350 0000 0000 0000 0000 0000 0000 0000 .....
55DH60 0000 0000 0000 0000 0000 0000 0000 .....
70 CAC8 CCF3 CDD5 4CCA 0035 7EFE 24CA 3435 .....L..58.$45
80 FE4E CAC0 3721 FC37 FE30 3809 FE38 3005 .N..7!.7.08..80.
90 D630 7718 2701 1000 21B0 50FE 5828 1809 .0w.'...!P.X(..
A0 FE56 2813 09FE 4828 0E09 FE53 2809 09FE .V(...H(...S(...
B0 5428 4D3E 34C9 0011 F037 EDB0 AFC3 C037 T(M)4....7.....7 Fehler-
FRS C0 6350 522E 1800 1818 0208 2B0B 0000 0000 cPR.....+..... Ausgang
3 D0 4E40 3D23 2302 2023 0208 2B0B 0400 0000 N5=##..#..+.....
3H E0 5F40 4A4E 190C 1014 0208 2B0B 0000 0000 _$JK.....+.....
F0 6840 4E22 1900 1818 0208 2B0B 0700 0000 h5N".....+.....

```



```

DRV 00 ED73 6E4D 315F 503E 09D3 FA01 0004 2100 .snM1_P>.....!.
0 10 01EA 0051 3C11 0040 7EF5 1A77 F112 0B13 ...Q<...$ß..w....
0H 20 2378 B120 F33E 05D3 FAED 7B6E 4DAF C900 #x...>.....änM... Der Teil
30 0000 0000 0000 0000 0000 0000 0000 0000 ..... des Frei-
DRS 40 0000 0000 0000 0000 0000 0000 0000 0000 ..... raums, in
137450 0000 0000 0000 0000 0000 0000 0000 0000 ..... den das
55EH60 0000 0000 0000 0000 0000 0000 0000 0000 ..... LWT-UP ver-
70 0000 0000 0000 0000 0000 0000 0000 0000 ..... schoben wird!
80 0000 0000 0000 0000 0000 0000 0000 0000 .....
90 0000 0000 C5F5 E5D5 E5ED 5BC2 4D21 943C .....Ä.M!< LWT-UP
A0 CD63 40E1 017D 3FED 42E5 D121 AE3C CD63 .c$..ü?..B..!<.c (alte
B0 40D1 E1F1 C1C9 0000 0000 0000 0000 0000 $..... Lage)
FRS C0 0000 0000 0000 0000 0000 0000 0000 0000 .....
4 D0 0000 0000 0000 0000 0000 0000 0000 0000 .....
4H E0 0000 0000 0000 0000 0000 0000 0000 0000 .....
F0 6A68 6C20 312E 3132 2E38 3200 0202 004D jh1.1.12.82....M

```

Änderungen in SYS23/SYS:

```

DRV 00 0102 004D FEF9 2030 0D28 710D CA63 500D ...M...0.(q..cP.
0 10 3EFD 2006 EF3E 5818 0800 0D20 1B3D EF3E >....)X.....=.)
0H 20 48CD B550 CDC9 01AF C900 0000 0000 0000 H..P.....
30 0000 0000 0000 0000 3E2A C900 0000 0000 .....)*.....
DRS 40 0000 0000 3E01 D3F6 3E40 D3F7 3E00 D3F5 .....>...$...>...
137050 C900 0000 3E01 D3F6 3E50 D3F7 AFC9 0000 .....>...>P.....
55AH60 0000 0000 0000 0000 0000 0000 0000 0000 .....
70 0000 0000 ED7B 6E4D AFC3 C037 CD40 4D00 .....änM...7.$M.
80 ED73 6E4D F316 0031 5F50 D5CD C901 210D .snM...1_P.....!
90 4E11 803E 0180 00ED B021 8D4E 1140 3C01 N..>.....!N.$<.
A0 8001 EDB0 D121 EC37 3A40 38FE 0420 06CD .....!7:$8..... Veränderter
B0 C901 FB18 BFCD E94D 0100 00CD E94D E52A .....M.....M.* CALL des
FRS C0 0D50 3E20 7721 586F AFED 42CD 1B51 3E3F .P>.w!Xo..B..Q?> LWT-UP's
0 D0 BC38 0E20 07CB 3DCB 3D00 1808 2100 3F18 .8....=,=...!?.
0H E0 0321 3F3F 3E2A 7722 0D50 E118 BB3A 1038 .!??)*w*.P....:8
F0 B720 1503 CB4E 2BF5 3A10 38B7 200A 03CB .....N<.:.8....

```

```

DRV 00 80B0 80B0 80B0 80B0 80B0 80B0 0102 0050 .....P
0 10 80B0 80B0 80B0 80B0 80B0 80B0 8FC0 3E00 .....>.
0H 20 0000 0000 0000 0000 0000 0000 0000 .....
30 0000 0000 0000 0000 0000 0000 0000 .....
DRS 40 0000 0000 0000 0000 0000 0000 0000 .....
137350 0000 0000 0000 0000 0000 0000 0000 .....
55DH60 0000 0000 0000 0000 0000 0000 0000 .....
70 CAC8 CCF3 CDD5 4CCA 0035 7EFE 24CA 3435 .....L..58$.45
80 FE4E CAC0 3721 FC37 FE30 3809 FE38 3005 .N..7!.7.08..80.
90 D630 7718 2701 1000 21B0 50FE 5828 1809 .0w.'...!P.X<...
A0 FE56 2813 09FE 4828 0E09 FE53 2809 09FE .U<...H<...S<...
B0 5428 4DC3 3D51 0011 F037 EDB0 AFC3 C037 T(M.=Q...7.....7 Anspruch
FRS C0 6350 522E 1800 1818 020B 2B0B 0000 0000 cPR.....+..... der Erweiter-
3 D0 4E40 3D23 2302 2023 020B 2B0B 0400 0000 N5=#...#...+..... ung statt
3H E0 5F40 4A4B 190C 1014 020B 2B0B 0000 0000 _$JK.....+..... Fehlerausgang
F0 6840 4E22 1900 1818 020B 2B0B 0700 0000 h$N*.....+.....

```

```

DRV 00 ED73 6E4D 315F 503E 09D3 FA01 0004 2100 .snM1_P>.....!.
0 10 01EA 0051 3C11 0040 7EF5 1A77 F112 0B13 ...Q<...$ß..w....
0H 20 2378 B120 F33E 05D3 FAED 7B6E 4DAF C9C5 #x...>.....änM... LWT-UP
30 F5E5 D5E5 ED5B C24D 2194 3CCD 6340 E101 .....Ä.M!<.c$... jetzt hier
DRS 40 7D3F ED42 E5D1 21AE 3CCD 6340 D1E1 F1C1 ü?..B..!<.c$.... im urspr.
137450 C9 ... Freiraum

```

Erweiterung für die Formate A bis G:

```

... 21 5C51 D641 3804 FE07 3804 3E34 B7C9 .!0Q.A8...8.)4..
55EH60 B7CA A750 093D 1BF9 5858 5858 5858 5858 ...P.=...XXXXXXXXX <- Füller
70 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA <- Formate
80 4242 4242 4242 4242 4242 4242 4242 4242 BBBBBBBBBBBBBBBBBB <- für
90 4343 4343 4343 4343 4343 4343 4343 4343 CCCCCCCCCCCCCCCC <- #,A
A0 4444 4444 4444 4444 4444 4444 4444 4444 DDDDDDDDDDDDDDDD <- bis
B0 4545 4545 4545 4545 4545 4545 4545 4545 EEEEEEEEEEEEEEEE <- #,G
FRS C0 4646 4646 4646 4646 4646 4646 4646 4646 FFFFFFFFFFFFFFFF
4 D0 4747 4747 4747 4747 4747 4747 4747 4747 GGGGGGGGGGGGGGGG
4H E0 0000 0000 0000 0000 0000 0000 0000 .....
F0 6A68 6C20 312E 3132 2E38 3200 0202 004D jh1.1.12.82....M

```

HEFT
20
August
1987

26

Ulrich Heidenreich
Werdenstr. 25
4300 Essen 1

Liebe Freunde aus dem Lande, in dem man die Maschinensprache spricht!

Und all ihr Übrigen, die sich nicht mit den Umständlichkeiten des BASIC begnügen wollen! Die ihr bereits ahnt, daß im "Nur-lesespeicher", auch kurz ROM genannt, so manches Handwerkzeug bereitliegt, auf das kein "USER MANUAL" hinweist, weshalb letzteres auch Handbuch genannt wird, weil man beim Lesen oft die Hände ringt und weil der Kopf dabei häufig leer bleibt (treffenderweise deshalb meist "Lehrbuch" genannt) - ihr unverdrossenen fröhlichen Programmierer, die ihr einen RÖCKRATH zu schätzen wißt, weil er euch den unentbehrlichen ROM-Rath erteilt - die ihr seine profunden Publikationen nun schon zum dritten Mal gekauft habt, weil das kostbare Werk vom pausenlosen Hin- und Herblättern schon so zerfleddert ist, daß es auseinanderfällt, als hätte einer darin verzweifelt nach seinem letzten S-DM-Schein gesucht (den es gar nicht mehr gibt) - statt nützlicher Routinen sieht man nur noch Ruinen...

ihr könnt aufatmen!

Denn nun sind die wichtigsten, meistgebrauchten Routinen, besser: Arbeitshilfen, herausgezogen und in knappster Form zusammengestellt! Ihr braucht nur noch Allerlei zu "CALLen!"

* SUPER möchte es möglich! * SUPER, die gemächlichste, aber flexibelste aller Datenbanken - aua! - Datenbanken, macht's auch möglich, daß jede Lücke, jeder Fehler, die/den ihr entdeckt (und das sind sicher viele!), auf schnellstem Wege nachgefüllt, nachgebessert werden kann. Sofern ihr sie mir meldet. Denn ich bin kein ROM-Experte, und ich überlasse euch auch die gesamte Verantwortung bei der Benutzung dieser Liste, die ich aus RÖCKRATH's ROM-Listing nur ganz knapp extrahiert habe, um mir die ständige Such- und Nachschlagearbeit zu ersparen.

Cum grano salis:

Wer R-R's ausführliches ROM-Listing nicht vorher gründlich studiert hat, wird wohl kaum in allen Fällen mit diesem Extrakt zurechtkommen! Drum:

LABORA CUM IRA ET STUDIO! Zu7ot



Korrekturmöglichkeiten

Name : (ohne) Einsprungadr./HEX 1936
 FDR/NEXT und GOSUB/RET. (Holt Daten v.Stack zur.)
 INPUT :
 OUTPUT:
 Veraend.Reg.

Name : ASC Einsprungadr./HEX 2A0F
 Ergibt ASCII-Code des 1. Zeichens eines Strings
 INPUT :
 OUTPUT:
 Veraend.Reg.

Name : BASIC Einsprungadr./HEX 06CC
 BASIC-Warastart
 INPUT :
 OUTPUT:
 Veraend.Reg.

Name : BLINK Einsprungadr./HEX 022C
 schaltet Stern in Bildschirmcke um
 INPUT :
 OUTPUT:
 Veraend.Reg. AF

Name : CALL Einsprungadr./HEX 03C2
 ruft I/O-Routinen ueber DCB auf
 INPUT : BC auss auf Stack sein (!) - B = DCB-Typ
 OUTPUT: (s.jeweilige Parameter u. Registerbeeinflussungen)
 Veraend.Reg. -

Name : CHECK Einsprungadr./HEX 1C96
 Syntax-Check (RST 08)
 INPUT : HL=Pointer; Byte nach RST 08
 OUTPUT: (HL)=Byte->RST08; HL->naechst.Zeich.,sonst SN-Err.
 Veraend.Reg. AF, HL=P

Name : CHR\$ Einsprungadr./HEX 2A1F
 erzeugt 1-Byte-String m.Zeichen; ASCII = Argument
 INPUT :
 OUTPUT:
 Veraend.Reg.

Name : CHRGET Einsprungadr./HEX 1D7B
 sucht nchst.sign.Zchn.u.gibt Info darueber (RST10)
 INPUT : -
 OUTPUT: A = Zeichen; Ziffer; CY=1; ':' od. 00: Z=1
 Veraend.Reg. AF, HL=P

Name : CLS Einsprungadr./HEX 01C9
 Bildschirm loeschen (CLS-Anweisung)
 INPUT :
 OUTPUT:
 Veraend.Reg. A=1F, F

ROM-Routinen des LEVEL 2 BASIC-Interpreters

Name : CPHLDE Einsprungadr./HEX 1C90
 vergleicht HL und DE
 INPUT : HL; DE: 16BIT Integer ohne Vorzeichen
 OUTPUT: HL>DE: Z=0,CY=0; =DE: Z=1,CY=0,A=0; <DE: Z=0,CY=1
 Veraend.Reg. AF

Name : CR Einsprungadr./HEX 20FE
 CR ausgeben
 INPUT : (409C)=Ausgabe-Flag
 OUTPUT: -
 Veraend.Reg. AF=0044

Name : CURPOS Einsprungadr./HEX 0348
 ermittelt Cursor-Position in der Zeile
 INPUT : (4020) = Cursor-Adresse, (4030) = Port-Status
 OUTPUT: A = Cursor-Position
 Veraend.Reg. AF

Name : DELAY Einsprungadr./HEX 0060
 Zeitschleife oder Warteschleife
 INPUT : BC bestimmt Zeitdauer (14.66 microsec x BC)
 OUTPUT: -
 Veraend.Reg. AF=0044, BC=0000

Name : DVAL Einsprungadr./HEX 0E65
 wandelt String in Zahl (Double) um
 INPUT : HL = Zeiger auf String
 OUTPUT: X = Zahl
 Veraend.Reg. AF, BC, DE, HL=P

Name : FILL0 Einsprungadr./HEX 1269
 Nullen in Puffer schreiben
 INPUT : A = Anzahl Nullen; HL = Pufferzeiger
 OUTPUT: HL=Pufferzeiger, um Anzahl der Nullen erhoert
 Veraend.Reg. AF=0044, HL

Name : HERE Einsprungadr./HEX 000B
 ermittelt Stand des PC
 INPUT : -
 OUTPUT: HL= Zeiger auf den Aufruf folgendes Byte
 Veraend.Reg. HL

Name : ILDXA Einsprungadr./HEX 27F8
 Zahl als Integer (ohne Sign) nach X
 INPUT : A=Zahl
 OUTPUT: HL, X=Zahl
 Veraend.Reg. AF=0044, HL

Name : INBUFF Einsprungadr./HEX 05D9
 Eingabe einer Zeile in den Puffer
 INPUT : B=max.Anzahl einzugebender Zeichen, HL=Puff.Anfang
 OUTPUT: B=Anz.d.tatsaechl.eingegebenen, C=max.Anz. Zeichen
 Veraend.Reg. AF, BC, DE=401D

ROM-Routinen des LEVEL 2 BASIC-Interpreters

Name : INCH Einsprungadr./HEX 002B
 Tastaturabfrage
 INPUT : -
 OUTPUT: A=ASCII-Code neu gedruckter Taste - oder 0
 Veraend.Reg. AF, DE=4015

Name : INCH2 Einsprungadr./HEX 035B
 Tastaturabfrage (s. INCH)
 INPUT : -
 OUTPUT: ASCII-Code gedruckter Taste oder 0
 Veraend.Reg. AF

Name : INCHW Einsprungadr./HEX 0049
 Tastaturabfrage (wartet, bis eine Taste gedrueckt)
 INPUT : -
 OUTPUT: A=ASCII-Code neu gedruckter Taste
 Veraend.Reg. AF, DE=4015

Name : INCHW2 Einsprungadr./HEX 03B4
 Tastaturabfrage (wartet, bis Taste gedrueckt)
 INPUT : -
 OUTPUT: A=ASCII-Code gedruckter Taste
 Veraend.Reg. AF

Name : INKEY Einsprungadr./HEX 019D
 Tastaturabfrage ohne Programmstop
 INPUT : -
 OUTPUT: -
 Veraend.Reg. -
 = RST 10

Name : INLINA Einsprungadr./HEX 1BB3
 gibt '?' aus und erwartet Eingabe einer Zeile
 INPUT : -
 OUTPUT: HL = I/O-Pufferanfang-1; F: CY=1, falls BREAK
 Veraend.Reg. AF, DE=4010, HL

Name : INLINE Einsprungadr./HEX 0361
 liest Zeile von Tastatur in I/O-Puffer
 INPUT : -
 OUTPUT: HL=I/O-Pufferanfang - 1
 Veraend.Reg. AF, DE=401D, HL

Name : INP Einsprungadr./HEX 2AEF
 Eingabe-Port lesen
 INPUT : -
 OUTPUT: -
 Veraend.Reg. -

Name : LEFT\$ Einsprungadr./HEX 2A61
 linken Teil eines Strings abtrennen
 INPUT : -
 OUTPUT: -
 Veraend.Reg. -


```
Name : LEN                               Einsprungadr./HEX 2A03
      ergibt Laenge eines Strings
INPUT :
OUTPUT:
Veraend.Reg.
```

```

Name : LETTER                               Einsprungadr./HEX 1E3D
      testet, ob Zeichen ein Buchstabe
INPUT : HL = Zeiger auf zu untersuchendes Zeichen
OUTPUT: wenn Buchstabe: CY=0, sonst CY=1
Veränd.Reg. AF

```

```
Name : LIST                               Einsprungadr./HEX 2B2E
      Programm auf Bildschirm auslisten (listen)
INPUT :
OUTPUT:
Veraend.Reg.
```

```
Name : LLIST                               Einsprungadr./HEX 2B29
      Programm auf Drucker auslisten (llisten)
INPUT :
OUTPUT:
Veraend.Reg.
```

```
Name : MEM                               Einsprungadr./HEX 27C9
      ergibt Groesse des freien Speichers

INPUT :
OUTPUT:
Veraend.Reg. A
```

```
Name : MID$                               Einsprungadr./HEX 2A9A
      mittleren Teil eines Strings abtrennen
INPUT :
OUTPUT:
Veraend.Reg.
```

```
Name : MROOM                               Einsprungadr./HEX 28BF
      Platz fuer String in Stringspace machen
INPUT : A=Stringlaenge
OUTPUT: DE = Stringspace-Adresse
Veraend.Reg. F, BC, DE, HL
```

```
Name : NEXTL                               Einsprungadr./HEX 20F9
      CR ausgeben, wenn Cursor nicht in Pos.0
INPUT : (409C)=Ausgabe-Flag; (40A6)=Cursor-Pos.
OUTPUT:
Veraend.Reg. AF=0044
```

```
Name : OUT                               Einsprungadr./HEX 2AFB
      Wert auf Ausgabe-Port ausgeben
INPUT :
OUTPUT:
Veraend.Reg.
```

```
Name : DUTCH                               Einsprungadr./HEX 0033
      Ausgabe eines Zeichens auf den Bildschirm
INPUT : A=ASCII-Code des auszugebenden Zeichens
OUTPUT:
Veränd.Reg. F, DE=401D
```

```
Name : DUTCH2                               Einsprungadr./HEX 033A
      gibt Zeichen auf den Bildschirm aus
INPUT : A=auszugebendes Zeichen
OUTPUT: (40A6) = Cursor-Position nachher
Veraend.Reg. F
```

```
Name : DUTCHF                               Einsprungadr./HEX 032A
      gibt Zeichen auf Bildschirm, Drucker od.Kass. aus
INPUT : A=auszugebendes Zeichen. (409C)=Ausgabeflag, s.u.
OUTPUT:
Verand.Reg. F
Ausgabeflag=0-->Bildschirm, 1-->Drucker, 80->Kass.
```

```
Name : PDS                                Einsprungadr./HEX 27F5
      ergibt Cursor-Position
INPUT :
OUTPUT:
Veraend.Reg. A
```

```
Name : PRCR                               Einsprungadr./HEX 0394
      CR auf Drucker ausgeben
INPUT : A=auszugebendes Zeichen; (409B)=Druckkopfposition
OUTPUT: (409B) = Druckkopfposition nachher
Veraend.Reg. A=0D
```

```
Name : PRINT                               Einsprungadr./HEX 003B
      Ausgabe eines Zeichens auf den Drucker
INPUT : A=ASCII-Code des auszugebenden Zeichens
OUTPUT: -
Veraend.Reg. AF, DE=4025
```

```
Name : PRINT1                               Einsprungadr./HEX 039C
      Zeichen auf Drucker ausgeben
INPUT : A=auszugebendes Zeichen; (409B)=Druckkopfposition
OUTPUT: (409B)=Druckkopfposition nachher
Veraend.Reg.
```

```
Name : PRINTS                               Einsprungadr./HEX 28A6
      druckt einen String
INPUT : (HL+1)...=String (durch '*' oder 0 beendet)
OUTPUT:
Veraend.Reg. AF, BC, D=0, E, HL=P
```

```
Name : PRRDY                                Einsprungadr./HEX 05D1
      testet, ob Drucker bereit
INPUT : -
OUTPUT: Z-1, wenn Drucker bereit
Veraend.Reg. AF
```


ROM-Routinen des LEVEL 2 BASIC-Interpreters

Name : RESET Einsprungadr./HEX 0000
 System-Initialisierung
 INPUT : -
 OUTPUT: -
 Veraend.Reg.

Name : RIGHT\$ Einsprungadr./HEX 2A91
 rechten Teil eines Strings abtrennen
 INPUT :
 OUTPUT:
 Veraend.Reg.

Name : RND Einsprungadr./HEX 14C9
 Zufallszahl erzeugen
 INPUT : X=Dummy-Argument (X=0) oder Intervall-Ende (X > 0)
 OUTPUT: X = Zufallszahl
 Veraend.Reg. AF, BC, DE, HL

Name : SADD Einsprungadr./HEX 0F89
 Summe bilden (summieren)
 INPUT : X=Summand; A=Summand (signed 8-Bit-Integer)
 OUTPUT: X = Summe
 Veraend.Reg. AF, BC, DE, HL

Name : SEOL Einsprungadr./HEX 1F07
 ELSE-Anweisung: sucht Ende der Zeile
 INPUT :
 OUTPUT:
 Veraend.Reg. AF, BC, D, HL=P
 Einsprung 1F09, wenn Trennzeichen2 = 0

Name : SEOS Einsprungadr./HEX 1F05
 DATA-Anweisung: sucht Ende der Anweisung
 INPUT :
 OUTPUT:
 Veraend.Reg. AF, BC, D, HL=P

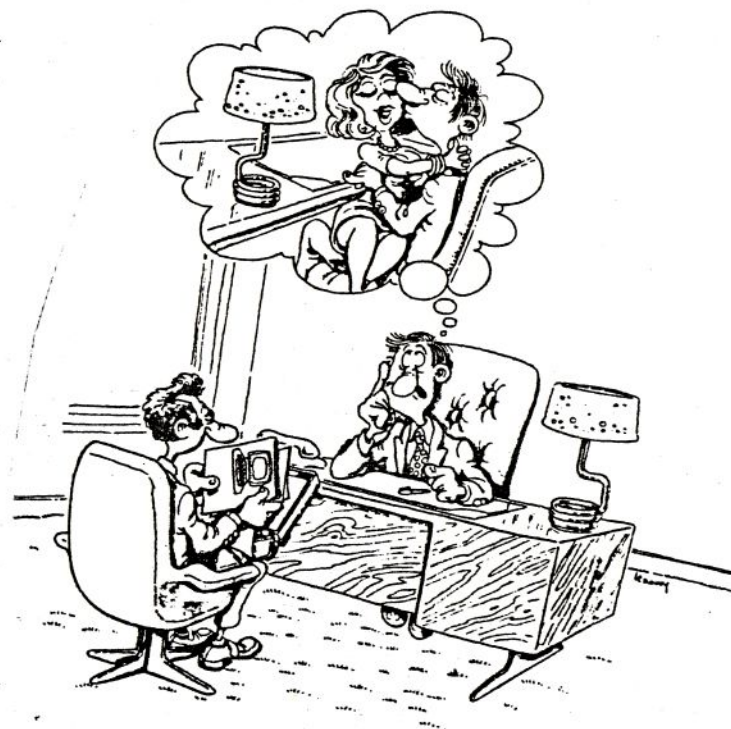
Name : STRING\$ Einsprungadr./HEX 2A2F
 erzeugt String aus
 gleichen Zeichen
 INPUT :
 OUTPUT:
 Veraend.Reg.

Name : TYPCHK Einsprungadr./HEX 25D9
 testet Typ des X-Registerinhaltes (RST 20)
 INPUT : (40AF)=Typ X
 OUTPUT: A=TypX-3; Int.:S=P=C=1; Single:C=1; Double: P=1
 Veraend.Reg. AF

Name : USR Einsprungadr./HEX 40BE
 User-Startadresse
 INPUT :
 OUTPUT:
 Veraend.Reg.
 FC-ERRDR

ROM-Routinen des LEVEL 2 BASIC-Interpreters

Name : VAL Einsprungadr./HEX 2AC5
 String in Zahl umwandeln
 INPUT :
 OUTPUT:
 Veraend.Reg. keine



Und Sie behaupten, Ihr Computer kann meine Sekretärin ersetzen...

- es hat ganze 6 (sechs!) Zeilen - und ich bin besonders stolz darauf! Aus zwei Gründen:

1) Meine bisherigen Programme besaßen 0 (Zero oder Null) Zeilen; es ist also das längste. (Die früher veröffentlichten 3 Programme zählen nicht, denn sie waren nur Lehr- oder, wenn ihr wollt, 'Leer'-stücke, da sie zu nichts weiter nütze waren;

2) Es füllt das erstmal eine erhebliche Lücke in unserem DOS aus und ist - mir jedenfalls - sehr viel nütze, wie ihr gleich erfahren werdet.

Jedesmal wenn ich vor dem Ausdrucken eines Directory (übrigens, wie mein dickes Dickschönäri mich neulich belächelte, spricht sich dies nicht "Direktorri", sondern zufällig so wie man's schreibt: "Direcktori" aus!), oder vor der Anwendung von JKL oder sonstwelchen Aktionen im DOS eine Überschrift, einen Hinweis oder sonstwas eben schnell mal ausdrucken wollte, ärgerte es mich, daß ich erst BASIC aufrufen, dann LPRINT schreiben, dahinter meinen Text setzen und anschließend mit CMD"S" wieder ins DOS zurückgehen mußte, bevor ich meine eigentliche Aktion durchführen konnte! Wie umständlich!!

Das ist nun endgültig vorbei. Dank meines Sechszeilers.

Zwar hat unser (Clubbe-)treuer Hartmut selbst eine DOS-Druckroutine ins SYS28/SYS eingebaut (s. seinen Beitrag im INFO unter der Überschrift "EXTENDED NEWDOS 80" sowie sein Betriebssystem "NEWDOS80+10"), die auf den Namen DR hört und als Library-Erweiterung eben diesen Befehl "DR" ziemlich schnell gehorcht; jedoch ist mir der freie Platz in den SYS-Files zu wertvoll für diese selten benötigte (s.o.) Routine! Aus der Not, in die ich beim Versuch, sie ins SYS-File einzubauen, geriet, wurde die Tugend, ein eigenes Maschinenprogramm zu "entwickeln"! Wie gesagt: Es ist mein erstes nützliches, und daher sowohl mein bestes als auch mein längstes...

Meine kleine Ruthine (ja, so heißt meine neue Freundin) hat noch einen Vorteil: Wenn man sich in dem unentbehrlichen Modus "DIS" befindet, muß man diesen erst verlassen, wenn man Library-Befehle aufrufen will wie "PDRIVE n", "SYSTEM n", "DIR-SORT n" o.a. Das ist bei einem SYS-freien .../CMD-File nicht nötig. Ich gehe also mit dem Cursor über die "Ruth" ('der darf dat') und schlage zu (ENTER); danach gebe ich gleich den Text ein, den ich ausdrucken möchte. Es dürfen alle Zeichen sein, auch Blanks und sogenannte Steuerzeichen. Und das in beliebiger Länge! (Nicht wie bei ähnlichen Routinen begrenzt auf 80 oder weniger Zeichen.) * Anschließend bleibt das System im DOS.

Wer Angst hat vor Maschinensprache spr. vor dem Programmieren in Assembler, der sei mit nachstehender kurzer Erläuterung des ebenso kurzen Programmchens getröstet:

"Faulheit" ist das oberste Gebot beim Programmieren!

Machen wir's uns bequem und überlassen das Denken den Herren in ROM! Hier im ROM liegt alles griffbereit:

a) eine Routine (auch UP = Unterprogramm genannt), die begierig

darauf wartet, Zeichen von der Tastatur entgegenzunehmen (Speicherplatz 49H = 73d);
b) eine Routine, die das so Empfangene auf den Bildschirm gibt (Speicherplatz 33H = 51d);
c) eine Routine, die es in den Druckerpuffer gibt (Speicherplatz 3BH = 59d);

Das ist bereits das Kernstück des "Programms". Jetzt muß man nur noch dafür sorgen, daß der Zeichenpuffer auch ausgegeben wird, denn das tut er erst, wenn er entweder eine Druckzeilenlänge enthält oder der Befehl "NUM ABER SCHLUSS!" erteilt wird (ODH = 13d). Das besorgen die beiden nächsten Zeilen. Solange dieser Befehl nicht erfolgt, wartet das Programm immer wieder auf ein weiteres Zeichen von der Tastatur (s. Rücksprung in der letzten, der 6. Zeile).

Die allerletzte sowie die allererste Zeile erwähne und zähle ich nicht, weil sie Eigentümlichkeiten eines jeden Assemblerprogramms sind, die ich den Experten abgeguckt habe...

Nun zum Schluß noch eine wichtige Anmerkung über eine Feststellung, die mich in allergrößtes Staunen versetzt hat: Das assemblierte Programm (siehe Listing) meldet ganz zum Schluß, daß diese 6 resp. 8 Zeilen "00000 TOTAL ERRORS" enthalten! Was für phänomenale Möglichkeiten ich da verpaßt habe! Es ist mir trotz größter Dummheit nicht gelungen, auch nur einen (00001) "ERROR" hineinzubringen - geschweige denn eine fünfstellige Menge...

HALLO Leute vom CLUB 80!
dieser Programmierer macht sich!
Gießet ruhig euren Spott
über'n Anfänger: Jolott!

* * * "D O S D R U C K" * * *

5200	00100	ORG	5200H	
5200	CD4900	00110 START	CALL	49H ; ZEICHEN-EINGABE
5203	CD3300	00120	CALL	33H ; BILDSCHIRM-AUSGABE
5206	CD3B00	00130	CALL	3BH ; ZEICHEN ZUM DRUCKERPUFFER
5209	FE0D	00140	CP	ODH ; WAR ES ENTER?
520B	CA2D40	00150	JP	Z,402DH ; DANN DRUCKEN UND INS DOS
520E	1BF0	00160	JR	START ; HOLE NAECHSTES ZEICHEN
5200	00170	END	START	
00000 TOTAL ERRORS (kaum zu glauben)				
34831 TEXT AREA BYTES LEFT				

START 5200 00110 00160 00170

[diese zusätzliche Information ist ein völlig überflüssiges Geschenk von EDTASM!]

Handwritten signature

Kleinschrift für den EDTASM/PLUS

Eigentlich hat ja Arnulf Sopp den Stein ins Rollen gebracht, als er neulich seinen Zeus-Assembler so in den Himmel lobte, um gleichzeitig den guten alten EDTASM/PLUS schlechtzumachen: Ja, wenn er wenigstens Kleinschrift beherrsche, der EdPlus, dann wäre er gar nicht mal so übel! Aber so nerve das ständige Hin- und Herschalten zwischen Groß- (für Mnemonics) und Kleinschrift (für Kommentare) doch.

Wer nun ein alter Hacker ist und zufälligerweise sowieso im EdPlus herumstochert, um ihm das 80x24-Format schmackhaft zu machen, läßt natürlich einen solchen Vorwurf nicht unbeantwortet, zumal Arnulf mit dem Kleinschriftproblem gar nicht mal so unrecht hat. Und schwerer als der 80er Patch kann die Operation wohl auch nicht werden: Ein paar "AND ODFH"s an die richtigen Stellen geschummelt und schon versteht der EdPlus auch das kleine Alphabet.

Gesagt, getan: Man nehme den 1-2-3-Debugger, einen Disassembler und n' Flasch' Bier dazu, suche die neuralgischen Punkte im EdPlus und erlebe dann seine erste "einfach-ein-AND-ODFH-Patchen"-Ernüchterung:

M75E1:	LD	A,(DE)	; Ein Zeichen aus der Befehlstabelle holen
	RLA		; Bit-7-Endemarkierung ins Carry schieben
	PUSH	AF	; das Carry retten
	OR	A	; das Carry löschen
	RRA		; RLA rückgängig machen
	CP	(HL)	; Mit dem Zeichen im Quelltext vergleichen
	JNC	DE	; Befehlstabellenzeigerweiterstellung
	JR	NZ,M75F7	; ungleich, nächsten Befehl vergleichen
	DEC	C	; Befehlslängenzähler (wo kommt der her?)
			; herunterzählen.
	JR	Z,M75FF	; Zero => Befehl komplett verglichen!
	POP	AF	; Befehls-Ende=Carry-Flag herholen
	JNC	HL	; Quelltextzeigerweiterstellung
	JR	NC,M75E1	; Weiter vergleichen

Maskieren Sie, geneigter Leser, hier einmal mit einem einfachen Patch die Kleinschrift heraus! Auch an weiteren Adressen schlägt Murphy mit solchen - eine einfache Maskierung so erschwerenden - Vergleichsbefehlen zu. Hinzu kommt zudem, daß ein und dasselbe "CP (HL)" teils in (HL), teils im Akku das Zeichen aus dem Quelltext übergeben bekommt. Wo maskiert man dann das Kleinschrift-Bit-5 heraus: aus A, aus (HL), gar nicht, immer, ... ?

Nach vielen unzähligen mißlungenen Versuchen, diesen In-welcher-Richtung-auch-immer-Vergleich immun gegen Kleinschriftunterschiede zu machen (unter anderem auch mit der zunächst genial erscheinenden Lösung "XOR (HL)" und "AND ODFH") blieb nichts anderes übrig, als doch etwas mehr Aufwand zu treiben:

Statt eines einfachen Patches entschied ich mich schweren Herzens, sowohl A als auch (HL) getrennt auf Kleinschrift zu untersuchen, um dann je nach Bedarf nur (HL), nur A, keins von beiden oder auch beide - letzteres nötig bei der Verwaltung der Symbol-Tabelle! - der ODFH-Maske zu unterziehen: Das Ergebnis findet sich im EDPATCH-Listing als Unterprogramm LOCOMP.

Nun warf sich aber die Frage auf, wie diese 1-Byte-Befehle sinnvoll durch entsprechende Aufrufe zum Unterprogramm LOCOMP zu ersetzen wären. Das Standard-Patch-Vorgehen, einfach drei passende Bytes mit einem CALL LOCOMP zu überschreiben und diese drei Bytes dann im Unterprogramm "nachzuholen", verbot sich aus einfachem Grund: Zunächst schien eine unüberschaubare Anzahl CP (HL)s dem Patch zu harren, was dann auch eine ebenso große Anzahl LOCOMP-Restaurations-Byte-Folgen notwendig gemacht hätte.

Erst eine uralte Erkenntnis über den EdPlus - damals entdeckt vom c't-Autor Heinz-Peter Heidinger - brachte die Lösung: Der EdPlus "verbiegt" einige RST-Vektoren. Warum also nicht dem guten Beispiel folgen und den weder von ihm noch außerhalb von BASIC benötigten RST 8 zum LOCOMP-Patch verwenden? Daß im Endeffekt dann doch nur drei OBEHs durch OCFHs zu ersetzen waren, rechtfertigt meiner Meinung nach aber nicht, im Nachhinein auf diese elegante Lösung zu verzichten: Als Schulbeispiel für den Fall, daß wirklich mal nur je ein Byte an-zig Patch-Stellen zur Verfügung steht, kann diese Methode nämlich allemal dienen!

Neben diesen verbleibenden drei Vergleichsbefehlen, nämlich an 62DDH zur Erkennung der Assembler-Direktbefehle, an 6E69H für "all purpose" und an 75E6H in der oben beschriebenen Mnemonic-Erkennung waren dann weitere Eingriffe nötig, die ich mangels Phantasie (?) dann doch mit der Drei-Byte-Ersatz-Methode erschlugh. Im einzelnen bergen diese letzten Patches nicht viel Interessantes, so daß ich zu deren Erläuterung auf die Kommentare im EDPATCH-Listing verweisen möchte.

Ulrich Heidenreich
Werderstr. 35
4300 Essen 1

HEFT
20
August
1987

38

37

0001

64B2

64B2 0C63

0000

8245

```

00000 TITLE EdPlus-Patches (C) 1987 by
00010 ;=====
00020 ;
00030 ; Patches in:
00040 ; *EDTASM+ Version 1.01 vom 4. September 1981*
00050 ; zur Verarbeitung von Kleinbuchstaben.
00060 ; Zusätzlich eventuelle Anpassung an das
00070 ; 80-Zeichen-Format des Genie III mit:
00080 ;
00090 GENIE3 EQU 1
00100 ;
00110 ; Durchführung des Patches:
00120 ;
00130 ; 1. Diese Quelle als EDPATCH/CMD assemblieren.
00140 ; 2. Zurück ins DOS
00150 ; 3. LOAD,EDPLUS/CMD
00160 ; (oder wie Sie auch immer Ihren
00170 ; EDTASM+ genannt haben!)
00180 ; 4. EDPATCH
00190 ; 5. Fertig!
00200 ;
00210 ;
00220 ;WICHTIGER HINWEIS:
00230 ;
00240 ; Diese zusätzlichen Patches residieren am Ende
00250 ; des EDTASM+ im ursprünglichen Textpuffer, der
00260 ; seinerseits dann wieder hinter diesen Patches
00270 ; beginnt.
00280 ;
00290 ; Nun erlaubt der EDTASM+ aber, mit "QZ" seinen
00300 ; Debugger Z-BUG aus dem Speicher zu entfernen,
00310 ; um den Textpuffer entsprechend zu vergrößern.
00320 ; Dies würde aber gleichzeitig den Speicher für
00330 ; diese Patches für den Quelltext freigeben.
00340 ;
00350 ; Um dieses zu verhindern, darf der Befehl "QZ"
00360 ; nicht mehr ausführbar sein:
00370 ;
00380 ORG 64B2H
00390 DEFW 630CH
00400 ;
00410 ; Stattdessen ist es aber hier möglich, durch
00420 ; Setzen des Flags:
00430 ;
00440 ;ZBUG EQU 1
00450 ;
00460 ; eine EDTASM+-Version mit aktivem Z-BUG, bzw.
00470 ; durch Löschen des Flags:
00480 ;
00490 ZBUG EQU 0
00500 ;
00510 ; eine Version ohne Z-BUG zu erzeugen!
00520 ;
00530 ;=====
00540 ;
00550 ORG 8245H ; Hinter EDTASM+ ohne Z-BUG
00560 COND ZBUG
00570 ORG 8FB9H ; Hinter EDTASM+ mit Z-BUG
00580 ENDC
00590 ;
00600 ;
00610 ;
00620 ;GETTYP Modifikation zur Annahme der Zeichen
00630 ; d,t,o,g und h als Typkennzeichner
00640 ;=====

```

8245 CD5A65

8248 C5

8249 F5

824A 47

824B FE61

824D 3807

824F FE7F

8251 3003

8253 E6DF

8255 47

8256 F1

8257 78

8258 C1

8259 C9

825A C5

825B 47

825C 4E

825D 79

825E CD4882

8261 77

8262 78

8263 CD4882

8266 BE

8267 71

8268 C1

8269 C9

826A CD8045

826D C34882

8270 216D4B

8273 C34882

8276 CD5A65

8279 C34882

00650

00660 GETTYP: CALL 655AH ; Original-CALL "nachholen" ! 40

00670

00680

00690

00700

00710

00720

00730

00740

00750

00760

00770

00780

00790

00800

00810

00820

00830

00840

00850

00860

00870

00880

00890

00900

00910

00920

00930

00940

00950

00960

00970

00980

00990

01000

01010

01020

01030

01040

01050

01060

01070

01080

01090

01100

01110

01120

01130

01140

01150

01160

01170

01180

01190

01200

01210

01220

01230

01240

01250

01260

01270

01280

01290

```

00660 GETTYP: CALL 655AH ; Original-CALL "nachholen" ! 40
00670
00680 ;=====
00690 ;LCMASK Maskiert Kleinschrift aus A,
00700 ; falls Kleinschrift in A
00710 ;=====
00720
00730 LCMASK: PUSH BC ; BC retten
00740 PUSH AF ; Flags retten
00750 LD B,A ; Akku-Inhalt retten
00760 CP 'a' ; unter Klein-"A" ?
00770 JR C,NO ; Ja, kein Kleinbuchstabe!
00780 CP 'ß'+1 ; über "ß" ?
00790 JR NC,NO ; Ja, kein Kleinbuchstabe!
00800 AND 0DFH ; Sonst: Großschrift maskieren.
00810 LD B,A ; Korrigierten Akku-Inhalt retten.
00820 NO: POP AF ; Flags restaurieren
00830 LD A,B ; Akku-Inhalt restaurieren.
00840 POP BC ; BC auch wieder her!
00850 RET ; Fertig.
00860
00870 ;=====
00880 ;LOCOMP Erweiterung der "CP (HL)"-Befehle zur
00890 ; Erkennung von Kleinbuchstaben.
00900 ; (Wird via RST 8H ausgeführt!)
00910 ;=====
00920
00930 LCOMP: PUSH BC ; BC retten
00940 LD B,A ; Vergleichsoperator A in B retten
00950 LD C,(HL) ; Vergleichsoperand => C
00960 LD A,C ; Vergleichsoperand auf
00970 CALL LCMASK ; Großschrift maskieren!
00980 LD (HL),A ; Neuen Operanden schreiben.
00990 LD A,B ; Vergleichsoperator restaurieren.
01000 CALL LCMASK ; dto. wie Vergleichsoperand
01010 CP (HL) ; Vergleich durchführen!
01020 LD (HL),C ; Vergleichsoperand restaurieren
01030 POP BC ; BC restaurieren.
01040 RET ; fertig!
01050
01060 ;=====
01070 ;GETHEX Weitere Modifikation zur Annahme von
01080 ; a ... f als gültige Sedezimal-Codes
01090 ;=====
01100
01110 GETHEX: CALL 6580H
01120 JP LCMASK
01130
01140 ;=====
01150 ;EDIT Modifikation im Zeileneditor zur Annahme
01160 ; von Kleinbuchstaben als Edit-Befehle
01170 ;=====
01180
01190 EDIT: LD HL,6B6DH ; Originalsequenz nachholen
01200 JP LCMASK
01210
01220 ;=====
01230 ;LANDS Modifikation in Load und Save zur
01240 ; Annahme auch der Kürzel d und c.
01250 ;=====
01260
01270 LANDS: CALL 655AH ; Originalsequenz nachholen
01280 JP LCMASK
01290

```



```

01300 ;-----
01310 ;STARS      Modifikation in der Auswertung
01320 ;          des *LIST OFF/ON
01330 ;-----
01340
827C 1A 01350 STARS: LD      A,(DE)
827D CD4882 01360 CALL    LCMASK
8280 FE4C 01370 CP      'L'
8282 C9 01380 RET
01390
01400 ;Neuer Text-Puffer-Beginn:
01410
8283 FFFF 01420 EOP:  DEFW  -1
01430
8285 218882 01440 INIT:  LD      HL,DMPCMD
8288 C31944 01450 JP      4419H
01460
01470 *LEBERKAS OFF
01500 *LEBERKAS ON
01510
01520 ;Zwischen den Leberkäsen steht:
01530
01540 ;DMPCMD:DEFW  'DUMP,EDNEU/CMD,5200H,8FF9H,5780H'
01550 ;          DEFB  0DH
01560
01570 ;-----
01580 ;Es folgen nun direkte Patches im EDTASM+.
01590
01600 ;Obwohl es durchaus genügen würde, lediglich die
01610 ;Befehlsargumente anzupassen, habe ich hier der
01620 ;Verständlichkeit wegen die vollständigen
01630 ;betroffenen Befehls-Sequenzen eingesetzt!
01640 ;-----
01650
01660 ;Umlegen des Text-Beginn-Zeigers hinter EdPatch:
01670
6231 01680 ORG      6231H
6231 218382 01690 LD      HL,EOP
01700
01710 ;RST 8-Zeiger für CP(HL)-Sequenzen verbiegen:
01720
01730 ORG      6237H
01740 LD      HL,LOCOMP
01750 LD      (4001H),HL
01760 LD      HL,4049H
01770 DEFW  0
01780 DEFW  0
01790
01800 ;Direkte Patches in den zugehörigen Teilroutinen:
01810
713D 01820 ORG      713DH
713D CD6A82 01830 CALL    GETHEX
01840
01850 ORG      7113H
7113 CD4582 01860 CALL    GETTYP
01870
01880 ORG      6B8FH
6B8F CD7082 01890 CALL    EDIT
01900
01910 ORG      5A29H
5A29 CD7682 01920 CALL    LANDS
01930
01940 ORG      7567H
7567 CD7C82 01950 CALL    STARS
01960

```

```

658C
658C FE7F

```

```

62DD
62DD CF

```

```

6E69
6E69 CF

```

```

75E6
75E6 CF

```

```

57DA
57DA FE0B

```

```

57DE
57DE FE7F

```

```

6490
6490 0B

```

```

64FC
64FC FE0B

```

```

69AC
69AC 011800

```

```

650D
650D F5
650E 3E1D
6510 CD3300
6513 F1
6514 00

```

```

01970 ;Ändern der Höchstgrenze für gültige Zeichen
01980 ;in Op-Codes von ursprünglich 5BH in 7FH:
01990

```

```

02000 ORG      658CH
02010 CP      7FH
02020

```

```

02030 ;Ersetzen der betreffenden CP (HL)-Befehle durch den RST 8:
02040

```

```

02050 ORG      62DDH
02060 RST      8
02070

```

```

02080 ORG      6E69H
02090 RST      8
02100

```

```

02110 ORG      75E6H
02120 RST      8
02130

```

```

02140 ;Änderungen für's Genie-III:
02150

```

```

02160 COND     GENIE3
02170

```

```

02180 ;Hochtasten mit Hochpfeil an Stelle des "A":
02190

```

```

02200 ORG      57DAH
02210 CP      11
02220

```

```

02230 ORG      57DEH
02240 CP      7FH
02250

```

```

02260 ORG      6490H
02270 DEFB     11
02280

```

```

02290 ORG      64FCH
02300 CP      11
02310

```

```

02320 ;Erhöhen des Zeilenzählers für den P-Befehl
02330 ;von 16 auf 24 Zeilen:
02340

```

```

02350 ORG      69ACH
02360 LD      BC,24
02370

```

```

02380 ;Bei Auf-/Abwärtstasten setzt der Original-EDTASM+ den
02390 ;Cursor durch "AND 0C0H" an den Zeilen-Anfang. Mit der
02400 ;80x24-Darstellung muß dies durch die Folge ...
02410

```

```

02420 ORG      650DH
02430 PUSH    AF
02440 LD      A,1DH
02450 CALL    0033H
02460 POP     AF
02470 NOP
02480

```

```

02490 ; ... ersetzt werden!
02500

```

```

02510 ENDC
02520

```

```

02530 ;-----
02540 ;Zum Schluß - der Vollständigkeit halber - noch einige
02550 ;Uralt-Patches, die vielleicht doch einigen Lesern
02560 ;noch nicht bekannt sind:
02570 ;-----
02580

```

```

02590 ;1. Anpassung der Datumskonversion an das deutsche Format
02600 ; "TT/MM/JJ" für die Kopfzeile des Assembler-Listings:
02610

```

HEFT
20
August
1987

42


```

5EA0      02620      ORG      5EA0H
5EA0 214640 02630      LD      HL,4046H      ; statt LD HL,4045H
                    02640
5EB2      02650      ORG      5EB2H
5EB2 2B     02660      DEC      HL      ; statt INC HL
                    02670
5EC5      02680      ORG      5EC5H
5EC5 00     02690      NOP              ; statt DEC HL
                    02700
                    02710 ;2. Erweiterung der Fehlermeldung TRSDOS ERROR xx durch
                    02720 ; Ausgabe des zugehörigen Fehlertextes:
                    02730
595B      02740      ORG      595BH
595B CD1D62 02750      CALL     621DH
                    02760
621D      02770      ORG      621DH
621D E680   02780      AND      80H
621F CD0944 02790      CALL     4409H
6222 212662 02800      LD      HL,NEWTXT
6225 C9     02810      RET
                    02820
6226 45     02830 NEWTXT: DEFM 'Err.'
6227 72
6228 72
6229 2E
                    02840
8285      02850      END      INIT
00000 TOTAL ERRORS

DMPCMD 828B
EDIT   8270
EDP    8283
GENIE3 0001
GETHEX 826A
GETTYP 8245
INIT    8285
LANDS   8276
LCMASK  8248
LOCOMP  825A
NEWTXT  6226
NO      8256
STARS   827C
ZBUG    0000

```



Vielleicht sollten wir doch lieber den Computer rausschmeißen
und den Chef zurückholen.

Um was geht's eigentlich?

MSDOS-Verwöhnte können dem R-Befehl unseres NEWDOS-/ GDOS-Betriebssystems nur ein müdes Lächeln abringen, haben sie doch mit den Funktionstasten F1 bis F4 nicht nur die Möglichkeit, DOS-Befehle zu wiederholen, sondern vorher noch beliebig zu ändern. Andererseits besitzt die BASIC-Version unserer Betriebssysteme mit dem EDIT-Befehl ein Werkzeug, das die Möglichkeiten des MSDOS weit in den Schatten stellen würde, wäre es auch unter DOS anwendbar!

Ein Editor fürs DOS!

Einer einfachen Implementation in die "Befehlseingabe" - also auf DOS-Ebene - der Interpreter-ROM-Routinen steht nun aber im Wege, daß diese Routinen doch zu sehr ihrer Verwendung als BASIC-Editor nahestehen: So werden zum Beispiel die BASIC-I/O-Routinen 0384H und 032AH an Stelle von 0049H bzw. 0033H hierin verwendet, die bekanntermaßen ohne Initialisierung der BASIC-Vektoren im Niemandsland verschwinden. Darüber hinaus nutzt zum Beispiel der I(nsert)-Unterbefehl die Interpreter-Routine an 1958H, die zwar brav Zeichen einfügt, jedoch aber anhand der BASIC-Programmzeiger durchaus auch einen ?OM Error zum Besten geben kann.

Aus diesen und weiteren Gründen ist schon etwas mehr Aufwand nötig, um MSDOS ein Schnippchen zu schlagen: Hier bietet sich an, den an sich schon vorhandenen R-Befehl des NEWDOS entsprechend aufzubohren und einem diesem Verwendungszweck angepaßten Zeileneditor eine /SYS-Datei zu spendieren. Wie das Einrichten einer /SYS-Datei und der zugehörige Patch in SYS1/SYS genau aussieht, mag der an Einzelheiten interessierte Leser im

"DOS-Buch",
Verlag Röckrath-Microcomputer,
Anhang "ID - Eine Idee zum NEWDOS/80"

nachlesen! Sollte das dort beschriebene Hilfsprogramm in SYS22/SYS bzw. SYS15/SYS bereits installiert sein, so beachten Sie bitte die diesbezüglichen Hinweise im Assemblerlisting. (Wer aber SYS22/SYS für seine eigenen DOS-Erweiterungen bereits "verschwendet" hat, sei auf SYS24/SYS aufmerksam gemacht: Das V24-Testprogramm wird wohl kaum einmal als System-Overlay benötigt werden!)

SYS1/SYS patchen

SYS1/SYS besitzt zwar in seiner Sprungleiste einen Eintrag für den Befehl R, der im Normalfall jedoch nie benötigt wird. Zwei Besonderheiten machen diesen Eintrag aber notwendig: Zum Einen braucht der Befehl LIB (GDOS auch:?) diesen Eintrag, um auch den R-Befehl anzuzeigen, zum Anderen wird bei gewählter System-Option BE=N - also: "R-Befehl verboten" - über diesen Eintrag die Fehlermeldung "unzulässige DOS-Funktion" des SYS1/SYS angesprochen. Der tatsächliche R-Befehl besteht dagegen lediglich darin, die beiden ersten Zeichen des DOS-Befehls-puffers - hier steht ja jetzt R,<cr> - zu restaurieren und mit dem so wieder hergestellten vorherigen Befehl normal fortzufahren.

Diese Tatsache, daß zur Beachtung der System-Option BE=N bereits ein an sich unnötiger Eintrag in der SYS1-Sprungleiste zur Verfügung steht, nutzen wir aus, um mit dem R-Befehl den Editor in SYS22/SYS zu aktivieren:

Alte BE-Option-Behandlung:

```
4E5F 3A6D43 SEARCH: LD HL,(FLAG4) ; System-Option BE:
4E62 CB6F BIT 5,A ; Bit 5 gelöscht =>
4E64 280F JR 2,NOCMDR ; Befehl "R" nicht erlaubt
```

Neue BE-Option-Behandlung:

```
4E5F 3A6D43 SEARCH: LD A,(FLAG4) ; System-Option BE wird erst
4E62 CB6F BIT 5,A ; in SYS22/SYS geprüft =>
4E64 180F JR NOCMDR ; Befehl "R" hier nicht beachten!
```

Alter Eintrag in der Sprungleiste:

```
5066 52 DEFM 'R' ; Befehlskürzel
5067 80 DEFB 80H ; Entry-Counter 0 in
5068 23 DEFB 23H ; SYS1/SYS => Fehlermeldung
```

Neuer Eintrag in der Sprungleiste:

```
5066 52 DEFM 'R' ; Befehlskürzel
5067 80 DEFB 81H ; Entry-Counter 1 in
5068 23 DEFB 0F8H ; SYS22/SYS => R-Editor
; (bzw. DEFB 0FAH für SYS24/SYS)
```

Nötige Patches in SYS1/SYS:

```
DRV 00 E311 D649 0102 004E C5D5 2A18 4322 A743 ...I...N...C*.C
0 10 064F 2118 43C3 4000 D1ED 7B9D 43CB 6928 .D!.C.$...ä.C.i(
0H 20 04CB 6020 BFC1 78B1 2003 23CB B6CB A6ED ..\...x...#.....
30 439D 43D5 F1C3 3351 31E0 41F5 F101 2D40 C.C...3Q1.A...-§
DRS 40 1108 44C5 D5EB 216A 43CB AE21 1843 0650 ..D...!jC...!.C.P
143150 7EFE 0DCB E51A 13CD B545 FE0D 7723 2807 B.....E..w#(.
597H60 10F3 F13E 3687 C93A 6D43 CB6F 180F 2A18 ...>6...mC.o...*. statt 28
70 4311 520D DF20 062A A743 2218 43E1 1158 C.R....*.C*.C.X
80 4FE5 1ABE 1323 28FA 2B1B 0730 05CD D54C 0....#(+..0...L
90 301D E11A 0713 30FB 1313 1AB7 20E3 0143 0....0.....C
A0 E416 417E FE2A 2010 3EEB 0E07 EFC1 C9C1 ..AP.*...).....
B0 1A4F 131A 4713 1A57 CB71 280F 3A6A 4307 .D..G..W.q(.!jC.
FRS C0 3009 E680 3E38 C28B 4DB7 C979 E61F 4FC5 0....>8..M..y..0.
1 D0 4A79 E6C0 C465 512D 20CB 6928 20CD D94C Jy...eQ...i(...L
1H E0 38CB C501 C551 CDC5 4CC1 2005 CDD9 4C38 8....Q...L.....LB
F0 BCD5 11E0 51CD 6851 D13E 3020 B0CB 61C4 ....Q.hQ.>0...a.
F0 BCD5 11E0 51CD 6851 D13E 3020 B0CB 61C4 ....Q.hQ.>0...a.
```

```
DRV 00 0049 802A 0049 4E46 4FB1 FF00 0102 0050 .I.*.INFO.....P
0 10 4A4B 4CB0 A510 4B49 4C4C 8045 904C 4388 JKL...KILL.E.LC.
0H 20 E500 4C46 81FE 004C 4942 82E3 004C 4953...LF...LIB...LIS
30 5485 F08B 4C4F 4144 80A4 504C 5754 81F9 T...LOAD..PLWT..
DRS 40 004E 81E4 804E 4446 C02B 0050 4155 5345 .N...NDF.(.PAUSE
143350 88EB 0050 4483 E900 504F 5254 82FF 0050 ...PD...PORT...P
599H60 5249 4E54 86F0 8850 524F 5486 E900 5055 RINT...PROT...PU statt:
70 5247 4589 E900 5281 F800 5381 E900 5354 RGE...R...S...ST 80 23
80 4D54 89EB 0055 4852 82E5 0056 2884 E500 MT...UHR...U+... (SYS24/SYS:
90 5632 3480 FA00 5A86 FF00 5A45 4954 8AE9 U24...Z...ZEIT.. 81 FA)
A0 0026 83E5 0021 83EB 8A3B 86E3 002F 85E3 .&...!...;.../..
B0 003F 82E3 003E C048 004D 3E82 EBB0 2323 .?....).H.M)...##
FRS C0 82F9 0046 2380 FB00 3830 83F9 0036 3484 ...F#...80...64.
3 D0 F900 4444 4581 F100 4944 80F8 0000 0021 ..DDE...ID.....!
3H E0 584F 0E40 0608 7ECB 7F23 2005 CDB7 5110 X0.$...P.#....Q.
F0 F523 237E B7CA B551 0DCC B551 28E4 CDAD .##...Q...Q(...
```


Der "neue" Editor

In SYS22/SYS muß dann natürlich genau diese Restauration der beiden ersten Zeichen nachgeholt werden, bevor die so vorbereitete Zeile dem EDIT übergeben wird. Da - wie oben beschrieben - der BASIC-Zeileneditor für diesen speziellen Zweck sowieso neu geschrieben werden muß, spricht nichts dagegen, bei dieser Gelegenheit gleich einige Verbesserungen einzubauen:

- Die Edit-Unterbefehle K(ill) und D(elete) klammern nicht wie im Original die gelöschten Zeichen mit Ausrufungszeichen; diese werden vielmehr sofort gelöscht und der Rest der Zeile entsprechend eingerückt.
- Ebenso geschieht I(nsert) nicht mehr "blind", sondern es wird der Rest der Zeile auf dem Bildschirm sichtbar ausgerückt.
- Bei Aufruf des R-Befehls gibt Edit selbsttätig einmal den L(ist)-Befehl.
- Der nicht-zerstörende-Backspace zur Positionierung des Cursors rückwärts löscht nun auch im Bildschirm keine Zeichen: Die zu ändernde Zeile bleibt somit immer vollständig mit sämtlichen aktuellen Änderungen sichtbar!
- Der Befehl C(hange) verbietet nun die Annahme von Steuerzeichen; im "alten" Editor fehlte diese Notbremse mit dem Mißerfolg, daß unsichtbare Steuerzeichen häufig unerklärliche Fehlermeldungen erzeugten!

Neben diesen "Schönheitsreparaturen" am Zeileneditor fand ich es sinnvoll, einen weiteren Befehl R(estore) den Edit-Unterbefehlen hinzuzufügen, da der Original-DOS-Befehl "R" die unschöne Eigenart besitzt, nach einem zufälligen ungewollten Zeilenvorschub auf DOS-Ebene den - dann vorletzten - Befehl zu "vergessen". Dies mag folgendes Beispiel verdeutlichen:

So soll's klappen:

Gegebener Befehl: Das steht im DOS-Eingabepuffer: Das sind die 2 ersten Zeichen:

```
DIR,0,I,S,A(cr)  DIR,0,I,S,A(cr)      DI
R(cr)           vorher: R(cr)R,0,I,S,A  DI
                nachher: DIR,0,I,S,A
```

So kann's aber passieren:

Gegebener Befehl: Das steht im DOS-Eingabepuffer: Das sind die 2 ersten Zeichen:

```
DIR,0,I,S,A(cr)  DIR,0,I,S,A(cr)      DI
versehentlich:   <cr>IR,0,I,S,A        <cr>I
dann:            <cr>IR,0,I,S,A        <cr>I
R(cr)           vorher: R(cr)R,0,I,S,A  <cr>I
                nachher: <cr>IR,0,I,S,A
```

Auf diese Weise hat DOS dann trotz Restauration der ersten beiden Zeichen den Befehl "vergessen", da das versehentlich gegebene <cr> (NEW LINE, ENTER) zu Beginn im DOS-Eingabepuffer verbleibt. Der oben erwähnte zusätzliche Edit-Unterbefehl R(estore) ermöglicht nun, eben dieses <cr> zu löschen, um anschließend das verlorengegangene "D" des Ursprungsbefehls wieder "davorzuhängen".

Alle weiteren Einzelheiten zum komfortablen R-Befehl entnehme der interessierte Leser bitte den Kommentarzeilen des Assemblerlistings!

```
00000 title R-Editor in SYS22/SYS
00010
00020
00030 ;*****
00040 ;*
00050 ;*   SYS22/SYS   Erweiterung des NEWDOS/6DOS-Befehls
00060 ;*   "R" um den bekannten BASIC-Zeileneditor
00070 ;*
00080 ;*****
00090
00100 ;-----
00110 ;
00120 ; Soll diese Befehlserweiterung zusätzlich zum
00130 ;
00140 ; "ID - Eine Idee zum NEWDOS/80"
00150 ;
00160 ; in SYS22/SYS untergebracht werden, so sind im Hilfsprogramm ID
00170 ; folgende Änderungen vorzunehmen:
00180 ;
00190 ; Alle Marken "EXIT" sind wegen des gleichlautenden Labels im
00200 ; Editor durch "IDEXIT" zu ersetzen!
00210 ;
00220 ; Die Zuweisung
00230 ;
00240 ; INCHAR      EQU      0049H
00250 ;
00260 ; ist aus dem ID-Kopf zu entfernen, da dieses Label auch
00270 ; im Editor Verwendung findet!
00280 ;
00290 ; Ab der Marke "ENDE" sind folgende Zeilen hinzuzufügen:
00300 ;
00310 ;ende: xor      a          ; System-Overlay als nicht aktiv
00320 ;       ld      (4317h),a  ; markieren, da Sektorpuffer den
00330 ;       pop     af         ; Editor überschreibt
00340 ;       ret
00350 ;
00360 ; An Stelle von
00370 ;
00380 ;buffer:dfs 256+16
00390 ;
00400 ; Tritt nun
00410 ;
00420 ;buffer equ $
00430 ;
00440 ; Die Marke "START" ist durch "S22ID" zu ersetzen;
00450 ; die Direktive END START entfällt selbstverständlich; ab hier
00460 ; ist stattdessen der Editor anzufügen!
00470 ;
00480 ;-----
00490
00495 page
```

Ulrich Heidenreich
Werderstr. 35
4300 Essen 1


```

4D00      00500
          00510      org      4d00h
          00520
4D00 0C   00530 start: inc      c          ; Entry-Counter prüfen:
4D01 0D   00540      dec      c
          00550
          00560 ; Diese Zeile nur einfügen, falls ID ebenfalls in SYS22/SYS:
          00570 ;
          00580 ;      jr      z,s22id      ; 0 = ID !
          00590 ;
4D02 0D   00600      dec      c          ; 1 = R !
4D03 CA204D 00610      jp      z,s22r
4D04 3E2A   00620 ildos: ld      a,2ah      ; Weder noch:
4D08 B7     00630      or      a          ; "Illegal DOS Function"
4D09 C9     00640      ret              ; melden!
          00650
4467      00660 outext equ      4467h      ; Text (HL) bis 0dh,3 ausgeben
4318      00670 cmdbuf equ      4318h      ; DOS-Befehlspuffer
43A7      00680 twocha equ      43a7h      ; 1. beiden Zeichen statt R,<cr>
          00690
          00700 ; -----
          00710 ;      Edit-Unterbefehle D(elete) und R(estore)
          00720 ; -----
          00730
4D0A 7E     00740 delete: ld      a,(hl)      ; Zeilen-Ende
4D0B FE0D   00750      cp      0dh          ; erreicht?
4D0D CB     00760      ret      z          ; ja => das darf nicht gelöscht werden!
          00770
4D0E 54     00780 restore: ld      d,h      ; Bei Restore jedoch!
4D0F 5D     00790      ld      a,l      ; DE: Lösch-Stelle
          00800      inc      de          ; plus einem Zeichen!
4D10 13     00810 delop: ld      a,(de)      ; Zeichen von eins hinter Lösch-Stelle
4D11 1A     00820      dec      de          ; an Lösch-Stelle
4D12 1B     00830      ld      (de),a      ; umschauflern!
4D13 12     00840      inc      de
4D14 13     00850      inc      de
4D15 13     00860      cp      0dh          ; Zeilen-Ende umgeschauflert ?
4D16 FE0D   00870      jr      nz,delop      ; Nein, weiter schauflern!
4D18 20F7   00880      call   dispt      ; Ja, Rest der Zeile anzeigen.
4D1A CD924E 00890      djnz   delete      ; Replikator beachten!
4D1D 10EB   00900      ret              ; Fertig!
4D1F C9     00910
          00920 ; -----
          00930 ;      Start-Ansprung des Editors. D(elete)
          00940 ;      und R(estore) beginnen davor, um
          00950 ;      für Relativ-Sprünge günstige Sprung-
          00960 ;      weiten zu erzielen!
          00970 ; -----
          00980
4D20 3A6D43 00990 s22r: ld      a,(436dh)      ; System-Option BE ?
4D23 CB6F   01000      bit      5,a          ; Ja, dann R-Befehl
4D25 CA064D 01010      jp      z,ildos      ; nicht erlaubt!
          01020
4D28 3E3E   01030      ld      a,3eh      ; Xtend/Hack-Flag auf nicht
4D2A 323E4E 01040      ld      (hxflag),a      ; zerstörenden backspace setzen.
4D2D 2AA743 01050      ld      hl,(twocha)      ; 1. beide Zeichen des DOS-Befehls
4D30 221843 01060      ld      (cmdbuf),hl      ; restaurieren, da durch R,<cr>
          01070
          01080 ; -----
          01090 ;      Neu-Start für Edit-
          01100 ;      Unterbefehl A(gain)
          01110 ; -----
          01120
4D33 211843 01130 again: ld      hl,cmdbuf      ; überschrieben. DOS-Puffer in
4D36 11C14E 01140      ld      de,edibuf      ; Edit-Puffer

```

```

4D39 D5     01150      push   de          ; übertragen.
4D3A 015000 01160      ld      bc,80
4D3D ED80   01170      ldir
4D3F E1     01180      pop     hl          ; Edit-Puffer-Inhalt
          01190
          01195      page

```

```

00000
00010      title   R-Editor in SYS22/SYS
00020
00030 ;*****
00040 ;*
00050 ;*      SYS22/SYS      Erweiterung des NEWDOS/GDOS-Befehls *
00060 ;*      "R" um den bekannten BASIC-Zeileneditor *
00070 ;*
00080 ;*****
00090
          00100 ; -----
          00110 ;
00120 ; Soll diese Befehlserweiterung zusätzlich zum
          00130 ;
00140 ; "ID - Eine Idee zum NEWDOS/80"
          00150 ;
00160 ; in SYS22/SYS untergebracht werden, so sind im Hilfsprogramm ID
          00170 ; folgende Änderungen vorzunehmen:
          00180 ;
00190 ; Alle Marken "EXIT" sind wegen des gleichlautenden Labels im
          00200 ; Editor durch "IDEXIT" zu ersetzen!
          00210 ;
          00220 ; Die Zuweisung
          00230 ;
00240 ; INCHAR      EQU      0049H
          00250 ;
00260 ; ist aus dem ID-Kopf zu entfernen, da dieses Label auch
          00270 ; im Editor Verwendung findet!
          00280 ;
          00290 ; Ab der Marke "ENDE" sind folgende Zeilen hinzuzufügen:
          00300 ;
          00310 ;ende: xor      a          ; System-Overlay als nicht aktiv
          00320 ;      ld      (4317h),a      ; markieren, da Sektorpuffer den
          00330 ;      pop     af          ; Editor überschreibt
          00340 ;      ret
          00350 ;
          00360 ; An Stelle von
          00370 ;
          00380 ;buffer:defs      256+16
          00390 ;
          00400 ; Tritt nun
          00410 ;
          00420 ;buffer equ      $
          00430 ;
          00440 ; Die Marke "START" ist durch "S22ID" zu ersetzen;
          00450 ; die Direktive END START entfällt selbstverständlich; ab hier
          00460 ; ist stattdessen der Editor anzufügen!
          00470 ;
          00480 ; -----
          00490
          00495      page

```



```

00500
00510 org 4d00h
00520
4D00 0C 00530 start: inc c ; Entry-Counter prüfen:
4D01 0D 00540 dec z
00550
00560 ; Diese Zeile nur einfügen, falls ID ebenfalls in SYS22/SYS:
00570 ;
00580 ; jr z,s22id ; 0 = ID !
00590 ;
00600 ; 1 = R !
4D02 0D 00610 dec c
4D03 CA204D 00620 jp z,s22r
4D06 3E2A 00630 ildos: ld a,2ah ; Weder noch:
4D08 B7 00640 or a ; "Illegal DOS Function"
4D09 C9 00650 ret ; melden!
00660
4467 00660 outext equ 4467h ; Text (HL) bis 0dh,3 ausgeben
4318 00670 cmdbuf equ 4318h ; DOS-Befehlsbuffer
43A7 00680 twocha equ 43A7h ; 1. beiden Zeichen statt R,<cr>
00690
00700 ; -----
00710 ; Edit-Unterbefehle D(elete) und R(estore)
00720 ; -----
00730
4D0A 7E 00740 delete: ld a,(hl) ; Zeilen-Ende
4D0B FE0D 00750 cp 0dh ; erreicht?
4D0D C8 00760 ret z ; ja => das darf nicht gelöscht werden!
00770
4D0E 34 00780 restore: ld d,h ; Bei Restore jedoch!
4D0F 5D 00790 ld e,l ; DE:Lösch-Stelle
4D10 13 00800 inc de ; plus einem Zeichen!
4D11 1A 00810 delop: ld a,(de) ; Zeichen von eins hinter Lösch-Stelle
4D12 1B 00820 dec de ; an Lösch-Stelle
4D13 12 00830 ld (de),a ; umschaufern!
4D14 13 00840 inc de
4D15 13 00850 inc de
4D16 FE0D 00860 cp 0dh ; Zeilen-Ende umgeschauferlt?
4D18 20F7 00870 jr nz,delop ; Nein, weiter schaufern!
4D1A CD924E 00880 call dispt ; Ja, Rest der Zeile anzeigen.
4D1D 10EB 00890 djnz delete ; Replikator beachten!
4D1F C9 00900 ret ; Fertig!
00910
00920 ; -----
00930 ; Start-Ansprung des Editors. D(elete)
00940 ; und R(estore) beginnen davor, um
00950 ; für Relativ-Sprünge günstige Sprung-
00960 ; weiten zu erzielen!
00970 ; -----
00980
4D20 3A6D43 00990 s22r: ld a,(436dh) ; System-Option BE ?
4D23 CB6F 01000 bit 5,a ; Ja, dann R-Befehl
4D25 CA064D 01010 jp z,ildos ; nicht erlaubt!
01020
4D28 3E3E 01030 ld a,3eh ; Xtend/Hack-Flag auf nicht
4D2A 323E4E 01040 ld (hxflag),a ; zerstörenden backspace setzen.
4D2D 2AA743 01050 ld hl,(twocha) ; 1. beide Zeichen des DOS-Befehls
4D30 221843 01060 ld (cmdbuf),hl ; restaurieren, da durch R,<cr>
01070
01075 page

```

```

01080 ; -----
01090 ; Neu-Start für Edit-
01100 ; Unterbefehl A(gain)
01110 ; -----
01120
4D33 211843 01130 again: ld hl,cmdbuf ; überschrieben. DOS-Puffer in
4D36 11C14E 01140 ld de,edibuf ; Edit-Puffer
4D39 05 01150 push de ; übertragen.
4D3A 015000 01160 ld bc,80
4D3D EDB0 01170 ldir
4D3F E1 01180 pop hl ; Edit-Puffer-Inhalt
01190
01200 ; -----
01210 ; Neu-Start für Edit-
01220 ; Unterbefehl L(ist)
01230 ; -----
01240
4D40 CD6744 01250 redo: call outext ; ausgeben
4D43 3E0E 01260 ld a,0eh ; Cursor einschalten
4D45 CDB34E 01270 call outcha
4D48 3E1B 01280 ld a,27 ; Cursor auf Zeilenanfang
4D4A CDB34E 01290 call outcha ; setzen.
01300
01310 ; -----
01320 ; Edit-Hauptschleife
01330 ; -----
01340
4D4D 114D4D 01350 edit: ld de,edit ; Rückkehradresse
4D50 05 01360 push de ; => Stack: d.h. RET immer hierhin!
4D51 0600 01370 ld b,0 ; Replikator löschen
4D53 CDBB4E 01380 inloop: call inchar ; Befehl/Replikator lesen
4D56 D630 01390 sub '0' ; ASCII-Korrektur
4D58 380E 01400 jr c,edcmd ; Kleiner 0 => Befehl
4D5A FE0A 01410 cp 10 ; Größer 9
4D5C 300A 01420 jr nc,edcmd ; ebenfalls Befehl
4D5E 4F 01430 ld c,a ; Replikator-Ziffer => C
4D5F 7B 01440 ld a,b ; A:alter Replikator
4D60 07 01450 rlca ; mal 2
4D61 07 01460 rlca ; mal 4
4D62 80 01470 add a,b ; plus 1 => mal 5
4D63 07 01480 rlca ; mal 10
4D64 81 01490 add a,c ; plus Replikator-Ziffer
4D65 47 01500 ld b,a ; => Replikator in Register B
4D66 18EB 01510 jr inloop ; weiter einlesen.
01520
01530 ; -----
01540 ; Ansprung der Edit-Unterbefehle
01550 ; -----
01560
4D68 05 01570 edcmd: dec b ; Replikator = 0
4D69 04 01580 inc b ; ?
4D6A 2001 01590 jr nz,rgz ; Nein, OK
4D6C 04 01600 inc b ; sonst: Replikator = 1
4D6D C630 01610 rgz: add a,'0' ; ASCII-Korrektur rückgängig machen.
4D6F FE0D 01620 cp 0dh ; NEW LINE ?
4D71 284D 01630 jr z,enter
4D73 FE08 01640 cp 08h ; BACKSPACE ?
4D75 2848 01650 jr z,bsp
4D77 FE20 01660 cp 20h ; SPACE ?
4D79 2859 01670 jr z,space
4D7B CBAF 01680 res 5,a ; Kleinschrift herausmaskieren.
4D7D FE44 01690 cp 'D' ; D(elete) ?
4D7F 2889 01700 jr z,delete
4D81 FE52 01710 cp 'R' ; R(estore) ?
4D83 2889 01720 jr z,restore

```



```

4D85 FE51 01730 cp 'Q' ; Q(uit) ?
4D87 2864 01740 jr z,quit
4D89 FE4C 01750 cp 'L' ; L(ist) ?
4D8B 2825 01760 jr z,list
4D8D FE53 01770 cp 'S' ; S(earch) ?
4D8F 2860 01780 jr z,search
4D91 FE49 01790 cp 'I' ; I(nsert) ?
4D93 CA1D4E 01800 jp z,insert
4D96 FE43 01810 cp 'C' ; C(hange) ?
4D98 CA804E 01820 jp z,change
4D9B FE45 01830 cp 'E' ; E(xit) ?
4D9D 2824 01840 jr z,exit
4D9F FE58 01850 cp 'X' ; X(tend) ?
4DA1 2861 01860 jr z,xtend
4DA3 FE4B 01870 cp 'K' ; K(ill) ?
4DA5 CA6B4E 01880 jp z,kill
4DAB FE48 01890 cp 'H' ; H(ack) ?
4DAA 2863 01900 jr z,hack
4DAC FE41 01910 cp 'A' ; A(fter) ?
4DAE C0 01920 ret nz ; nein, zurück nach "edit"
4DAF D1 01930 pop de ; Rückkehrradresse vom Stack
4DB0 1881 01940 jr again ; "Same procedure as last year!"
01950
01960 ; -----
01970 ; Edit-Unterbefehl L(ist)
01980 ; -----
01990
4DB2 D1 02000 list: pop de ; siehe Kommentar zu quit!
4DB3 CD6744 02010 call outext ; (rest der) Zeile ausgeben
4DB6 3E0D 02020 ld a,0dh ; Zeilenvershub
4DB8 CDB34E 02030 call outcha ; ausgeben
4DBB 21C14E 02040 ld hl,edibuf ; Mit Zeilenbeginn Editor
4DBE 1880 02050 jr redo ; erneut starten.
02060
02070 ; -----
02080 ; Edit-Unterbefehle ENTER / NEW LINE
02090 ; und E(xit)
02100 ; -----
02110
4DC0 CD6744 02120 enter: call outext ; (Rest der) Zeile ausgeben
4DC3 21C14E 02130 exit: ld hl,edibuf ; Inhalt des Edit-Puffers
4DC6 111843 02140 ld de,cmdbuf ; in den DOS-Befehlspuffer
4DC9 D5 02150 push de ; (Adresse retten)
4DCA 015000 02160 ld bc,B0 ; kopieren.
4DCD EDB0 02170 ldir
4DCF E1 02180 pop hl ; Befehl aus diesem
4DD0 D1 02190 pop de ; - nach Stackkorrektur! -
4DD1 C31944 02200 jp 4419h ; ans DOS übermitteln!
02210
02220
02230 ; -----
02240 ; Edit-Unterbefehl "Leertaste"
02250 ; -----
02260
4DD4 7E 02270 space: ld a,(hl) ; Ende der Zeile
4DD5 FE0D 02280 cp 0dh ; erreicht?
4DD7 C8 02290 ret z ; ja, "rechter" geht's nicht!
4DD8 23 02300 inc hl ; sonst: Textzeiger weiter
4DD9 CDB34E 02310 call outcha ; Zeichen ausgeben
4DDC 10F6 02320 djnz space ; evtl. Replikator beachten.
4DDE C9 02330 ret ; Fertig!
02340
02345 page

```

```

02350 ; -----
02360 ; Edit-Unterbefehl "Rückpfeil"
02370 ; -----
02380
4DDF 11C14E 02390 bsp: ld de,edibuf ; Anfang der Zeile
4DE2 DF 02400 bsplop: rst 18h ; erreicht?
4DE3 C8 02410 ret z ; ja "backer" gehts nicht!
4DE4 2B 02420 dec hl ; sonst: Textzeiger zurück
4DE5 3E18 02430 ld a,24 ; dto. Cursor
4DE7 CDB34E 02440 call outcha
4DEA 10F6 02450 djnz bsplop ; Replikator ?
4DEC C9 02460 ret
02470
02480 ; -----
02490 ; Edit-Unterbefehl Q(uit)
02500 ; -----
02510
4DED D1 02520 quit: pop de ; Rückkehrradresse nach edit vom Stack
4DEE C30044 02530 jp 4400h ; alles ignorieren!
02540
02550 ; -----
02560 ; Edit Unterbefehl S(earch)
02570 ; -----
02580
4DF1 CDBB4E 02590 search: call inchar ; Was sollen wir suchen?
4DF4 5F 02600 ld e,a ; Such-Zeichen => E
4DF5 7E 02610 sloop: ld a,(hl) ; Zeilen-Ende bei Suche
4DF6 FE8D 02620 cp 0dh ; erreicht?
4DF8 C8 02630 ret z ; Ja, nicht gefunden, zurück nach edit!
4DF9 CDB34E 02640 call outcha ; sonst: Zeichen ausgeben
4DFC 23 02650 inc hl ; Textzeiger weiterstellen
4DFD 7E 02660 ld a,(hl) ; Suchzeichen
4DFE 8B 02670 cp e ; erreicht?
4DFF 20F4 02680 jr nz,sloop ; Nein, weiter suchen!
4E01 10F2 02690 djnz sloop ; Wenn gefunden, Replikator beachten!
4E03 C9 02700 ret ; Dann aber fertig!
02710
02720 ; -----
02730 ; Edit-Unterbefehle X(tend) und H(ack)
02740 ; -----
02750
4E04 7E 02760 xtend: ld a,(hl) ; Zeilen-Ende
4E05 FE0D 02770 cp 0dh ; erreicht?
4E07 280D 02780 jr z,bspdel ; Ja, dann hier wie I(nsert) weiter!
4E09 CDB34E 02790 call outcha ; Sonst: Zeichen ausgeben
4E0C 23 02800 inc hl ; Textzeiger weiter
4E0D 18F5 02810 jr xtend ; Zeilen-Ende suchen!
02820
4E0F 360D 02830 hack: ld (hl),0dh ; Genau hier Zeilenende einsetzen
4E11 3E1E 02840 ld a,30 ; Im Bildschirm ab hier löschen
4E13 CDB34E 02850 call outcha ; weiter wie I(nsert), jedoch
02860
4E16 3E36 02870 bspdel: ld a,36h ; bei Hack/Xtend zerstörenden
4E18 323E4E 02880 ld (hxflag),a ; Backspace aktivieren.
4E1B 1805 02890 jr bcont
02900
02905 page

```

HEFT
20
August
1987


```

02910 ;-----
02920 ; Edit-Unterbefehl I(nsert)
02930 ;-----
02940
4E1D 3E3E 02950 insert: ld a,3eh ; Bei Insert: Statt Backspace nur
4E1F 323E4E 02960 ld (hxflag),a ; Cursor-Rückstellung erlauben.
02970
4E22 CDBB4E 02980 bcont: call inchar ; Was wollen wir I(nserten) ?
4E25 FE0D 02990 cp 0dh ; NEW LINE bitteschön nicht einfügen,
4E27 2897 03000 jr z,enter ; sondern wie Edit-Befehl behandeln!
4E29 FE1B 03010 cp 1bh ; ESC = Ende Einfügen
4E2B 2006 03020 jr nz,noesc ; Nein!
4E2D 3E3E 03030 ld a,3eh ; Ja, auf nicht zerstörenden Backspace
4E2F 323E4E 03040 ld (hxflag),a ; zurückschalten
4E32 C9 03050 ret ; und zurück zu edit!
03060
4E33 FE08 03070 noesc: cp 8 ; Backspace?
4E35 2019 03080 jr nz,nobsp ; Nein!
4E37 11C14E 03090 ld de,edibuf ; Geht's "backer" ?
4E3A DF 03100 rst 18h ; Na?
4E3B 28E5 03110 jr z,bcont ; Nein, war wohl nichts!
03120
4E3D 2B 03130 dec hl ; Textzeiger zurückstellen
4E3E 3E0D 03140 hxflag: ld a,0dh ; Bei Hack & Xtend hier: LD (HL),0dh =
4E40 3A3E4E 03150 ld a,(hxflag) ; Zeilenende einsetzen
4E43 FE36 03160 cp 36h ; und Zeichen
4E45 3E08 03170 ld a,8 ; löschen!
4E47 2802 03180 jr z,delbsp ; Sonst nur:
4E49 3E18 03190 ld a,24 ; Cursor zurückstellen
4E4B CDB34E 03200 delbsp: call outcha
4E4E 1BD2 03210 jr bcont ; weiter "inserten"
03220
4E50 FE20 03230 nobsp: cp 20h ; Steuerzeichen?
4E52 3BCE 03240 jr c,bcont ; Bloß nicht einfügen!
4E54 47 03250 ld b,a ; Zeichen retten
4E55 11104F 03260 ld de,edibuf+79 ; Edit-Puffer-Ende
03270
4E58 1B 03280 inslop: dec de ; Ein Zeichen davor
4E59 1A 03290 ld a,(de) ; um
4E5A 1B 03300 inc de ; eine Stelle nach
4E5B 12 03310 ld (de),a ; hinten schieben
4E5C 1B 03320 dec de
4E5D DF 03330 rst 18h ; Einfügstelle erreicht?
4E5E 20F8 03340 jr nz,inslop ; Nein, immer weiter so!
4E60 70 03350 ld (hl),b ; Ja, Zeichen einfügen.
4E61 78 03360 ld a,b ; Auf'm Bildschirm wollen
4E62 CDB34E 03370 call outcha ; wir's aber auch sehen!
4E63 23 03380 inc hl ; Textzeiger weiterstellen.
4E66 CD924E 03390 call dispt ; Rest der Zeile anzeigen.
4E69 1BB7 03400 jr bcont ; weiter "inserten"
03410
03420 ;-----
03430 ; Edit-Unterbefehl K(ill)
03440 ;-----
03450
4E6B CDBB4E 03460 kill: call inchar ; Was soll's denn bitteschön sein?
4E6E 4F 03470 ld c,a ; Das retten wir mal in C
4E6F C5 03480 kiloop: push bc ; Replikator, Zeichen retten
4E70 D601 03490 ld b,1 ; Replikator auf 1 setzen
4E72 CD0A4D 03500 call delete ; 1 Zeichen löschen
4E75 C1 03510 pop bc ; Kill-Replikator, -Zeichen her!
4E76 7E 03520 ld a,(hl) ; welches Zeichen hat "delete"
4E77 FE0D 03530 cp 0dh ; herangeschaufelt? Etwa Ende?
4E79 C8 03540 ret z ; ja, weitermachen = Unsinn!
4E7A B9 03550 cp c ; Ist Zeichen erreicht?

```

```

4E7B 20F2 03560 jr nz,kiloop ; Nein, weiter K(ill)en!
4E7D 10F0 03570 djnz kiloop ; Replikator ?
4E7F C9 03580 ret ; Ende!

```

```

03590 ;-----
03600 ; Edit-Unterbefehl C(hange)
03610 ;-----
03620
03630
4E80 7E 03640 change: ld a,(hl) ; Zeilen-Ende erreicht?
4E81 FE0D 03650 cp 0dh ; Na?
4E83 C8 03660 ret z ; Ja, bloß nicht ändern!
4E84 CDBB4E 03670 call inchar ; Neues Zeichen anfordern,
4E87 FE20 03680 cp ' ' ; aber Steuerzeichen
4E89 D8 03690 ret c ; ignorieren!
4E8A 77 03700 ld (hl),a ; sonst: Zeichen einsetzen
4E8B 23 03710 inc hl ; Textzeiger weiterstellen
4E8C CDB34E 03720 call outcha ; und Zeichen ausgeben.
4E8F 10EF 03730 djnz change ; Wie oft denn noch?
4E91 C9 03740 ret ; Na, also: Fertig!
03750
03760 ;-----
03770 ; UP für I(nsert) und D(elete):
03780 ; Rest der Zeile ausgeben und Cursor
03790 ; an aktuelle Änderungsstelle
03800 ; zurückstellen.
03810 ;-----
03820
4E92 E5 03830 dispt: push hl ; Textzeiger retten
4E93 DE00 03840 ld c,0 ; Zeichenzähler nullsetzen.
4E95 7E 03850 ll: ld a,(hl) ; Textende erreicht?
4E96 FE0D 03860 cp 0dh ; Na?
4E98 2807 03870 jr z,e1 ; Ja!
4E9A CDB34E 03880 call outcha ; Sonst: ein Zeichen der restlichen
4E9D 0C 03890 inc c ; Zeile ausgeben und mitzählen.
4E9E 23 03900 inc hl ; Textzeiger weiterstellen.
4E9F 1BF4 03910 jr ll ; Weiter auf Zeilen-Ende hoffen.
4EA1 3E1E 03920 e1: ld a,30 ; Zeilen-Ende erreicht: Für alle
4EA3 CDB34E 03930 call outcha ; Fälle: Bildschirm säubern.
4EA6 E1 03940 pop hl ; Textzeiger restaurieren.
4EA7 0C 03950 inc c ; Waren überhaupt
4EA8 0D 03960 dec c ; Zeichen übrig?
4EA9 C8 03970 ret z ; Nein, alles so OK!
4EAA 3E1B 03980 ld a,24 ; sonst:
4EAC CDB34E 03990 e2: call outcha ; Cursor auf Edit-Position
4EAF 0D 04000 dec c ; zurück-
4EB0 20FA 04010 jr nz,e2 ; transferieren!
4EB2 C9 04020 ret
04030
04040 ;-----
04050 ; I/O-Unterprogramme
04060 ;-----
04070
4EB3 D9 04080 outcha: exx
4EB4 F5 04090 push af
4EB5 CD3300 04100 call 033h
4EB8 F1 04110 pop af;
4EB9 D9 04120 exx
4EBA C9 04130 ret
04140
4EBB D9 04150 inchar: exx
4EBC CD4900 04160 call 49h
4EBF D9 04170 exx
4EC0 C9 04180 ret
04190
04195 page

```

```

04200 ;-----
04210 ; Edit-Zeilenspuffer
04220 ; (Der Editor arbeitet
04230 ; nicht direkt im DOS-
04240 ; Befehlspuffer, da sonst
04250 ; der A-Befehl nicht
04260 ; möglich wäre!)
04270 ;-----
04280
04290 edibuf: defs 80
04300
04310 end start

```


Tape-Games auf dem Model III/4/4p

Viele der alten Model 1-Spiele sind auch auf dem TRS 80 Model 3 und damit auf dem 4/4p im M.3-Modus lauffähig. Ein großes Problem gab und gibt es aber mit den Spielen, die mit LMOFFSET auf einem Model 1 von Tape auf Disk übertragen wurden. Wenn man die vorgeschriebene Prozedur einhält, stürzt die Maschine ab bzw. bootet neu. Eine sehr einfache Methode, diese Spiele doch zum Laufen zu bringen, soll hier erklärt werden.

Nachdem man auf dem Model 1 das Programm gestartet hat, muß man RESET drücken und gleichzeitig BREAK halten um ins ROM-BASIC zu kommen. Dieser Vorgang bleibt beim Model 3/4/4p natürlich der gleiche. Auch SYSTEM <ENTER> muß man wie gehabt eingeben. Danach kann man aber nicht einfach durch Eingabe von "/" das Programm starten, sondern man muß zusätzlich die Startadresse mit angeben.

Das ist leichter gesagt als getan. Woher soll man die Startadresse nehmen!? Ganz einfach, man sucht sie sich mit SUPERZAP, DDE, FED oder einem ähnlichen Diskettenmonitor aus dem /CMD-File. Die Startadresse findet man direkt vor der Meldung "Do the following:..." (siehe dazu auch Info Nr. 9, Seite 9/10 "Do it yourself - LMOFFSET with Disable DOS"). Dort steht die Bytefolge C3 xx xx, wobei C3 der Hex-Code ist, der die Z80-CPU dazu veranlasst, ihre Arbeit bei der Stelle fortzuführen auf die die beiden folgenden Bytes zeigen. Die eigentliche Startadresse wird also durch die beiden Bytes nach C3 repräsentiert. Sie sind in der, für die Maschinensprache typischen Reihenfolge LowByte/HighByte geschrieben.

Ein Beispiel verdeutlicht das Ganze. In einem Programm findet man die Bytefolge: C3 90 7A. C3 repräsentiert wie wir wissen den Sprung, die Adresse zu der gesprungen wird lautet 7A90h. Nun braucht man nur noch die Adresse ins dezimale Zahlensystem umzurechnen (wozu hat man seinen Computer!?) und, wie oben beschrieben, nach dem "/" als Startadresse anzugeben. Schon läuft (hoffentlich) das Spiel!!!

Damit man sich nicht jedesmal die Adresse wieder neu suchen muß, kann man sie mit dem Zapper in den "Do the ..." -Text mit einbauen. Das ganze kann dann so aussehen!

Tue folgendes:

```
<BREAK> halten und RESET drücken um L2-BASIC zu starten
<BREAK> loslassen und zwei mal <ENTER> drücken
SYSTEM <ENTER> und dann
/53248 <ENTER> eingeben
```

Umkehr !

Unser Clubkamerad Gerald Dreyer stellte mir vor kurzem eine Aufgabe, bei deren Lösung ein Problem auftrat, welches mich ein paar Stunden stark beschäftigte. Da ich glaube, daß dieses Problem und die/meine Lösung zumindest für die an der Maschinensprache Interessierten und die Tüftler unter uns von Interesse ist, will ich beides hier kurz erläutern.

Die mir gestellte Aufgabe bestand darin Datenfiles, wie sie von dem Grafikprogramm DGRAF (erstellt Grafiken aus DBase-Dateien) erzeugt werden, in die HRG des Model 4p einzulesen. Diese Files sind folgendermaßen organisiert:

	y1	y2	y3	y4	y5	y6	y450
x1	:	:	:	:	:	:	:
	:	:	:	:	:	:	:
x2	:	:	:	:	:	:	:
	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:
x40x	:	:	:	:	:	:	:
	:	:	:	:	:	:	:
	:	:	:	:	:	:	:

Wie man deutlich sieht, handelt es sich um ein Datenformat, welches relativ leicht auf Drucker ausgegeben werden kann. 450 Bytes (=Bildpunkte) pro Zeile mit logischerweise jeweils 8 Bit (=Bildpunkte) in der Höhe. Insgesamt ergibt sich also eine Auflösung von $450 * 320$ Bildpunkten. Leider ist das Format für die HRG nur sehr bedingt geeignet. Das hat folgende Gründe:

1. Die HRG ist byteweise in y-Richtung organisiert, das Bild liegt also, wenn man es direkt einliest, auf der Seite.
 2. Die HRG hat zwar eine Auflösung von 640 Punkten in x-Richtung aber nur 240 Punkte in y-Richtung.
- Es sind also zwei Dinge zu tun, um die DGRAF-Files einlesen zu können:
1. das Bild um 90 Grad zu drehen und
 2. die Bildhöhe um die Hälfte verringern (auf 160 Punkte).

Dabei trat nun das eigentliche Problem auf, daß die Datenbytes einzeln "umgekehrt" werden mussten. Beispiel:
Folgendes Byte (1) muß nach der Umkehrung wie (2) aussehen!

1	0	1	2	3	4	5	6	7
Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
(1)	1	0	0	1	0	1	1	0

$$\begin{array}{r} (1) \quad 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \\ (2) \quad 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \end{array}$$

Bit 7 muß also auf Bit 0 verschoben

usw. 0 auf 7. Nochmal ein Beispiel

UZW. 0 auf 7. Nochmal ein Beispiel:
 Bit 7 6 5 4 3 2 1 0

Bit	7	6	5	4	3	2	1	0
(1)	0	1	0	1	0	1	0	1

$$(2) \quad 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0$$

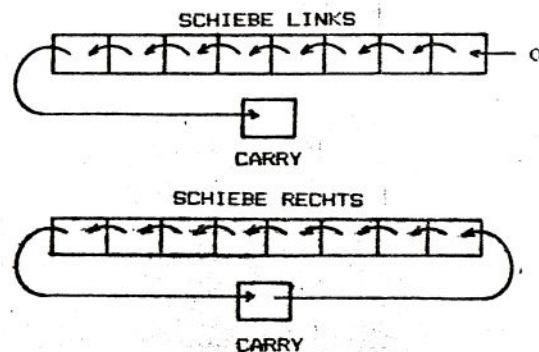
vor der Umkehr Bit 0 1 2 3 4 5 6 7!!!

HEFT
20
August
1987

58

Zunächst gedachte ich, dem Problem mathematisch auf den Grund zu gehen und versuchte allerhand Operationen von der Addition bis zum Negieren, einschließlich aller Kombinationen zusammen mit Schiebefehlen links und rechts (die Problemlösung wurde in TurboPascal gesucht!). Leider brauchte ich sehr lange bis ich zu der Erkenntnis kam, daß es so nicht zu schaffen ist(?). Hätte ich die Lösung gleich in einer Maschinenspracherroutine gesucht, wären mir wahrscheinlich etliche Zweifel an meinem (so hoffe ich) klaren Menschenverstand erspart geblieben. Auf alle Fälle kam ich, nachdem ich eine Hochsprachenlösung verworfen hatte, recht schnell zu folgendem Ergebnis.

Die Z80-CPU (und natürlich auch andere Prozessoren) stellt dem Programmierer verschiedenartige Schiebe- und Rotationsoperationen zur Verfügung. Dabei wird der Inhalt eines Registers oder einer Speicherstelle verändert, indem man den Inhalt nach links oder rechts um jeweils ein Bit verschiebt. Den prinzipiellen Unterschied zwischen Schieben und Rotieren könnt ihr am besten folgender Grafik entnehmen:



Wie ihr seht, wird beim Schieben Bit 0 (7) mit einer 0 aufgefüllt, alle anderen Bits wandern eine Stelle nach links (rechts) wobei Bit 7 (0) aus dem/dem Register/Speicherst. in das Carry-Flag wandert. Wird ein zweitesmal geschoben, geschieht das gleiche, dabei geht das ursprüngliche Bit 7, welches beim ersten Schieben nach CARRY gewandert war, verloren!

Das Rotieren ist im Prinzip nichts weiter als eine Schiebeoperation, bei der allerdings nicht mit einer 0 aufgefüllt wird, sondern mit dem Wert, der in CARRY stand. Wurde diese Operation neunmal durchgeführt, stellt sich das/die Register/Speicherstelle so dar, als wäre nichts weiter passiert!

Die Schiebe- und Rotieroperationen gibt es in einer großen Vielfalt und in unterschiedlichsten Variationen. Über den ganzen Umfang dieser CPU-Befehle und auch über die Anwendungsgebiete möchte ich mich hier nicht auseinandersetzen und verweise daher auf einschlägige Literatur (wie z.B. Programmierung des Z80 von Rodney Zaks, in der Bibliothek des CLUB 80 enthalten).

Zwei der Rotierbefehle des Z80 lauten RL s (Rotiere links durch das Carryflag) und RR s (Rotiere rechts durch das Carryflag). "s" steht dabei für eines der Register (A,B,C,D,E,H,L) oder eine durch HL, IX+d oder IY+d adressierte Speicherstelle. Mit diesen Befehlen und ein wenig drumherum kann man ein Byte sehr schnell "umkehren"! Dazu erst einmal folgendes Listing:

```

1 LD      A,(Byte)
2 LD      B,A
-----
3 RL      B      diese beiden Anweisungen müs-
4 RR      A      sen achtmal wiederholt werden!
-----
5 LD      (Byte),A

```

Die folgende Darstellung macht die Sache wahrscheinlich schon erheblich deutlicher! In Byte soll Beginn die Zahl 202 (Binär 11001010) stehen.

Inhalt von	Register A	Carry Bit	Register B
Nach Anweisung Nr. 1	Bit 7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0
1	1 1 0 0 1 0 1 0	?	?
2	1 1 0 0 1 0 1 0	?	1 1 0 0 1 0 1 0
3	1 1 0 0 1 0 1 0	1	1 0 0 1 0 1 0 ?
4	1 1 1 0 0 1 0 1	0	1 0 0 1 0 1 0 ?
B mal wiederholen!	1 1 1 0 0 1 0 1	1	0 0 1 0 1 0 ? 0
3	1 1 1 1 0 0 1 0	0	0 0 1 0 1 0 ? 0
4	1 1 1 1 0 0 1 0	0	0 1 0 1 0 ? 0 1
3	0 1 1 1 1 0 0 1	0	1 0 1 0 ? 0 1 0
4	0 0 1 1 1 1 0 0	1	1 0 1 0 ? 0 1 0
3	0 0 1 1 1 1 0 0	1	0 1 0 ? 0 1 0 1
4	1 0 0 1 1 1 1 0	0	0 1 0 ? 0 1 0 1
3	1 0 0 1 1 1 1 0	0	1 0 ? 0 1 0 1 0
4	0 1 0 0 1 1 1 1	0	1 0 ? 0 1 0 1 0
3	0 1 0 0 1 1 1 1	1	0 ? 0 1 0 1 0 0
4	1 0 1 0 0 1 1 1	0	0 ? 0 1 0 1 0 0
3	1 0 1 0 0 1 1 1	0	0 1 0 1 0 0 0 0
4	0 1 0 1 0 0 1 1	1	? 0 1 0 1 0 0 0

Nach der letzten Anweisung (5) steht in Byte 83, was, schaut man sich die Zahl einmal binär an (01010011), genau die Umkehrung des Anfangswertes ist! Das Wunder vollbringen die jeweils acht RL- und RR-Operationen, bei denen die Bits, jeweils im Carryflag zwischengespeichert, aus dem Register B in das Register A wandern. Dadurch, daß einmal links und einmal rechts geschoben wird, steht der Wert dann im Zielregister in umgekehrter Reihenfolge zur Verfügung. Im Prinzip ist es übrigens egal, welche Register benutzt werden und ob zuerst links und dann rechts geschoben wird oder umgekehrt. Das Ergebnis bleibt das gleiche!

Natürlich kann man das ganze auch recht elegant in einer Schleife programmieren und muß so nicht acht mal die RL/RR-Operationen hintereinander schreiben. Dabei ist aber zu beachten, daß man bei Verwendung einer Schleife die Ausführungszeit erheblich erhöht. Die Rotierbefehle müssen nämlich auf alle Fälle achtmal durchlaufen werden und zusätzlich kommt dann noch das decrementieren eines Zählers und die Schleifenabfrage dazu. Wo also mehr auf die Ausführungszeit und weniger auf den Platzbedarf geachtet werden muß (was bei der Einbindung in ein TurboPascal-Programm der Fall ist), sollte man die einfache Form einer Schleife vorziehen!

Nun noch ein praktisches Beispiel für die Anwendung dieser Routine. EPSON- und ITOH-Drucker unterscheiden sich in ihrer Grafikfähigkeit besonders darin, daß EPSON die oberste Nadel des Druckkopfes als Bit 7, ITOH jedoch als Bit 0 eines Grafikbytes ansieht. Mit der vorgestellten Routine kann man diese Inkompatibilität ausschalten!

Zuletzt zwei Fragen. Liebe Assembler-Freaks, gibt es eine Möglichkeit das beschriebene Problem schneller zu lösen? Liebe BASIC-, Pascal- und sonstigen Programmierer, gibt es für das vorgestellte Problem eine Hochsprachenlösung, welche in akzeptabler Zeit das gleiche Ergebnis bringt? Viel Spaß beim Bit-Tüfteln, Euer

Karsten Quernann

m4/4p ecke

* Keine Angst vor Haifischen! *

Auch von Leuten, die des Lesens & Schreibens durchaus kundig, liest man oft lange Texte, deren Wörter offenbar starr und "unzerreißbar" sind...

Gemeint ist die Scheu vor Trennungen.

Ist es etwa eine besondere Schwäche von TSCRIPS?

Jahrelang schrieb ich hiermit und ärgerte mich stets über mein Unvermögen, die Zeilen optimaler zu füllen, weil das herrische, aber doch wohlgemeinte TSCRIPS-Kommando "J=Y" (Augen rechts! Rrrricht't euch!!) lieber Super-Vacua zwischen die Wörter bläst als zuzulassen, daß ein einziger Schütze A...h-aha - wollte sagen: Sad Sack, aus der Reihe tanzt. Zugegeben: Mein Unvermögen wurde durch eine höchst mystische sogenannte "Benutzer-Anleitung" legitimiert, die mehr verdunkel- als erhellte. Zu tiefergehendem Studium aber war ich zu faul -

bis Arnulf SOPP,
ganz salopp,
selber ob
dieser Crux mit Recht empört,
mich das Haifischen gelehrt!

Wieso "Hai"?

Lautschrift erleichtert das Leben!

(Stabreie versehentlich....)

"HYPHEN" = engl. "Trennstrich", gesprochen "HAIFN", war es, was da fehlte und den ernstesten Schreiber quälte!

"HYPHENATION" ist jener Vorgang, der in einem ästhetischen Schreibwerk geübt - ja, ich möchte sagen: gepflegt wird. Und für den TSCRIPS nach "BREAK" auf das Befehlskürzel "H" in der Kommando- (oder Status-)zeile wartet!.

Verständlich: Schlechte Erklärungen können einem Angst machen, können einen frustrieren; zumal die Anwendung ein wenig Übung erfordert.

Aber dann, erst gekonnt, ist es ein Genuß!

Ich kann gar nicht genug lange Wörter bilden, Wortschlangenausdrückeherstellungsgier ist seitdem mein neuestes Hobby, um möglichst viele Haifische zu fangen! Wie die gegen ihren Willen getrennten Wörter an den Zeilenenden ihre Zungen rausstrecken, als wollten sie mir zurufen: "Bäh - ...und wir gehören doch zusammen!" - Jawohl, das sollt ihr auch! Aber die Zeilen sind schöner, sind voller geworden. Kein (?) gähhnender gap mehr!

Nun gut: Auch TSCRIPS ist nicht vollkommen. Es gibt nach wie vor Abstände, die mir unverständlich sind; wenn ich z.B. die überschüssig-überflüssigen Abstände zusammenzähle und feststellen muß, daß da das kurze Wort, mit dem die nächste Zeile beginnt, doch noch hineingepaßt hätte... Dies zu erklären, ist mir zu hoch. Sicher kann das einer unserer ML-Experten.

Hier folgt nun eine kurz gefaßte

Gebrauchsanleitung zum Trennen in TSCRIPS.

da Arnulf seine Hilfe sicher nicht auf mich beschränkt wissen wollte:

Trennen in TSCRIPS

- 1) Setze Bildschirmbreite "Width" = W = RM - LM.
- 2) Setze den Cursor an den Anfang des zu trennenden Textes.
- 3) Drücke den Klammeraffen gleichzeitig mit Q und danach den Bindestrich '-'. Hierauf wird der Text vorübergehend ein wenig nach rechts gerückt und es erscheint ein Pfeil nach rechts.
- 4) Jetzt gehe ans Textende und drücke wieder das Klammeräffchen gleichzeitig mit Q, anschließend - bei festgehaltenem Äffchen! - den Abwärtspfeil. (Das geht schneller, als es sich anhört, weil Q und Abwärtspfeil fast beieinander liegen.)
- 5) Drücke die BREAK-Taste, danach das 'H' (für 'Hyphenation' = Trennung). Wenn jetzt ENTER (RETURN oder NEW LINE) gedrückt wird, erscheint die bange Frage nach der gewünschten "Heißen Zone". Das bedeutet:
Wähle eine Zahl zwischen 2 und 9 - am zweckmäßigsten die 2 (das heißt, daß bereits nach zwei Buchstaben getrennt werden kann; dann geht am meisten in die Zeile hinein).
- 6) Nach ENTER wird jetzt ein erster Ort zum Trennen angeboten (und zwar liegt er links vom Cursor). Falls er orthographisch falsch ist, kann der Cursor mit dem Linkspfeil solange nach links gerückt werden, bis eine trennbare Stelle erreicht ist. Dann taste man den "Bindestrich" '-' (der in diesem Fall zu einem Trennstrich wird). Falls es hier keine trennbare Stelle gibt, gehe man mit ENTER weiter.
Merke: Nach rechts Rücken ist nicht möglich. Das würde den Abzähl-Algorithmus des Systems durcheinander bringen.
- 7) Wenn es nichts mehr zu trennen gibt - d.h. wenn der Text durchgelaufen ist - meldet sich das System von selbst mit "HYPHENATION COMPLETE".

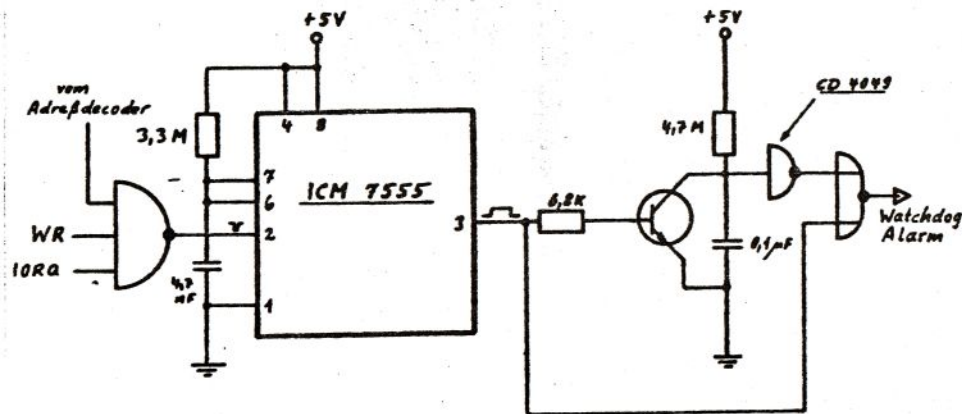
Xa7ot



Eine Schaltung, die man aufbaut in der Hoffnung, daß man sie nie braucht, ist der Watchdog. Für den Fall, daß Hardware oder Software außer Tritt kommen, muß verhindert werden, daß unkontrollierte Schaltzustände mit schlimmen Folgen für den Prozeß entstehen. Die Schaltung, die als "Notbremse" funktioniert, erwartet in regelmäßiger Folge (einige Millisekunden Abstand) Impulse vom Hauptprogramm. Bleiben die Impulse aus, wechselt ein Signalpegel, mit dem man z.B. einen NMI-Interrupt auslösen kann, der die CPU in eine definierte Fehleroutine zwingt. Zusätzlich kann man über einen Leistungstransistor die Stromversorgung der Ausgangsrelais abschalten und damit sämtliche Aktionen im Prozeß stoppen. Gleichzeitig erscheint ein optisches oder akustisches Alarmsignal, das dem Bedienungspersonal anzeigt, daß etwas schief gelaufen ist.

Zum Schluß ein Schaltungsvorschlag für einen Watchdog, der mit wenigen gängigen Bauteilen auskommt, bereits auf einen einzigen ca. 100 ns breiten Eingangsimpuls reagiert und dessen Zeitkonstante sich in einem sehr weiten Bereich durch die Wahl eines Kondensators einstellen läßt.

Heinrich Betz



CLUB 80 HARD - - - CLUB 80 HARD 64



Auslesen des Graphikspeichers

GDP64-Karte des NDR-Klein-Computers im TRS80 und GENIE

Helmut Bernhardt, Jörg Seelmann-Eggebert

Zwischen der in den Computern TRS80 und GENIE wohl am weitesten verbreiteten HRG 1B und der GDP64 des NDR-Klein-Computers bestehen folgende gravierenden Unterschiede:

Feature	HRG 1B	GDP64
Bildformat	192x384	256x512
Mischbarkeit des Video-signals mit dem des Text-bildschirms	ja	nein
Geschwindigkeit der Gra-phikausgabe	langsam	sehr schnell
Auslesbarkeit des Graphik-speichers	ja	nein
Anzahl Bildschirmseiten	1	4
Programmierbarkeit	umständlich	komfortabel
Anschluß an den Computer	beide frei	verdrahtet
BASIC-Treiber vorhanden	ja	ja
Verschleiß an Portadressen	128	32

Die wesentlichen Nachteile der GDP64 bestehen in der nicht vorhandenen Auslesbarkeit des Graphik-Speichers, wodurch ein Abspeichern von Bildern auf Diskette, ein Ausdrucken der Graphik, ein Spiegeln und Invertieren des Bildes und die Implementation eines Graphik-Cursors nicht möglich sind.

Die nicht vorhandene Mischbarkeit des Videosignals mit dem des normalen Textbildschirms kann dadurch umgangen werden, daß entweder ein zweiter Monitor benutzt wird oder eine einfache softwaregesteuerte (und/oder handgesteuerte) Umschaltung des Videosignals für den Monitor dazugestrickt wird. Ein möglicher Schaltungsvorschlag wird zum Schluß noch vorgestellt.

Zunächst aber soll das Hauptproblem, das Auslesen des Graphikspeichers, gelöst werden. Dazu sei vorweg bemerkt, daß wegen des kritischen Timings des EF9366-Graphikprozessors nicht garantiert werden kann, daß jeder Chip dabei funktioniert. Mit einem Pulldown-Widerstand kann der abfallenden Flanke des entsprechenden Signals zwar etwas Dampf gemacht werden, ob das bei dem jeweiligen Exemplar des 9366 dann ausreicht, muß probiert werden. Durch Verringern des Widerstandswertes läßt sich wahrscheinlich auch noch was raus-holen.

Für das Auslesen des Graphik-Speichers der GDP64 benötigt man 2 zusätzliche ICs und einen 2K2-Widerstand. Hauptbestandteil ist ein 74LS374-Latch, in das der EF9366 den Inhalt der ihm benannten 8 Bits des Bildwiderholungspeichers einträgt und aus dem sich die CPU dieses Datum abholen kann. Außerdem sind noch zwei OR-Gatter zur Ansteuerung des Latch nötig.

Und was muß gemacht werden ?

Nicht viel. Bei dem zusätzlichen 74LS374 werden außer den Pins 10 und 20 alle Pins um 45 Grad hochgebogen. Dann wird IC 20, 74LS245 der GDP64-Karte aus

der Fassung gezogen und das 74LS374 mit den Pins 10 und 20 auf den 74LS245 an dessen Pins 10 und 20 angelötet.

Dann wird IC 19, 74LS273 aus der Fassung genommen und auf dem Tisch so neben den eben gebauten Doppeldecker gelegt, wie die ICs 19 und 20 nebeneinander auf der Karte gesteckt haben. Mit möglichst kurzen isolierten Drahtstücken werden dann alle in der folgenden Tabelle aufgeführten Verbindungen zwischen Huckepack-74LS374 und 74LS273 hergestellt. Beim 74LS273 sollten die Drahtenden möglichst weit oben am Pin angelötet werden, damit die Pins nachher auch wieder in die Fassung passen. Danach kann das Verbundsystem wieder in die Fassungen gesteckt werden.

Nun werden die Verbindungen des Doppeldeckers-74LS374 mit den Pins des 74LS-166 entsprechend der rechten Hälfte der Tabelle hergestellt, wobei dann der 74LS166 aus der Fassung zu ziehen und locker auf diese zu stellen ist (damit sich die Drahtlängen richtig bemessen lassen und andererseits beim An-löten keine dauerhafte Verbindung zwischen IC und Fassung hergestellt wird; das entspräche dann auch nicht mehr dem Sinn einer Fassung).

Verbindungen zwischen dem Huckepack-LS374 und IC 19 sowie IC4

Signal	IC19, LS273	Huckepack-LS374	IC4,LS166
	Pin	Pin	Pin
D1	3 ----- 2	3 ----- 11	
D3	4 ----- 5	4 ----- 10	
D5	7 ----- 6	7 ----- 4	
D7	8 ----- 9	8 ----- 2	
D6	13 ----- 12	13 ----- 3	
D4	14 ----- 15	14 ----- 5	
D2	17 ----- 16	17 ----- 11	
D0	18 ----- 19	18 ----- 14	

Nun wird ein zusätzliches 74LS32 genommen und dessen Pins 1-4, 6 und 8-13 ebenfalls um 45 Grad hochgebogen. Mit den Pins 5, 7 und 14 wird es auf dem aus der Fassung genommenen IC8, 74LS32 an dessen Pins 5, 7 und 14 angelötet. Nun kann auch dieser Doppeldecker wieder in seine Fassung gesteckt werden. Eventuell müssen die ab gespreizten Pins der beiden Huckepack-ICs noch etwas zurechtgebogen werden, damit sie sich nicht gegenseitig berühren und auch keinen Kontakt mit ihren Träger-ICs haben. Dann werden folgende Verbindungen hergestellt:

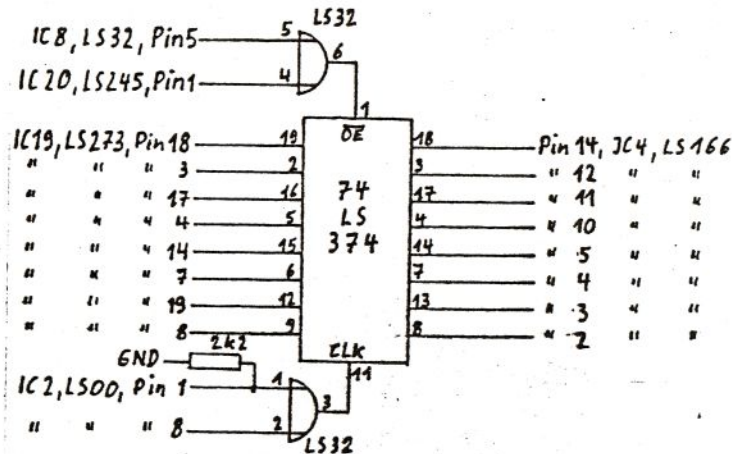
Huckepack-LS32	Verbinden mit
Pin	
1	IC2, 74LS00, Pin1
2	IC2, 74LS00, Pin8
3	Huckepack-LS374, Pin11
4	IC20, 74LS245, Pin1
6	Huckepack-LS374, Pin1

Und schließlich werden noch die Pins 1 und 7 des IC2, 74LS00 über einen 2K2-Widerstand verbunden (Pulldown des MFREE-Signals des EF9366).

Damit ist alles getan, um die dem EF9366 vorgegebene Speicherstelle auslesen zu können.

Wenn danach das Auslesen des HRG-Speichers nicht funktioniert, ist das nicht verwunderlich, weil man zunächst etwas falsch gemacht hat; zumindest ich (Helmut) bin das so gewöhnt. Dieser Fehler kann aber durch Vergleich des Strippengewirrs mit der Einbauanleitung schnell gefunden werden.

Möglicherweise treten dann beim Auslesen immer noch Fehler auf. Die Ursache liegt dann beim oben beschriebenen Timing des EF9366 mit seinem MFREE-Signal. Man kann dann noch mit einem kleineren Widerstand als 2k2 Versuche machen oder eventuell einen anderen EF9366 ausprobieren. Letztere Variante kann natürlich recht kostspielig werden, bis man ein funktionierendes Exemplar gefunden hat. Alternativ kann man sich damit abfinden, daß es nicht funktioniert und alle Umbauten wieder abreißen (weniger erstrebenswert).



Und was bringt die GDP64 gegenüber der HRG 1B mehr?

An Software-Unterstützung gibt es dank der Bemühungen von Christof Ueberschaar ein 4K-Byte großes BASIC-Treiberprogramm, das ins HIMEM geladen wird und nach Einbinden in das Disk-BASIC sehr komfortable Graphik-Befehle bereitstellt. Gegenüber der Variante des Treibers für die HRG 1B, wo die Graphik-Befehle durch ein # eingeleitet werden und daher der BASIC-Interpreter bei jedem Befehl erst prüfen muß, ob ein # vorliegt, ist hier der CMD-Vektor verbogen worden. Es wird dann nur noch bei Auftauchen des CMD-Tokens im Programmtext eine zusätzliche Abfrage gemacht, ob das nach CMD obligatorische (") folgt und nur dann, wenn das nicht der Fall ist, weiter untersucht, ob das Byte ein gültiger Graphik-Befehl ist, der dann abgearbeitet wird.

Diese Philosophie hat zwar den Nachteil, daß man anstelle des einfachen # jedesmal 3 Zeichen eintippen muß, bevor der eigentliche Graphik-Befehl kommt, andererseits wird dadurch nicht die Abarbeitung der normalen BASIC-Befehle verlangsamt, womit ein Teil des Geschwindigkeitsvorteils der GDP64 verschenkt wäre.

Außerdem ist die Punktdressierung der GDP64 bedeutend einfacher als die der HRG 1B. Es werden dem EF9366 einfach die X- und Y-Koordinaten des Punktes als 16-Bit-Wörter in entsprechende Register geschrieben und dann einer der Befehle "Punkt setzen", "Punkt löschen" oder "Punkt abfragen" ausgeführt. Bei der HRG 1B bedeutet das ein umständliches Berechnen der Speicheradresse und des entsprechenden Bits im Byte.

Noch gravierender ist die Eleganz beim Zeichnen von Linien über Vektorbefehle. Hier werden die Punktkoordinaten (X1/Y1) für den Anfangspunkt einer Linie in die entsprechenden Register eingetragen und in zwei weitere Register werden die Werte Delta X und Delta Y eingetragen. Es sind dies die Differenzen X2-X1 und Y2-Y1 (Koordinaten von Anfangs- und Endpunkt einer Linie). Diese Differenzen müssen stets positive Zahlen sein. Das Vorzeichen (Vektorrichtung) wird durch entsprechende Bits in den Vektorbefehlen vorgegeben. Es läßt sich auch noch festlegen, ob die Linie durchgezogen, gepunktet oder punktrichliert ausgegeben werden soll, oder ob die Linie überhaupt gezeichnet werden soll oder eine vorher hier gezeichnete Linie wieder gelöscht werden soll.

Wenn die Linie dann mit einer Geschwindigkeit von 1,5 Mio Punkten pro Sekunde gezeichnet ist, stehen automatisch in den X- und Y-Koordinatenregistern die Koordinaten X2 und Y2 der eben gezeichneten Linie, so daß von deren Endpunkt durch Vorgabe neuer Werte für Delta X und Delta Y gleich ein weiterer Vektor gezeichnet werden kann, ohne daß die Endpunktkoordinaten der vorherigen Linie noch als Anfangskoordinaten der neuen Linie ausgegeben werden müssen.

Dann hat der EF9366 noch einen eingebauten Zeichengeneratort (der aber leider nur die amerikanischen Zeichen kennt), mit dem ein komfortables Einbinden von Texten in die Graphik möglich ist. Außer Aufrecht- und Schrägschrift sowie in Breite und Höhe voneinander unabhängig in 16 Stufen in der Größe variierbaren Zeichen läßt sich vorgeben, ob normal von links nach rechts, auf dem Kopf stehend von rechts nach links oder um 90 Grad gedreht von unten nach oben oder von oben nach unten geschrieben werden soll. Für die Textausgabe ist vor der Ausgabe der Textzeichen neben dem Festlegen dieser Modi auch noch der Fußpunkt des ersten Zeichens in die Koordinatenregister einzutragen. Nach der Ausgabe eines Zeichens erhöht sich der Pointer auf den Fußpunkt des dahinter auszugebenden Zeichens, wobei die oben beschriebenen Modi natürlich berücksichtigt werden.

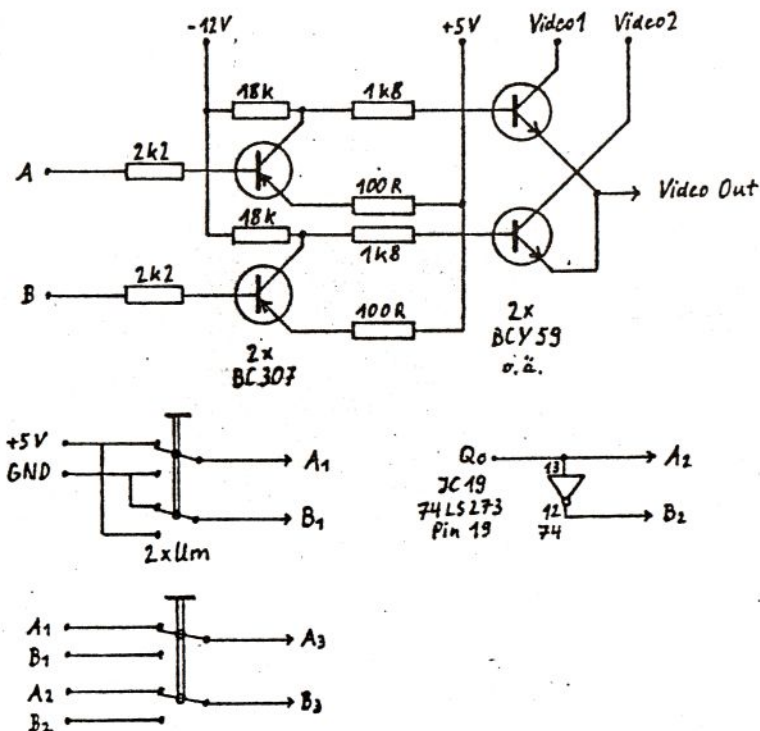
Diese Bemerkungen sollen keinesfalls als erschöpfende Beschreibung der Leistung der GDP64 zu werten sein. Für die Programmierung des EF9366 kommt man ohne Datenblatt nicht aus. Es sollte hier nur ein ungefährer Eindruck vermittelt werden, wofür der Aufwand überhaupt getrieben wurde, bei der unsicheren Erfolgsaussicht, den Hardware-Patch überhaupt anzufangen.

Das Umschalten der Video-Signale

Da die Video-Signale des Text/Bauklötzchen-Graphik-Interface des Computers und der GDP64-Karte nicht direkt miteinander mischbar sind, muß mit einer Umschaltung zwischen den Signalen gearbeitet werden. Für Experimentierzwecke reicht dazu ein einfacher Umschalter, mit dem abwechselnd eines der beiden Video-Signale durchgeschaltet werden kann. Die Bezugsmassen der beiden Signale sind miteinander und mit der Abschirmung des Kabels zum Monitor zu verbinden.

Als Dauerlösung ist aber eine Softumschaltung wesentlich eleganter. Dafür kann das Bit 0 des unter Port 60H decodierten Latch (IC19,LS273), mit dem auch die HRG-Speicherbanks angewählt werden, mißbraucht werden. Es ist dann

noch ein Inverter nötig, den IC9, 7404, Pins 13 und 12 liefert. Dabei ist zu beachten, daß mit einem OUT-Befehl an Port 60H dann sowohl die für Auslesen und Schreiben einzustellenden Banks als auch die Selektion des Video-Signals gesteuert werden. Da dieses Latch sich nicht auslesen läßt, muß über dessen Inhalt im RAM Buch geführt werden, damit beim Ändern der Bankwahl nicht das Video-Signal umgeschaltet wird und umgekehrt. Außerdem sollte der Pin 1 des IC 19, 74LS273 nicht mit +5V sondern mit RESET* (oder einem aus RESET* und NMI* erzeugten SYSRES*) verbunden werden, damit durch Löschen des Latch beim Einschalten (bzw. Drücken des RESET-Knopfes, wodurch ein NMI* erzeugt wird) ein definierter Zustand vorliegt.



In Abb.2 ist oben eine prinzipielle Schaltung zur Umschaltung der Video-Signale (Video1 und Video2) gezeigt, deren Steuereingänge A und B entgegengesetzte logische Pegel haben müssen. Wenn anstelle von -12V nur -5V zur Verfügung stehen, kann auch -5V verwendet werden, wenn anstelle der beiden 18K-Widerstände solche von 7K5 genommen werden.

Unten in Abb.2 sind zwei Möglichkeiten für die Beschaltung der Steuereingänge A und B gezeigt, womit sich entweder durch Hand- oder durch Softwareumschaltung das gewünschte Video-Signal selektieren läßt. Und schließlich ist noch die Umschaltung der Umschaltungsvarianten per Hand gezeigt, wofür dann noch ein zweiter 2fach-Umschalter nötig ist.

GRAPHIC-TREIBER FÜR EF9366

© 1984 UE-SOFT

Start: im NEWDOS/80 mit 'DO GRAPHIC'

Befehle:

Linie: CMDLINE(x1,y1)-(x2,y2),S/R[I]D]
x1,y1=von-Koordinaten, x2,y2=bis-Koordinaten
S=set, R=reset,
P=punktierte Linie, D=gestrichelte Linie

Box: CMDLINE(x1,y1)-(x2,y2),S/R[B/BF]
x1,y1=Eckpunkt, x2,y2=gegenüberliegender
Eckpunkt
B=Box, BF=gefüllte Box

Kreis: CMDSCR(x,y),r,S/R[F]
x,y=Mittelpunkt, r=Radius
F=gefüllter Kreis

Set: CMDSET(x,y)
x,y=Koordinaten des zu setzenden Punktes

Reset: CMDRESET(x,y)

Position: CMDPOS(x,y)

Print: CMDPRINTI(x,y)]"Text";[(x,y)]Variable
x,y=Print at-Koordinaten

Löschen: CMDCLEARI(x,y)]"Text";[(x,y)]Variable

Schriftart: CMDSIZE(b,h),r
b=Breite (1-16), h=Höhe (1-16),
r=Richtung (A-D)

CLS: CMDCLS

Invers: CMDNOT

Teil invers: CMDFIELD(x1,y1)-(x2,y2)

Hardcopy: CMDLPRINT

Sichern: CMDSAVE"Filename"

Laden: CMDLOAD"Filename"

GDP-Regs: CMDRSET

Spiegeln: CMDFLEX

Linie bis: CMDDRAW(x,y),S/R[I]D]

HEFT
20
August
1987

70

Tastatur- und Druckerplatine (ECB)

Bernd Drowälder

Aufbau des Tastaturteiles:

Die Tastatur des TRS80 ist in Form einer 8x8 Matrix aufgebaut. Dabei werden die Adressleitungen A0-A7 über Inverter mit Open-Kollektorausgängen auf die Matrixzeilen geführt und die Matrixspalten werden über 8 invertierende Treiber auf den Datenbus des TRS80 gelegt. Die Tastatur wird vom Rechner als Speicherbereich gesehen. Sie belegt den Speicher von 3800H bis 38FFFH. Mit Hilfe des Keyboard-Signales werden die Tastaturspalten über die Treiber auf den Datenbus geschaltet. Das KYBD-Signal decodiert den Bereich 38XXH. Dieses geschieht mit Hilfe eines 74LS138, einem 3-Bit Binärdecoder.

Erzeugung des KYBD-Signales:

Der Ausgang Y7 des Binärdecoders nimmt nur L-Pegel an, wenn A11, A12 und A13 H-Pegel und die Adressleitungen A8-A10, A14 und A15 auf L-Pegel liegen. Da aus der Tastaturmatrix nur ausgelesen werden kann, wird außerdem das Signal RD* zur Decodierung benutzt.

Die Adressleitungen A0-A7 wurden bisher nicht berücksichtigt. Diese Adressen werden benötigt, um die Tastaturzeilen auszuwählen, die ausgelesen werden sollen. Über eine der Adressen A0-A7 wird eine der Matrixzeilen ausgewählt. Über den Inverter mit Open Kollektor-Ausgang wird auf die Matrixreihe Low-Potential gelegt. Wird jetzt eine Taste gedrückt, zieht das Low-Potential am O.K.-Inverterausgang die Spalte, die über einen Pull-Up-Widerstand auf 5V liegt, auf L-Potential herab. Über den invertierenden Treiber gelangt ein H-Pegel auf die der Spalte entsprechenden Datenleitung. Über eine Tabelle erkennt der Rechner dann, welche Taste vom Benutzer gedrückt wurde.

Aufbau des Druckerteiles:

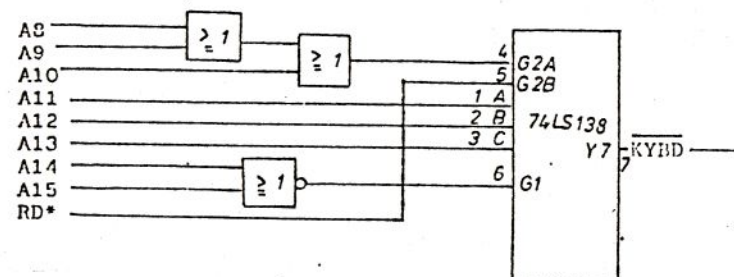
Der TRS80 hat die Möglichkeit, eine Centronics 8-Bit Parallel-Schnittstelle anzuschließen. Die Steuerung dieser Schnittstelle erfolgt Memory-Mapped über die Adresse 37E8H. Die Adresse 37E8H wird mit einem 74LS133, drei NOR-Gattern und einem Inverter ausgewählt.

Der 74LS133 ist ein NAND-Gatter mit 13-Eingängen. Der Ausgang eines NAND wird nur LOW, wenn alle Eingänge auf H-Pegel liegen. Die Adressen, die schon auf HIGH liegen werden direkt mit dem NAND-Gatter verbunden, die anderen werden über die NOR-Gatter und dem Inverter auf H-Pegel gebracht. Der Ausgang wird also nur LOW, wenn die Adresse 37E8H anliegt. Um den Drucker steuern zu können wird das Ausgangssignal mit den Leitungen RD und WR verknüpft, um sowohl über die Adresse 37E8H Schreiben als auch Lesen zu können. Das Signal 37E8H WR gibt einen Puls an das 8-Bit D-Register, welches daraufhin die anliegenden Daten übernimmt und an den Ausgang des D-Registers weiterleitet. Außerdem wird mit diesem Signal über ein Monoflop ein 1 us langer STROBE Puls erzeugt, der dem Drucker mitteilt, daß gültige Daten zum Drucken anliegen.

Mit dem 37E8H RD Signal werden 4 Treiber auf den Datenbus des Rechners durchgeschaltet. Der Rechner kann über diese Leitungen den momentanen Status des Druckers abfragen.

Um die Druckerkarte auch alleine mit dem normalen TRS80 nutzen zu können, werden 2 Dioden eingefügt, die über die GDT-Leitung die Treiber zum Rechner durchschalten.

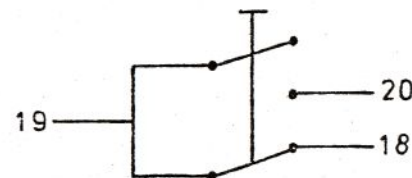
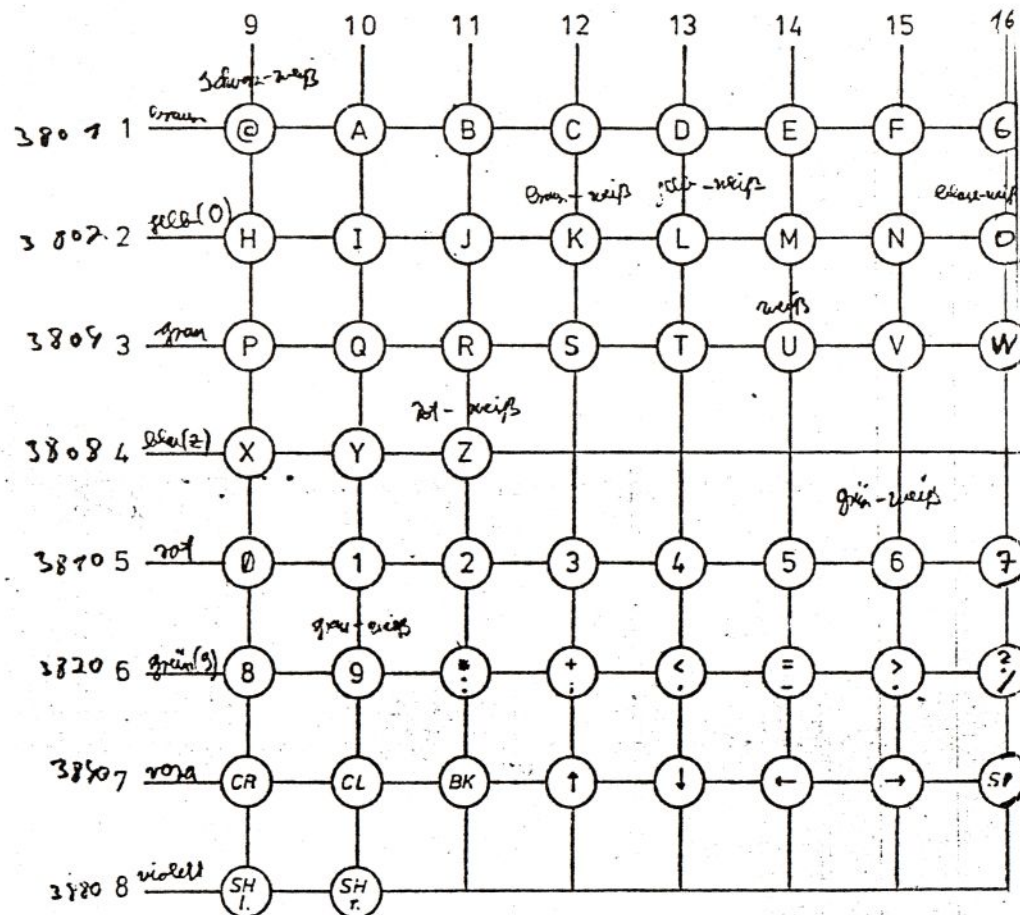
Erzeugung des KYBD-Signales:



4.1 Anschlußplan des Druckersteckers

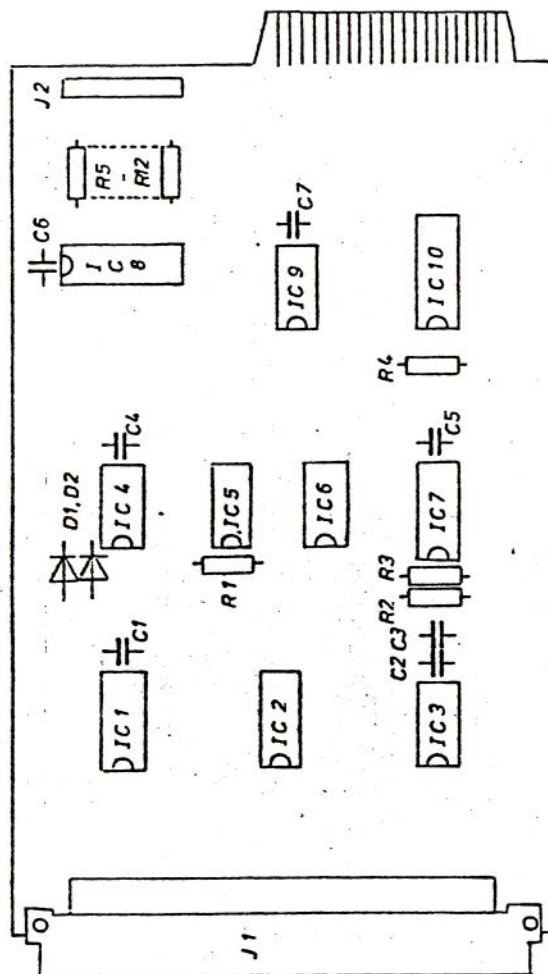
Leitung	Signalname	Beschreibung des Signals
1	DATA STROBE	Gültige Daten liegen an
2	GND	
3	D1	
4	GND	
5	D2	
6	GND	
7	D3	DATENLEITUNGEN
8	GND	
9	D4	D1 - D8, die zum Drucker geführt werden
10	GND	
11	D5	
12	GND	
13	D6	
14	GND	
15	D7	
16	GND	
17	D8	
18	GND	
19	NC	Nicht verbunden
20	GND	
21	BUSY	Es werden noch Daten verarbeitet
22	GND	
23	OUT OF PAPER	Kein Papier mehr
24	GND	
25	UNIT SELECT SLCT	Ein Drucker ist ausgewählt
26	PRIME*	Löscht den Drucker-Buffer
27	GND	
28	FAULT*	Fehlermeldung
29	NC	Nicht verbunden
30	NC	Nicht verbunden
31	GND	
32	NC	Nicht verbunden
33	GND	
34	GND	

4.1.1 Tastaturmatrix





4.3 Bestückungsplan Tastatur- und Druckerkarte



4.4 Bestückungsliste Tastatur- und Druckerkarte

Halbleiter:

IC1	74LS138
IC2	74LS133
IC3	74LS02
IC4	74LS32
IC5	74LS05
IC6	74LS05
IC7	74LS123
IC8	74LS240
IC9	74LS125
IC10	74LS273

Widerstände:

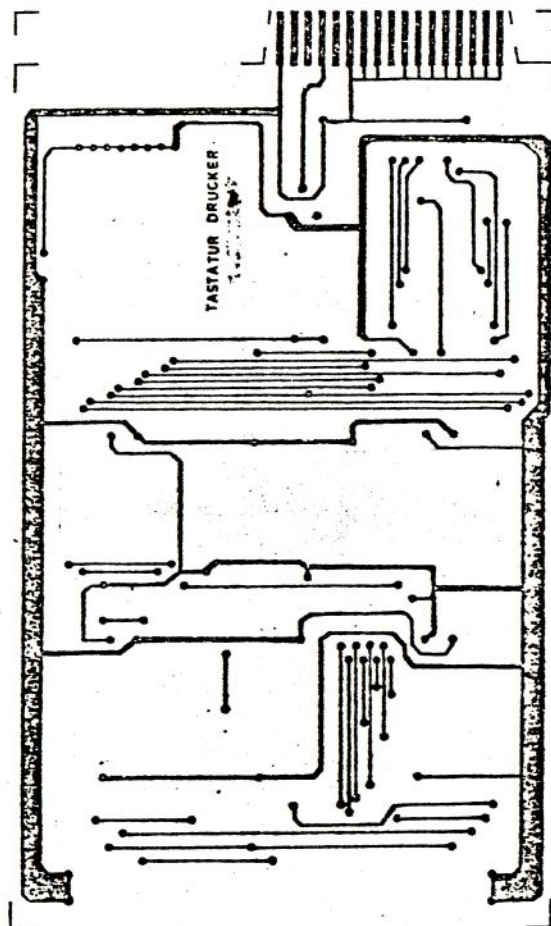
R1, R3, R4, R5-12	4,7 kOhm
R2	20 kOhm

Kondensatoren:

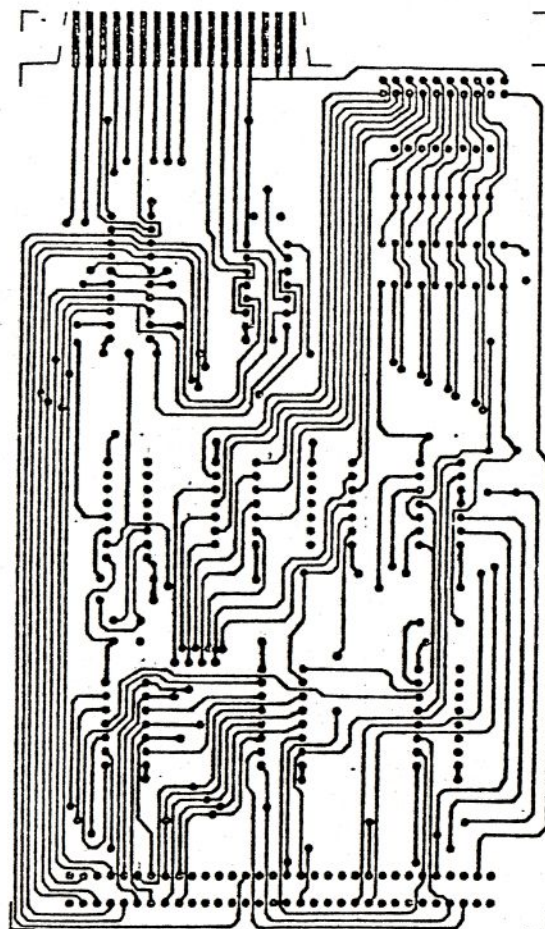
C1, C2	100 nF
C3	200 pF
C3-C7	100 nF

Sonstige Bauelemente:

D1, D2	1N4148
J1	64 pol. Steckerleiste
J2	20 pol. Steckleiste, abgewinkelt



Platinenlayout
Lötseite



Unser Clubfreund Richard Rensch hat vor einiger Zeit, ich glaube es war im Info Nr. 18, um Hilfe beim Anschluß eines Laufwerks an sein Genie III gebeten. Nach einiger Zeit und einem regen Brief- und Schaltplanwechsel konnten wir das Laufwerk, das eigentlich für den Anschluß an einen PC-Kompatiblen gedacht war, doch noch dazu überreden am Genie III zu funktionieren. Seit dieser Zeit wollte ich möglichst viele der Informationen, die ich in den verschiedensten Quellen entnommen habe, zu einem Beitrag über die Shugart-Schnittstelle zusammenfassen. Dies hat sich nun erledigt! Der folgende Artikel aus der c't läßt wohl keine Fragen mehr offen und dürfte für die Allgemeinheit interessant sein. Viel Spaß bei der Lektüre, euer

Karsten Obermann

Anschluß gesucht

Floppy-Laufwerke zum Laufen gebracht – vom sogenannten Shugart-Bus und ähnlichen 'Standards'

Willi Wagemuth, Dettlef Grell

'Shugart-Bus', neuerdings auch 'ANSI-Bus', und 'Standard' sind die Schlagworte, die fast immer als erste fallen, wenn die Rede auf den Anschluß von Floppy-Laufwerken an Rechner kommt. Folglich sollte es doch ein leichtes sein, ein Floppy-Laufwerk erfolgreich an einem Floppy-Controller zu betreiben, wenn beide mit eben diesem standardisierten Bus-Anschluß versehen sind. Im Prinzip ja, aber mit dem armseligen Rüstzeug Standard ist es bei weitem nicht getan.

lediglich anhand des Begriffes 'Floppy-Bus' beschreiben werden.

Nicht ohne Papiere!

Wichtigste Vorbedingung, die es in fast allen Fällen ermöglicht, auch mit unterschiedlichsten Laufwerken und Controllern zum Zuge zu kommen, ist eine ausreichende Dokumentation von beiden. Dabei reicht beim Controller im allgemeinen die Beschreibung der wirklich angeschlossenen und bedienten Bus-Signale aus.

Beim Laufwerk braucht man zunächst dasselbe, es sind aber vor allem die Steckbrücken, die einen entscheidenden Einfluß auf die Funktion des Laufwerks und zum Teil auch auf die Belegung der Bus-Schnittstelle haben. Wenn das Laufwerk nicht auf Anhieb 'spielt', und man vor dem Problem steht, zehn oder manchmal bis zu dreißig Jumper auf gut Glück zu variieren, sind Frust oder gar Beschädigungen der Hardware vorprogrammiert.

Zwar werden wir Ihnen im folgenden eine Fülle von Tipps ge-

ben, die einen teilweisen Dokumentationsmangel kompensieren können, aber geben Sie nicht uns die Schuld, wenn sich Ihre 50-Mark-Glückskauf-Floppy als böser Flop entpuppt.

Alles Abweichler

Die Inkompatibilitäten zwischen Floppy-Drives (und in ähnlicher Weise zwischen Controllern) sind aus dem langsamen Wachsen der Anforderungen an die Laufwerke und den steigenden Möglichkeiten derselben zu erklären. Der Urvater des Floppy-Bus ist der 50polige Bus für 8-Zoll-Laufwerke, wobei man gleich anfügen muß, daß nur die Hälfte als Signalleitungen benutzt wird, da die anderen Leitungen Massepotential führen.

Bereits damals wurde die Firma Shugart mit ihrem recht sinnvollen Bus-System bekannt, das den meisten Erfordernissen genügte. Allerdings hat dieser Bus nicht sofort in der Branche Nachahmer gefunden. Control Data zum Beispiel hatte recht abweichende Vorstellungen davon, wie 8-Zoll-Laufwerke be-

dient werden müssen – Besitzer gebrauchter 8-Zoll-Laufwerke wissen das zu beklagen.

Aber auch wenn man tatsächlich ein Laufwerk mit angeblichem Shugart-Bus findet, so passen zwar oft die Stecker, seltener jedoch alle Signale zusammen. Und wenn man Pech hat, paßt zwar all das Genannte, aber die zeitliche beziehungsweise logische Ansteuerung durch den Controller erweist sich als abweichend.

Die Vielzahl von Steuerleitungen (die zum Teil bei den neueren 5.25-Zoll-Drives entfallen sind) und eine Fülle von Optionen, die per Steckbrücken einstellbar sind, erfordern bei 8-Zoll-Laufwerken genaue und ausführliche Unterlagen, will man sie in 'endlicher Zeit' in Betrieb nehmen. Um diesen Beitrag nicht unnötig aufzublähen, werden wir aber nur noch mit einigen Randbemerkungen auf die inzwischen doch weitgehend veralteten 8-Zöller eingehen.

Etwas günstiger ist die Situation auf dem Markt für 5.25-Zoll-Laufwerke. Hier konnten sich die Hersteller tatsächlich auf ein etwas wie einen Standard einigen. Durch Wegfall einiger Optionen, die sich schon bei den 8-Zoll-Drives als entbehrlich erwiesen hatten, beziehungsweise durch Erhöhung der Laufwerks-Intelligenz konnte man den eh nie ganz ausgenutzten 50poligen Bus auf 34 Leitungen abspecken.

Die ersten 'kleinen' Laufwerke boten noch nicht so viele Betriebsvarianten wie heutige Drives (auf der Laufwerksplatte war gerade Platz für die grundlegende Schreib-/Lesen- und Steuer-Logik). So wurden anfangs nur die Signale an den Pins 8 bis 30 (siehe Kasten) und bald auch 32 belegt, allerdings von so vielen Laufwerksherstellern einvernehmlich, daß Abweichungen von der hier beschriebenen Form quasi nicht mehr zu finden sind. Kann man sich bei den heutigen Laufwerken auf diese Grundfunktionen beschränken, so hat man auch wenig Probleme: Fast alle Laufwerke und Controller arbeiten auf dieser Ebene zusammen.

Dennoch soll nicht vergessen werden, daß die Firma Philips anfangs einen recht stark abweichenden Bus für 5.25-Zoll-Systeme propagiert hat. Die meisten Signale auf diesem Bus ent-

sprechen zwar den bisher genannten, liegen aber auf anderen Pins (der Stecker ist nämlich 50polig, wenn auch nicht etwa zum 8-Zoll-Shugart-Bus kompatibel). Die wichtigste Abweichung ist, daß für jedes Laufwerk eine eigene 'Ready'-Leitung zur Verfügung steht (Tabelle 3).

Mehrfachbelegung

Der 'Ärger' begann, als statt ursprünglich maximal drei nun auch vier Laufwerke selektiert werden sollten und die Laufwerke ihren Status (Ready-Signal) übermitteln konnten. (Bis dahin hat man die Steuersoftware im Rechner gemäß den Angaben der Laufwerkshersteller halt so lange warten lassen, bis man sicher sein konnte, daß das Laufwerk bereit war.)

Hier kochten nun verschiedene Hersteller wieder ihr eigenes Süppchen. So findet man die Ready-Leitung mal auf Pin 34, mal auf Pin 6 oder aber auch gar nicht. Vielleicht liegt auf Pin 6 aber auch das Select-Signal für das vierte Laufwerk (siehe Kasten). Ebenso ungewiß ist, ob das Signal 'Head Load' auf Pin 2 oder Pin 4 oder überhaupt zu finden ist (manche einfachen Laufwerke lassen den Kopf nach Einlegen der Diskette dauernd auf der magnetischen Oberfläche 'schleifen').

Dann gibt es noch das Signal 'In Use' (in Gebrauch), mit dem das Laufwerk in Zugriffspausen, wenn der Rechner anderweitig beschäftigt ist, bei Laune gehalten werden kann, damit der nächste Zugriff schneller geht (Kopf bleibt geladen). Neuere Laufwerke kennen oft noch eine Formatumschaltung von normalen Disketten auf solche mit hoher Schreibdichte, und manche haben Signale, mit denen ein Diskettenwechsel signalisiert wird.

Für all diese Funktionen stehen aber nur vier Leitungen (2, 4, 6, 34) zur Verfügung. Diese vier genannten Leitungen werden also je nach Leistungsvermögen des Laufwerks und Laune des Herstellers belegt und führen typischerweise zu Anschlußproblemen – aber wie soll ein Hersteller auch die nicht standardisierten Zusatzfunktionen anders unterbringen?

Ist es schon schwierig genug herauszufinden, wie man nur ein solches Laufwerk an einem Floppy-Controller zum Laufen

bringt, so beginnt beim Anschluß eines weiteren Drives mit anderen Bus-Varianten das Spiel aufs neue. In beiden Fällen sind bei arglosem Anschluß ('Ist doch alles Shugart-Bus, oder?') durchaus Beschädigungen der Elektronik möglich, wenn nämlich Controller-Ausgänge (Select, Head Load) mit Floppy-Ausgängen (Ready, Disk Change) zusammenprallen.

Problematisch kann es auch werden, wenn ein Laufwerk kein Ready-Signal liefert, dieses auf dem Controller aber benötigt wird. Da hilft nur 'Tricksen', indem man etwa das Select-Signal des betreffenden Laufwerks dem Controller selbst (über ein Open-collector-Gatter, nichtinvertierend, oder eine Germaniumdiode) wieder als Ready anbietet.

Oder, wenn das Laufwerk sehr lange zum Start braucht, bastelt man eine etwas aufwendigere Logik, die die Indexpuls auswertet und Ready meldet, wenn diese in einer Geschwindigkeit folgen, die man bei Nenndrehzahl erwarten muß. Wichtig ist, daß jedes Laufwerk ein unabhängiges Ready meldet, es also mit dem Select verknüpft wird.

Qual der Wahl

Damit aber noch nicht genug. Je moderner die Laufwerke, desto flexibler sind sie meistens, was die Belegung dieser 'neutralen Pins' betrifft: Diverse Steckbrücken (Jumper) auf den Laufwerken ermöglichen – in Grenzen – die Anpassung von Laufwerken unterschiedlicher Hersteller bezüglich eben dieser Signale.

Da bei der Bezeichnung dieser Steckbrücken allerdings absolut keine Eintracht herrscht, nicht mal unbedingt bei verschiedenen Laufwerken des gleichen Herstellers, kommt man hier nicht mehr ohne Laufwerks-handbuch weiter. Zwar gibt es inzwischen bei einigen Herstellern gleiche Jumper-Bezeichnungen (HL, HM oder ähnliches), aber man kann nie sicher sein, in welchen Kombinationen mit anderen Jumpern sie was bewirken. Vor allem ist nie ersichtlich, welche Kombinationen verboten sind, weil sie unter Umständen zu Beschädigungen führen.

Eine dieser Optionen ist besonders hervorzuheben: Die Funktion 'Head Load' sollte man

stets mit 'Motor on' verknüpfen, so daß der Kopf beim Einschalten des Motors immer geladen wird. IBMs PCs zum Beispiel verlangen diese Einstellung definitiv.

Kühlen Kopf bewahren

Moderne 3.5-Zoll-Laufwerke zum Beispiel bieten sehr kurze Kopf-Ladezeiten beziehungsweise Beruhigungszeiten. Neuere Rechner, etwa Atari oder Amiga, sind dann im allgemeinen vollständig in ihrer Controller-Software auf diese Zeiten abgestimmt. Probleme gibt es, wenn man an solche Rechner zu Konvertierungszwecken ältere 5.25-Zoll-Drives mit längeren Kopf-Ladezeiten anschließt. Da das Ready-Signal keine Aussage über den Lade-Zustand des Kopfes macht, kann also ein noch nicht eingeschwengender Kopf zum Schreiben oder Lesen veranlaßt werden – was typischerweise zu Fehlern führt.

Außerdem trennt man damit bei den meisten Laufwerken die Leitung 'Head Load' vom Bus ab. Das kann notwendig sein, wenn möglicherweise ein anderes Laufwerk im System ist, das über diese Leitung zum Beispiel zwischen normaler und doppelter Schreibdichte umschaltet wird. Zum andern erspart es vielfach das 'Kopfgeklapper', das einige Floppy-Controller beim Positionieren veranstalten. Auch zum Beispiel Assembler mit intensiven Diskettenzugriffen (meist unter CP/M) verdröseln oft so viel Zeit zwischen einzelnen Disk-Zugriffen, daß der Controller schon wieder den Kopf abfallen läßt.

Manchmal kann es sogar hilfreich sein, das floppy-interne Head-Load-Signal per Lötkolben fest an Motor on zu ketten: Zum Beispiel ließ sich ein 8-Zoll-Drive (Siemens) trotz zig Optionen nicht an einem Controller mit µPD 765 zum Steppen überreden. Der 765 (wie auch einige Controller von Western Digital) läßt – Disketten schonend – nämlich auch ohne 'Head Load' den Kopf positionieren. Das bewußte Laufwerk stiept aber partout nicht ohne aktives Head-Load-Signal.

Und da wir schon mal dabei sind: Dieses Laufwerk gibt sich auch nicht mit den kurzen Select-Impulsen zufrieden, die der

765 an die Drives eins bis drei ausgibt. Die Sache funktioniert ohne Zusatzhardware nur, wenn das Laufwerk als viertes im Bunde angesprochen wird, weil der Controller-Chip hier nämlich länger selektiert.

Die Lampe des Laufwerks koppelt man am besten mit dem Signal 'Drive Select', denn dann sieht man sicher, welches Laufwerk gerade angesprochen wird. Auf andere Einstellmöglichkeiten kommen wir bei speziellen Problemen noch zurück.

Breite Spuren

Manche 80-Spur-Laufwerke lassen eine Umstellung für 40-Spur-Betrieb zu. In c't wurde auch eine kleine Schaltung veröffentlicht, mit der man jedes 80-Spur-Laufwerk zu Doppel-Stepping überreden kann (c't 5/86, S. 46. MSDOS-Disketten auf dem Atari ST). In beiden Fällen wird aber nur die Positionierung des Kopfes an die 40-Spur-Diskette angepaßt, die Spurbreite bleibt erhalten.

Das kann unter folgenden Umständen zu Problemen führen: wenn eine Diskette, die in einem echten 40-Spur-Laufwerk formatiert und beschrieben wurde, auf einem 'umgebauten' 80-Spur-Drive beschrieben wurde und diese Diskette später wieder auf einem echten 40-Spur-Laufwerk weiterverwendet werden soll.

Dabei werden nämlich in die 'breiten' 40er Spuren der Diskette schmalere von 80-Spur-Laufwerk eingebettet, und es können sehr wohl Reste der ursprünglichen 'Breitspur' übrigbleiben. Wird diese Diskette jetzt wieder dem 40-Spur-Laufwerk angeboten, so kann dieses möglicherweise das 'Gemisch' nicht mehr lesen. Resultat: Die Diskette wird als physikalisch defekt gemeldet, ist also auch mit Tricks nicht mehr auf dem 40-Spur-Drive lesbar (wohl aber noch auf dem 80er).

Normalerweise gibt es aber keine Probleme, wenn eine 'jungfräuliche', also unformatierte Diskette auf dem 80-Spur-Laufwerk formatiert wird, so daß keine 'Breitspurstre' vorhanden sind.

Es gibt allerdings noch etwas zu beachten, wenn man 80-Spur-Drives doppelt so große Schritte nehmen läßt, als sie üblicherweise sollen: Sie brauchen auch die doppelte Zeit dafür! Wenn

HEFT
20
AUGUST
1987

82

81

Wir haben nichts gegen Standards, ganz im Gegenteil! Nur wenn sich keiner dran hält, kann man gut drauf verzichten – weshalb wir im folgenden auch genau das tun und die Realität

Ihr Laufwerk also die ersten Spuren im allgemeinen recht zuverlässig lesen kann, danach aber sporadisch Fehler aufzuweisen, so könnte die Steptrate, also die verfügbare Zeit, um die nächste Spur aufzusuchen, zu knapp sein. Die Steptrate läßt sich üblicherweise per Software (Betriebssystem) über den Controller-Chip einstellen (Rechnerhandbuch!).

Higher and Higher

Etwa 1983 kamen die ersten 5,25-Zoll-Laufwerke mit 80 Spuren und – für damalige Zeiten – sehr hoher Schreibdichte (High Density) auf den Markt. Ihre Kapazität ließ sich bis etwa 1,4 MByte hochtreiben, abhängig von der Formatierung. Zur Erzielung der hohen Schreibdichte wurden ein paar Anleihen bei den 8-Zoll-Laufwerken aufgenommen. So sind diese Drives zum Beispiel aus Controller-Sicht, also steuerungstechnisch, kompatibel, obwohl sie den 34poligen Anschlußstecker der normalen 5,25-Zoll-Drives aufweisen.

Die wichtigsten Unterschiede zu normalen 5,25-Zoll-Drives liegen in der Datenübertragungsrates und der Motordrehzahl. Die herkömmlichen 5,25-Zoll-Laufwerke arbeiten bei einfacher Schreibdichte (single Density, FM-Aufzeichnung) mit 125 KBit/s, ein Format, das nur bei ganz wenigen, restlos veralteten Rechnern zu finden ist, denn auf einer einseitigen 40-Spur-Scheibe lassen sich keine 100 KByte unterbringen. Die heute fast immer anzutreffende doppelte Schreibdichte (double Density, MFM-Verfahren) bietet bei doppelter Transferrate (250 KBit/s) auch die doppelte Speicherkapazität. Die Diskette macht dabei 300 Umdrehungen pro Minute.

8-Zoll-Laufwerke hingegen arbeiten seit jeher schon bei einfacher Schreibdichte mit 250 KBit/s Übertragungsrates und mit 500 KBit/s bei doppelter Dichte. Längere Spuren (größerer Durchmesser), eine daraus und aus der Motordrehzahl von 360 U/min resultierende etwa verdoppelte Geschwindigkeit zwischen Schreib-/Lesekopf und Disk stellen dabei etwa die gleichen Anforderungen an das Diskettenmaterial wie bei den einfachen 5,25-Zoll-Drives.

Die 5,25-Zoll-Drives mit High-Density-Modus arbeiten ebenfalls mit 360 U/min und einer Transferrate von 500 KBit/s (bei double Density). Daß diese damit de facto vorgenommene 'Verkleinerung' der 8-Zoll-Scheibe auf 5,25 Zoll überhaupt möglich war, lag an den Fortschritten bei der Datenträgerherstellung.

Diese ließ sich der anfangs einzige Anbieter (Maxell) auch fürstlich bezahlen, und normale 80-Spur-Disketten waren auf diesen Laufwerken nicht einsetzbar. Aber umgekehrt liefen übrigens auch diese ersten High-Density-Disketten nicht in herkömmlichen Laufwerken (andere magnetische Daten).

Damit fristeten diese Laufwerke zunächst ein sehr tristes Schattendasein. Sie konnten quasi nur in Rechner eingebaut werden, die in der Lage waren, Laufwerke mit 8-Zoll-Double-Density zu bedienen. Den Einbau konnten nur technisch versierte Leute vornehmen, da hierzu ja auch eine Konvertierung des 50poligen 8-Zoll-Bus an den 34poligen für 5,25-Zoll-Drives vorgenommen werden mußte; und die teuren Disketten waren zu nichts in der Welt kompatibel.

High and low

Etwa ein Jahr später erkannten die Hersteller die Nachteile solch einer 'Insellösung'. Sie schufen eine neue Generation von High-Density-Laufwerken, die dem Anwender auch Zugang zur alten 5,25-Zoll-Welt gewährte. Diese neuen HD-Laufwerke avancierten zu sogenannten Multifunktionslaufwerken (abgekürzt MF), die zwischen 'normalem' 80-Spur-Betrieb (300 U/min, 250 KBit/s, herkömmliche Magnetisierung) und High-Density-Betrieb (360 U/min, 500 KBit/s, hohe Magnetisierung) umschaltbar sind.

Die Umschaltung zwischen hoher und normaler Schreibdichte ist nicht einheitlich gelöst. Einige Laufwerke haben einen Jumper, mit dem man eine der beiden Betriebsarten fest wählen kann. Andere lassen sich nur über eine Bus-Leitung umschalten, wobei man aber die Polarität des Umschaltsignals wählen kann.

Auch bezüglich der Umschaltung der Motordrehzahl findet

man Unterschiede. So gibt es Laufwerke, etwa für den IBM AT, die grundsätzlich nur mit 360 U/min arbeiten. Um damit herkömmliche Disketten bearbeiten zu können, braucht man – wie der AT – einen Floppy-Controller, der mit der recht ungewöhnlichen Datentransferrate von 300 KBit/s aufwarten kann. Zum Thema AT können wir aber noch ausführlich.

Andere Laufwerke hingegen schalten zusammen mit der Schreibdichte auch stets die Drehzahl um. Noch andere, und neuerdings wohl die meisten, überlassen dem Anwender per Jumper die Wahl, ob er Betrieb in beiden Schreibdichten nur mit 360 U/min (für ATs also) oder mit Drehzahlumschaltung möchte.

Es ist wohl einzusehen, daß die Jumper-Stellungen sehr sorgfältig anhand der Dokumentation den jeweiligen Bedürfnissen anzupassen sind. Eine mit 360 U/min, einfacher Dichte und 250 KBit/s beschriebene 80-Spur-Diskette ist von keinem Nicht-Multifunktionslaufwerk lesbar, und umgekehrt sind 'regulär' beschriebene 80-Spur-Scheiben in einem schneller rotierenden MF-Laufwerk nur mit einer Datenrate von 300 KBit/s zu verarbeiten. Die meisten fabrikneuen MF-Laufwerke werden übrigens heute standardmäßig mit einer Jumperung für den IBM AT ausgeliefert.

Generationsprobleme

Wie erwähnt war die erste Generation HD-Laufwerke und -Disketten völlig unverträglich mit den herkömmlichen. Mit Aufkommen der MF-Drives ist eine Annäherung technischer Daten von HD-Laufwerken und -Disketten an die alte 5,25-Zoll-Welt zu beobachten. Es gibt also HD-Scheiben, die es in MF- und in Standardlaufwerken tun, aber es klappt nicht unbedingt bei jeder Disk-Sorte mit jedem Laufwerks-Fabrikat. Wichtig ist, daß sich solche Unverträglichkeiten nicht von Leseproblemen unterscheiden lassen, die etwa durch falsche Formate (Disk-Parameter) oder fehlerhaften Anschluß des Laufwerkes hervorgerufen werden.

Wenn Ihnen also ein Bekannter eine auf seinem Rechner Laufwerk erprobte High-

Density-Scheibe zum Testen mitgibt, diese aber in Ihrem Laufwerk nur Schreib-/Lesefehler produziert, so besorgen Sie sich sicherheitshalber zunächst einmal eine HD-Scheibe eines Fabrikats, das der Laufwerkshersteller empfiehlt, ehe Sie wochenlang Bus-Signale oszilloskopieren.

Ein weiterer 'kleiner Unterschied' zwischen Nur-HD-Drives und Multifunktionslaufwerken liegt darin, daß erstere eine sogenannte Schreibvorkompensation benötigen (unter anderem auch, damit sie gleichzeitig mit 8-Zoll-Drives an einem Controller laufen können). Diese Write Precompensation wird vom Controller ab Spur 43 (über das Signal 'Track greater 43') vorgenommen. Die inneren (kürzeren) Spuren ließen sich dadurch später besser lesen. Das Problem ist: Reine HD-Laufwerke müssen mit Vorkompensation betrieben werden. MF-Laufwerke hingegen verkraften diese in der Regel nicht!

Wenn das Formatierungsprogramm gleich ein Verify (Kontrolle der Formatier-Information) durchführt, merkt man schon beim Formatieren, wie es das eigene Laufwerk ab Spur 43 lieber hat. Hat man einen 'dummen' Formatter, erlebt man möglicherweise erst bei mehr als zur Hälfte mit Daten gefüllter Scheibe 'merkwürdige' Schreib-/Lesefehler mit herben Datenverlusten.

Extrawurst – Industriestandard PC

Wenn Sie glauben, daß der 'Industriestandard PC' jetzt eitel Kompatibilität beschert, müssen wir Sie bitter enttäuschen: Jetzt geht's nämlich erst richtig los.

PCs und ATs beziehungsweise die Schar der damit Kompatiblen bieten zwei besondere Probleme. Zum einen haben einige (!) dieser Geräte ein teilweise manipuliertes Anschlußkabel. Man erkennt es meistens schon daran, daß das Flachbandkabel nicht völlig unverseht, also 'in einem Stück' von Stecker zu Stecker geführt wird, sondern einige Adern 'umgelegt' wurden.

Aus nicht ganz überzeugenden Gründen bewirkt man damit, daß alle angeschlossenen Laufwerke (übrigens auch die Fest-

Der 34polige Floppy-Bus

Die derzeit am weitesten verbreiteten Floppy-Laufwerke sind (noch) die 5,25-Zoll-Laufwerke mit ihrem 34poligen Bus. Erfreulicherweise verfügen auch die neueren 3,5-Zoll-Laufwerke, die die 5,25-Zöller langsam ablösen, und sogar die 3,0-Zoll-Laufwerke (die sich aber wohl nicht auf breiter Ebene durchsetzen werden) über den gleichen Bus mit 34 Leitungen.

Bei diesem Bus führen alle Leitungen Masse (0 V), die mit Anschluß-Pins mit ungeraden Nummern verbunden sind. Das heißt, alle benachbarten Signalleitungen (an Pins mit geraden Nummern) werden bei Verwendung der üblichen Flachbandkabel durch Masseleitungen voneinander getrennt. Die Verbindung zum Floppy-Laufwerk erfolgt bei 5,25- und 3,0-Zöllern mit einem Platinen-Direktstecker, bei 3,5-Zoll-Drives auch vielfach über einen Pfostenstecker (weiblich). Der aktive Zustand sämtlicher Leitungen ist 'low', also etwa 0 Volt.

Pin 2

Dieser Anschluß ist bei verschiedenen Laufwerken unterschiedlich belegt:

1. Head Load
Laufwerkseingang. Damit wird der Schreib-/Lesekopf des Laufwerks geladen (an die Diskette gedrückt und vormagnetisiert). Nach diesem Vorgang sind Wartezeiten ähnlich wie nach einem Step-Impuls einzuhalten (siehe Pin 20).
2. In Use
Laufwerkseingang. Damit kann dem Laufwerk mitgeteilt werden, daß weitere Zugriffe erfolgen werden (oft kann man damit aber auch nur die Lampe an der Frontblende schalten).
3. High-/Normal-Density-Umschaltung
Laufwerkseingang. Mit diesem Signal wird bei Multifunktionslaufwerken die Betriebsart zwischen normaler Schreibdichte (300 U/min, 250 KBit/s Datenrate) und hoher Schreibdichte (360 U/min, 500 KBit/s) umgeschaltet. Die Polarität dieses Signals ist im allgemeinen auf dem Laufwerk per Jumper einstellbar.
4. Disk Change
Laufwerkseingang. Er signalisiert bei selektiertem Laufwerk, daß die Diskette gewechselt wurde. Meistens wird dafür ein Schalter am Diskettenschacht herangezogen beziehungsweise das 'Index'- oder 'Write-Protect'-Signal wird ausgewertet.

Pin 4

Auch dieser Anschluß ist bei verschiedenen Laufwerken unterschiedlich belegt.

1. Head Load
(siehe Pin 2, Punkt 1)
2. In Use
(siehe Pin 2, Punkt 2)
3. Disk Change
(siehe Pin 2, Punkt 4)

platten) stets über die Leitung 'Drive Select 1' aktiviert werden, also deren Jumper für die physikalische Laufwerksauswahl auf 'DS1' (zweites Laufwerk, da meist von 0 gezählt wird) gesteckt werden müssen.

Weiterhin hat man festgelegt, daß über das interne Kabel nur maximal zwei Laufwerke bedient werden. Die beiden verbliebenen Select-Leitungen werden – höchst ungewöhnlich – als separates Motor-on-Signal für jedes der beiden Laufwerke ausgeführt. Im Gegensatz zu anderen Rechnern, die mit einer Motor-on-Leitung also die Mo-

toren aller angeschlossenen Laufwerke gemeinsam einschalten, bedienen PCs und Abkömmlinge ihre Drives einzeln.

Allerdings läßt der PC nach außen (über eine Cannon-Steckverbindung) den Anschluß weiterer zwei Laufwerke zu, liefert also Select 3 und 4 nebst den Motor-on-Signalen. In einfachen PCs und XTs findet man gewöhnlich nur Controller, die Standardlaufwerke mit 250 KBit/s Transferrate bedienen können. Controller, die auch Multifunktionslaufwerke im High-Density-Modus unterstützen und 500 KBit/s Daten-

Pin 6

Auch dieser Anschluß hat bei verschiedenen Laufwerken unterschiedliche Belegung.

1. Drive Select 3
(siehe Pin 10, 12, 14)
2. Ready
(siehe Pin 34)

Pin 8: Index/Sector

Dieser Laufwerksausgang geht auf Low-Pegel, wenn das Indexloch der Diskette (bei 3,5-Zoll-Drives ist es eine Kerbe im Laufwerks-Teller) den Sensor passiert. Bei Standardlaufwerken (300 U/min) erscheinen daher fünf Impulse pro Sekunde, bei Laufwerken mit hoher Schreibdichte (360 U/min) sechs. Der Impuls kennzeichnet den Beginn einer Spur bei Soft-Sektorisierung und dient zur Synchronisierung des Floppy-Controllers. (Nur sehr wenige ältere Computer verwenden noch Disketten mit Hard-Sektorisierung, bei der für jeden Sektorbeginn ein Loch vorhanden ist.)

Diese Impulse können auch zur Drehzahlbestimmung vom Rechner ausgewertet werden. Die meisten Laufwerke generieren daraus intern das Ready-Signal, das anzeigt, daß der Motor die zum Betrieb nötige Drehzahl erreicht hat. Dieses Signal wird nur aktiv, wenn die dem Laufwerk zugeordnete Select-Leitung aktiviert wird.

Pin 10: Drive Select 0

Pin 12: Drive Select 1

Pin 14: Drive Select 2

Laufwerkseingang. Das folgende gilt auch für die Leitung 'Drive Select 3', falls diese vorhanden ist. Laufwerksintern kann über drei (oder vier) Steckbrücken entschieden werden, über welche Select-Leitung das Laufwerk angesprochen und aktiviert werden soll. Der Controller darf nur dann mehr als eine Select-Leitung aktivieren, wenn sichergestellt ist, daß dabei nicht zwei Laufwerke gleichzeitig über dieselben Ausgangsleitungen antworten (Bus-Crash). Ebenso dürfen auf einem Laufwerk nur dann mehrere Select-Jumper gesteckt sein, wenn ausgeschlossen werden kann, daß dadurch mehr als ein Laufwerk aktiviert wird.

Einige Laufwerke können auch in einen All-Select-Modus versetzt werden (wirkt, als wären alle Brücken gesteckt). Das kann sinnvoll sein, wenn nur ein Laufwerk angeschlossen ist. Bei einigen älteren Laufwerken (vorwiegend allerdings achtzölligen) ist auch eine Multiplex-Betriebsart möglich. Dabei aktiviert der Controller mehrere Select-Leitungen, die dann aber als binäre Adresse ausgewertet werden. Dadurch lassen sich über drei Leitungen acht, über vier 16 Laufwerke auswählen.

Pin 16: Motor on

Laufwerkseingang, über den der Motor des Laufwerks eingeschaltet wird. Hier kann bei vielen Laufwerken durch Jumper bestimmt werden, ob der Motor unabhängig von den Select-Leitungen eingeschaltet wird oder nicht. Bei einigen Disk-Controllern sollte man die Stellung für unabhängiges Einschalten des Laufwerks wählen, da die Select-Signale nicht kontinuierlich antehen (etwa beim µPD 765).

Normaldisketten (40-Spur-Betrieb, 360 KByte) nicht auf 300 U/min zurück, sondern lassen diese weiterhin mit 360 U/min rotieren.

Damit man kompatibel zu herkömmlichen 40-Spur-Laufwerken schreiben und lesen kann, muß dazu nur die Datentransferrate nicht auf die üblichen 250 KBit/s, sondern auf 300 KBit/s heruntergeschaltet werden, was der Controller auch tut. Dieser durchaus clevere Maßnahme verdankt der AT einen überdurchschnittlich schnellen Zugriff auf diese Disketten.

Alleskönner AT

Die meisten Exemplare dieser Gattung (zum Beispiel das Original oder der Kaypro 286i) schalten – wie bereits angedeutet – die Geschwindigkeit der MF-Laufwerke bei Betrieb mit

Außerdem kann es oft vorteilhaft sein, die Option einzustellen, bei der zusammen mit 'Motor on' der Kopf des Laufwerks geladen wird (siehe Text).

Pin 18: Direction Select

Über diesen Eingang des Floppy-Laufwerkes bestimmt der Controller, in welche Richtung der Schreib-/Lesekopf des Laufwerkes bei einem Spurwechsel bewegt wird. Dabei bewirkt Low-Pegel die Bewegung zur Mitte der Diskette (höhere Spurnummern), High-Pegel die Bewegung nach außen (zur Spur 0). Dieses Signal wird nur dann abgefragt, wenn gleichzeitig ein Step-Impuls vorliegt.

Pin 20: Step

Laufwerkseingang. Jeder Impuls (Low-Pegel) an diesem Anschluß bewirkt bei selektiertem Laufwerk eine Bewegung des Schreib-/Lesekopfes in die durch das Signal 'Direction Select' vorgegebene Richtung. Die Bewegungsgeschwindigkeit wird durch die Wiederholrate (Step Rate) des Step-Impulses bestimmt. Die Zeiten liegen bei modernen Laufwerken zwischen drei und sechs Millisekunden, bei älteren und besonders bei Billiglaufwerken können sie bis zu 30 Millisekunden betragen. Die aktuellen Zeiten sind dem jeweiligen Datenblatt des Laufwerkes zu entnehmen und gegebenenfalls in der Steuerungssoftware des Floppy-Controllers (meist im Betriebssystem) an die Laufwerke anzupassen.

Durch die Abhängigkeit vom Select-Signal können die Step-Impulse für mehrere Laufwerke verschaltet werden, so daß diese gleichzeitig ihren Kopf positionieren können. Dies wird zwar von manchen Disk-Controllern unterstützt, doch die meisten preiswerten Betriebssysteme nutzen solche Spezialitäten nicht.

Nach einer Positionierung des Kopfes sind beim Lesen und Schreiben Wartezeiten einzuhalten, da der Kopf sich erst in seiner neuen Position stabilisieren muß. Bei Lesevorgängen ist dies nicht so wichtig, da hier Lesefehler durch den Disk-Controller oder das Betriebssystem des Rechners (erneutes Lesen) ausgeglichen werden, aber beim Schreiben ist Vorsicht geboten, da durch Positionsfehler der beschriebene Sektor unleserlich werden kann. Die Wartezeiten sind dem Datenblatt des Laufwerkes zu entnehmen (notfalls kann man mit 15 bis 25 Millisekunden experimentieren).

Pin 22: Write Data

Laufwerkseingang. Über diese Leitung werden die zu speichernden Daten seriell ans Laufwerk übertragen. Voraussetzung ist, daß das Signal 'Write Gate' aktiv und das Laufwerk 'ready' ist.

Es gibt unterschiedliche Übertragungsverfahren. Allen gemeinsam ist, daß immer ein kompletter Sektor übertragen werden muß. Die Datentransferrate hingegen variiert je nach Verfahren und Taktrate zwischen 125 und 500 KBit/s (siehe Text).

Pin 24: Write Gate

Laufwerkseingang. Mit diesem Signal wird das zum Schreiben der Daten nötige Löschen der alten Daten eingeschaltet und die Schreib-

daten (Write Data) werden freigegeben. Voraussetzung ist wieder, daß das Laufwerk selektiert und 'ready' ist.

Pin 26: Track 0

Dieser Laufwerkseingang signalisiert bei selektiertem Drive, daß der Kopf auf Spur 0 positioniert ist.

Pin 28: Write Protect

Laufwerkseingang. Signalisiert bei selektiertem Drive, daß der Schreibschutz der Diskette aktiviert ist und das Laufwerk keine Daten auf Diskette schreiben (und damit ändern) kann.

Pin 30: Read Data

Laufwerkseingang. Wenn das Drive selektiert und 'ready' ist, 'Write Gate' hingegen inaktiv, werden die Daten von der Diskette gelesen und zum Floppy-Controller im Rechner übertragen.

Pin 32: Side 1 Select

Laufwerkseingang. Über ihn wird bei zweiseitigen Laufwerken (Schreib-/Lesekopf für jede Diskettenseite) bestimmt, welcher Kopf benutzt wird. Mit High-Pegel wird die Seite 0 (die auch von einseitigen Laufwerken benutzt wird), mit Low-Pegel die Seite 1 angesprochen.

Nach einem Umschalten des Kopfes sollte mit einem Schreibvorgang einige Zeit gewartet werden, damit die Schreiblogik des Laufwerkes Zeit hat, auf den anderen Kopf umzuschalten (im Datenblatt des Laufwerkes nachsehen oder 1 bis 2 Millisekunden warten). Wird diese Zeit nicht eingehalten, so kann der Schreibvorgang einen Sektor zerstören.

Pin 34:

Auch dieser Pin wird im allgemeinen unterschiedlich belegt:

1. Ready

Laufwerkseingang. Diese Leitung übermittelt bei selektiertem Laufwerk die Einsatzbereitschaft: eine Diskette ist eingelegt, der Motor läuft auf Nenndrehzahl. Das Laufwerk ist also (bis auf 'Headload') bereit, eine Funktion wie Lesen, Schreiben oder Positionieren des Kopfes auszuführen. Dabei ist bei manchen Laufwerken das Ready-Signal für die Positionierung des Kopfes nicht unbedingt erforderlich, und es gibt Controller, die den Kopf positionieren, ohne daß er geladen sein muß. Diese Leitung ist bei Laufwerken, die Anschluß 6 als Ready-Signal benutzen, meistens frei.

Erzeugt ein Laufwerk kein Ready-Signal, muß man es für einige Controller künstlich erzeugen (Zusatzhardware, siehe Text). Ist es dem Controller 'egal', ob er ein Ready-Signal bekommt oder nicht, muß die Steuerungssoftware nach dem Ansprechen des Laufwerkes ('Motor on' und 'Select' aktiv) die Hochlaufzeit des Motors und die Kopfplazeit durch geeignete Zeitschleifen abwarten (Laufwerksdaten dem Handbuch entnehmen).

2. In Use

(siehe Pin 2, Punkt 2)

3. Disk Change

(siehe Pin 2, Punkt 4)

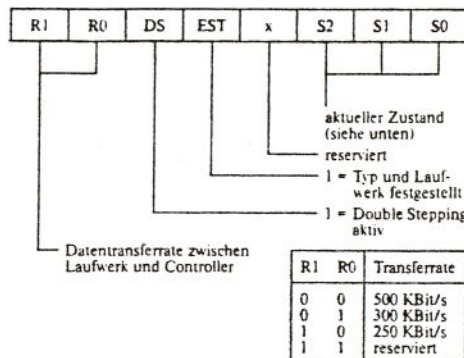
Signal liefert. Alle Probierelei mit SETUP oder Treibern erlöst einen nicht: Solange der AT hier Signale empfängt, behandelt er das neue Drive als MF-Laufwerk. Erst wenn diese Leitung gekappt wurde, enthält der AT neue Controller-Fähigkeiten: kein Double-Stepping mehr, 250 KBit/s Datenrate. Bei PC DOS 3.1 läßt sich das neue Laufwerk formatieren, indem man nicht mehr die Option '/4' angibt. Ab DOS 3.2 wird's komplizierter. Hier stellt sich FORMAT.COM ohne '/4' auch bei dem einfachen Laufwerk auf das High-Density-

Format ein, und mit der Option auf 300 KBit/s und Double-Stepping. Abhilfe: Man muß den mitgelieferten Treiber DRIVER.SYS mit dem Aufruf

DEVICE = DRIVER.SYS

D:1 F:0

in CONFIG.SYS eintragen. Das bezeichnet das zweite physikalische Laufwerk, und F:0, daß es sich um ein echtes 360-KB-Laufwerk handelt. Achten Sie unbedingt auf das Leerzeichen vor dem ersten Schrägstrich, sonst gibt es eine Fehlermeldung. Der Treiber gibt als Meldung aus, unter wel-



S2	S1	S0	aktueller Zustand
0	0	0	360-KB-Disk in 40-Spur-Laufwerk, Status ungeprüft
0	0	1	360-KB-Disk in MF-Laufwerk, Status ungeprüft
0	1	0	1,2-MB-Disk in MF-Laufwerk, Status ungeprüft
0	1	1	360-KB-Disk in 40-Spur-Laufwerk, Status ungeprüft
1	0	0	360-KB-Disk in MF-Laufwerk, Status geprüft
1	0	1	1,2-MB-Disk in AT-Laufwerk, Status geprüft

chem Laufwerksbuchstaben er zu finden ist.

Jetzt wird FORMAT.COM mit Angabe des entsprechenden Laufwerksbuchstaben aufrufen, aber ohne Zuhilfenahme weiterer (verlockender) Parameter, die das DOS-Handbuch bereithält. Schreiben und lesen kann man anschließend wieder über Laufwerk B:.

80 Spuren / 720 KB

Und da wir nun mal dabei sind: Natürlich kann man auf diese Art (aber nur ab DOS-Version 3.2!) auch einfache 80-Spur-Laufwerke beziehungsweise 3,5-Zoll-Drives anschließen. Zu letzteren kommen wir noch, aber hier sei noch mal darauf hingewiesen, daß auch hier wieder Pin 34 (Ready oder Disk Change) stillgelegt werden muß! Das Formatieren klappt ebenfalls wieder nur über Treiber, die bei Standard-80-Spur-Drives folgenden Aufruf enthalten müssen (aus Platzgründen schreiben wir in zwei Zeilen):

DEVICE = DRIVER.SYS

D:1 F:2 T:80

Auch bei diesem Format kann man nach dem Formatieren weiter unter dem Buchstaben B: arbeiten, da sich die neuen DOS-Versionen die vollständige Formatinformation beim ersten Zugriff aus dem Bootsektor holen und den Rechner korrekt darauf einstellen.

Tabelle 1. Die Bedeutung der einzelnen Bits in den Disk-Status-Zellen des AT. Die Speicherstellen liegen im RAM an Adresse 0040:0091 (1. Laufwerk) und 0040:0092 (2. Laufwerk). 'Status geprüft' (established) heißt, daß dieser Status durch einen Zugriff verifiziert wurde.

liegen für die beiden physikalischen Laufwerke zwei Speicherzellen, die Auskunft über die eingestellte Laufwerksbetriebsart geben.

Wenn man diese manipuliert, so kann man den AT zu allem überreden, was Tabelle 1 und das jeweilige Laufwerk hergeben – und damit natürlich auch das 80-Spur-Format mit 720 KB Speicherplatz mittels Multifunktionslaufwerk erzwingen. Letzteres allerdings nur bis zum nächsten Öffnen der Laufwerksklappe (Disk Change); denn dann datiert der AT diese Zellen wieder mit den ihm aus dem ROM-BIOS geläufigen Werten auf.

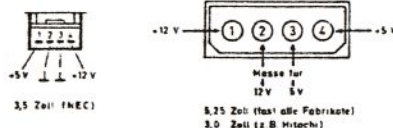
Nach dem Kaltstart steht bei 40:90 (und 91, wenn zwei Laufwerke im System sind) üblicher-

weise 61h. Nach einem Zugriff auf 360-KB-Scheiben im Multifunktionslaufwerk wird der Wert auf 74h geändert, also sind Double-Stepping und 300 KBit/s aktiv. Nach dem Zugriff auf eine 1,2-MB-Floppy findet man folgerichtig 15h. Nachdem eine Diskette in einem 360- oder 720-KB-Laufwerk gelesen wurde, wird dieser Wert auf 93h gesetzt.

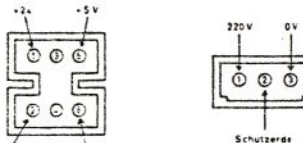
Interessant dabei ist, daß auch der Startwert 61h eine Rate von 300 KBit pro Sekunde einstellt, aber dem Controller scheint es nichts auszumachen, wenn das Laufwerk den Bootsektor mit 250 KBit/s anliert. Das Double-Stepping stört beim Lesen von Spur 0 ohnehin nicht.

Verändert man aber diesen Wert wesentlich auf 15h, etwa durch Aufruf von FORMAT ohne Parameter auf Drive B:, dann ist es aus. Das ROM-BIOS des AT probiert in Multifunktionslaufwerken nach Erkennen eines Disk-Wechsels zunächst aus, ob eine 360-KB-Disk einliegt, dann wechselt es zu High Density, aber nicht mehr zurück. Da ein Laufwerksstatus 15h durch das fehlende Disk-Change-Signal von einem einfachen Laufwerk nicht aufgehoben werden kann, bleibt nur die Korrektur im RAM beziehungsweise ein Warmstart.

Um nun mit MF-Laufwerken 80 Spuren schreiben und lesen zu können, braucht man nur die bewußte Diskette einzulegen und den Status auf 54h zu ändern, denn man sollte das mit 360 U/min laufende Drive natürlich mit 300 KBit/s bedie-



Betriebsanschlüsse für 8-Zoll-Drives neuerer Bauart (z.B. Shugart SA 810/860)



Ältere 8-Zoll-Laufwerke besitzen den Laufwerksmotor mit Netzspannung 220V. So ist zum Beispiel das Siemens FDD 105-80 mit dieser beiden Steckern ausgestattet

Vielfalt statt Standard, auch bei den Steckverbindungen für die Betriebsspannungszuführung. Man kann die Anschlußstecker für 5,25- und 3,5-Zoll-Laufwerke zwar nicht an die jeweiligen anderen Drives anschließen, aber die vertauschte Zuordnung von Pin-Nummern zu Betriebsspannungen kann tödlich in die Irre führen. Die dargestellten Steckverbindungen sind die Versionen am Floppy-Laufwerk, sozusagen 'aus der Sicht des Steckers'.

HEFT
20
August
1987

86

Bus-Pin	BASF 6106/08	Tandon TM 50-1/2	Shugart SA 400 L	Phillips X 3111 bis 3114	BASF 6128/38	NEC 1035	Hitachi HFD 305 SX	Mitsubishi M 4854	TEAC 55 GFV
2	Head Load	Connector Clamp	-	Ready/ Head Load	Head Load	-	-	Two Sided/ Drive Selected	High/ Normal Density
4	-	Spare	-	In Use/ Select 4	In Use (opt.)	Head Load/ In Use	In Use	In Use/ Head Load	In Use/ Head Load/Open
6	Ready	Drive Select 3	Drive Select 4	Ready/Motor on	Ready/Select 4	Drive Select 3	Drive Select 3	Drive Select 3	Drive Select 3
8	Index	Index/Sector	Index/Sector	Index	Index	Index	Index	Index	Index
10	Select 1	Drive Select 0	Drive Select 1	Select 1	Select 1	Drive Select 0	Drive Select 0	Drive Select 0	Drive Select 0
12	Select 2	Drive Select 1	Drive Select 2	Select 2	Select 2	Drive Select 1	Drive Select 1	Drive Select 1	Drive Select 1
14	Select 3	Drive Select 2	Drive Select 3	Select 3/In Use	Select 3	Drive Select 2	Drive Select 2	Drive Select 2	Drive Select 2
16	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on
18	Direction in	Direction Select	Direction Select	Direction	Dir Sel	Direction Select	Direction Select	Direction Select	Direction Select
20	Step	Step	Step	Step	Step	Step	Step	Step	Step
22	Write Data	Composite Write Data	Write Data	Write Data	Write Data	Write Data	Write Data	Write Data	Write Data
24	Write Gate	Write Enable	Write Gate	Write Gate	Write Gate	Write Gate	Write Gate	Write Gate	Write Gate
26	Track 00	Track 0	Track 00	Track 00	Track 00	Track 0	Track 00	Track 00	Track 00
28	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect
30	Read Data	Composite Read Data	Read Data	Read Data	Read Data	Read Data	Read Data	Read Data	Read Data
32	Side Select	Side Select	-	Head Select	Side Select	Side Select	Side 1 Select	Side One Select	Side One Select
34	In Use/ Change Disk	Connector Clamp	-	In Use/Disk Change/Ready	In Use/Disk Change/Ready	Ready	Ready	Ready	Ready/ Disk Change
Anzahl Steck- brücken	10	8	8	29	27	bis zu 25	4	11	17
Laufwerks- Typen	5,25 Zoll, 40 Spuren, ein-/zweiseitig	5,25 Zoll, 40 Spuren, ein-/zweiseitig	5,25 Zoll, 40 Spuren, einseitig	Vier verschie- dene 5,25-Zoll- Drives, 40 und 80 Spuren, ein- und zweiseitig	5,25 Zoll, zweiseitig, 40/80 Spuren	3,5 Zoll, div. 80-Spur- Drives	3,0 Zoll, 40 Spuren, einseitig	5,25 Zoll, 80 Spuren, zweiseitig, nur High Density	5,25 Zoll, 80 Spuren, zweiseitig High/Normal Density

Tabelle 2. So verschieden können Floppy-Bus-Anschlüsse sein. Hier ein bunter Querschnitt recht weit verbreiteter 5,25- und 3,5-Zoll-Floppy-Laufwerke verschiedensten Typs und Alters. Wir haben auch bewußt die teilweise recht unterschiedlichen Originalschreibweisen für die verschiedenen Signale übernommen.

nen. Auch Formatieren geht, wenn auch nur wieder über den Treiber. Aufruf: wie oben /D:0 (erstes Drive) oder /D:1 (zweites) und /F:2/T:80. Wie man das Ganze so bastelt, daß der Status nicht verlorengeht, beschreiben wir in einem der nächsten Hefte.

Die Kleinen

Nun zu den 3- und 3,5-Zöllern. Der offensichtlichste Unterschied des Anschlusses liegt darin, daß hier eine Pfostenleiste statt eines Stricks Platine zur Kontaktierung benutzt wird. Ein Pfostenstecker statt des üblichen Platinensteckers angepreßt, und das war's. Die Signalbezeichnungen stimmen (mit allen Anmerkungen) mit

Die wirklichen Probleme beginnen damit, daß die meisten neuen Laufwerke mit stromsparenden und etwas leistungsschwachen CMOS-ICs ausgerüstet sind. Dadurch sind sie nicht mehr in der Lage, die bei den anderen Drives bislang üblichen Abschlußwiderstände (zur Bus-Terminierung) von 150 Ohm zu treiben. Sie brauchen zunächst leichtere Lasten auf dem Disk-Controller (Werte von 1,5 kOhm sind angebracht).

Hat man ein gemischtes System aus 3,5- und 5,25-Zöllern, so müssen auch die Widerstands-Arrays auf letzteren vergrößert werden. Können die meisten 5,25-Zoll-Laufwerke und Controller sogar zwei bis drei Laufwerke mit 150-Ohm-Arrays ver-

tungen verwenden, nämlich bei Laufwerkeinstellung 1 und 2. Bei einer weiteren Sorte von Laufwerken wird man mit einer Motordrehzahl von 600 U/min überrascht! Diese Exoten (Sony) sind aber höchst selten. Sie bieten nicht etwa eine erhöhte Kapazität, sondern nur einen doppelt schnellen Datentransfer, vorausgesetzt, man hat einen darauf eingerichteten Controller (500 KBit/s).

Die neuen High-Density-Drives, die beispielsweise in IBMs neuer Personal-System/2-Familie Dienst tun und 1,44 MByte fassen, begnügen sich wieder mit den altbekannten 360 U/min.

Den krönenden Abschluß bildet die Firma Schneider bezie-

die Anschlußstecker einfach spiegelbildlich belegt. Um an einem CPC an den externen Stecker ein 'Fremdlaufwerk' anzuschließen, ist man gezwungen, sich einen Stecker mit vertauschten Anschlüssen zu löten. (Kleiner Tip für Ungläubige: man kann die Preßsteckverbindungen beliebig aufsetzen, die Spiegelung bleibt!)

Grenzenlose Vielfalt

Die aufgezeigten Probleme machen es wohl überdeutlich: unbekannte Laufwerke ohne detaillierte Unterlagen zum Laufen bringen zu wollen kann einigermaßen spannend sein. Dies gilt besonders für gebrauchte 8-Zoll-Laufwerke, da bei diesen derart viele Varianten (auch mit verschiedenen Bus-Anschlüssen) im Umlauf sind und bei einigen Typen so viele Jumper-Optionen möglich und nötig sind, daß man sich selbst mit passender 'Landkarte' kaum zurechtfindet.

Aber auch bei 5,25-Zöllern sollte man nicht wahllos zugreifen. Für den Apple II und dessen TTL-Controller umgebaute Laufwerke laufen meist nicht an normalen Floppy-Controllern, umgekehrt geht's fast nie. Während die Aufzeichnungsverfahren mit Standard-Controllern unabhängig von den Daten eine relativ konstante Frequenz aufweisen, auf die die meisten Laufwerke mit Filtern abgestimmt sind, variiert dies beim Apple-Aufzeichnungsverfahren sehr stark. Ohne Umbauanleitung ist man hier meist aufgeschmissen.

Des weiteren gibt es sehr alte 5,25-Zoll-Laufwerke, die nur 35 statt 40 Spuren haben. Der Unterschied ist quasi nicht zu sehen, aber natürlich im Betrieb deutlich zu merken. Haben diese Laufwerke schon einen Spindelantrieb zur Kopfpositionierung, so kann man unter Umständen den Anschlag entfernen und noch ein paar Spuren rausschinden. Wird aber über eine Plastikscheibe positioniert, so hat man Pech gehabt.

Auch 3,5-Zöllern mit nur 40 Spuren sind im Umlauf. Diese lassen sich dann zwar gut dort einsetzen, wo bisher etwa 5,25-Zöller mit 40 Spuren verwendet wurden. Mit Datenaustausch Richtung Atari oder Amiga ist's dann aber erst mal nichts.



Laufen, was heißt laufen? Wenn es laufen würde, brauchte ich den Fehler ja nicht zu suchen.

Bus-Pin	Siemens FDD 100-8 D	Shugart SA 810/860	Controller-Belegung Janich & Klass, FDC 8/5	
			8"-Shugart-Bus	5,25"-Philips-Bus
2	-	Ext. Wrt. Current Switching (opt.)	Low Current	RD Data
4	-	-	-	Head Load
6	-	-	-	Track
8	-	True Ready (opt.)	-	Index
10	-	Two Sided (opt.)	Two Sided	Low Current/Terminator
12	Disk Change	Disk Change (opt.)	Fault	Step
14	-	Side Select (opt.)	Side Select	Direction
16	In Use	In Use (opt.)	Fault Reset	Write Gate
18	HDL/D	Motor on/Head Load	Head Load	WR Data
20	Index	Index	Index	Select 1
22	Ready	Ready	Ready	Select 2
24	Sector	Sector	-	Select 3
26	Select 0	Drive Select 1/Side Select	Select 1	Motor on
28	Select 1	Drive Select 2/Side Select	Select 2	Rdy 1
30	Select 2	Drive Select 3/Side Select	Select 3	Rdy 2
32	Select 3	Drive Select 4/Side Select	Select 4	Rdy 3
34	Step in	Direction Select/Side Select	Direction	Rdy 4
36	Step	Step	Step	Write Protect
38	Wrt Data	Write Data	WR Data	Terminator
40	Write	Write Gate	Write Gate	Side Select
42	Track 00	Track 00	Track 0	-
44	Wrt Protect	Write Protect	Write Protect	-
46	Raw Data	Read Data	RD Data	Terminator
48	Sep Data (opt.)	Sep Data	-	-
50	Sep Clk (opt.)	Sep Clock	-	-
Laufwerks-Typ	8 Zoll, 77 Spuren, einseitig	8 Zoll, 77 Spuren, ein-/zweiseitig	-	-

denen bei 5,25-Zoll-Laufwerken überein.

Lebenswichtig aber ein Unterschied bei den Betriebsspannungssteckern! Bei 5,25- und bei 3,5-Zoll-Laufwerken kommen vierpolige Stecker zum Einsatz. Diese Stecker sind zwar völlig inkompatibel, aber ihre Pins sind bei beiden zur Kennzeichnung der anliegenden Betriebsspannungen jeweils von 1 bis 4 durchnummeriert - allerdings entgegengesetzt (siehe Bild)!

kraften (obwohl nach Vorschrift immer nur das letzte am Kabel eines haben soll), so ist das für 3,5-Zöller und übrigens auch für ganz neue 5,25-Zoll-Drives zuviel des Guten. Ein gemeinsames Array von 470 Ohm wird im allgemeinen von allen Drives als vertretbarer Kompromiß hingenommen.

Als nächstes tauchen die Laufwerke mit kodierten Drive-Select-Leitungen auf. Hier lassen sich meist nur zwei Select-Lei-

Tabelle 3. Ohne tiefere Erklärung auch ein paar typische Bus-Belegungen für 8-Zoll-Laufwerke. Nicht zu vergessen der 50polige 5,25-Zoll-Anschluß von Philips.

hungsweise Amstrad. Obwohl die 3,0-Zoll-Drives in den CPCs etwa so kompatibel zu 5,25-Zöllern sind wie letztere untereinander (bis auf die gleichfalls schwächeren Treiber), hat man

BÜRSE -- BÜRSE -- BÜRSE

Ich kann folgende Sachen günstig besorgen:

1 x HRS 1B Grafik mit 384x192 Punkten von RB-Elektronik
früher kostete der Fertigbaustein 369,- DM
jetzt nur 100,- DM (incl. Software)

1 x Computer Netzteil mit +/- 5V und +/- 12V
60 Watt, komplett fertig
50,- DM

2 x TEAC A Laufwerke 40 Track SS/DD, same line
hervorragend für TRS-80
je Laufwerk 90,- DM

1 x 80 Zeichen-Karte von Schmitke-Elektronik für den
VIDEO GENIE I mit Banking-Logik
und Software
200,- DM

nagelneue Drucker, originalverpackt, 6 Monate Garantie:
198,- DM

Star DP 510

Star Gemini 10 xi

Star Delta 10 x

100 Zeichen/s

120 Zeichen/s, IBM-Zeichensatz

298,- DM

160 Zeichen/s, BK-Buffer, parallel +
seriell: 398,- DM

Andreas Rychlik, Königsberger Allee 120; 4100 Duisburg 1

Angebot für den Bastler:
2 Cassettendrive's von ADAM
1 Tastatur (ADAM)
sowie 3 Programm-Cassetten mit Programmen
für ADAM-Rechner zu verschenken. (gestiftet von Peter Spies)
Gesamt oder einzeln bei der Redaktion erhältlich.

Verkaufe:
TRS 80 M I
MDX 2 Expansio Interface (defekt) incl. Doubler
Floppystation mit 1 TEAC FD55FU (2x80,DS,DD)
ohne Netzteil 2 BASF 6106 (40,DD)
Drucker DHR 88
gegen Gebot gesamt oder einzeln

Weiterhin abzugeben:
Manuals: TRSDOS, NEWDOS 80, PASCAL 80, Mumath, Musimp,
SUPER DMS, MZAL-Assembler, Scripsit, ZBASIC,
VISICALC, Alcor Pascal, Fortran 80
gesamt oder einzeln gegen Versandkosten
Josef Konrad — Anzengruber Straße 35 — 8838 Gröbenzell

Suche RS232 für Modell 4
Rüdiger Sörensen, ☎ 06131/32860

Suche MS-DOS Programm
zum Lesen von NEWDOS80-
Disketten unter MS-DOS.

P. Spieß
Trugenhofenerstr. 27
8859 Rennertshofen 1

* = * In VISICALC sortieren * = * ===== Zu INFO 19 S.29-31

VISICALC bietet ein recht nützliches Instrumentarium: Man kann damit bequem sehr umfangreiche Tabellen schreiben; man kann Formeln eingeben - sogar trigonometrische -, deren Ergebnisse von VISICALC bei jeder beliebigen Änderung automatisch neu berechnet werden - eben weil hierfür nicht Zahlenwerte, sondern Formeln eingegeben werden - man kann (Version 3) Relationen zu Bezugsskalen herstellen lassen, man erhält auf einfache und schnelle Weise Summen oder Durchschnitte (Mittelwerte) von Zeilen oder Spalten, man kann beliebig lange Kommentare eingeben und hierfür die Spaltenbreite anpassen - eine bestimmte Version arbeitet sogar auf Wunsch mit unterschiedlichen Spaltenbreiten im gleichen Rechenblatt! - man kann die Daten in Kombination mit dem Programm "VISIGRAF" grafisch darstellen (Diagramme zeichnen); man kann VISICALC bis zu einem gewissen Grade sogar als kleine übersichtliche Datenbank benutzen (ich verwalte hiermit meine Eingaben und Ausnahmen); man kann... man kann vieles, aber eben nicht alles, was man mit BASIC machen könnte - zum Beispiel: sortieren!

Für unseren Bremerhavener Schwesterclub - Verzeihung: Bruderclub ("Club" ist maskulin...) - hatte ich vor Jahren ein Programm geschrieben, mit dem man eine VISICALC-Datei in BASIC-Variablen einlesen, diese manipulieren und danach wieder in eine VC-Datei zurückschreiben kann. Voraussetzung hierfür ist, daß die VC-Datei im sogenannten "DIF-Format" vorliegt; das ist das system- und sprachunabhängige "DATA INTERCHANGE FORMAT". Wer es näher studieren möchte, lese

Hergert, "Erfolg mit VISICALC" (SYBEX), oder
 "DIF: A Format for Data Exchange between Applications Programs"
 in BYTE, Nov. 81, S.174-206, oder
 D.Haan, "What's the DIF?", 80 MICRO Sept. 83, p.108-114

BASIC besitzt die sehr vielseitige Sortier-Routine CMD "Q". Hiermit können sowohl Numeri als auch Strings sortiert werden, sowohl auf- wie abwärts, nach einer oder gleichzeitig mehreren Feldvariablen, sowohl direkt wie auch indirekt (indiziert). Das vermißte ich bei VISICALC.

Ich habe deshalb meine damals veröffentlichte "Utility" dazu verwendet, ein Sortierprogramm für VC-Dateien zu bauen. (Danke für den Hinweis, lieber Leser, daß es dafür den "VCSORTER/BAS" gibt; aber der funktioniert nicht! Er setzt außerdem voraus, daß man die VC-Datei vor seiner Anwendung vergewaltigt, da man die zu sortierende Zeile oder Spalte zuerst an den Blattrand "noven" muß; das habe ich nicht so gern...!)

Hat man die VC-Datei erst einmal mit "/S#S.Filename, untere rechte Ecke" im DIF-Format gespeichert, so besorgen Menu und Dialog in meinem Programm alles weitere. Es können unabhängig voneinander Zeilen oder Spalten sortiert werden, auf- oder abwärts, Numeri oder Strings: das Programm erkennt, worum es sich handelt. Da die VISICALC-Spalten keine Nummern, sondern eine alphabetische Kennzeichnung haben (wobei das Alphabet zweieinhalbmal durchlaufen wird, d.h. nach "Z" folgt "AA, AB, ..." usw.), kann man statt einer Nummer auch diese Alpha-Kennzeichnung eingeben; das Programm rechnet sie in den entsprechenden Zahlenwert um.

Das Programm kann auch dazu dienen, eine VISICALC-DIF-Datei lediglich auf dem Bildschirm anzusehen oder auszudrucken. Ohne VISICALC zu laden (oder zu besitzen). Dabei werden gleichzeitig Zeilen- und Spaltennummer jeder Eintragung angegeben. Selbstverständlich steht das Programm in unserer Programm-Disketothek zur Verfügung (jetzt bei Clubkamerad Werner Förster).

X.-T. Kühlenwein

Fehler!

In seinem vorletzten Brief (den letzten könnt ihr auf Seite 2 lesen) hat mich Kajott auf ein paar Fehler in meinem Artikel "Extended NewDOS" aufmerksam gemacht. Natürlich darf und will ich diese Fehler nicht unter den Tisch kehren, da eventuell Maschinensprache-Anfänger diese Texte studieren und dann durch falsche Informationen verwirrt werden können! Aus diesem Grund hier ein Ausschnitt aus dem Brief von Kajott:

Ach so, noch das PS wegen der Fehler in vorgenannten Beitrag:
 2.Seite...untere: Eine Granule hat 1985 5 Sektoren, egal bei welcher Disk-Type: zehn Sektoren sind also stets gleich zwei Granules, nicht nur bei SS/SD.
 4.Seite...hinter der Directory-Tabelle...in unteren Drittel: die Bit-Folge muß 10110111 lauten und entspricht B7h, nicht B5h.
 "Platz...in SYS-1...untere: In den aufgeführten 36 Bytes sind nicht "nur relative" Sprünge (JR); in Zeile 50CD steht ein absoluter Sprung (JP 7, M519D). Dieser führt allerdings sowieso aus diesem Bytebereich hinaus und ist deshalb für die Verschiebung der 36-Byte-Folge irrelevant. Nächste Seite...2.Absatz: Du schreibst, "Der Platz dort ist nur 32 Bytes lang". Das stimmt nicht, er ist sogar nur 23 Bytes lang. (Das kann natürlich auch ein Vertipper sein, ist aber für den interessierten Leser, der daraus lernen will, leider verwirrend, weil er ja (wie ich als Anfänger z.B.) zunächst alles glauben muß, was in einer Info steht! (So geht es mir aber dauernd mit den berühmten Lehrbüchern, denen ich jedes Wort glaube, weil ich, solange ich noch daraus lerne, die vielen Fehler darin ja noch nicht erkenne. Inzwischen habe ich sogar in "GROSSER" einige gravierende Fehler gefunden! Wie soll da einer weiterkommen...
 Gleiche Seite...5.Absatz: Da steht: "...muß man im Sektor 0 rel. die Bytes 25/26 von B750 in B51 ändern." M.E.muß es B51 (statt B51) heißen! Denn die Verschiebung beträgt 264 Bytes, nicht 260!

Mit schlechtem Gewissen gelobe ich Besserung

Karlmut Obermann

Ergänzung/Berichtigung zu "RESET-feste MEMDISK!"

Bei meinem MEMDISK-Patch am Montezuma-CP/M (Info 19 Seite 43/44) ist mir ein Fehler(chen) unterlaufen. Der Fehler resultierte aus dem Umstand, daß ich so kurze Patches normalerweise nicht mit dem Assembler, sondern per Hand und Liste "assembliere". Um die Sache im Info veröffentlichten und von Fall zu Fall einen Sektordump.

Hier ist mir der Fehler unterlaufen, daß ich im Assemblerlisting CP "N" statt CP "n" eingetragen habe. Danach habe ich auf einer Musterdiskette das Ergebnis in die Sektoren eingetragen und ausgedruckt. Ausprobiert habe ich diese Version des Patches nicht mehr, da ja meine per Hand (und mit CP "n") übersetzte Idee funktionierte!

Natürlich funktioniert, rein vom Programm her gesehen, auch das abgedruckte Programm, aber man muß jedesmal SHIFT drücken und das kann man sich wirklich ersparen! Wie Ihr seht, entstehen Fehler manchmal auf sehr seltsame Weise und können nur durch konsequente Dokumentation dessen, was man wirklich gemacht hat, vermieden werden!

Karlmut Obermann

HEFT
20
August
1987

94

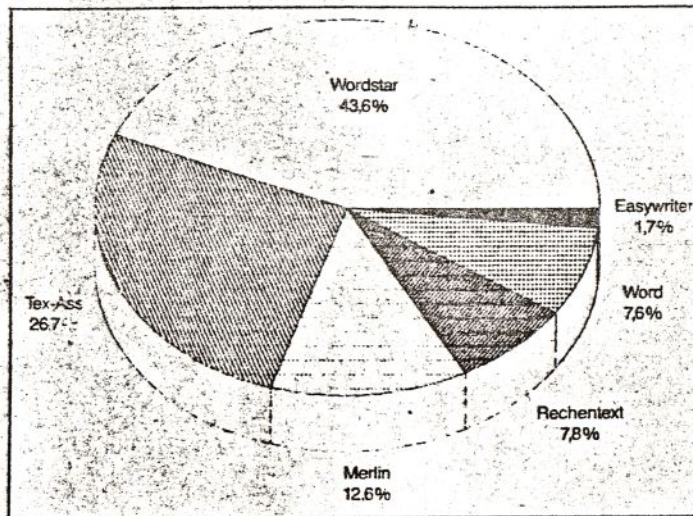
Im Schatten des Dinosauriers

Auf jedem vierten Personal Computer läuft hierzulande unter anderem auch ein Textverarbeitungssystem. Nach dem Marktforschungsinstitut IDC, das zuletzt 1985 eine Analyse dieses Software-Bereichs vornahm, existierten zu dieser Zeit entsprechende Programme im Gesamtwert von 204 Millionen Mark. Jedes Jahr kommen rund 40 000 Textverarbeitungspakete hinzu.

Die Marktanteile und Umsätze gelten als stabil, was nicht immer ein gutes Zeichen sein muß. Im Bereich Textverarbeitung bedeutet dies, daß von über 130 000 installierten Programmen in der Bundesrepublik knapp 70 Prozent auf ein Textverarbeitungssystem namens Wordstar entfallen. Böse Zungen behaupten, besonders die noch weit verbreiteten älteren Versionen des Programms besäßen echte Dinosaurier-Qualitäten: groß, schwerfällig, unflexibel und dabei alles beherrschend.

Ein paar Eigenschaften stehen tatsächlich in keinem Zusammenhang mit der enormen Marktdurchdringung des Systems. Nobody is perfect: Unbestreitbar haben alle auf dem

München (sg) — Trotz integrierter Programme und, seit einiger Zeit auch Desktop Publishing, sagen Marktforscher für den Bereich Textverarbeitungs-Software auch zukünftig stabile Verhältnisse voraus. Die Zeiten des stürmischen Wachstums wie zu Beginn der 80er Jahre sind jedoch vorbei — nur noch wenige PC-Anwender setzen ihr System ausschließlich als Ersatz für die Schreibmaschine ein.



Markt erhältlichen Programme ihre »Problemstellen«, doch deren Anteil an der Installationsbasis ist vergleichsweise gering — nach Wordstar kommt eine Weile nichts, dann tauchen Konkurrenzprodukte wie TexAss,

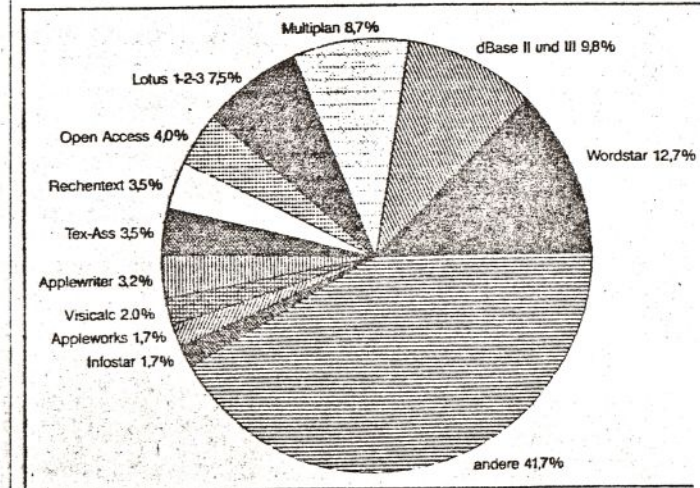
Word und Trend mit deutlich unter 10 Prozent auf.

Hersteller Micropro räumt denn auch freimütig ein, bei der Weiterentwicklung seines Paradeprodukts geschlafen zu haben. Mit der Vorstellung von

Wordstar-Erweiterungen und neuen Releases sei man aber nun aufgewacht. Zeit wurde es offenbar, denn die Nummer eins im Markt für Textverarbeitung steckte bis vor kurzem in den roten Zahlen. Dies hat sich laut Leon Williams, President der Micropro Corporation, im vergangenen Quartal geändert — bei einem Umsatz von 9,3 Millionen Dollar habe man einen Gewinn von 550 000 Dollar erwirtschaftet.

Die deutsche Niederlassung behauptete laut Marktforscher IDC 1985 mit Wordstar seine Spitzenposition. Von den insgesamt 38 346 ausgelieferten Textverarbeitungspaketen entfiel ein Anteil von 43,6 Prozent auf das GmbH-Produkt. An zweiter Stelle lag das Programm Tex-Ass der Firma Bongartz & Schmidt (26,7 Prozent), gefolgt von Brains Merlin, MIS Rechentext (7,8 Prozent), Microsofts Word (7,6 Prozent) sowie Easywriter von Computer Associates (1,7 Prozent).

Wie häufig der Personal Computer als Ersatz für die Schreibmaschine eingesetzt wird, verdeutlicht eine Studie der Intelligenz Electronics Europe.



Textverarbeitung hält die Spitze — Anteil am deutschen Softwaremarkt nach Stückzahlen
Quelle: Intelligent Electronic Euro

Der bundesdeutsche Gesamtmarkt für Software wurde Mitte '85 von einem Textprogramm angeführt — Wordstar hielt nach Stückzahlen einen Anteil von 12,7 Prozent. Erst auf den weiteren Plätzen folgten integrierte Programme wie dBase II und III (9,8 Prozent), Multiplan (8,7 Prozent), Lotus 1-2-3 (7,5 Prozent) sowie das Programm Open Access (4,0 Prozent).

Allerdings, und dies können den Anbietern von reinen Textverarbeitungspaketen mittelfristig Probleme bereiten, setzen immer weniger Anwender ihr PC ausschließlich als Textsystem ein. Nach dem stürmischen Wachstum zu Anfang der 80er Jahre wird sich dieser Bereich laut IDC als eigenständiges Segment für Basis-Applikation lediglich behaupten.

Auf zu neuen Ufern!

Liebe Clubfreunde!

Im letzten Info hat unser Freund Hans Martin Stephan einen Vorschlag zur Diskussion gestellt, der mir schon öfter von ausscheidenden Mitgliedern (die meisten "Aussteiger" sind "Aufsteiger" in die PC/Atari-Welt) angetragen wurde. Ich möchte hier kurz meinen Standpunkt dazu erläutern und hoffe, daß sich auch andere zu Wort melden!

Ich habe das Problem der Abwanderung von Tandy TRS80-Besitzern in die "Kompatibilität" schon lange beobachtet und mir immer wieder die Frage gestellt, wie man dieser Entwicklung entgegenreten könnte. Eine Möglichkeit die Attraktivität der TRS 80-Computer zu steigern ist meiner Meinung nach die vermehrte Unterstützung von CP/M im CLUB 80 und im Info.

Früher bildeten die TRS 80-Besitzer für Softwareanbieter einen interessanten Kundenkreis. Das sah man daran (und sieht man noch heute), daß sehr viele Sprachen auf den Tandy-Maschinen implementiert wurden. Nicht nur Tandy hat sich da, zusammen mit Microsoft, sehr engagiert, auch und vor allem kleinere Entwickler versuchten sich den TRS80-Markt zu erobern. Wäre z.B. der heute sehr verbreitete Turbo-Pascal-Compiler vor fünf oder sechs Jahren auf den Markt gekommen, er wäre ohne Zweifel auch unter TRS-/NewDOS zu haben, vielleicht wäre er sogar für dieses Betriebssystem entwickelt worden. Heute ist das anders geworden! Kaum eine(s) der neuen Sprachen(Programme) wird noch auf unsere Maschinen übertragen; wir stehen ziemlich im Abseits!!!

Durch die vermehrte Verbreitung von CP/M-Maschinen (Schneider CPC/Joyce, Commodore C128 usw.) ist dieses Betriebssystem, das sich schon in Gefahr befand, durch MS-DOS u.ä. verdrängt zu werden, wieder zum Leben erweckt worden. Wie groß die Aktivitäten in diesem Bereich sind, kann man sich kaum vorstellen! Nicht nur die sehr umfangreiche Public-Domain-Software, die erhältlich ist, auch die neuesten Sprachen (u.a. das schon erwähnte Turbo-Pascal, Modula, Prolog, Lisp) und viele kommerzielle Programme (WordStar, dBase, Multiplan usw. usw.) verleihen dem eher schlechten Betriebssystem CP/M große Attraktivität. Viele unserer TRS80-Maschinen können (Model 1 und III sowie Genie I und II nach Umbauten; Model 4/4p, Genie III, IIs, IIs sofort) auch unter CP/M betrieben werden und damit am Aufschwung dieses DOS teilhaben.

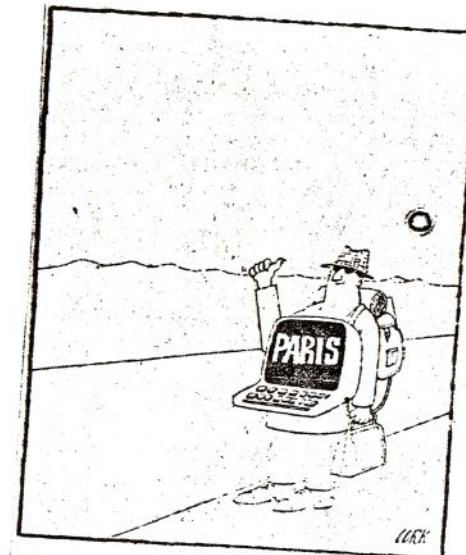
Darin sehe ich auch die Chance eines Brückenschlags zu denen, die sich trotz dieser neuen Aussichten nicht davon abhalten lassen, auf die Kompatiblenwelle aufzuspringen. Turbo-Pascal, "C", Prolog, dBase und vieles andere mehr läuft auf beiden Systemen, sowohl auf CP/M als auch auf MSDOS. Es ließen sich also durchaus Gemeinsamkeiten finden, trotz der, auf den ersten Blick unüberwindlichen Kluft!

Ich (und viele weitere Clubmitglieder) arbeite schon einige Zeit mit CP/M und werde dieses Thema in nächster Zeit durch entsprechende Veröffentlichungen im Info mehr unterstützen. Natürlich hoffe ich dabei auf Hilfe der anderen CP/M-User. Auch bin ich jederzeit dafür, im Info eine MS-DOS-Ecke einzurichten und zu unterstützen. Diese Aktivitäten dürfen aber nicht dazu führen, daß sich der CLUB 80 weg von einem "Club der Tandy-Freunde" und hin zu einer CP/M-UserGroup oder einem MS-DOS-Verein entwickelt. Vielmehr muß es so sein, daß die neuen Zweige zu zusätzlichen Standbeinen des CLUB 80 werden!

Zu guter Letzt möchte ich noch ein rein organisatorisches Problem in Punkto MS-DOS-Ecke ansprechen. Selbstverständlich wird schon jetzt kein Artikel abgewiesen, weil er sich mit diesem Thema beschäftigt. Wenn also jemand über MS-DOS, Turbo-Pascal oder ähnliches schreiben will, kann er das jederzeit tun! Die Frage ist nur die, ob es für eine solche Einrichtung auch Leute gibt, die bereit sind etwas zu schreiben. Gelesen wird es auf alle Fälle!

Damit genug zu diesem Thema! Ich hoffe auf eine angeregte Diskussion, Euer

Ralfaut Obermann



Längst nicht alle Einsatzbereiche für Personalcomputer sind hinreichend bekannt

HEFT
20
August
1987

98

Seit ich mein GENIE I 1983 kaufte, ist der Umgang mit dem Computer zu meinem gewohnten Arbeitsmittel geworden. Aus meinem GENIE habe ich zuletzt auch wohl so viel wie möglich herausgeholt: 3,5 MHz, verbessertes DOS etc.

Dennoch schreibe ich diesen Text jetzt auf meinem neuen PLANTRON PC XT mit 20 MByte Festplatte, den ich vor kurzem für 3.500,- DM kaufte. Wie ich aus verschiedenen Texten und Gesprächen entnehme, denkt auch von Euch der ein oder andere in dieser Richtung nach. Deshalb möchte ich etwas zum "Informationshaushalt" beitragen:

Insgesamt ist ein IBM-Kompatibler schon als Verbesserung zum GENIE I und II anzusehen; dennoch sind die Unterschiede gerade im Vergleich zum IIs und IIIs beim PC nicht so wesentlich. Wer hier zuschlagen will, sollte also nicht überzogene Erwartungen hegen. Die folgenden Erfahrungen habe ich bei mir gemacht:

Geschwindigkeit

Mein PC arbeitet ursprünglich mit dem 8088-2 bei wahlweise 4,7 und 8 MHz. Diese "Geschwindigkeit" wird allerdings mehr oder weniger durch ein ineffizientes MS-DOS müheelos vernichtet. Ich habe deshalb sogleich den Prozessor durch den NEC V20, der Z80 kompatibel ist (!), ersetzt. Dies läßt sich leicht durchführen und damit läuft meine Mühle ca. 3 mal schneller als der Ursprungs-PC. Dies reicht für die meisten Anwendungen aus, allerdings ist dies wenig im Vergleich zu den 5,4 (7.2) des AT mit 6 (8) MHz oder ca. 18 fachen Geschwindigkeit eines 80386-Rechners. Ein Vergleich: Eine BASIC-Anwendung auf dem GENIE benötigte für 100.000 Durchläufe rund 3 Minuten; der PC brauchte dafür 8 in BASIC und sogar - wär hätte das gedacht - 11 Minuten in "C".

Erweiterungen

Im Hinblick auf den Systemaufbau mit vielen Erweiterungssteckplätzen wäre diese Mühle allerdings ideal für einen Freak. Wenn man was basteln will, geht dies hier wesentlich einfacher, als beim GENIE. Ich habe beim GENIE z.B. nach dem Einbau der seriellen Schnittstelle, Speed up etc. eine ungeheure Menge von Drähten legen müssen. Viel mehr ist kaum mehr zu

überschauen.

Der PC kann dagegen leicht mit Steckkarten ausgebaut werden, z.B. existieren 2 MB RAM Extensionskarten, Grafikkarten und Schnittstellen. Der RAM kann leicht intern bis auf 1 MByte ausgebaut werden; diese werden vom Prozessor noch direkt adressiert z.B. als RAM-Disk. Für mathematische Berechnungen habe ich einen 8087 Co-Prozessor installiert. Leider benutzen nur wenige Standardprogramme die 20-30fache Rechengeschwindigkeit und höhere Genauigkeit wirklich aus.

Zuverlässigkeit, Handhabung

In dieser Hinsicht ist gerade der PLANTRON vorbildlich. Die AT-Tastatur ist sehr Bedienungsfreundlich, getrennte Cursor-Tasten, ergonomische Tastenform, 12 Funktionstasten, Leuchtanzeigen für GROSSCHRIFT, Num Lock, Power. Die Disketten sind nahezu unverwundlich. Selbst die billigen Marken laufen ohne Schwierigkeiten. Der hochauflösende bernsteinfarbene Monitor ist absolut flimmerfrei und endlich können auch 80 Zeichen pro Zeile dargestellt werden, was bei uns ja nur bei den modernen GENIE's funktioniert. Die eingebaute Festplatte gehört zwar nur zu den Mittelschnellen, dennoch ist es ein großer Geschwindigkeitsunterschied gegenüber den Diskettenlaufwerken. Mehr Leistung ist auch erforderlich, da schon dieses Textprogramm über 200 kB benötigt; Overlays noch nicht mitgerechnet. Probleme, wie Abstürze durch Hardware-Fehler, wie sie mein GENIE im Sommer manchmal hatte, kommen - wohl auch Dank eingebautem Gebläse - nicht mehr vor.

Betriebssystem

MS-DOS ist mehr oder minder ein "aufgemotztes" CP/M-80 und läßt in vielen Eigenschaften selbst gegenüber dem recht guten NewDOS 80 Lücken. Z.B. gibt es kein LIB; kein BOOT, kein CLOCK (UHR). Wenn man beim formatieren eine Taste festhält (was möglich ist) meckert das DOS und verabschiedet sich schließlich durch einen HALT, d.h. Ausschalten wird ohne RESET-Knopf erforderlich. Positiv an der bei mir installierten Version 3.2 ist allerdings, daß viele Ergänzungen bereits implementiert sind und manches besser läuft. Z.B. gehört die

RAM-Disk zum Standard. Auf der Festplatte können viele Unterverzeichnisse verwaltet werden.

Zum Betriebssystem werden auch eine Reihe von ergänzenden Programmen angeboten, z.B. zum sortieren des Inhaltsverzeichnisses, zum Zurückholen von versehentlich gelöschten Dateien usw. Diese Programme müssen aber zusätzlich teuer erworben werden.

Neben MS-DOS können auch andere Betriebssysteme verwendet werden, z.B. CP/M-86 und Concurrent PC-DOS.

Anwendungs-Software

Aufgrund des größeren Arbeitsspeichers und des vorwiegend kommerziellen Einsatzes bieten Softwarehäuser komplexere Programme an, als bei uns üblich und möglich sind. Allerdings sind die Unterschiede nicht überall groß und finden sich für viele Eigenschaften Vorläufer aus der CP/M und NewDOS-Welt.

Wegen WordStar kann man dabei ruhig bei CP/M bleiben, es läuft nur unwesentlich benutzerfreundlicher. Ich kann mir vorstellen, das im GENIE IIs sogar mehr Power dafür bereit steht.

WordPerfect ist ein neues amerikanisches Textprogramm, mit einer Reihe von Eigenschaften, die ich schon länger suchte: erweiterbares Wörterbuch, über 200 Druckertreiber, gleichzeitige Bearbeitung von 2 Texten, beliebige Textlänge, also auch z.B. 200 Seiten im Arbeitsspeicher, automatische Sicherungskopien z.B. nach 5 Minuten, ohne daß der Benutzer eine Taste drücken muß (!) (welcher GENIEText-User würde sich dies nicht wünschen).

Weitere Eigenschaften: Rechnen im Text, Spaltenverarbeitung, Macros, PS-Schrift mit Randausgleich.

Wie aber z.B. Arnulf mit seinen vielen Verbesserungen gezeigt hat, kann vieles davon auch auf dem GENIE entwickelt werden.

Auf dem Markt werden neben Grafik-Programmen auch sogenannte integrierte Programme angeboten (OPEN ACCESS II, Framework II, SYMPHONY). Regelmäßig werden dabei Tabellenkalkulation, Textverarbeitung und Datenbank unter einer Benutzeroberfläche zusammengeführt, was insbesondere jeden freuen wird, der z.B. Serienbriefe drucken möchte oder Tabellen und Grafiken in einen Text integrieren möchte. Die einzelnen Programmfunktionen sind dabei mittlerweile recht brauchbar

fortentwickelt, dennoch sind spezielle Programme für jede Einzelaufgabe i.d.R. mächtiger.

Transfer

Dateien können mit speziellen Programmen von NewDOS, G-DOS und CP/M-80 nach MS-DOS transferiert werden. Insofern spart man die nochmalige Erfassung.

Programmiersprachen

Unter MS-DOS werden inzwischen alle gängigen Programmiersprachen angeboten und auch noch laufend fortentwickelt.

Zusammenfassung

Mit der Maschine bin ich sehr zufrieden. Dabei ist sicherlich die berufliche Nutzung von wesentlicher Bedeutung: Ich schreibe viele Stunden wöchentlich daran. Im Vergleich zu allen übrigen Maschinen auf dem Markt hat mir insbesondere die sehr solide Verarbeitung und der günstige Preis gefallen. Die Basisversion mit 1 Laufwerk kostet rd. 1300,- DM wesentlich weniger, als ich für mein GENIE damals bezahlt habe! Wer allerdings mit dem Geld nicht so knapp ist und eher mit kommerziellen Programmen umgehen möchte, sollte unbedingt den AT vorziehen oder besser noch den immer billiger werdenden 386.

Dagegen kann ich nicht zu Schneider PC und verschiedenen Billigst-PC raten; diese Geräte sind entweder nicht für Dauerbelastung ausgelegt oder haben sonstige Nachteile, z.B. schlechte Tastatur, geringe Erweiterungsmöglichkeiten usw.

Wer sich für meine sehr preisgünstige "Quelle" interessiert, dem stehe ich gerne zur Verfügung.

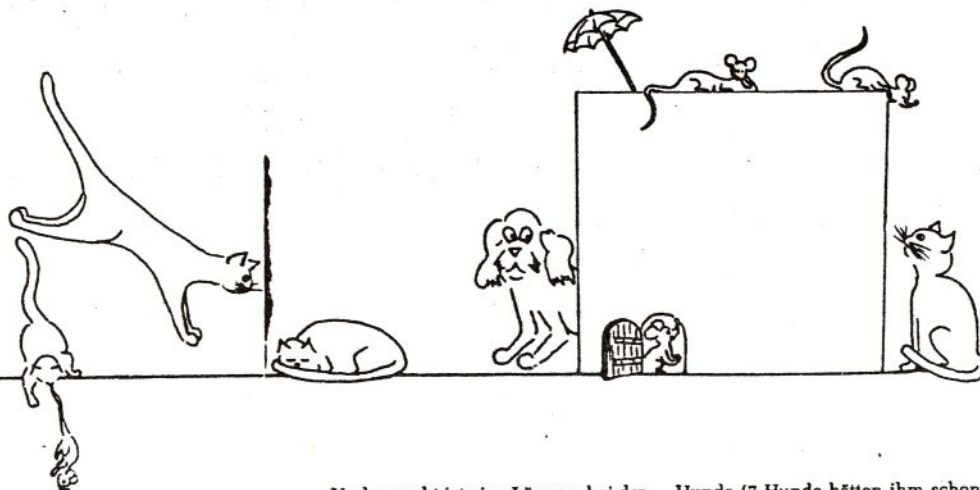
Zum Club

Ich bin an einem Erfahrungsaustausch zum PC interessiert, meine aber auch, daß in unserem Tollen Info hier nicht kostbarer Platz verschwendet werden sollte. Über Kontakte zu diesem Thema freue ich mich allerdings.

Paul-Jürgen Schmitz
5. Juli 1987

Mathematik für Feinschmecker

Heute: Der tüchtige Tierhändler
von Wolfgang J. Weber



Im folgenden werden wir drei Hauptpersonen kennenlernen. Nomen est omen – manchmal jedenfalls. Da ist zunächst Herr Witzig, ein schneller Kopf, der allerdings gelegentlich gehörig neben das Ziel schießt. Herr Witzig programmiert gerne, auch in solchen Fällen, in denen es andere Wege zum Erfolg gäbe. Zweitens ist da Herr Denkmann, der ziemlich genau das Gegenteil von Herrn Witzig verkörpert: bedächtig in Wort und Tat, möchte er immer erst einen Vorgang vollständig verstanden haben, bevor er etwas unternimmt. Nur böswillige Beobachter interpretieren dies als Mangel an Initiative. Als Dritten treffen wir auf Herrn Rechenberg. Ihm macht die Mathematik großen Spaß, und er stellt seinen Mitmenschen gerne Denksportaufgaben mathematischer Natur. Am liebsten sind ihm übrigens solche Aufgaben, für die ihm schon gleich zu Beginn die Lösung mitgeteilt wurde, daher ist es auch nicht verwunderlich, daß Herr Rechenberg sich auf diese Weise nicht immer Freunde schafft. Hingegen dürfte vielen Lesern von PASCAL-International sein Hobby nicht ganz fremd sein, denn das Programmieren und das Problemlösen sind ja eng miteinander verwandt. Hören wir, was er auf Lager hat.

An diesem Abend sitzt Herr Rechenberg als Gastgeber gemütlich mit seinen Freunden Denkmann und Witzig

bei einem Glas Wein beisammen. Man hat gerade seine jeweiligen Meinungen über die Politik, das Wetter, Virusprogramme und andere Landplagen dargestellt und zwanglos weitgehende Einigkeit erzielt. Das Gespräch droht zu verstummen, doch da hätte man Rechenberg unterschätzt.

»Stellen Sie sich mal einen Tierhändler vor, der an einem Tag 100 Tiere verkauft;«

»Aber lieber Rechenberg, es wurde gerade gemütlich, Ihr Wein ist auch ganz hervorragend, und jetzt wollen Sie uns vermutlich wieder eine erbauliche Geschichte über die unerschöpflichen Geheimnisse der Arithmetik erzählen«, protestiert Witzig.

»Ja, und zwar wird es Sie vielleicht sogar interessieren. Also, unser Tierhändler verkauft 100 Tiere, und er verdient dabei genau 100 Mark.«

»Und jetzt sollen wir sagen, wie alt er ist?« blödelte Denkmann.

Rechenberg: »Meine Herren, wo bleibt das Niveau? Nein, er verkauft 100...«

»Gut, gut, das wissen wir«, unterbricht ihn Witzig, der sich damit abgefunden hat, daß er wieder einen Beweis seiner stets wachen Intelligenz erbringen muß, um sein Glas Wein zu verdienen. Es gibt eben nichts umsonst im Leben. »Was hat er verkauft und was hat er im einzelnen eingekommen?«

»Verkauft hat der Tierhändler Hunde, Katzen und Mäuse. Er hat an jedem Hund 15 DM, an jeder Katze 1 DM und an jeder Maus 25 Pfennige verdient.«

»Lausige Gewinnspannen, wenn Sie mich fragen. Der wäre besser Apotheker geworden. Aber was sollen wir jetzt rauskriegen?« fragt Witzig.

»Vermutlich, wieviele Tiere er von jeder Sorte verkauft hat«, meldet sich Denkmann.

»Stimmt, meine Herren, das ist die Aufgabe, der Sie sich doch sicher gewachsen zeigen werden?« Der Tonfall Rechenbergs beschwört böse Ahnungen in den beiden anderen herauf, daß sich dieser Abend noch etwas in die Länge ziehen könnte. Vielleicht ist die Aufgabe sogar unlösbar?

»Drei unbekannte Größen, aber nur zwei Gleichungen. das ist nicht fair«, protestiert Witzig.

»Doch. Bedenken Sie, daß wir nicht irgendeine Lösung finden sollen, sondern eine ganzzahlige«, belehrt ihn Denkmann. »Also wir haben, wenn ich zusammenfassen darf, zwei Gleichungen.

Gleichung 1 lautet:
Hunde + Katzen + Mäuse = 100

Gleichung 2 lautet:
 $15 \times \text{Hunde} + 1 \times \text{Katzen} + 0,25 \times \text{Mäuse} = 100$

Und gesucht ist eine Lösung, bei der keine Drittelhunde oder Viertekatzen auftreten, und natürlich auch keine negativen Zahlen.«

Alle schweigen; Rechenberg blickt gespannt auf seine Freunde, die grübelnd mit ihren Kugelschreibern hantieren und Zettel bekrakeln. »Darf ich Ihnen noch etwas Schreibpapier anbieten? Bitte bedienen Sie sich doch«, will Rechenberg gerade sagen, doch da schaut Witzig triumphierend in die Runde.

»Ich hab's. Er hat Null Hunde, Null Mäuse und 100 Katzen verkauft;«

»Ach, da enttäuschen Sie mich aber, lieber Witzig.« Rechenberg ist echt überrascht, denn an diese triviale Lösung hatte er selbst noch nicht gedacht. »Streng genommen ist auch das eine Lösung, doch wurde ja gesagt, daß der Tierhändler tatsächlich Hunde, Katzen und Mäuse verkauft hat, also von jeder Sorte mindestens ein Exemplar.«

Witzig gibt sich geschlagen, er hatte zu früh triumphiert. »Ja, dann darf ich Sie, lieber Rechenberg, bitten, mir Ihren Computer zur Verfügung zu stellen. Ich werde jetzt alle möglichen Fälle durchgehen lassen und testen, bei welchen die gegebenen Bedingungen zutreffen.«

»Wird das nicht sehr lange dauern?« gibt Denkmann zu denken.

»Das glaube ich nicht, denn der Händler kann nur zwischen 1 und 6

Hunde (7 Hunde hätten ihm schon mehr als 100 DM eingebracht), zwischen 1 und 100 Katzen und zwischen 1 und 400 Mäuse verkauft haben. Das macht $6 \times 100 \times 400$, also 240000 Fälle, für einen Computer ist das keine problematische Sache. Also wo ist der Hauptschalter von diesem Kasten?« Denkmann: »Viel Spaß Herr Kollege, ich finde, daß unser Prozessor im Kopf zur Lösung der gestellten Aufgabe ausreichen sollte. Ich brauche nur Papier und Bleistift.«

An dieser Stelle wollen wir das gelehrte Kollegium verlassen und uns unsere eigenen Gedanken machen. Wie wird Witzig das Problem programmieren und wie wird Denkmann an die Sache herangehen?

Als Tip zum Programmieren: Die Anzahl der verkauften Mäuse muß durch 4 teilbar sein, und sie darf natürlich 100 nicht übersteigen (weil nur 100 Tiere insgesamt verkauft werden). Damit sind wesentlich weniger Fälle zu untersuchen als Witzig oben vermutete. Aber auch Denkmanns Ansatz ist sinnvoll: die Aufgabe kann durch schlichtes Nachdenken gelöst werden, bei dem als Hardware Papier und Bleistift ausreichen.

HEFT
20
August
1987

102

Lösung auf Seite 89

Lösung von Seite 86

Die Aufgabenstellung lautete: Ein Tierhändler verkauft 100 Tiere und realisiert dabei einen Gewinn von DM 100. Bei den verkauften Tieren handelt es sich um Hunde (Gewinn je 15 DM), Katzen (Gewinn je 1 DM) und Mäuse (Gewinn je 1/4 DM). Wieviele Tiere von jeder Art wurden verkauft?

Zur Lösung sind zwei Zugänge möglich. Erstens Witzigs Ansatz: Man kann alle möglichen Kombinationen aufzählen und jeweils überprüfen, ob die beiden Bedingungen erfüllt sind. Das wird im folgenden Pascal-Programm getan (s. Listing).

Zweitens Denkmanns Lösung: Man kann die Gleichungen genauer betrachten und noch mehr Schlüsse aus ihnen ziehen. Die Gleichungen sind:

Gleichung (1):

Hunde + Katzen + Mäuse = 100

Gleichung (2):

$15 \times \text{Hunde} + 1 \times \text{Katzen} + 0,25 \times \text{Mäuse} = 100$

Es wurde bereits erwähnt, daß die Anzahl der verkauften Mäuse durch 4 teilbar sein muß, denn in Gleichung (2) treten je überall ganze Zahlen auf, und deshalb muß auch der Summand $0,25 \times \text{Mäuse}$ ganzzahlig sein. Die Gleichung (2) kann mit der Gleichung (1) kombiniert werden:

$\text{Hunde} + 1 \times \text{Katzen} + 0,25 \times \text{Mäuse} = \text{Hunde} + \text{Katzen} + \text{Mäuse}$

Die Katzen fallen algebraisch nicht mehr ins Gewicht, damit erhalten wir eine Gleichung mit nur noch zwei Unbekannten

$15 \times \text{Hunde} + 0,25 \times \text{Mäuse} = \text{Hunde} + \text{Mäuse}$

oder nochmals umgeformt

$14 \times \text{Hunde} = 3/4 \times \text{Mäuse}$

was besser in der folgenden Form geschrieben wird:

$56 \times \text{Hunde} = 3 \times \text{Mäuse}$

Nun betrachten wir die Primteiler der linken und der rechten Seite: Die Anzahl der Hunde muß durch 3 teilbar sein, weil 56 nicht durch 3 teilbar ist. Also kommt als Wert für die Unbekannte Hunde nur noch 3 oder 6 in Frage. (0 scheidet aus, weil ja mindestens ein Hund verkauft wurde.) 6 kommt als Lösung nicht in Frage, weil dies mit 112 verkauften Mäusen

verknüpft wäre, was laut Aufgabenstellung nicht der Fall sein kann. Aus Hunde = 3 folgt Mäuse = 56. Weil insgesamt 100 Tiere im Spiel sind, wurden 100 - 56 - 3 = 41 Katzen verkauft. Wir machen die Probe:

Gleichung 1:
 $3 \text{ Hunde} + 41 \text{ Katzen} + 56 \text{ Mäuse} = 100$

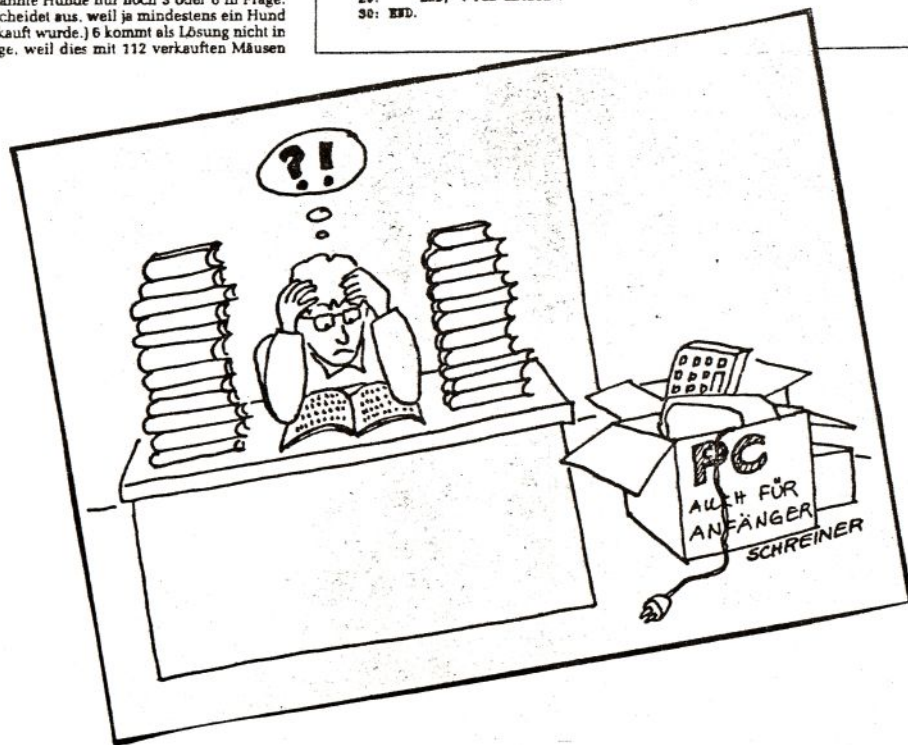
Gleichung 2:
 $15 \times 3 + 1 \times 41 + 0,25 \times 56 = 45 + 41 + 14 = 100$

Damit ist die Aufgabe gelöst. Wir wollen uns der Bewertung enthalten, welcher der zwei vorgestellten Zugänge besser ist:

wenn man einen Computer zur Verfügung hat, dann darf man ihn auch benutzen, wie Herr Witzig das getan hat. Andererseits ist es aber auch nicht verboten, den eigenen biologischen Computer zu benutzen, sofern man ihn bedienen kann, wie Herr Denkmann es uns vormacht. Zu einem sollte uns unser Kopf aber ganz bestimmt zu schade sein: alle möglichen Fälle einfach stumpf und blöde der Reihe nach durchzugehen und die Aufgabe durch direktes Probieren zu lösen. Wozu gibt es schließlich Computer?

(Wolfgang J. Weber)

```
1: PROGRAM tierhandel;
2:
3: (* 100 Tiere werden verkauft, der Gewinn beträgt insgesamt 100 DM.
4: Gewinnspannen: pro Hund: 15 DM, pro Katze 1 DM, pro Maus 0,25 DM.
5: Wie viele Tiere von jeder Art wurden verkauft?
6: Programmiersprache: Turbo-Pascal, Autor: W.J. Weber, Dezember 1986 *)
7:
8: VAR hunde, Katzen, maeuse: INTEGER;
9:
10: BEGIN
11:   ClrScr;
12:   GotoXY(10,10); Writeln('Der tsechtige Tierhaendler');
13:   GotoXY(10,11); Writeln('-----');
14:   GotoXY(1,15);
15:   FOR hunde := 1 TO 6 DO
16:     FOR Katzen := 1 TO 100 - hunde DO
17:       BEGIN
18:         maeuse := 4;
19:         REPEAT
20:           IF (hund + Katzen + maeuse = 100) AND
21:              (15 * hund + Katzen + (maeuse DIV 4) = 100) THEN
22:             BEGIN
23:               Write('Lösung gefunden: Es wurden');
24:               Write(hunde:3, ' Hunde, ', Katzen:3, ' Katzen und');
25:               Writeln(maeuse:3, ' Mäuse verkauft. ');
26:             END;
27:             maeuse := maeuse + 4;
28:           UNTIL Katzen + maeuse >= 100;
29:         END; (* FOR Katzen *)
30:       END.
```



Mathematik für Feinschmecker

Heute:
Wie man durch Würfeln theoretisch vielleicht reich, praktisch aber meistens arm wird

von Wolfgang J. Weber

Wieder treffen wir auf die Herren Rechenberg, Denkmann und Witzig, die regelmäßigen Lesern von PASCAL keine Unbekannten mehr sind. Heute diskutieren sie über ein Problem der Wahrscheinlichkeitsrechnung, bei der ja bekanntlich selbst Fachleute gelegentlich Trugschlüssen erliegen.

Rechenberg hat seinen letzten Urlaub in Südamerika verbracht. Dort lernte er in einem Spielcasino ein verblüffend einfaches Gewinnspiel kennen, bei dem nur gewürfelt wird. Spielregel Nr 1 ist: der Spieler darf viermal würfeln. Spielregel Nr 2 ist: wenn dabei (mindestens) eine Sechse erwürfelt wird, dann hat der Spieler gewonnen. Die Rate der Gewinnauszahlung ist 5:3, das heißt, wenn er z.B. umgerechnet DM 3 einsetzt und gewinnt, erhält er DM 5 = DM $3 \times (5/3)$ ausgezahlt. Bei einem Einsatz von DM 60 ergibt sich im Gewinnfall eine Auszahlung von DM 100, usw.

Unser Held Rechenberg hatte hier sofort seine Chance gesehen. »Mag sein,« dachte er sich, »daß diese Latinos noch nichts von Wahrscheinlichkeitstheorie gehört haben, jedenfalls haben sie sich hier offensichtlich verkalkuliert. Jedes Kind weiß, daß die Wahrscheinlichkeit, bei einem Wurf eine Sechse zu werfen 1/6 ist. Bei vier Versuchen steigt die Wahrscheinlichkeit logischerweise auf den vierfachen Wert, also auf 4/6 oder 2/3. Das heißt,« folgert Rechenberg nun messerscharf, »daß man durchschnittlich in zwei von drei Spielen einen Gewinn erwarten darf.« Auf lange Sicht könne er daher nur gewinnen, glaubte Rechenberg. Zu Beginn wollte er mit einem festen Einsatz von umgerechnet jeweils DM 60 spielen. Bei drei Spielen würde er durchschnittlich zwei Gewinne zu je DM 100 einstecken und dabei drei Einsätze zu je DM 60 verlieren: das macht einen erwarteten Gewinn von DM 200 - DM 180 = DM 20. Pro Spiel würde er dabei also im Durchschnitt DM 6,66 gewinnen, was zwar nicht dramatisch viel aber immerhin mehr als 10% des Einsatzes wäre.

So weit die Theorie. Leider sah die Praxis nicht ganz so golden aus. Eher trübe, um ganz offen zu sprechen. Denn Rechenbergs Rechnung ging nicht auf - er verlor mehr und mehr Geld, und die Wende zum stetigen Gewinn wollte sich, entgegen der Theorie, nicht einstellen. Anfangs dachte er noch, es handle sich um zufallsbedingte Schwankungen, die sich über kurz oder lang mehr als ausgleichen würden. Er war sich doch so sicher, daß er das Spiel richtig analysiert hatte! Nachdem sich seine Verluste aber auf weit über hundert DM beliefen, verlor er dann auch

noch etwas die Nerven und erhöhte seinen Einsatz unkontrolliert. Das Ergebnis des Casinoabends war schließlich: er hatte einiges Geld verloren, seine Nerven waren ruiniert... und seine Freunde hatten etwas zu lachen. Rechenberg konnte sich nämlich nicht enthalten, nach seiner Rückkehr, bei geeigneter Gelegenheit, seinen Freunden Witzig und Denkmann von diesem Mißgeschick zu berichten.

»Das überrascht mich gar nicht, lieber Rechenberg. Diese Casinoleute sind doch Profis. Und Sie, seien Sie mir nicht böse, sind ein Dilettant, jedenfalls, was das Glücksspiel angeht. Es ist doch klar, wie ein solcher Zweikampf ausgehen muß!« Vor Witzigs guter Laune und seinen klugen Bemerkungen gibt es kein Entrinnen.

»Und wo, bitte schön, liegt der Fehler in meiner Argumentation?« Rechenberg findet seinen Freund heute gar nicht witzig.

»Ein kleiner Rechenfehler, mehr nicht,« murmelt Denkmann. »Sie sind da aber in guter Gesellschaft, in bester adeliger Gesellschaft sogar. Haben Sie schon einmal vom Chevalier de Méré gehört?«

»Franzose, aha. Baut er vielleicht Wein an? Chateau de Méré«

»Nicht ganz. Der Herr ist schon länger tot, es sind ziemlich genau 300 Jahre. Dieser Mr.ing in die Geschichte der Mathematik dadurch ein, daß er Blaise Pascal eine interessante Frage stellte. Auch Denkmann befindet sich offenbar in guter Laune. »Mir wäre Rechenbergs Mißgeschick sicher erspart geblieben,« denkt er sich. »Manchmal ist selbst Bildung etwas wert.«

»Nun spannen Sie uns doch nicht derart auf die Folter,« drängt ihn Witzig. »Um ehrlich zu sein, mir

leuchtet die Argumentation unseres Freundes Denkmann ganz und gar ein. Nur die allgemeine Lebenserfahrung hätte mich eventuell von einem gleichartigen Verlust bewahrt.«

»Meine Herren, die Lösung sollten Sie selbst finden.« Denkmann ist unerbittlich. »Es geht sogar auf zwei Arten. Als einen Weg könnten Sie einfach am Computer simulieren, daß jemand über viele Spiele hinweg je viermal würfelt. Dabei zählen Sie, wie oft mindestens eine Sechse auftritt. Wenn Sie ihr Ergebnis haben, dann erzähle ich Ihnen, wo der Pferdefuß in Rechenbergs Argumentation liegt. Denn auch auf theoretischem Weg, ganz ohne Computer, kann man die richtige Lösung herleiten.«

Wir verlassen an dieser Stelle die Gesprächsrunde, denn der Leser wird vermutlich selbst gerne über die aufgeworfenen Fragen nachdenken wollen. Bei der Computersimulation braucht man einen Zufallszahlengenerator, der Werte zwischen 1 und 6 liefert. Hat man z.B. in Pascal eine Funktion RANDOM, die zufällige Real-Zahlen zwischen 0 und 1 erzeugt, dann ergibt der Ausdruck $(\text{TRUNC}(6 \times \text{RANDOM})) + 1$ die gewünschten Werte zwischen 1 und 6. In Turbo Pascal können derartige Werte einfach durch den Funktionsaufruf $\text{RANDOM}(6) + 1$ gewonnen werden. Für die theoretische Behandlung findet man Hinweise in Lehrbüchern der Wahrscheinlichkeitslehre unter dem Stichwort »bedingte Wahrscheinlichkeiten«, und häufig wird dort auch die Frage des Chevalier de Méré explizit behandelt.

Die Lösungen zu allen angeschnittenen Fragen finden sich im Anschluß.

Lösung:

Die Aufgabenstellung lautete: Jemand würfelt viermal. Wie groß ist die Wahrscheinlichkeit, daß dabei mindestens eine Sechse auftritt? Es wurde vermutet, daß diese Wahrscheinlichkeit 2/3 betrage, das ist aber nicht richtig. Im folgenden Pascal-Programm wird eine variable Anzahl von Würfelspielen simuliert und über die Zahl der gewonnenen und verlorenen Spiele genau Buch geführt. Am Ende werden die simulierten Werte ausgegeben und zusätzlich zur Illustration der Gewinnbetrag, der sich bei einem konstanten Einsatz von DM 60, ergeben hätte.

PROGRAM würfeln;

Wiederholtes Ausführen des Programms zeigt, daß zwar im Durchschnitt mehr Spiele gewonnen als verloren werden, daß die Wahrscheinlichkeit zu gewinnen aber nur geringfügig größer ist als 1/2. Ein typisches Ergebnis eines Programmlaufs bei z.B. 10.000 Spielen ist:

5221 Spiele gewonnen,
4779 Spiele verloren.

Die Annahme, daß 2/3 aller Spiele im Durchschnitt gewonnen werden, ist somit unhaltbar geworden.

Neben der praktischen Simulation dieses Problems durch ein Computerprogramm gibt es auch einen theoretischen Zugang. In der Argumentation des Herrn Rechenberg steckt ein Denkfehler. Die Wahrscheinlichkeit, in vier Würfeln eine Sechse zu werfen, ist nicht $1/6 + 1/6 + 1/6 + 1/6$. Eine solche einfache Addition von Wahrscheinlichkeiten ist hier nicht erlaubt, es widerspricht der Logik der Wahrscheinlichkeitslehre. (Das ist einfach einzusehen: angenommen, man darf siebenmal werfen. Wäre dann die Wahrscheinlichkeit 7/6? Nein, denn Wahrscheinlichkeiten, die größer als 1 sind, gibt es nicht!)

Rechenbergs Analyse hätte hier das Kalkül für sogenannte bedingte Wahrscheinlichkeiten erfordert. Wir gehen die korrekte Argumentation Schritt für Schritt durch.

Erstens: Wie groß ist die Wahrscheinlichkeit für Erfolg sofort im 1. Wurf? Antwort 1/6, klar.

Weiter: Wie groß ist die Wahrscheinlichkeit für einen Erfolg im 2. Wurf? Antwort $1/6 \cdot 5/6$, denn Erfolg beim zweiten Anlauf setzt Mißerfolg im ersten Wurf voraus. Dessen Wahrscheinlichkeit ist $5/6$, und beide Werte sind zu multiplizieren. Analog ergibt sich die Wahrscheinlichkeit für einen Erfolg im 3. Wurf zu $1/6 \cdot 5/6 \cdot 5/6$.

Die Wahrscheinlichkeit für einen Erfolg im 4. Wurf ist $1/6 \cdot 5/6 \cdot 5/6 \cdot 5/6$.

Zusammen ergeben die errechneten Zahlen den Wert $671/1296 = 0,51775$.

Dies ist die Gewinnwahrscheinlichkeit. Sie liegt näher bei 1/2 als bei 2/3, wie es ja auch die Computersimulation gezeigt hat. Es gibt übrigens auch einen vergleichsweise einfacheren Weg zur richtigen Lösung. Man fragt: »Wie wahrscheinlich ist es, in vier Würfeln KEINE Sechse zu werfen?« (Mathematiker sagen, man berechnet das Gegenereignis.) Die Antwort ist, daß man viermal Pech haben muß, was jeweils mit der Wahrscheinlichkeit $5/6$ der Fall ist. Die gefragte Wahrschein-

keit für das Gegenereignis ist also $5/6 \cdot 5/6 \cdot 5/6 \cdot 5/6 = 625/1296$.

Die eigentlich interessierende Wahrscheinlichkeit ist dann die Differenz zu 1. Man erhält in diesem Fall natürlich dasselbe Ergebnis wie oben: 0,51775.

Abschließend noch ein Ausflug ins Historische: Der erwähnte Chevalier de Méré stellte Blaise Pascal folgende Frage: Ist es wahrscheinlicher bei vier Würfeln mit einem fairen Würfel (mindestens) eine Sechse zu werfen oder bei 24 Würfeln mit je zwei Würfeln (mindestens) eine »Doppelsechse«, d.h. einen Sechserpasch.

Den ersten Teil der Frage haben wir schon beantwortet. Pascal fand (natürlich!) die richtige Antwort: die erste Wahrscheinlichkeit ist größer als die zweite für einen Sechserpasch, die etwas weniger als 1/2 beträgt.

Wer will, kann heutzutage etwas tun, was Blaise Pascal nicht konnte: auch das zweite Problem am Computer simulieren.

Herr Rechenberg hat das Wagnis einer Auslandsreise auf sich genommen und befindet sich zur Zeit im Vereinigten Königreich Ihrer Majestät. England besitzt ja einen eigenartigen Ruf: vieles geht dort gemüthlicher vonstatten als in Deutschland, dafür funktioniert aber auch nicht alles unbedingt auf Anhieb.

Dies erfährt auch gerade Rechenberg, der von seinem Telefonapparat im Hotelzimmer nach Hause telefonieren will. Die internationale Vorwahl der BRD ist 0049. Das klappt! Vorwahl Eschwege 5651 klappt auch noch. Doch die Teilnehmernummer 8702 geht nicht mehr.

Denkman gibt nicht auf und ruft seinen Freund Witzig in Frankfurt an, der die Nummer 69-81932 hat. Das funktioniert. Er versucht es noch mit einer anderen Nummer 22627 und hat wieder kein Glück. Was ist hier los?

Einige Minuten Nachdenken bringen ihn auf den Gedanken, daß der Telefonapparat defekt sein könnte. Bei den funktionierenden Ziffernkombinationen treten alle Ziffern auf, außer der 7. Die Fehler treten bei Ziffernkombinationen auf, die eine 7 enthalten. Da es sich um ein Tastentelefon handelt, vermutet Denkman messerscharf, daß die Taste 7 defekt sein könnte. Was tun? Beschwerde beim Manager, schließlich müßte der froh sein, daß ihm einer seiner Gäste Hinweise auf Mängel in seinem Hause gibt.

Der ist aber gar nicht froh, denn der bestellte Service läßt auf sich warten und Denkman könnte schließlich noch 90% der ihn interessierenden Anschlüsse erreichen, da es zehn Tasten seien und doch nur eine davon defekt. Die Tatsache, 90% der Nummern wählen zu können, ist kein Grund ein größeres Theater zu veranstalten und so räumt

Mathematik für Feinschmecker

Heute: Telefonieren mit Hindernissen
Von Wolfgang J. Weber

Denkman das Feld. Er legt sich schlafen und denkt noch ein wenig über die 90% nach. Da kommt ihm ein Zweifel ...

Auch den Lesern von PASCAL International sollten Zweifel gekommen sein. Wie viele sechsstellige, siebenstellige und achttellige Ziffernkombinationen kann man mit dem defekten Telefon noch erzeugen? Gefragt ist wieder nach einem Computerprogramm einerseits und einer theoretischen Analyse andererseits.

Lösung:
Die Aufgabenstellung lautete: Zur Verfügung stehen alle Ziffern außer der "7". Wie viele sechsstellige, siebenstellige und achttellige Ziffernkombi-

nationen kann man mit diesen neun Ziffern erzeugen? Es wurde vermutet, daß noch jeweils 90% aller Kombinationen möglich sind, das ist aber nicht richtig.

Im folgenden Pascal-Programm werden alle gültigen siebenstelligen Kombinationen durchgezählt, indem eine passende Zahl von FOR...TO-Schleifen ineinandergeschachtelt wird. Die Logik des Programms ist sehr einfach. Eine Anpassung an die Frage nach sechsstelligen bzw. achttstelligen Ziffernfolgen kann leicht vorgenommen werden.

Das Ausführen des Programms erfordert unangemessen viel Zeit. Hier liegen programmtechnisch noch einige Mög-

lichkeiten zur Verbesserung. Das Programm zeigt aber deutlich, daß nicht etwa 90% der siebenstelligen Folgen mit den funktionierenden 9 Zifferntasten erzeugt werden können, sondern noch nicht einmal die Hälfte, genau 47,82969%.

Von allen sechsstelligen Ziffernfolgen können noch 53,1441% und von den achttstelligen nur magere 43,046721% erzeugt werden.

Neben dem direkten Abzählen durch ein Computerprogramm gibt es auch eine theoretische Lösung. Rechenberg sollte hier das mathematische Teilgebiet der Kombinatorik benutzen. Wir wollen uns schrittweise das Prinzip verdeutlichen. Mit den zur Verfügung stehenden neun Ziffern können erzeugt werden:

9 einstellige Folgen (90% aller Folgen),

$9 \times 9 = 81$ zweistellige Folgen (81% aller Folgen),

$9 \times 9 \times 9 = 729$ dreistellige Folgen (72,9% aller Folgen),

$9 \times 9 \times 9 \times 9 \times 9 \times 9 \times 9$ siebenstellige Folgen.

Insgesamt gibt es 10000000 (10⁷ oder 1E7) verschiedene siebenstellige Ziffernfolgen. Das Produkt 9999999 hat den Wert 4782969, was etwa 47,8% der insgesamt möglichen Ziffernfolgen entspricht. Ganz entsprechend kann der Fall der achttstelligen Ziffernfolgen erledigt werden: $9^8 \cdot 10^6 = 43046721/10000000 = 43,0\%$.

Alles in allem kann Rechenberg nur ungefähr die Hälfte seiner Telefonate führen, nicht etwa 90%. (Dies ist natürlich eine statistische Aussage, die für große Zahlen recht genau, nicht aber den Einzelfall gilt.) Auch an diesem Beispiel sieht man wieder, daß Computererwer nicht immer zu einem Erfolg führt als Brainpower. Im sinnvollen Verhältnis der beiden zueinander liegt die Würze und die Kürze!

```

1:
2: PROGRAM wurfeln;
3:
4:
5: (* Es wird je viermal gewürfelt, und geprüft, ob dabei eine Sechse
6: auftritt. Wenn ja, wird der um das Faktor 5/3 vergrößerte Einsatz
7: ausbezahlt, bei DM 60 Einsatz wird z.B. DM 100 ausbezahlt. Wenn
8: nicht, ist der Einsatz verfallen.
9: Wie oft gewinnt (oder verliert) man dabei auf lange Sicht?
10:
11: Programmiersprache: Turbo Pascal, Autor: W.J. Weber, Januar 1987 *)
12:
13: CONST gewinnzahl = 6;
14:
15: VAR spielanzahl, i, gewonnen, verloren, nettogewinn: REAL;
16:
17: PROCEDURE Anfang;
18: BEGIN
19:   CLEREK;
20:   WRITELN('4 x Würfel:');
21:   WRITELN('-----');
22: END;
23:
24: FUNCTION Wurf(i: INTEGER): INTEGER;
25: BEGIN
26:   Wurf := RANDOM(6) + 1; (* i ist nur Dummy-Parameter *)
27:   (* Die Augenzahl muss zwischen 1 und 6 liegen *)
28: END;
29:
30: BEGIN (* Hauptprogramm *)

```

```

31:   Anfang;
32:   WRITE('Wie oft soll gespielt werden? ');
33:   READ(spielanzahl);
34:   WRITE('mal: ');
35:   gewonnen := 0;
36:   verloren := 0;
37:   i := 0; (* Schleife wird durch Real-Variable *)
38:   WHILE i < spielanzahl DO (* kontrolliert, ob eine grosse Zahl *)
39:     BEGIN (* von Wiederholungen zu gestatten. *)
40:       i := i + 1;
41:       IF gewinnzahl IN (Wurf(1), Wurf(2), Wurf(3), Wurf(4))
42:       THEN gewonnen := gewonnen + 1;
43:       ELSE verloren := verloren + 1;
44:     END;
45:     nettogewinn := 100 * gewonnen - 50 * spielanzahl;
46:     GOTOXY(1, 10);
47:     WRITE('Es gab ', gewonnen:0:1, ' Gewinne');
48:     WRITE(' und ', verloren:0:1, ' Verluste. ');
49:     WRITELN;
50:     WRITE('Bei konstantem Einsatz von DM 60 führt das zu einem ');
51:     IF nettogewinn < 0
52:     THEN WRITE('Gesamtverlust');
53:     ELSE WRITE('Gesamtgewinn');
54:     WRITE(' von DM ', abs(nettogewinn):0:1, ' ');
55:     WRITELN;
56:     IF nettogewinn < 0
57:     THEN WRITE('Traurig, traurig!');
58:     ELSE WRITE('Hurra!');
59:   END;

```

```

PROGRAM Ziffernkombinationen;
(* Es werden alle siebenstelligen Ziffernfolgen darauf untersucht,
ob sie keine Ziffer "7" enthalten.
Der Prozentsatz der so gefundenen Ziffernfolgen wird ausgegeben.
Programmiersprache: Turbo Pascal, Autor: W.J. Weber, März 1987 *)
VAR funktioniert: REAL; (* Ist erforderlich weil u.U.
Maximal überschritten wird *)
L1, L2, L3, L4, L5, L6, L7: INTEGER; (* Laufvariablen *)
BEGIN
  funktioniert := 0;
  FOR L1 := 0 TO 9 DO
    IF L1 <> 7 THEN
      BEGIN
        WRITE(' ');
        FOR L2 := 0 TO 9 DO
          IF L2 <> 7 THEN
            FOR L3 := 0 TO 9 DO
              IF L3 <> 7 THEN
                FOR L4 := 0 TO 9 DO
                  IF L4 <> 7 THEN
                    FOR L5 := 0 TO 9 DO
                      IF L5 <> 7 THEN
                        FOR L6 := 0 TO 9 DO
                          IF L6 <> 7 THEN
                            FOR L7 := 0 TO 9 DO
                              IF L7 <> 7 THEN
                                funktioniert := funktioniert + 1;
                            END;
                          WRITE(' ');
                        END;
                      WRITE('100-funktioniert/1E7:6:1:');
                      Write('9 aller 7-stelligen Ziffernfolgen ');
                      WriteLn('können erzeugt werden. ');
                    END;

```

HEFT
20
Aug
1987
10

Nun können im Dualsystem a_n , b_n und u_{n-1} in 8 verschiedenen Kombinationen auftreten und beeinflussen mit diesen auch u_n und z_n in verschiedener Weise. Diese 8 Kombinationen und deren Einfluß sozusagen als Funktion für u_n und z_n sind in der nachfolgenden sogenannten Boole'schen Wahrheitstafel dargestellt:

a_n :	0	0	0	0	1	1	1	1
b_n :	0	0	1	1	0	0	1	1
u_{n-1} :	0	1	0	1	0	1	0	1
u_n :	0	0	0	1	0	1	1	1
z_n :	0	1	1	0	1	0	0	1

u_n wird also unter folgenden Bedingungen gesetzt:

$$\begin{aligned} & (\text{NOT } a_n \& b_n \& u_{n-1}) \mid \\ & (a_n \& \text{NOT } b_n \& u_{n-1}) \mid \\ & (a_n \& b_n \& \text{NOT } u_{n-1}) \mid \\ & (a_n \& b_n \& u_{n-1}). \end{aligned}$$

Der erste Ausdruck in Klammern stellt die vierte Spalte der Wahrheitstafel dar, der zweite Ausdruck in Klammern die sechste Spalte, der dritte Ausdruck die siebente Spalte und der vierte Ausdruck die letzte Spalte. Diesen langen logischen Ausdruck kann man zusammenfassen:

$$u_n = a_n \& b_n \mid (u_{n-1} \& ((\text{NOT } a_n \& b_n) \mid (a_n \& \text{NOT } b_n))).$$

Desgleichen erhält man für z_n :

$$\begin{aligned} & (\text{NOT } a_n \& \text{NOT } b_n \& u_{n-1}) \mid \\ & (\text{NOT } a_n \& b_n \& \text{NOT } u_{n-1}) \mid \\ & (a_n \& \text{NOT } b_n \& \text{NOT } u_{n-1}) \mid \\ & (a_n \& b_n \& u_{n-1}) \\ & = (\text{NOT } a_n \& ((\text{NOT } b_n \& u_{n-1}) \mid (b_n \& \text{NOT } u_{n-1}))) \mid \\ & (a_n \& (\text{NOT } (b_n \mid u_{n-1}))) \mid \\ & (b_n \& u_{n-1}). \end{aligned}$$

Wenn Sie nun bedenken, daß in den beiden oben berechneten Ausdrücken die einzelnen Buchstaben nichts weiter bedeuten als elektrische Signale, und die Operationszeichen nichts weiter als Schaltungen, die die Eingangssignale steuern und verarbeiten, so daß die endgültigen Ausgangssignale als Ergebnis anzusehen sind, dann wissen Sie jetzt, wie Rechner rechnen – im Prinzip wenigstens. Abbildung 4 zeigt nochmals das Schema des Addiervorgangs im Rechner anhand der in Abbildung 3 dargestellten logischen Grundbausteine. Es liegt nun an Ihnen, sich vorzustellen (angenommen, Sie

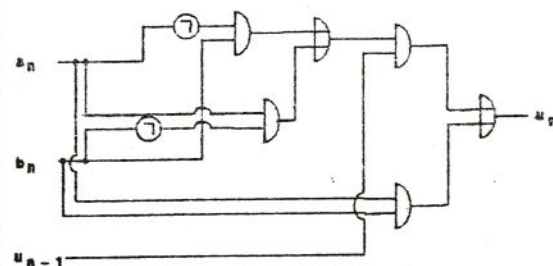
lassen von ihrem Computer eine Multiplikation durchführen), welche komplizierte Wege die Signale geben, die man über die Tastatur eingibt und die dann Sekundenbruchteile später bearbeitet auf dem Bildschirm flimmern.

Leider bin ich in dieser Folge nicht mehr dazu gekommen, die Fehler der Rechner-Rechnerei wie angekündigt zu behandeln, doch hielt ich es für wichtiger, einmal einen etwas längeren Exkurs in die Bereiche der Mathematik zu bringen, da doch die Funktionsweise eines Computers wie gesehen (bzw. gelesen) auf der wohl logischsten der Naturwissenschaften basiert.

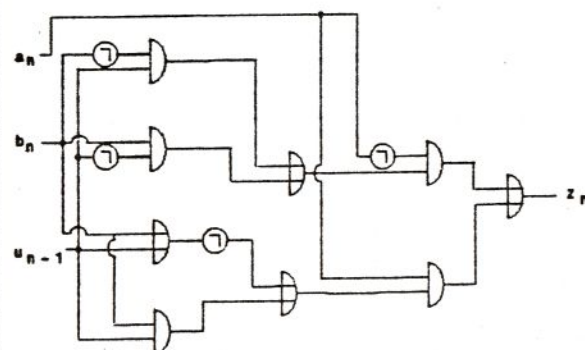
(Jörg Gondermann)

Abbildung 4: Additionsschaltung

a) Berechnung von u_n



b) Berechnung von z_n



Kosinus

von GUBA & ULLY

Ein typischer Computerfreak, dieser Kosinus: Tag und Nacht programmiert und bastelt er. Sport, Disco oder gar Mädchen interessieren ihn nicht so sehr. Kosinus ist ein kleines, manchmal weltfremdes Genie. Angst hat er nur vor einem: Stromausfall...



UND... UND NIEMAND VERSTAND DICH SO GUT WIE ER!



ABER ÜBERTREIBST DU NICHT DOCH EIN WENIG? ER WAR IMMERHIN NUR EIN COMPUTER!

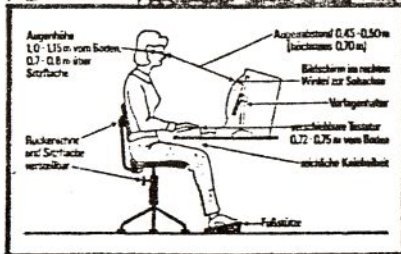


MANN... ICH WEISS NICHT, WAS ICH ZU DIESEM QUATSCH SAGEN SOLL!



Ich arbeite am Computer Bin ich strahlengefährdet?

Wissenschaftler haben jetzt
Computer getestet. Ergebnis: Die Arbeit
vor dem Bildschirm soll mit ganz bestimmten
Risiken verbunden sein



Eine Datentypistin muß täglich bis zu
30 000mal mit den Augen zwischen
Schreibvorlage, Tastatur und Bildschirm
hin- und herwechseln. Die Zeichnung
links zeigt Ihnen, wie der optimale Computer-
Arbeitsplatz aussehen sollte

Klagen darüber, daß die Arbeit am Computer krank macht, gibt es schon, seit die ersten Geräte 1968 bei uns eingeführt wurden. Jetzt soll es auch Beweise dafür geben! Drei Bremer Physiker haben 16 der meist-verkauften Personal- und Homecomputer untersucht und herausgefunden, daß keiner der Apparate strahlungsfrei ist. Alle Geräte sind von sogenannten elektrischen Feldern umgeben.

Unser zentrales Nervensystem reagiert darauf mit Augenflimmern und Kopfschmerzen. Langfristig kann es zu Herzbeschwerden, Kurzsichtigkeit und Atemnot kommen, bei schwangeren Frauen können Fehlgeburten oder Mißbildungen nicht ausgeschlossen werden.

HÖRZU sprach mit Dr. Ute Boikat vom Bremer Forschungs-Team.

Die meisten sitzen
zu nahe vor dem Bildschirm



Dr. Ute Boikat (35), Physikerin bei der Gesundheitsbehörde

HÖRZU: Wodurch wird die Strahlung verursacht?

Dr. Boikat: Eigentlich sind es keine Strahlen, sondern elektrische Felder. Sie werden durch die Bildröhre ausgelöst, die unter Hochspannung (etwa 15 000 Volt) arbeitet. Die Felder beeinflussen unser zentrales Nervensystem. Hinzu kommen Nervenreizungen durch Magnetfelder, die von den Spulen in den Geräten erzeugt werden und durch statische Felder, die dadurch entstehen, daß ständig Elektronen von innen auf die Computerscheibe geschleudert werden. Diese Felder laden den Computer oft so stark auf, daß wir leichte Stromschläge kriegen, wenn wir sie berühren.

HÖRZU: Welche Felder sind am gefährlichsten?

Dr. Boikat: Unter Fachleuten ist umstritten, ob elektrische Felder mehr auf den Menschen einwirken als magnetische oder statische. Schwedische Wissenschaftler halten die Magnetfelder für besonders gefährlich, die Kanadier die

elektrischen. Konkrete Beweise hat keiner.

HÖRZU: Eigentlich müßten die artverwandten Fernsehstrahlen dann genauso gefährlich sein wie Computerschalen?

Dr. Boikat: Sind sie auch. Wir haben in unseren Untersuchungen einen Farbfernseher einbezogen. Er hat die stärksten Magnetfelder erzeugt. Die Gefahr wird allerdings dadurch stark reduziert, daß man viel weiter vom Fernseher entfernt sitzt als vom Computer. Wer etwa drei Meter weg



Professor J. Bernhardt (48) vom Bundesgesundheitsamt

„Nicht die Bildschirme sind das Problem, sondern die Bildschirm-Arbeitsplätze, das beweisen über 400 Untersuchungen auf der ganzen Welt. Ein Beispiel von vielen: In Skandinavien erkrankten vier Frauen an Gesichtsausschlägen. Ihre Computer hatten sich durch extreme Trockenheit in den Räumen statisch aufgeladen und verschleuderten Staubpartikel. Durch mehr Luftfeuchtigkeit und Erdung der Computer war das Problem sofort beseitigt.“

Trotzdem bin ich dafür, daß bei uns endlich gesetzliche Grenzwerte für Computereinträge festgelegt werden.“

ist, bekommt kaum etwas ab. Die typische Arbeitsentfernung zum Computer-Bildschirm beträgt jedoch nur 30 Zentimeter.

HÖRZU: Sind die gesundheitlichen Schäden nur Vermutungen, oder gibt es auch wissenschaftliche Beweise?

Dr. Boikat: Die gibt es. Es ist allerdings nicht erwiesen, daß die von uns untersuchten Felder allein krank machen. Oft kommt Streß hinzu, ein falsch angelegter Computer-Arbeitsplatz – also die richtige Anordnung des Lichteinfalls auf den Bildschirm –, flimmernde Bilder...

HÖRZU: Sind Menschen am Computer auch krebgefährdet?

Dr. Boikat: Mediziner behaupten das. Eindeutige Beweise

dafür fehlen allerdings noch.

HÖRZU: Wer ist häufiger von Computer-Krankheiten betroffen, Frauen oder Männer?

Dr. Boikat: Eindeutig die Frauen. Weil mehr von ihnen an den Geräten arbeiten, aber auch, weil ihr Organismus, insbesondere während einer Schwangerschaft, feinfühler auf elektrische Felder reagiert.

HÖRZU: Und die Kinder? Wie gesundheitsgefährdend sind ihre Spielcomputer?

Dr. Boikat: Genauso gefährlich wie Arbeitscomputer. Die Billig-Geräte mit viel Plastik schirmen die Felder am wenigsten ab. Kinder sollten auf keinen Fall länger als eine Stunde täglich vor dem Bildschirm sitzen.

HÖRZU: Gibt es keine ungefährlichen Computer?

Dr. Boikat: Doch, meist sind Geräte mit Metallgehäuse besser. Aber sie werden weniger benutzt, weil sie bis zu 2500 Mark teurer sind.

Computer-Arbeitsplatz: Was Sie dazu wissen sollten



Wichtig vorm Bildschirm: korrekte Haltung, richtiger Abstand

Man sollte nicht länger als vier Stunden täglich vor dem Bildschirm sitzen, pro Stunde zehn Minuten Pause einlegen. Gegen die statischen Felder schützt ein Abschirmgitter aus feinem Metallgewebe. Es wird an einem Kabel gerätet. Die feinen Staubpartikel lagern sich nicht mehr auf Gesicht oder Schirm ab, sondern bleiben in der Luft hinter dem Gitter. Das Gaze-Filter kosten 160 bis 300 Mark. Vor elektrischen und magnetischen Feldern schützen Metallgehäuse. Ersatzweise kann man den Monitor auch mit Haushalts-Alufolie einwickeln.

HEFT
20
August
1987

110

Im Heft 12, erschienen im März 1986, also vor mehr als einem Jahr, erschien zum letztenmal eine Bücherliste des CLUB 80. Seither ist im Club einiges geschehen, aber bei den Büchern hat sich nicht viel getan!

Leider sind einige der Publikationen, wie schon mehrfach im Info beklagt, bei einer unkontrollierten Entnahmeaktion während des 86'er Clubtreffens in Holzhausen abhanden gekommen. Dafür wurde aber die Bibliothek wieder von anderer Seite (Peter Spieß, Bernd Retzlaff, Gerald Dreyer) mit neuen Büchern aufgefüllt.

Viel zu wünschen übrig, ließ bis jetzt nur das Interesse der Mitglieder an dem Angebot, sich die Bücher beim Club auszu-leihen, statt sich die manchmal doch recht teuren Veröffentlichungen selbst zu kaufen. So liegen die Bücher größtenteils bei mir im Keller herum und warten auf Leihler und Leser!

Ein Grund für die nur sehr schwache Nachfrage ist ohne Zweifel der oben angesprochene Umstand, daß seit über einem Jahr keine aktuelle Liste mehr erschienen ist. Dies soll sich nun ändern! Die folgend abgedruckte Liste ist auf dem neuesten Stand und enthält alle zur Zeit verfügbaren Bücher. Bei Anforderungen, die sich in nächster Zeit hoffentlich häufen, ist natürlich Paragraph 8 Absatz 2 der Clubsatzung zu beachten, der kurz gesagt folgendes festlegt:

1. Der Besteller hat die Kosten für den Versand in beide Richtungen zu tragen (bei Bestellungen 3,— DM in Briefmarken beilegen)!
2. Die Ausleihzeit beträgt vier Wochen (auf Antrag auch länger, wenn das Buch nicht von jemand anderem angefordert wurde)! Wird die Ausleihfrist überschritten erfolgt eine kostenpflichtige Mahnung (5,— DM, für jede weitere Mahnung ebenfalls 5,— DM).
3. Verlorenegegangene Bücher müssen ersetzt werden!
4. Weitergabe an Dritte (insbesondere Nichtmitglieder) ist untersagt und kann zum Ausschluß aus dem Club führen!

Ich hoffe, daß das doch recht gute Angebot bald mehr genutzt wird und damit ein zur Zeit brachliegender Teil der Serviceleistungen des CLUB 80 wieder reaktiviert wird!

Karsten Obermann

Nr. 0002: Computerwissen

Michael Scharfenberger

--- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmöglichkeiten von Hard- und Software sowie Erklärung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knebeleien programm. in BASIC

Ruedeger Baumann

--- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

-

--- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel

--- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80

Chris Palmer

--- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0008: More Basic Computer Games

David H. Ahl

--- Creative Computing Press, USA

84 Spiele fuer den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist fuer Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse

Busch Rudolf

--- Franzis-Verlag GmbH, München

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (als o mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen

Busch Rudolf

--- Franzis-Verlag GmbH, München

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (als o mit Lern-Effekt).

Nr. 0011: TRS-80 PROGRAMS

Tom Rugg und Phil Feldman

--- Dilithium Press, Beaverton, USA

32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012: Programme und Tricks fuer Genie I und Genie II

Clemens Becher, Franz Seger --- ?

Viele Programme, Tips und Tricks fuer den Genie.

Nr. 0013: BASIC: Alles ueber PEEK und POKE

Heiko Requardt --- Franzis-Verlag

Eine Software-Sammlung in BASIC (mit vielen guten Tips und Tricks fuer den 'Amateur').

Nr. 0014: TRS-80 und Video Genie ROM-Listing fuer Level II

Luidger Roeckrath --- ?

ROM-Listing, RAM-Adressen, I/O-Adressen, Unterprogramme, Basic-Anweisungen und Funktionen, Aufzeichnungsformate auf Cassette, ...

Nr. 0016: Das Modem-Sonderheft (Datenübertr. mit Mikrocomp.)

verschiedene --- MC, Franzis-Verlag, München

Grundlagen, Schaltungstechnik, DFÜ-Programme

Nr. 0017: GENIE DATA (Heft 1/83 bis Heft 3/85)

--- Heinz Hueben Verlag

Eine Sammlung von 13 Computerheften.

Nr. 0020: The Custom TRS-80 & Other Mysteries

Dennis Bathory Kitz --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das sich mit Soft- und Hardware für unseren Computer beschäftigt.

Nr. 0022: TRSDOS 2.3 Decoded & Other Mysteries

James Farvour --- IJG Inc., 1953 West, USA

Ein kommentiertes TRSDOS 2.3 in englischer Sprache.

Nr. 0023: TRS-80 Disk & andere Geheimnisse

H.C. Pennington (G. Daubach) --- IJG Inc., 1953 West, USA

Das ins Deutsche übersetzte Buch beschreibt die Diskettenorg., die Zuordnung freier Speicherbereiche, die Behebung v. Diskettenfehlern etc.

Nr. 0024: TRS-80 Data File Programming (Model I/III)

Finkel & Brown ---

Ein Lehrbuch, wie man in BASIC Datenfiles programmiert und benutzt. Mit sehr vielen Beispielen (englisch-sprachig).

Nr. 0025: Programming Techniques for Level II BASIC

William Barden ---

Ein gutes Buch über schnelle Graphiken, Suchroutinen und Sortier Routinen im Level II BASIC (englisch-sprachig).

Nr. 0026: Advanced Level II BASIC

Don Imman, Albrecht, Zamora ---

Ein gutes Lehrbuch mit vielen Beispielen zu allen Themen der Computerei (englisch-sprachig).

Nr. 0027: Level II BASIC

Don Imman, Albrecht, Zamora ---

Ein gutes Lehrbuch für den Anfänger in BASIC (englisch-sprachig).

Nr. 0028: Programming In Style

Th. Dwyer, Margot Critchfield ---

Einführung in BASIC, strukturiertes Programmieren, Tips zur besseren Aufmachung der eigenen Programme, Tips für den Profi (englisch-sprachig)

Nr. 0029: The First Book Of 80-US

verschiedene --- Hofacker Verlag, München

Sammelband aus dem TRS-80 Users Journal.

Nr. 0030: The Rest Of 80

verschiedene --- Wayne Green Books, USA

Viele Beiträge aus der 80-Micro (Allgemeines, Graphik, Utilities).

Nr. 0031: Z-80-Applikationsbuch

Michael Klein --- Franzis Verlag, München

Eine Einführung in die Programmier- und Interfacetechniken des Mikroprozessors Z 80.

Nr. 0032: Z 80 Assembler Handbuch

--- Hofacker, München

Die Z 80 Befehle mit Kommentar und je einem Beispiel.

Nr. 0033: Programmierung des Z 80

Rodnay Zaks (Bernd Ploss) --- Sybex-Verlag, Duesseldorf

Als Lehr- und Lernmittel konzipiert ist dieses ins Deutsche übersetzte Buch ein umfassendes Nachschlagewerk und eine gründliche Einführung.

HEFT

20

August

1987

114

Nr. 0034: Daten - Disketten - Dateien

Hans-Joachim Sacht

--- CHIP, Vogel-Verlag, Wuerzburg

Anregungen und Hinweise, wie Programme mit komfortabler Dateiverw. geschr.
ieben werden koennen und was bei den versch. DOS zu beachten ist.

Nr. 0035: Anwenderprogramme fuer TRS-80 und Video Genie

M. Stuebs

--- Hofacker, Muenchen

Viele (meist einfache) BASIC-Programme fuer den TRS-80.

Nr. 0036: Programmieren mit TRS-80

M. Stuebs

--- Hofacker, Muenchen

Eine Einfuehrung in den TRS-80 und viele (meist einfache) BASIC-Programme

Nr. 0038: Textsystem in BASIC und Z80-Assembler

Gert Scheller

--- Luther-Verlag, Sprendlingen

Das Buch beschreibt ein fertiges Textsystem fuer den TRS-80, das in BASIC
und Z80 modular aufgebaut ist, und leicht erweiterbar ist.

Nr. 0039: Textsystem in BASIC und Z80-Assembler - Band 2

Wolfgang Weber

--- Luther-Verlag, Sprendlingen

Der Band 2 beschreibt Erweiterungen zum Programm aus Band 1.

Nr. 40 : Einfache Peripheriegeräte im Selbstbau

Owen Bishop

--- Birkhäuser

Verbinden Sie Ihren Microcomputer mit seiner Umwelt! Einfache und nachbau-
sichere Schaltungen zur Erweiterung des Computerhorizonts!

Nr. 41 : Start in die Computergrafik

Fred Wagenknecht

--- Vogel-Verlag

Grundlagen und Programme der Computergrafik für TRS 80, VideoGenie und Co-
lorGenie.

Nr. 42 : Level II BASIC Reference Manual

--- Radio Shack

Dieses Buch wird normalerweise von Tandy zum Model I dazugegeben. Für Gen-
ie-Besitzer ev. interessant!

Nr. 43 : BASIC-Code

Wiegand/Fillinger

--- Ravensburger

Die kompatible Computersprache für über 25 Homecomputer. Mit 14 Programml-
istings und Kassette. Natürlich auch für TRS 80 Model I!!!

Nr. 44 : Kleiner BASIC-Wortschatz

Busch

--- Franzis-Verlag

Die wichtigsten BASIC-Begriffe einfach erklärt und gelistet!

Nr. 45 : BASIC Computer Games

David Ahl

--- Creative Computing Press

101 Computerspiele für Homecomputer!



1. Vorsitzende **Hartmut OBERMANN**
 Schwalbacher Straße 6
 6289 Heidenrod 1
 ☎ 06124 /3913

2. Vorsitzende **Gerald SCHRÖDER**
 Am Schützenplatz 14
 2185 Seevetal 1
 ☎ 04185 /2682

Hardwarekoordinator **Eckehard KUHN**
 Im Dorf 14
 7443 Frickenhausen 1
 ☎ 07022 /45417

Diskothekar **Werner FÖRSTER**
 Christoph-Krebs-Straße 9
 8720 Schweinfurt
 ☎ 09721 /21841

Redaktion **Jens NEUEDER**
 Panoramastraße 21
 7178 Michelbach /Bilz
 ☎ 0791 /42877

Autoren Die Redaktion bedankt sich bei
 den im INHALTSVERZEICHNIS genannten
 Autoren für die Mitarbeit an der
 Club-INFO.

Druck **Peter Spieß**
 Trugenhofenstraße 27
 8859 Rennertshofen 1
 ☎ 08434 /454

Bankverbindung des CLUB 80
 Postgirokonto Peter STEVENS
 Sonderkonto CLUB 80
 Konto-Nummer 285 491 - 465
 Postgiroamt Dortmund
 BLZ 440 100 46

Hallo Club-80er,

trotz Sommerloch, Dauernebenbeschäftigung als "Häuselsbauer" und Dank Eurer Mitarbeit konnte ich Euch ein dickes INFO fabrizieren.

In der Ferienzeit habt Ihr nun sicher auch einmal die Gelegenheit es ausgiebig und in Ruhe zu lesen. Es ist ja auch diesmal wieder recht umfangreich und interessant geworden. Nicht zuletzt dadurch, daß wir diesmal einen "clubfremden" Schreiber dabei haben. Ihm möchte ich auf diesem Wege meinen besonderen Dank für seine Mitarbeit an unserem Clubinfo aussprechen.

Als nächstes möchte ich nun einen Punkt in unserem Clubleben ansprechen, bei dem Ihr mich sehr enttäuscht habt.

Die Entscheidung des Zehnzeilerwettbewerbes !!

Ganze 2 (in Worten: zwei) Clubmitglieder hatten sich zur telefonischen Abstimmung gemeldet. -- Waren die anderen alle im Urlaub ??

Ich finde, das daß kein gutes Zeichen für eine Interessengemeinschaft, wie wir es eine sind, ist, sich am Clubleben nicht zu beteiligen. Immerhin haben sich die Programmschreiber -auch für Euch- bemüht etwas auf die Beine zu stellen. über eine Reaktion von Eurer Seite aus hätten sie sich sicher gefreut!

Da wir in nächster Zeit ein Vorstadstreffen haben, werden wir Vorstandsintern eine Entscheidung über eine Belohnung der Wettbewerbs-Programmierer treffen.

Ich finde Eure Art der aktiven clubinternen Mitarbeit echt mangelhaft. Leichter und einfacher hättet Ihr eine Abstimmung (Hörer abheben, wählen und Eure Stimme abgeben) wirklich nicht mehr haben können.

Aber nichts desto Trotz wünsche ich Euch viel Freude mit Eurer neuesten INFO.

Jens

Das INFO erscheint zweimonatlich.
 Es erfolgt keine Zensur oder Kontrolle
 der jeweiligen eingeschickten Infobeiträge
 durch die Redaktion.

Club 80
Aktuell

Club 80
Aktuell

Club 80
Aktuell

Alle waren voll!

-zählig erschienen! Wo!? Natürlich beim Vorstandstreffen vom 14.-16.08. beim 1. Vorsitzenden Hartmut Obermann. Dies war bemerkenswerterweise das erste Vorstandstreffen, bei dem wirklich der ganze Vorstand beisammen war. Außerdem konnten wir uns noch über mehrere Gäste freuen (Ulrich Böckling, Gerald Dreyer und Rüdiger Sörensen).

Während des Treffens konnten wir viele brennendheiße Themen abhandeln. Hier nun eine kurze Zusammenfassung (die Ergebnisse werden in einzelnen Artikeln erläutert):

- Schaffung einer MS-DOS-Gruppe bzw. -Ecke
- Anschaffung von
 - Reinigungsset für Diskettenlaufwerke
 - Büchern über CP/M, Sprachen usw.
 - neuer PD-Software aus den USA
- Einrichtung einer CP/M-Programmbibliothek
- Aktualisierung des Grafik-Sonderheftes
- Auflage weiterer Sonderhefte
- Auflage eines Sammelbandes der Clubinfos für Neuzugänge
- mehr Werbung für den Club durch
 - Beilagen zu Firmen-Werbesendungen
 - Plakate auf Messen, Ausstellungen und Flohmärkten (Neugestaltung eines Werbe-Plakates)
 - Mundpropaganda der Club-Mitglieder
- Termin- und Ortsplanung für das Clubtreffen 1988 (vom 12.5. bis 15.5., Raum Frankfurt-Darmstadt)
- allgemein Administratives.

Nach diesem strapaziösen Wochenende mit vielen ungesunden Sachen wie schwarzem Kaffee und wenig Schlaf wird der Vorstand eine Kur antreten müssen. Da im Moment die Clubfinanzen dies zulassen, wird die Vorstandsadresse auf unbestimmte Zeit nach Florida verlegt.

Bis zu unserer Rückkehr fröhliches Hacken wünscht

Der gesamte Vorstand.

Name	Vorname	Straße	PLZ	Stadt	Telefon	privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Nistedt	04182	/8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204	/62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431	/241907	// 0431 /74047
Betz	Heinrich	St. Wolfgangstraße 13	8551	Hausen	09191	/31698	// 09191 /611108
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069	/5601621	// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451	/7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261	/69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Niesbaden-Bierstadt	06121	/508218	// -
Drowälder	Bernd	Hügel 1	4441	Wettringen	05233	/4320	// 02557 /1236
Emmerich	Helmut	Waldstraße 5	6682	Ottweiler	06824	/4114	// -
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721	/21841	// 09721 /51256
Gromotka	Uwe	Lange Reihe 40	2003	Meyhe	0421	/ 88496	// 0421 /300-6870
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144	/6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127	/70024	//
Hill	Peter	Bergstraße 65	6754	Otterberg	-		// -
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948	/542	// 02921 /70431
Krispin	Michael	Schwanstraße 8	4130	Moers 3	02841	/73690	// -
Krüger	Karl-Herbert	Bruchweg 65	4920	Lemgo	05261	/13686	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhäusen 1	07022	/45417	// 07022 /77474
Lose	Gerhard	Viefhaushof 42	4306	Essen 13	0201	/212608	//
Mand	Harald	Kleinflintbeker Straße 7	2302	Flintbek bei Kiel	04347	/3629	// 0431 /3013500
May	Holger	Marienstr. 9	5760	Sundern 2	02935	/1668	// -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911	/506051	// 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 20	6940	Weinheim -Lützelstaden	06201	/55052	// -
Müller	Kurt	Soltstraße 24a	2050	Hamburg 80	040	/7246083	// 04151 /8891-37
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791	/42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124	/3913	// -
Obscheringkat	Helmut	1 RUE DES BRUYERES	F-68340	Soultz	0033089/762690	//	-
Perschbach	Patrick	Waldstr. 52	5000	Köln 91	0221	/872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01	/7047418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141	/603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8000	Fürstfeldbruck	08141	/43791	//
Reichelt	Dieter	Philipp-Schmitt-Straße 30	6902	Sandhausen	06224	/52906	// -
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133	/4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122	/43551	// 04103 /605310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203	/331383	// 0203 /331383
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	-		// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221	/70/044	// -
Schreie	Christian	Fliederweg 32	4000	Düsseldorf 31	0203	/740097	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105	/2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025	/1631	// 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228	/643853	// -
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451	/791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8059	Rennertshofen 1	08434	/454	// 08431 /7041604
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a. TM.	05401	/99585	// 05401 /30096
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231	/593883	// 0231 /593883
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153	/1564	// -
Sörensen	Rüdiger	Thomas-Mann-Straße 30	6500	Mainz 1	06131	/32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362	/42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711	/731285	//
Wackor	Fred	Postfach 2246	7550	Rasstatt	07222	/52574	// -
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035	/3361	// -
Wucherer	Jürgen	Manzelstraße 1	7750	Konstanz	07531	/54686	// -

Stand: Juli 1987

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion