

CLUB INFO

31. AUSGABE

ADRESSE: CLUB 80 / FRITZ CHWOLKA / SAARSTRASSE 34 / 5173 ALDENHOVEN
TEL.: 02484 / 8920

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

Seite:
und Autor:

Seite:
und Autor:

Clubinternes

Neues vom Vorstand	01 - 03	
		Gerald Schröder
Clubtreffen in Spremberg/DDR	04 - 05	
Nachbemerkung zu Spremberg	06	
Bildenseite vom Clubtreffen	07 - 08	
		Arnulf Sopp
Termine	46	
		Redaktion

Software

Turbo-Pascal und ZCPK	09 - 11	
		Alexander Schmid
ZEUS und der HD64180 oder Z180/280 .	12 - 13	
Programme relocieren	15 - 16	
		Arnulf Sopp
Harddisk-Einbindung ins CP/M-Bios ..	17 - 22	
		Helmut Bernhardt, Peter Petersen

Hardware

Die Clubeigene Grafikkarte	23 - 24	
		Andreas Magnus
Club 80 Projekt	25 - 32	
		Helmut Bernhardt
IMB-Banker mit IMB	33	
		Alexander Schmid

Röme

Wer hat was -- wer will was	34	
-----------------------------------	----	--

Konstruktives

Schnittstellen-Programmierung	35 - 41	
Von 80 auf 180 oder 280?	43 - 44	
		Gerald Schröder

Club Bibliothek

Buchvorstellung	45	
-----------------------	----	--

Die letzten Seiten

Impressum	47	
Schluß	48	
		Redaktion
Bildseiten Sprembergtreffen	am INFO-Ende	
		Arnulf Sopp
Mitgliederadressenliste	am INFO-Ende	
		Fritz Chwolka

Club 80-Hauptversammlung 1990 - das Ereignis des Jahres

Vom 28.4. bis 1.5. fand die diesjährige Hauptversammlung statt, deren Ergebnisse ja schon im letzten Info kurz wiedergegeben wurden. Hier nochmal ein etwas längerer Bericht.

Anwesend waren (und haben damit das Klassenziel erreicht):

Helmut Bernhardt, Heinrich Betz, Jörg Brans, Fritz Chwolka, Manfred Held, Jürgen Kemmer, Andreas Magnus, Kurt Müller, Jens Neueder, Stefan Nitschke, Hartmut Obermann, Alexander Schmid, Gerald Schröder, Arnulf Sopp, Rüdiger Sörensen, Oliver Volz. (sowie kurzzeitig Horst-Dieter Schroers)

Versammlungsort war diesmal Weilburg, ein kleines Städtchen an der Lahn, das der mitgereisten holden Weiblichkeit genügend Sehenswürdigkeiten und das richtige Wetter zum Erkunden bot. Wir Herren/Hacker dagegen bekamen nur den Keller in unserem Hotel zu sehen, denn da tagten (?) wir bis spät in die Nacht. Da das Hotel gleichzeitig ein italienisches Restaurant beherbergte, brauchten wir nicht mal zum Essen das Gebäude verlassen.

Im Tagungsraum fand sich eine bunte Mischung aller möglichen Rechner, aber es war diesmal kein TRS-80 Model I/III/Genie I mehr zu sehen. Das zeigt m.E. den allmählichen Wandel des Club 80 in Richtung der neueren Nachbauten (Model IV(p), Genie IIs/IIIs) und der CP/M-Welt. Natürlich wurden zu Vergleichszwecken auch einige "Gastrechner" neben "die schnellsten 111-Rechner der Welt" (c't anno dazumal) gestellt.

Im einzelnen waren vertreten: Model IVp (3x), Genie IIIs (2x), Genie IIs (2x), Model IV, Prof-180X, Tatung (CP/M). Als Gäste: Acorn Archimedes, MS-DOS (2x), Atari-Mega-ST.

Bemerkenswert finde ich, daß 5 Rechner hauptsächlich unter CP/M liefen, wobei drei als Oberfläche ZCPR benutzten. Außerdem konnten wir in Arnulfs Genie IIIs einen Z180-Prozessor bewundern, während Fritz schon Werbung für einen Z280-Rechner machte.

Es stellte sich als eine der wichtigsten Aufgaben des Treffens heraus, das CP/M-Format von Fritz' Tatung-Rechner zu knacken, was Rüdiger und Hartmut dann auch gelang. Übrigens halten nach den Megabyte RAM auch Harddisks in die Z80-Welt Einzug: Jörgs Model IVp, Hartmuts Prof-180X sowie Arnulfs und Andreas' Genie IIIs sind damit ausgestattet. (Der Tatung steht kurz davor.)

Neben dem regen Gedankenaustausch fand natürlich auch ein (kurzer) offizieller Teil statt. Der wurde aber auf der Terrasse im strahlenden Sonnenschein durchgeführt, weil unsere Herren Hacker sonst gar nicht hätten sagen können, wie das Wetter draußen eigentlich aussieht. Also brieten 16 Club-Mitglieder (und einige beratende bessere Hälften) in der Sonne und lauschen den direkt nebenan vorbeirauschenden Pkws und Lkws. Alle Abstimmungen fielen dementsprechend 16:0 aus, d.h. es gab einfach keine Gegenstimme, wobei nachher einige Teilnehmer von den Ergebnissen überrascht waren (dafür aber sehr interessante Gespräche mit ihren Nachbarn geführt hatten).

Zuerst mußten wir entsetzt zur Kenntnis nehmen, daß die Kassenlage katastrophal ist. Falls die letzten 20 säumigen Mitglieder ihren Beitrag auch noch bezahlen sollten, würden wir fast DM 7000 (siebentausend) in der Kasse haben, was ein Rekord ist. Allerdings müssen wir je (erscheinendem) Info im Moment mit ca. DM 600 Kosten rechnen. Aber die Hoffnung, daß wir die Kohle so wieder loswerden, schwindet zusehends, denn unser Club-Redakteur Jens Neueder mußte uns berichten, daß keine Artikel mehr vorliegen. Daraufhin wurde beschlossen, ab jetzt solange "Not-Infos" (Deckblatt & Impressum) herauszubringen, bis Ihr Euch wieder bequemt, Eure Gedanken, Erfahrungen und Telefongespräche auf Papier zu bringen.

Eine andere Möglichkeit, diesen Geldberg abzubauen, wurde abgelehnt: Die Beiträge werden nicht gesenkt und es werden auch keine Staffel-Beiträge oder Honorare für fleißige Autoren eingeführt. Diese Maßnahmen hätten nur zur Verwirrung geführt und zur Belustigung beigetragen, aber sonst kaum Ergebnisse gezeigt.

Die Zahl der Mitglieder hält sich relativ konstant bei 70 Mann und einer Dame (Emanzipation, wo bist Du geblieben?). Allerdings könnte sich diese Zahl durch "verdeckte" Austritte, indem einfach der Beitrag nicht bezahlt wird, noch verringern.

Wie in allen aktuellen Diskussionen, mußten wir uns natürlich auch mit der DDR befassen. Wir haben ja schon die ersten Ost-Mitglieder, hatten bisher aber Probleme mit Disketten-Tausch und persönlichen Besuchen (vielleicht erinnert sich noch jemand an "damals"?). Das sollte sich jetzt grundlegend ändern und in absehbarer Zeit wird auch die Bezahlung des Clubbeitrages, wenigstens von der Währung her, einfacher werden. Ergebnis: Bis zur nächsten Hauptversammlung gilt: alle Interessenten aus der DDR können kostenlos und beitragslos Mitglied werden. Im nächsten Jahr sehen wir weiter.

Übrigens vermuten wir ein größeres Mitglieder-Potential in den Ost-Gebieten, denn dort ist der Z80 ja noch up-to-date, was man (leider oder glücklicherweise?) hier nicht mehr sagen kann. Schätzungsweise wird "drüben" auch mehr gebastelt als in unserer "Wegwerf-Gesellschaft", was zu interessanten Einblicken führen könnte.

Nochmal kurz zum Info: Das Drucken klappt, wobei Jens sogar weniger Arbeit als vorher hat, weil die Infos schon gebunden werden. Eine zweite Druck-Möglichkeit hat Helmut aufgetan, wobei die Kosten ähnlich günstig sind wie die jetzigen. Dementsprechend war der Kopierer-Kauf auch kein Thema mehr. Über die Artikel-Ebbe hatte ich ja schon weiter oben berichtet.

Weiterhin wurde beklagt, daß auf veröffentlichte Artikel selten eine Reaktion erfolgt. Auch Aufrufe (so der von Rüdiger zur Mitarbeit an einem erweiterten BIOS) liefen ins Leere. Wir möchten uns/Euch deshalb bitten, die Artikel nicht nur zu lesen, sondern auch zu überlegen, ob sie eine Antwort wert sind. Ähnlich sieht es bei unseren anderen Anlaufstellen aus: Sowohl im Hardware- wie auch im Software-Bereich seid Ihr scheinbar wunschlos glücklich bzw. voll ausgestattet. Glückwunsch!

Womit wir schon beim Vorstand bzw. den mehr oder weniger offiziellen Posten des Clubs wären. Das Gros behält seine Posten, nur die Spitze wurde ausgetauscht und Kurt zum neuen Bücher-Menschen ernannt (weil nicht bekannt war, ob Christian das weitermachen möchte). Die Verteilung sieht jetzt so aus:

1. Vorsitzender	: Fritz Chwolka
2. Vorsitzender	: Gerald Schröder
Kassenwart	: Familie Obermann
Info-Redakteur	: Jens Neueder
Hardware-Koordinator	: Andreas Magnus
Bücher	: Kurt Müller
Newdos-Programme	: Oliver Volz
CP/M-Programme	: Rüdiger Sörensen
Model-IV(p)	: Jörg Brans

HEFT
31
Juli
1990

02

Der letzte "Posten" ist neu: Jörg hat sich angeboten, die Model IV(p)-User zu betreuen. Durch Jörgs Kontakte zum großen Bruder (USA) hat er einige interessante Erweiterungen (RAM, Textverarbeitung, Festplatte) und Literatur aufgetan, die auch für die anderen Model IV(p)-Besitzer interessant sein könnten. Vielleicht meldet Ihr Euch mal bei ihm?

Der neue Vorstand hat hoch und heilig versprochen, genauso viel Arbeit in den Club zu stecken, wie das die Mitglieder offensichtlich tun (z.B. beim Schreiben von Artikeln). Weiterhin wird unser Erster auch noch etwas Werbung in seiner CP/M-Gemeinde (er betreut eine Liste mit ca. 100 CP/M-Usern) und in der DFÜ-Szene machen. Begleitet wird diese Aktion von Veröffentlichung der neuen Adresse in diversen Zeitschriften. Obri-gens könnt Ihr auch von Euch aus Werbung für den Club (z.B. in Anzeigenblättern oder Mailboxen) machen. Oder teilt dem Vorstand mit, wo wir noch inserieren sollten.

Ach ja, zwei Dauerbrenner habe ich noch vergessen:

1. Die Eintragung ins Vereinsregister wird endgültig nicht stattfinden.
 - a) dürfte das inzwischen verjährt sein
 - b) meint unser clubeigener Steuerprüfer, daß wir so nicht prüfungswürdig sind
 - c) machen wir auch noch Gewinn (s. Kassenstand)
2. Die offizielle Einführung von MS-DOS in den Club wurde gar nicht erst diskutiert.

Durch wundersame Wendungen des Schicksals (bzw. die Weisungen unserer geliebten Hardthöhe) hat es den agilen Hartmut Obermann nach Nord-Italien (nach Günzburg) verschlagen, so daß wir jetzt guten Mutes auf das erste Südländertreffen warten. Im Norden wird das schon traditionelle Nordlichter- bzw. Fischköpfe-Treffen (Nr. 4) diesmal von Kurt Müller organisiert. Vielen Dank an diese beiden Aktivisten!

Mehr fällt mir zum Clubtreffen nicht ein, außer, daß es wieder super-affengeil-abgefahren-sahnenmäßig war. Wie jedes Clubtreffen eben.

Gerald



Spremburg? Nie zuvor davon gehört. Es liegt in der DDR. Seine Umgebung, die landschaftlich etwa an das Ruhrgebiet erinnert, erinnert auch optisch daran, wie jenes früher war: Ach nein, wir wollen in diesen Zeiten nur Nettes sagen. Um nicht lügen zu müssen, darf ich dann allerdings nur über die wenigen Spremberger berichten, die ich kennenlernte. Die Stadt (und nicht nur sie) hat das alte Regime nämlich geschafft.

Ringsum Tagebau. Im Süden die Stadt Schwarze Pumpe mit einem Braunkohle-Energiekombinat. Frank erzählt, daß man manchmal noch in Spremberg seine hellen Fußabdrücke im Ruß auf der Straße sieht.

Es gibt Oasen. Da wäre zum Beispiel Franks Wohnung (siehe Mitgliederliste). Mein Feinsliebchen und ich hatten in den drei Tagen, die wir uns dort aufhielten, immer das Gefühl, willkommen zu sein. Frank war vorbildlicher Gastgeber, Fremdenführer, auch Mitsäufer und ein amüsanter Gesellschafter. - Eigenartig; da erzählt einer, wie er als Angehöriger seines Bürgerkomitees die Stasi ausgenommen hat, und man amüsiert sich darüber, ignoriert seine eigene Gänsehaut. Psychische Notwehr?

Ach so, das Clubtreffen. Nein, tut mir leid, noch nicht, denn Computer sehe ich hier pausenlos. Was mir von dieser Reise in Erinnerung bleiben wird, sind nicht die flimmernden Bildschirme, das hektische Programm-tauschen, die Fachsimpelei über Entwicklungen, von denen man bestenfalls gelesen hat (kommt uns das alles nicht sehr bekannt vor?). Es ist die unbeschreibliche Exotik der Situation, der Mief von Braunkohleöfen und Trabis, Franks Mutter, die in 40 Jahren DDR nicht gealtert zu sein scheint, es ist ein Museumsdirektor-Ehepaar, das ebenso schlitzohrig wie mutig die Machthaber verarschte und damit den Bismarckturm (scheußlich-schöner, aber authentischer Historismus) wenigstens weitgehend rettete, es ist die Kellnerin, die uns das Essen am liebsten an den Kopf geschmissen hätte, weil Gäste nun mal die Ruhe stören, es ist aber auch die andere Kellnerin anderswo, die sich offenbar tatsächlich dafür interessierte, ob es uns geschmeckt hat. Es ist Spremburg, es ist die DDR.

Es war einmal ein Clubtreffen. Sollte ich gottbehüte jemals Enkel auf meinem Schoß sitzen haben, werde ich ihnen das ungefähr so erzählen. Wirklicher als ein Märchen kam es mir nämlich nicht vor: Elf oder zwölf Besitzer eines Computers. Sagt da gerade jemand, "Na und?" Freunde, wir sind in der DDR! Elf oder zwölf Besitzer eines Computers, die sich seit ein paar Jahren als Club verstehen. Normal? Wer einen Rechner besitzt, kann damit selbstredend nur beabsichtigen, das Regime zu stürzen. Wie, das wußte das Regime auch nicht so genau. Wenn aber im Prinzip schon die Steinzeittechnik des Küchenmessers dazu ausreicht, dann ist ab der Glühbirne äußerste Wachsamkeit geboten! Wenn sich ein paar real existierende Computeraner treffen, ist das eine Zusammenrottung. Wer diese kriminelle Vereinigung beherbergt, ist auch mit dran. Pardon: War. Elf oder zwölf Digitalpartisanen.

Kleiner Seitenhieb nach Westen: Computer, auch in dieser Größenordnung, stehen immer noch auf der Cocom-Liste. Man kann ja nie wissen.

Merkwürdig - nur zwei von ihnen sprechen sächsisch. Ist das nicht die Hochsprache der DDR? Ich sollte meine Vorurteile überdenken. Die anderen haben den örtlichen Lausitzer Tonfall, der sehr an Berlinerisch erinnert.

Die Kneipe ist nach hiesigem Sprachgebrauch das Vereinslokal der örtlichen Laubenpieper. Im lokalen Idiom hört sich das so an: "Spartengaststätte der Sparte 'Eigener Aufbau'". Ach so.

Auf den Tischen standen ungefähr zehn Meter Computer, hauptsächlich Sinclair ZX-Spectrum. Ein vermutlich IBM-kompatibler Laptop mit Plasma-Display schien irgendwie nicht dazuzugehören. Wer soeben mitlächelnd lächelte,

wird später möglicherweise vor Neid erblassen. Abenteuerliche Floppy-Controller, sämtlich Eigenbau, sämtlich ohne Gehäuse, standen daneben. Die Laufwerke wieder eine Kabellänge weiter. Die meisten haben ebenfalls kein Gehäuse. Diese Maschinen kriegen CP/M auf die Reihe. Mit unseren TRS-80 ist das schon schwieriger.

Computer, Controller, Floppy, Bildschirm. Vier Verbraucher. Franks Steckdosenleiste ist ein Holzbrett mit vier Aufputzdosen. "Der DDR-Bürger ist Weltmeister im Improvisieren," sagt er. Natürlich gibt es auch fertige Leisten, wie wir sie kennen. Zur Zeit vielleicht beispielsweise in Leipzig oder in Königs Wusterhausen.

Uwe hat seinen Schneider Joyce nicht mitgebracht. Wenn ein Computer den Transport nicht übersteht, ist erst mal Feierabend. HO hat keine Teile. Alles muß man sich umständlich und ziemlich langwierig "besorgen". Ein Wort, das man verdächtig oft hört.

Einer ist ziemlich sicher, daß eine Drahtbrücke umgepolt gehört. Der Wessi lötet aus und um und läßt es darauf ankommen. Der Ossi hat jetzt ein Problem. Natürlich gibt es Lötkolben. Aber was, wenn dabei ein Teil die weiße Fahne schwenkt? Der Wessi kauft ein neues, falls er es nicht schon auf Halde liegen hat. Der Ossi hat jetzt ein Problem.

Auf einem Tisch steht ein Fotokopierer. Nein, keine Pershing 2. Ein Fotokopierer. Sein Eigner hat ein Joint Venture mit einem Bundi gemacht. Zu beiderseitigem Nutz und Frommen (vermutlich, solange es dem Bundi nutzt und frommt). Karlheinz kopiert wie besenzt. Schon mal beim R19 die Türen durch die Scheibe per Miniatur-Infrarot-Fernbedienung im Autoschlüssel auf- und zugeschlossen? Es muß für Karlheinz ungefähr dasselbe Feeling sein. Bislang waren Kopierer hier etwa so einfach zugänglich wie die Goldvorräte in Fort Knox.

Frank hat zwei Floppy-Laufwerke dabei. Wer einen Artikel schreibt, kriegt eins. Er nimmt seine Laufwerke wieder mit. Frank hat dasselbe Problem wie unser Jens-80. Schade, Jens zahlt nicht dieselben Honorare wie Frank.

Er hat auch einen Stapel elektronischer Bauteile aus ehemaligen Stasi-Abhöranlagen mitgebracht. Wer möchte, kann sich kostenlos bedienen. Auch hiervon bringt er das meiste wieder mit nach Hause. Hat man die Brauchbarkeit dieser Teile nicht erkannt? Oder fürchtet man das kalte Grausen, jedesmal, wenn man eins davon einlötet?

Zwischendurch gibt es Essen. Das Schnitzel ist gebraten. Meine sind meistens gekocht. Die Bundi-Bauern haben das unheimlich drauf, mit Hormonen oder Hortensien oder Wasweißich das Fleisch mit Wasser anzureichern, um Gewicht zu machen. Wasser zum Fleischpreis ist ein enormes Geschäft.

Das Clubtreffen hatte auch einen offiziellen Teil. Vereinsmeierei hier wie da. Wesentlicher Punkt der erfrischend kurzen Tagesordnung: Sollen wir dem Club-80 überhaupt, einzeln, en bloc oder nicht beitreten? Können wir ihm gem. Artikel 23 GG einfach mitteilen, daß Franks Mitgliedschaft jetzt für alle von Franks Club gilt? Was kostet das? Und für uns?

Und wenn sie keinen System Crash hatten, dann hacken sie noch heute.

Natürlich werden es mir meine Enkel nicht glauben, aber ihre Ihnen auch nicht. Die Zeiten sind vorbei. Oder?

Arnulf

Nachbemerkung zu Spremberg

Das Clubtreffen fand am 21. 4. dieses Jahres statt. Wir haben heute den 5. 7. 90. Selther hat sich allerhand geändert, verbessert. "Ossi-Raten", hier in Lübeck, direkt an der Grenze, noch vor kurzem ein populärer Sport, ist schwierig geworden: Die verräterische blaß-zweifarbige Windjacke aus Dederon, die Prinz-Heinrich-Mütze (dort ausgerechnet "Schmidt-Mütze" genannt), der Trabbi und die fünf prallvollen Plastetüten sind immer seltener das ehemals unveränderliche Kennzeichen der Brüderundschwwestern.

Auch auf dem technischen, natürlich auch dem digitalen Sektor hat sich eine Menge getan. Die DM hat offiziell Einzug gehalten, nachdem sie schon seit Monaten inoffizielles Zahlungsmittel für "Bückware", Schmierstoff für Handwerker und Eintrittskarte für die Intershops mit Westware war. (Dabei fällt mir ein, daß Schalck-Intershopski immer noch frei in der BRD herumläuft, denn er hat ja nicht falsch geparkt.)

Zur Sache: Wir dürfen uns auf die 15 neuen Clubmitglieder aus der DDR freuen. Nicht nur, daß viele von ihnen jetzt einen neuen, einen "richtigen" Computer kaufen werden, der besser in die Hardware-Landschaft des Club-80 paßt. Sie haben auch vier Jahrzehnte lang trainiert, aus nichts Erstaunliches zu zaubern. Auf ihre Bastel- und Programmieranleitungen, die wir demnächst im Info lesen werden, kann man gespannt sein.

Auf unserer eigenen Jahreshauptversammlung haben wir beschlossen, daß die 15 Interessierten aus Franks Dunstkreis und jeder andere DDR-Bürger, der dem Club beitreten will, für ein Jahr beitragsfrei geführt wird.

Arnulf

*Eine Fotosseite zum Spremberg-Treffen
ist am Ende dieses INFÜ's beigelegt.*

HEFT
31
Juli
1990

07



08



Unser Erster und wir.
Einige Eindrücke vom letzten Clubtreffen in Weilburg.



Turbo-Pascal und ZCPR

Turbo-Pascal auf der einen Seite ist neben dem Assembler in CP/M-Kreisen wohl die am weitesten verbreitete Programmiersprache und liefert eine ganze Menge mächtige Werkzeuge, die vom direkten Systemzugriff bis zu informatischen Höhenflügen ziemlich alles abdecken. Auf der anderen Seite steht das ZCPR-System, das mit seinen Suchpfaden, Shells und Error-Handlern usw. alles das bietet, was einem zum vollkommenen Glück noch fehlen kann. Leider hat Turbo keinen blassen Schimmer, in welcher komfortabler Umgebung es laufen kann, aber dem kann abgeholfen werden.

Zuerst kurz ein paar Grundlagen. Ein ZCPR-Utility ist dadurch gekennzeichnet, daß der Code am Anfang folgendermaßen aussieht:

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
100 C3 xx xx Z 3 E N V 01 00 00 xx xx xx xx xx
```

Bei Adresse 100 steht ein Sprung, der in das eigentliche Programm führt und von 103-107 steht genau dieser Text als Erkennungsmerkmal. Das Byte bei 108 gibt an, um welche Art Programm es sich handelt, 01 bedeutet hier 'externes Utility', sprich gewöhnliches COM-File. Beim Laden des Programms schreibt der Lader in die Bytes 109 und 10A die Adresse des sog. Environments und da beginnt die Sache interessant zu werden. Wenn man nämlich diese Adresse kennt, sind alle anderen Adressen, wo z.B. der Suchpfad, die Prozessorgeschwindigkeit, die Steuerzeichen für das verwendete Terminal stehen, auch zugänglich.

Turbo-Pascal kommt einem in dieser Beziehung in bemerkenswerter Weise entgegen, hinter dem Sprung bei 100 kommt nämlich gleich eine Copyright-Notiz, die entsprechend überschrieben werden kann. Wem das nicht gefällt, der kann sich auch ein kleines Alias (eine Art SUBMIT) bauen, das den Code erst beim Laden patcht, sodaß das Copyright unverletzt bleibt:

```
GET 100 TURBO.COM          <- lädt Turbo
POKE 103 5A 33 45 4E 56 01 00 00 <- macht Patches
GO                          <- startet Turbo
```

Im Pascal-Programm selber kann man nun mit

```
VAR Z3EADR : Integer Absolute $109;
```

auf diese Adresse zugreifen und nach Lust und Laune mit den Möglichkeiten des ZCPR arbeiten. Um auf die einzelnen Werte einfacher zugreifen zu können, ist ein Record eine ganz brauchbare Lösung. Das sieht dann so aus:

```
TYPE ENVREC = Record
  env       : Byte;
  cbios     : Integer;
  z3id      : Array $1..50 of Char;
  envtyp    : Byte;
  expath    : Integer;
```

```
  expaths : Byte;
```

```
VAR ZENV : ^ENVREC;
```

```
ZENV := Ptr(Z3EADR);
```

Mit WriteLn('Länge des Pfads = ', envrec^.expaths); kann man sich z.B. bequem die Länge des Suchpfades ausgeben lassen. Damit kann man Turbo endlich beibringen, Files auf mehreren Laufwerken zu suchen. Natürlich kann man durch Schreiben auch gezielt Eingriffe ins System vornehmen, z.B. den Suchpfad so zu verändern, wie man ihn gerade braucht oder die diversen Flags bearbeiten.

Das Grundgerüst für die Zusammenarbeit von Turbo-Pascal mit ZCPR ist in einer Library namens NZ-TURBO.LBR zu finden, die auf zahlreichen PD-Disketten im Umlauf sein dürfte. Wenn nicht, Rudiger hilft bestimmt gerne weiter (ich übrigens auch).

Hier noch ein kleines Demoprogrammchen, wie man den Suchpfad abklappern kann.

```
(* Testprogramm fuer den Zugriff auf das ZCPR-Environment *)
```

```
(* $I NZ-TURBO.CON *)
(* $I NZ-TURBO.TYP *) (* hier stehen die Records usw. drin *)
(* $I NZ-TURBO.VAR *) (* " " " " " " *)
(* $I HEX.FUN *)
```

```
TYPE pointer=File;
  filename=String$130;
```

```
VAR aktiw : Integer;
  suchlw : Char;
  laufw : String$170;
  start : Integer;
  p_elem : Integer;
  fileptr : pointer;
  fehler : Integer;
```

```
FUNCTION Oeffnen(name:filename;laufw:Char):Integer;
BEGIN
```

```
  (* $I-*)
  Assign(fileptr,suchlw+'_'+name);
  Reset(fileptr);
  (* $I+*)
  Oeffnen:=IOResult;
END;
```

```
BEGIN
  envptr:=Ptr(z3eadr);
```

```
  Write('Path-Adr. : ',envptr^.expath,' = ');
```

```
WriteLn(HEX(envptr^.expath,1),'H');
WriteLn('Path-Len. : ',envptr^.expaths);
```

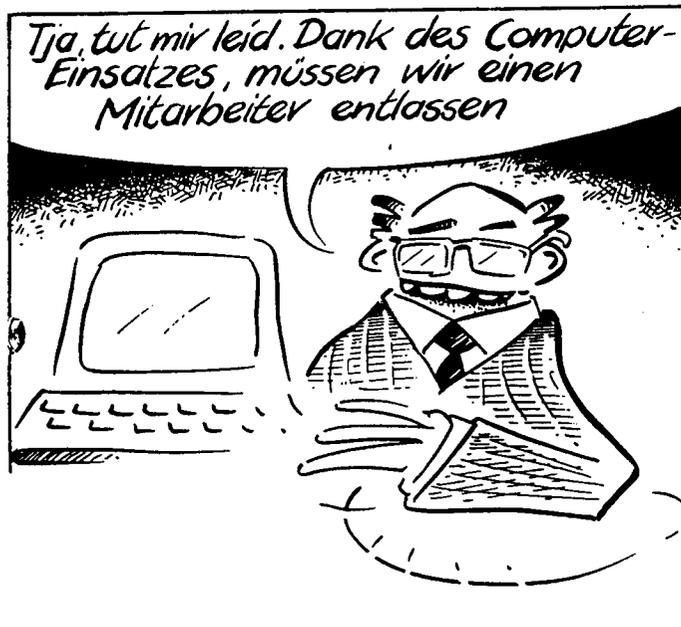
```
laufw:='ABCDEFGHJKLMNOP';
aktlw:=Bdos($19)+1;
WriteLn('Aktuelles Laufwerk: ',LaufwAktlwU);
```

```
WriteLn('Suchpfad      : ');
start:=envptr^.expath-2;
```

```
p_elem:=aktlw;          (* zuerst aktuelles Lw durchsuchen *)
WHILE (p_elem<>0) DO    (* Ende des Suchpfades *)
BEGIN
  IF (p_elem<>36) THEN  (* $ bedeutet auch akt. Lw im Pfad *)
  BEGIN
    suchlw:=laufwAp_elemU;
    fehler:=0effnen('SYS.ENV',suchlw);
    Write('Laufwerk ',suchlw,':');
    IF fehler=0 THEN WriteLn('gefunden') ELSE WriteLn('nix');
  END;
  start:=start+2;
  p_elem:=MemAstartU;
END;
END.
```

Eine weitere Arbeitserleichterung ist der mögliche Zugriff auf die auf dem jeweiligen Terminal vorhandenen Steuerzeichen. Damit kann sich jedes Programm vollautomatisch selber installieren und die ewige Sucherei und Tipperei hat ein Ende.

Alexander Schmid



Ein paar Computer in unserem Club laufen inzwischen nicht mehr mit dem guten alten Z80, sondern einem seiner Nachfolger HD64180 oder Z180. Außerdem läuft gerade ein Selbstbauprojekt an, bei dem der Z280 das Herz bildet.

Diese neuen Mikroprozessoren haben einige Befehle, die der Z80 noch nicht kannte. Ihre Codes, die sog. Illegals, befolgte Jener zwar auch, aber die neueren CPUs stellen damit etwas anderes an. Helmut hat das im letzten Info eingehend beschrieben und in einer Tabelle zusammengefaßt. Deshalb sei es hier nicht mehr erklärt.

Diese Befehle haben, ihrer anderen Funktion gemäß, neue Namen. Und die sind ZEUS leider unbekannt.

Bisher schrieb ich diese Befehle umständlich in der folgenden Form:

```
00029 ;      OTDMR
00030         DB          0edh,9bh
```

Der Befehl mit seinem richtigen Namen, den aber ZEUS nicht kennt, steht in einer Kommentarzeile. Die nächste Zeile enthält ausführbaren Code, der aber leider als DB (DEFB) eingegeben werden muß (dieser Zwei-Byte-Befehl hätte auch DW 9bedh lauten können).

Dessen leid, schrieb ich nun einfach alle neuen Befehle in eine kleine Library von Hilfslabels (s. Listing). Dort stehen sie als EQUates. Sie können in den ZEUS-Editor geladen werden. In der Source erscheinen nun die Befehle mit ihrem richtigen Namen, aber ohne sie mit dem Semikolon als Kommentare entschärfen zu müssen:

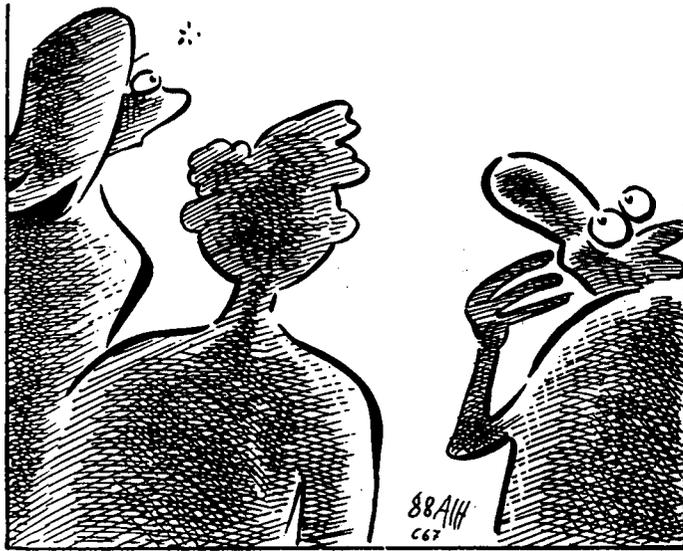
```
00029         DW          otdmr
```

Einige der neuen Befehle bestehen aus 3 Bytes. Da ist es mit einem DEFW leider nicht mehr getan. Das dritte Byte, in allen Fällen die Angabe eines I/O-Ports, muß dann doch noch in einer weiteren Zeile als DEFB-Statement nachgereicht werden (s. Listing). Dennoch wird die Eintipperei damit immer noch etwas vereinfacht gegenüber der alten Methode.

Natürlich ist das auch nur eine Krücke. Eleganter wäre es, den Befehl OTDMR in ZEUSens Befehlsvorrat einzubauen. Vor Jahren haben Gerald und ich das mal versucht. Es ist aber reichlich kompliziert, deshalb blieb die Arbeit irgendwann liegen. So geht es schließlich auch.

Arnulf

```
00001 ;Hilfslabels für Befehle des Z180 bzw. des HD64180, die ZEUS nicht kennt
00002
00003 ;Inputs in Register q vom I/O-Port n in der I/O-Zero-Page
00004 ;(die Angabe des Ports muß als DEFB-Statement folgen)
00005 in0_b EQU 00edh
00006 in0_c EQU 08edh
00007 in0_d EQU 10edh
00008 in0_e EQU 18edh
00009 in0_h EQU 20edh
00010 in0_l EQU 28edh
00011 in0_f EQU 30edh
00012 in0_a EQU 38edh
00013
00014 ;Outputs aus Register q auf den I/O-Port n in der I/O-Zero-Page
00015 ;(die Angabe des Ports muß als DEFB-Statement folgen)
```



```

00016 out0_b EQU 01edh
00017 out0_c EQU 09ech
00018 out0_d EQU 11ech
00019 out0_e EQU 19e2h
00020 out0_h EQU 21ech
00021 out0_l EQU 29ech
00022 out0_a EQU 39ech
00023
00024 ;(wiederholte) Block-Outputs mit Inkrement des Ports in der I/O-Zero-Page
00025 otir EQU 83edh
00026 otimr EQU 93edh
00027 otdm EQU 8beah
00028 otdmr EQU 9bedh
00029
00030 ;Multiplikationen (wwL * wwH = ww)
00031 mlt_bc EQU 4cedh
00032 mlt_de EQU 5cedh
00033 mlt_hl EQU 6cedh
00034 mlt_sp EQU 7cedh ;SPL * SPH = SP
00035
00036 ;non-destruktives AND
00037 tst_b EQU 04edh
00038 tst_c EQU 0cedh
00039 tst_d EQU 14edh
00040 tst_e EQU 1cedh
00041 tst_h EQU 24edh
00042 tst_l EQU 2cedh
00043 tst_hl EQU 34edh ;A AND (HL)
00044 tst_a EQU 3cedh
00045 tst_m EQU 64edh
00046 tstio EQU 74edh ;(C) AND m (I/O-Port m als DEFB-St. anzufügen)
00047
00048 ;SLEEP-Befehl
00049 slp EQU 76edh
00050

```



Häufig kommt es vor, daß ein Programm nicht gleich dorthin geladen werden kann, wo es laufen soll. In günstigen Fällen kommt man darin mit relativen Sprüngen aus, die an jeder Adresse ohne Änderung funktionieren. Denn sie erstrecken sich über eine Sprungdistanz, sie peilen kein unveränderliches Sprungziel an.

Gibt es jedoch JPs, CALLs oder Speicher-Ladebefehle, ist die Adresse leider ORG-abhängig. Wenn also eine Routine unter G-DOS an 5200h oder unter CP/M an 0100h geladen wird, dort aber gar nicht laufen soll, müssen diese Adressen relociert, also der endgültigen Ladeadresse angepaßt werden.

ZEUS leistet für einfachere Fälle eine gute Hilfe: Da mit seinen Labels ein paar logische und mathematische Operationen erlaubt sind, kann beispielsweise ein Offset eingeführt werden:

```

offset    EQU    38f4h
          LD     A, (byte-offset)
          ...
byte      DB     00h
    
```

Das Byte "byte" steht nach dem Laden von Diskette meinetwegen an 5FD3h, soll aber später 38F4h Bytes tiefer liegen. Der Offset mit dem Label "offset" wird von ZEUS korrekt mit der ersten Ladeadresse von "byte" verrechnet.

Schwieriger ist es, wenn ein und dasselbe Programm an mehreren Stellen laufen soll, z. B. in mehreren Banks an verschiedenen Adressen. Genau dieser Fall trat jetzt bei mir auf, als ich eine TRAP-Routine für den Z180 sowohl in Bank 0 im unbenutzten BASIC-Bereich von G-DOS, als auch in Bank 1 ganz unten ab 0000h bereitlegen wollte.

Die kritischen Adressen habe ich zunächst mit den Labels addr1 - addr15 sowie dstlog versehen. Diese Adressen wiederholten sich dann in einer Relokationstabelle (s. Listing). Danach brach ich mir fürchterlich einen ab, um mit HL als Quellzeiger, DE als Zielzeiger, BC als Hilfszeiger, Helmut als Bergsteiger usw. die Adreß-Operanden der Befehle mit den neuen Werten zu beglücken. Es entstand ein umfängliches Programm, bei dem es mir einfach um den Speicherplatz leidtat. Das ging dann wiederum mir auf den Zeiger.

In der jetzigen Form ist die Routine erfrischend kurz: Zweimal wird der Stack verbogen. Zunächst wird er in die Tabelle der zu relocierenden Adressen gelegt. Mit POP gelangt die Adresse nach HL. Der Stackpointer hat sich dabei auf die nächste Adresse in der Tabelle erhöht. Er wird nun aus HL mit der Programmstelle geladen, an der eine Adresse relociert werden soll. Was dort bereits steht, wird wieder mit POP erforscht. Die Subtraktion mit dem zweiten Offset (SBC HL,DE) ergibt den neuen Wert. Der wird mit PUSH gepatcht. Fertig.

Der ursprüngliche Stack und der zweite in der Relokationstabelle müssen natürlich gerettet werden, um den Super-Soft-GAU zu vermeiden. Leider existiert dafür nur ein Speicherladebefehl, was etwas unelegant ist (LD (nn),SP) und ein paar Nanosekunden länger dauert als Registerladebefehle. Da es in meinem Programm aber nur 16mal vorkommt, kann man damit gut leben.

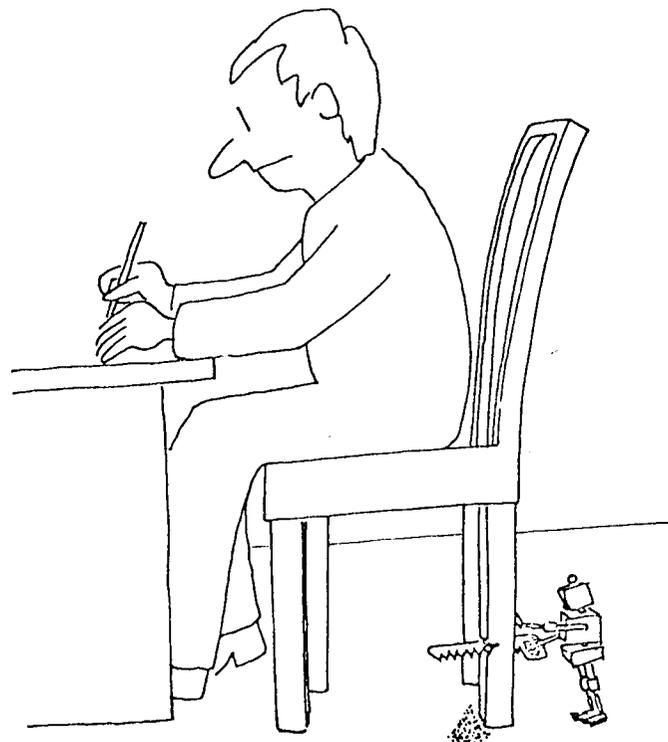
Die TRAP-Routine ist die von Gerald. Da sie hier nicht (sonst aber sehr) von Interesse ist, habe ich im Listing fast alles davon mit LIST OrF ausgespart. Nur ein wenig Code mit den Labels addr... ist übrig. Das erkennt man an den springenden Zeilennummern. Auch in meinem eigenen Programm ist an dieser Stelle vieles nicht von Belang. Der gelistete Teil stellt lediglich die Relokation vor.

Arnulf

```

5200      00001      ORG      5200h
          00002
          00037 target EQU      1650h      ;im BASIC-Bereich (vorläufig)
38CF      00038 offs  EQU      $-target    ;Abstand zur Arbeitsadresse
5260 328016 00097 gotrap LD      (const-offs),A ;evtl. Konstante in den Programmtext
526E      00098 addr5 EQU      $-2
5270 78      00099      LD      A,B        ;DD/FD zurück
5271 32AC16 00100      LD      (ddfd1-offs),A ;in den Programmtext setzen
5272      00101 addr6 EQU      $-2
5274 32B216 00102      LD      (ddfd2-offs),A ;dort auch
5275      00103 addr7 EQU      $-2
530B 115016 00204      LD      DE,target    ;Offset der Adressen von target zu 0000
530E 0610    00205      LD      B,endtab-addrtab/2 ;Anzahl zu relocierender Adressen
5310 ED732753 00206      LD      (spbuff2),SP ;Stack retten
5314 315253 00207      LD      SP,addrtab    ;Stack auf Adressentabelle setzen
5317 E1      00208 addrlop POP     HL        ;neue Adresse abholen
5318 ED732253 00209      LD      (spbuff1),SP ;Tabelle als Stack retten
531C F9      00210      LD      SP,HL        ;dortige Adresse = Stack
531D E1      00211      POP     HL        ;dortiges Word abholen
531E ED52    00212      SBC     HL,DE      ;auf Basisadresse 0000 umrechnen
5320 E5      00213      PUSH    HL        ;neues Word ablegen
5321 310000 00214      LD      SP,0000h     ;Stack wieder in der Tabelle
5322      00215 spbuff1 EQU      $-2
5324 10F1    00216      DJNZ   addrlop     ;bis alle Adressen bearbeitet sind
5326 310000 00217      LD      SP,0000h     ;alten Stack restaurieren
5327      00218 spbuff2 EQU      $-2
          00256
          00257 ;Tabelle der zu relocierenden Adressen in der TRAP-Routine
5352 2852    00258 addrtab DW     addr1,addr2,addr3,addr4,addr5,addr6,addr7,addr8
5362 8952    00259      DW     addr9,addr10,addr11,addr12,addr13,addr14,addr15,dstlog
          00260 endtab
    
```

00000 Fehler



Harddisk-Einbindung ins CP/M3-BIOS

hier am Beispiel des OMTI 5527 RLL Controllers und einer Platte mit 615 Zylindern und 4 Köpfen (32MB im RLL-Format)

Helmut Bernhardt, Peter Petersen

Die Hardware

Der Aufwand in Hardware, um einen PC-Harddisk-Controller wie die verschiedenen OMTI-Typen an den ECB-Bus oder auch an andere Z80-Bus-Systeme anzuschließen, ist minimal. Beim ECB-Bus bestehen die Probleme hauptsächlich in der ungeeigneten Geometrie des Controllers, der die Breite der Europakarten von 10cm überschreitet. Die Montage auf eine ECB-Bus-Trägerkarte, die auch die beiden zur Busanpassung nötigen ICs aufnimmt, muß für jedes Gerät individuell gelöst werden.

Die Busanpassung beschränkt sich auf die Umverlegung der ECB-Adressen an die Slot-Adressen des Controllers, damit bei einem Zugriff der CPU unter einer für den Controller vorgesehenen Portadresse am Controller die Portadresse 320H(-323H), unter der der Controller im PC decodiert ist, anliegt. Außerdem müssen aus den ECB-Signalen /RD, /WR und /IORQ die PC-Bus-Signale /IOR und /IOW mit zwei OR-Gattern eines 74LS32 erzeugt werden. Auch das low aktive /RESET des ECB-Bus muß über einen 74LS04-Inverter in das high aktive RESET des PC-Bus umgesetzt werden.

Da 4 aufeinanderfolgende Portadressen benötigt werden, müssen A0 und A1 direkt an den Controller gelangen. Für eine Adreßlage bei DOH-D3H am ECB-Bus ist in Abb.1 eine Beschaltung gezeigt, die folgende Pegelübersetzung bewirkt:

Adresse	:	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ECB Port DOH	:	-	-	1	1	0	1	0	0	x	x
PC Port 320H	:	1	1	0	0	1	0	0	0	x	x

Außerdem ist an AEN und A10-A19 des Controllers GND und an /DACK3 und /MEMR +5V zu legen. Außer dem OMTI 5510 benötigen alle OMTI-Controller am Slot-Pin b9 +12V Versorgungsspannung. Selbstverständlich müssen die Daten D0-D7 an die gleichen Eingänge des Controllers gelegt werden.

Die Software

Wesentlich anspruchsvoller ist die Software-Einbindung der Festplatte in das BIOS des CP/M plus.

Mit den unter CP/M bestehenden Möglichkeiten zur Verwaltung der Files auf einer Platte (Unterteilung in User 0 bis User 15) lassen sich nur begrenzte Mengen an Files übersichtlich handhaben. Deshalb ist es bei größeren Platten sinnvoll, diese in mehrere logische Laufwerke zu partitionieren. Noch handhabbar sind dabei Größen von einigen Megabyte Kapazität. Danach könnte eine 32MB-Platte in 4 Partitionen aufgeteilt werden. Die sich anbietende Aufteilung, daß alle Spuren unter einem Kopf als ein Laufwerk behandelt werden, bedeutet, daß nach je 26 Sektoren der Kopf eine Spur weiterbewegt werden muß. Der Spurwechsel ist bei Festplattenzugriffen aber die zeitraubendste Aktion und sollte so wenig wie möglich vorkommen.

Deshalb wurde die Partitionierung hier auch anders durchgeführt: Eine logische Spur umfaßt alle 4 Spuren (Köpfe) eines Zylinders. Es werden nacheinander erst die 4 Spuren unter allen 4 Köpfen gelesen/geschrieben, bevor der Kopfschlitten um eine Position weiterbewegt wird. Dadurch können nacheinander 104 Sektoren verholt werden, ohne zwischendurch Steppen zu müssen. Einer Partition wird dabei ein zusammenhän-

gender Bereich von Zylindern zugeordnet; die Nummer des letzten absoluten Zylinders einer Partition ist die Anzahl Spuren und die Nummer des letzten absoluten Zylinders der vorherigen Partition ist die Anzahl reservierter Systemspuren.

Spur-Nr.	0	1	154	307	460	615
Partit. 0	-----					
Partit. 1	-----					
Partit. 2	-----					
Partit. 3	-----					

----- reservierte Systemspuren
 ===== Datenspuren der Partition

Das BDOS kennt für Disk-I/O nur die Adresse einer Tabelle mit dem Namen @dtbl (drive table). In dieser Tabelle stehen die Adressen von 16 Disk Parameter Headern (DPHs) für 16 Laufwerke in der Reihenfolge ihrer logischen Namen (A, B, ...P). Nicht existierende Laufwerke erhalten hier den Eintrag 0. Wenn in einem Sourcecode-Modul nur die Drive Table und die DPHs und DPBs für RAM-Floppy und Harddisk enthalten sind, könnte das so aussehen:

```
public @dtbl ; Drive Table
public fdsd0, fdsd1, fdsd2, fdsd3 ; dph's fuer 4 Floppies
public fdram ; dph RAM-Floppy
public hsk0, hsk1, hsk2, hsk3 ; dph's fuer harddisk 0 - 3
```

```
extrn hdlogi, hdinit, hdlinit ; Harddisk Routinen
extrn hdwrit, hhread
```

```
maclib xcpm3 ; mit hdpb-Macro fuer grosse Drives
```

```
cseg ; Drivetable und dpb's im Common
```

; Drive Table (zeigt auf die DPBs der verfügbaren Laufwerke)

```
@dtbl dw fdsd0, fdsd1, fdsd2, fdsd3 ; Floppy Drives
dw fdram ; RAM-Floppy
dw hsk0, hsk1, hsk2, hsk3 ; Harddisk
dw 0,0,0,0,0,0 ; nicht benutzt
```

; Die Einträge in @dtbl weisen auf Adressen der DPHs, wobei die Labels nicht am Anfang der xDPHs stehen. Der Aufbau der xDPHs sieht so aus:

```
dseg ; die DPHs koennen in gebanktem RAM stehen
```

; extended disk parameter header (xDPH) der Harddisk-Partitionen

```
drva: dw hdwrit ; Routine zum Schreiben
dw hhread ; und Lesen eines Sektors
dw hdlogi ; Einloggen des Drives
dw hdinit ; Initialisierung des HDC
db 0 ; relative drive 0
db 0 ; media flag : not used
hsk0 dph 0, hddpb0 ; Skew-Tabelle und xDPB
```

```
drvb: dw hdwrit, hhread, hdlogi
dw hdlinit ; kann auf RET zeigen
db 1 ; relative drive 1
db 0 ; media flag : not used
hsk1 dph 0, hddpb1
```

```
drvc: dw hdwrit, hhread, hdlogi
dw hdlinit ; kann auf RET zeigen
```

HEF
 31
 Juli
 1990

18

wie der von beiden genutzte Ausgang 'termcmd1', der wieder die Bank 0 einschaltet, machen zusammen knapp 60 Bytes des kostbaren Common-Speichers aus, dem 512 Bytes eines Zwischenspeichers bei der anderen Variante gegenüberstehen.

Bei einem Prof mit 9.2MHz Systemtakt ist damit ein Interleave 1:1 möglich. Wegen insgesamt bereitgestellter Data-Buffer von 64K RAM erscheint ein erster Zugriff auf die Festplatte recht langsam, weil dann immer gleich riesige Datenmengen bewegt werden. Nachfolgende Zugriffe können dann aber aus dem Buffer leben und erreichen dann die Geschwindigkeit einer RAM-Floppy.

Weiteres Tuneup am BIOS

Da die Speicheraufteilung des Prof 180 recht ungünstig ist und dadurch bedingt auch die RAM-Floppy-Routinen umständlich mit '?xmove' über einen 128 Byte Zwischenspeicher im common arbeiten, wurde auch die ganze Speicherkonfiguration nach Conitec's Vorstellung aufgegeben und durch eine sinnvollere ersetzt.

Speicherbereich	Adreßlage nach Conitec	neue Adreßlage
Systembank	00000h-0efffh	00000h-0efffh
Common	0f000h-0ffffh	0f000h-0ffffh
TPA	40000h-4efffh	10000h-1efffh
CCP-Buffer	4f000h-4ffffh	1f000h-1ffffh
weitere Banks	-----	20000h-2ffffh
RAM-Floppy	10000h-3ffffh	30000h-7ffffh
	+ 50000h-7ffffh	

Die ursprüngliche Einteilung von Conitec ermöglicht den Einsatz von 4164-RAMs, wobei die Selektion der beiden 64K-Blöcke durch den Pegel von A18 erfolgt (was bei 41256-RAMs auch sinnvoll ist). Entsprechend liegen diese beiden Blöcke auf den physikalischen Adressen 0xxxxh und 4xxxxh. Da bei den heutigen RAM-Preisen der Einsatz von 4164-RAMs unwahrscheinlich ist und hier 41256-RAMs stecken, wurde auf die Möglichkeit des Abrüstens auf 128K verzichtet, so daß der Bereich, der für Banks und/oder RAM-Floppy zur Verfügung steht, zusammenhängt. Das vereinfacht die RAM-Floppy-Routinen gewaltig.

Der Zugriff auf die RAM-Floppy erfolgt nun im DMA-Betrieb und zur Vereinfachung (und Beschleunigung) wurde die Sektorgroße von 128 auf 1024 Bytes erhöht und die Größe einer Spur auf 64K festgelegt. Die Track-Nr kann dann direkt (durch Addition des Offsets des RAM-Floppy-Anfangs) in das Adreß-Zusatzbyte für den DMAC umgesetzt werden und aus der Sektor-Nummer kann durch Shiften um 2 Bits der Adreßanteil A15-A10 erhalten werden; A9-A0 sind immer 0 für die Anfangsadresse eines Sektors.

Die RAM-Floppy-Routinen sind in RFLO.MAC versammelt. DPH und DPB der RAM-Floppy sowie die Routinen und Tabellen für Bankumschaltung und DMA sind neben den DPHs und DPBs für die Festplatte in MOVE.ASM enthalten.

Durch die neue Speichereinteilung muß GENCPM bei der Erzeugung der Memory Segment Table auch mit anderen Parametern versorgt werden, als den in GENCPM.DAT bisher enthaltenen. Bei 64K Data-Buffer sind insgesamt 3 Segmente anzugeben (Common und TPA sowie CCP-Buffer werden hier nicht mitgezählt). Für jedes Segment sind dann je 3 Bytes anzugeben, die folgende Bedeutung haben:

- 1) high Byte der logischen Anfangsadresse der Bank
- 2) high Byte der Länge der Bank
- 3) Bank-Nummer

Eine Angabe '00,80,02' bedeutet, daß die Bank Nr 2 von 0000h bis 7fffh

reicht (logische Adressen). Wo diese Bank physikalisch im Speicher liegt, spielt hier keine Rolle.

Für die Bank 0 sind diese Angaben (speziell beim Prof 180) etwas komplizierter. Der Prof 180 benutzt 4KByte des Boot-ROMs (bzw. des auf gleiche Adressen ins RAM kopierten ROM-Inhalts) vom BIOS des CP/M aus. Deshalb muß für die Systembank als high Byte der Anfangsadresse 10 angegeben werden, damit diese ROM-Routinen nicht überschrieben werden. Als Länge ist die Differenz des von GENCPM vorher ausgegebenen Anfangs des gebankten BDOS abzüglich dieser Adresse anzugeben. Die Bank Nr ist natürlich 00.

Um unseren CP/M-Diskothekar nicht mit Format-Konvertier-Problemen zu überlasten, bitte ich darum, bei Interesse an den BIOS-Sources für die Festplatte und die neue Variante der RAM-Floppy mir eine formatierte Diskette mit den gerade verfügbaren Angaben zum Format und möglichst einigen Text-Files darauf (damit ich prüfen kann, ob ich das Format richtig eingestellt habe) zu schicken. Beiliegendes Rückporto wird auch gerne gesehen. Helmut

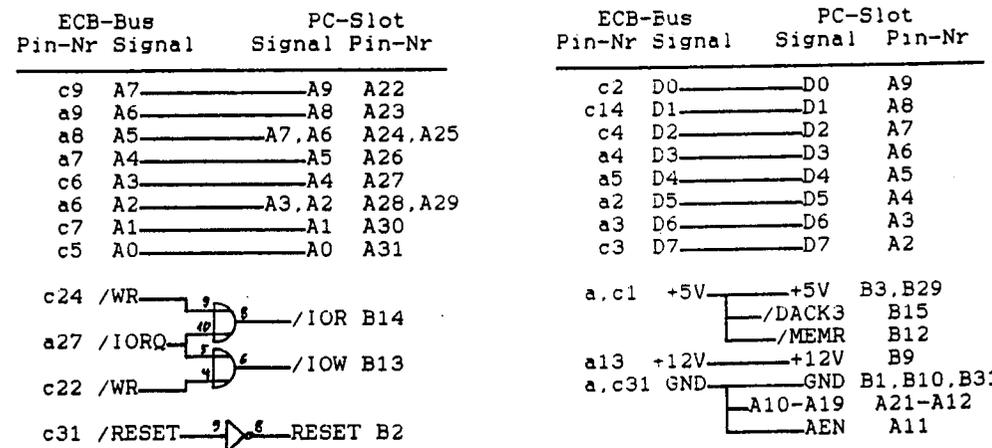
Zum Thema Harddisk unter CP/M geben folgende Artikel etwas her:

Andreas Zippel 'Hart, schnell und sicher', c't 8/86, S.52ff; c't 9/86, S.105ff; c't 10/86, S.144ff

Matin Rost, Detlef Grell 'Harddisk-Controller einmal anders', c't 4/87, S.138ff; c't 9/87, S.138ff

Christian Persson 'OMTI-Alternative', c't 5/89, S.244

Anschluß des OMTI-Controllers 5527 an den ECB-Bus



HEFT
31
Juli
1990

Nachdem im vorletzten INFO mein Artikel erschienen ist, habe ich mit einer Flut von Zuschriften gerechnet. Dem war leider nicht so. Dem konnte ich entnehmen, daß Ihr alle sehr schreibfaul seid, oder kein Interesse an Projekten habt. Ich will hoffen, daß es der erste Grund ist. Aber immerhin habe ich ja zwei Zuschriften bekommen. Aber vielleicht waren euch ja die Angaben auch zu dürftig. Jetzt sieht es anders aus. Auf dem Clubtreffen habe ich mich mit Helmut über eine Grafikkarte unterhalten. Wir sind zu dem Entschluß gekommen es so einfach wie möglich zu machen und vor allem das Rad nicht neu zu erfinden. Man kann nicht wiederlegen, das auch IBM seine Vorteile hat, auch wenn es nur der ist, dafür gesorgt zu haben 'billige' Grafikkarten auf den Markt zu schmeißen. Und diesen Vorteil wollen wir nutzen.

Aber nun im einzelnen. Wir haben beschlossen einen Grafikkadappter zu entwickeln. Dieser hat die Aufgabe eine Herkules- oder EGA-Karte über Port's an jeden Rechner anzuschließen. Der Adapter wird 'Intelligent', d.h. er bekommt eine eigene CPU. Alles was zur Ansteuerung der Grafikkarte notwendig ist, erledigt ein Z-80, genauso wie die Kommunikation mit dem HOST-Rechner.

Die Kommunikation mit der Karte erfolgt über zwei Port's, einer lesend, einer schreibend, die mit ein paar Bauteilen leicht in jeden Rechner einzubauen sind. Über diese Port's werden Daen und Befehle ausgetauscht. Die Adapterkarte enthält eine CPU, 32Kb RAM, 32Kb ROM sowie die Anschaltung der Grafikkarte. Es wird also nicht zu viel. Für ganz eilige: Die genaue Schaltung ist noch nicht fertig, weil Helmut und ich noch ein genaues Konzept ausarbeiten müssen, aber das ist jetzt auch noch nicht ganz so schlimm, weil noch ein paar andere Dinge festzulegen sind, die aber die Software angehen. Wer aber trotzdem mit Rat und Tat zur Seite stehen will, wende sich bitte an Helmut oder mich, auch was die Zoffware angeht.

Aber nun zu der Karte selber. Wir haben uns zwar schon ein paar Gedanken gemacht was das Teil können muß, aber wie schon gesagt, es ist noch in der Entwicklung, und außerdem sollt Ihr eure Meinung auch dazu beitragen. Die Liste erhebt keinen Anspruch auf Vollständigkeit, sondern ist ein erster Überblick. Wenn Ihr eigene Ideen habt, laßt sie mich wissen, auch wenn Ihr nicht wißt wie sie zu bewerkstelligen sind, denn darüber machen wir uns dann später Gedanken.

- Programm Download (Eigene Programme die die Karte erledigt)
- Punkt setzen
- " löschen
- " testen
- Linie ziehen mit Farbangabe
- Viereck zeichnen mit Farbangabe
- Fläche ausfüllen mit Farbangabe
- Kreis zeichnen
- Schirm invertieren

- Bereich invertieren
- Fläche invertieren
- Bereichsfarbe ändern
- Flächenfarbe ändern
- Bereich spiegeln
- Bereich drehen
- Bereich verschieben
- Objekt verschieben
- ASCCI anzeigen
- Zeichensatz laden
- Terminalfunktion
- Letzten Befehl wiederholen
- Letzten Befehl zurücknehmen
- HRG auslesen (zum abspeichern)
- HRG einschreiben (von Disk)
- HRG mischen (MERGE)
- beliebigen CURSOR erzeugen
- CURSOR bewegen (plazieren)
-

Das ist das, was mir auf die schnelle so einfiel. Ich will es erstmal dabei belassen, um mir über den genauen Aufbau Gedanken zu machen. Vielleicht gibt es im nächsten INFO schon Schaltungen oder Programme dazu. Aber wie gesagt, Eure Mitarbeit und Eure Anregungen sind ausdrücklich erwünscht.

Bis bald, Euer Hardware-Koordinator.



Club 80 Projekt:

Bau einer Terminal-Karte auf Basis der PC-Hercules-Karte

Der GENIE 1 mit Hercules-Karte als Video-Interface ist ja ganz witzig aber für die meisten Club-Mitglieder uninteressant, weil die Strippenzieherei einfach zu aufwendig ist und die Festlegung auf die GENIE-Kompatibilität zu beengend ist.

Um das günstige Preis/Leistungs-Verhältnis der Hercules allgemeiner zugänglich zu machen, wurde angeregt, als Projekt im Club 80 eine Terminal-Karte zu bauen, die zusammen mit der Hercules in jedem Computer als Konsole eingesetzt werden kann. Das Terminal soll mit Z80-Power und entsprechender Betriebssoftware fast ein eigenständiger Text-/Grafik-Computer sein, der in Arbeitsteilung mit der CPU Video-Ausgaben erledigt.

Da eine EGA- oder VGA-Karte ähnlich wie die Hercules-Karte funktioniert (erweiterter I/O-Adreßbereich, anderer Memory-Adreßbereich), ist das Prinzip auch auf diese Karten anwendbar. Es bestehen dort aber bei den über die IBM-Vorgaben hinausgehenden Features unterschiedliche Lösungen, die nicht ausreichend dokumentiert sind. Außerdem benötigen diese Karten Monitore, deren Preise weit über den Aktionsradius des Bastlers hinausgehen, so daß wir uns zunächst auf die Hercules-Karte beschränken sollten. Wenn wir das im Griff haben, können wir immer noch überlegen, ob wir nach den Sternen greifen wollen.

Von Andreas Magnus habe ich als letzte Meldung vor längerer Zeit erfahren, daß er auch schon einen entsprechenden Artikel ins Info gesetzt hat; ich weiß allerdings nicht, was er dort schon angekündigt hat. Deshalb möge man mir nachsehen, wenn einige Aussagen hier zum zweiten mal erscheinen. Da inzwischen schon etwas festere Vorstellungen bestehen, ist dieser Beitrag aber unumgänglich.

Die Kommunikation der Terminal-Karte mit der CPU soll über einen bidirektionalen Parallelport (Datenaustausch) und einen Statusport (Handshaking) erfolgen. Zwischen CPU und Terminal werden Befehle der CPU an das Terminal vereinbart, die das Terminal zur selbständigen Bearbeitung der Hercules anregen. Die bisherige Vorstellung sieht dabei im Textmodus die Emulation eines Televideo 950 Terminals und im Grafikmodus die eines Tektronix 4014 Grafikterminals vor.

Während für die Hardware des Terminals schon feste Vorstellungen bestehen, ist für die Betriebssoftware noch alles offen. Zur Erstellung eines Betriebsprogramms für das Terminal sind alle Club-Mitglieder herzlich eingeladen. Der Umfang der zu erstellenden Software ist so groß, daß jedes Gramm Gehirnschmalz im Club gefragt ist.

Durch das Herumwerkeln mehrerer Programmierer ist die Funktion eines Software-Koordinators nötig (Freiwillige bitte vortreten!). Dieser Job bedeutet, einen Überblick über die Gesamtarbeit zu gewinnen, diese in Module zu unterteilen und die Zusammenarbeit der einzelnen Module durch Festlegen von gemeinsam genutzten Datenfeldern und zu übernehmen und zu übergebenden Parametern der einzelnen Module (in den CPU-Registern) zu gewährleisten. Diese Vorgaben sind von den Programmierern der einzelnen Module einzuhalten. Wer dazu Vorstellungen hat, möge bitte nicht durch bescheidene Zurückhaltung versuchen, einen guten Eindruck zu machen.

25 Auch wenn jemand zu Teilaspekten der Betriebssoftware Ideen hat und die Programmierung dieses Parts übernehmen möchte (oder auch nur seine Idee einbringen möchte), kann er einen wertvollen Beitrag zu dem Projekt leisten.

Als Bestechung in Naturalien bekommt jeder, der sich an dem Projekt

beteiligt, von mir eine geätzte Terminalkarte geschenkt, die er allerdings selbst und mit eigener Finanzierung der Bauteile aufbauen muß. Hercules-Karten kann ich auch günstig beibringen (zur Zeit für 40,- DM). Wer den Job des Software-Koordinators übernehmen will, bekommt auch die Hercules geschenkt (wenn der Club mit seinem gewaltigen Konto das nicht übernehmen will, stiftet ich die Karte).

Um das ganze Projekt etwas konkreter darzustellen, will ich im folgenden die bestehenden Vorstellungen zur Hardware und Software, wie Andreas Magnus und ich uns das vorstellen, breittreten. Wie gesagt, davon ist noch nichts endgültig. Insbesondere bei der Software-Vorstellung kann noch alles ganz anders gestaltet werden. Das Folgende ist also nur als Diskussionsgrundlage zu werten.

So könnte die Hardware aussehen:

Das Terminal ist ein Z80-Computer mit 32K Arbeitsspeicher und 32K Hercules-Video-RAM. Nach dem Einschalten besteht der Arbeitsspeicher beim Lesen aus EPROM und beim Schreiben aus RAM. Durch ein LDIR wird der EPROM-Inhalt bei der Initialisierung ins RAM kopiert. Danach wird das EPROM abgeschaltet und beim Lesen und Schreiben in den unteren 32K erreicht der Z80 das RAM.

Die oberen 32K des Z80 sind dem Video-RAM der Hercules vorbehalten. Hier können die beiden 32K HRG-Pages der Hercules bzw. das Text-/Attribut-Video-RAM eingeblendet werden. Im Textmodus bietet die Hercules-Karte 8 linear hintereinander liegende Seiten mit je 4K Text-/Attribut-Speicher.

Im I/O-Adreßraum des Terminal-Z80 sind die Hercules-internen Ports, das Interface zum Host-Computer, ein Latch zur Einstellung der Konfiguration des Terminals und ein Tastatur-Interface unterzubringen. Zur Vereinfachung der Hardware scheinen folgende Portadressen die sinnvollsten zu sein:

BOH-BFH	interne Ports der Hercules-Karte (siehe Artikel zur Hercules im GENIE)
00H	Status-Register des Interface zum Host-Computer
01H	Daten-Register des Interface zum Host-Computer
02H	Konfigurations-Latch
03H	Tastatur

Das Konfigurations-Latch unterstützt bislang folgende Befehle:

OUT	(02H),0	Einschalten des Text-/Attribut-Speichers im Textmodus bzw. der HRG-Seite 0 im Grafik-Modus
OUT	(02H),1	Einschalten der HRG-Seite 1; im Text-Modus ist der Text-Speicher nicht verfügbar
OUT	(02H),2	im Bereich 0-32K ist bei Leseoperationen das EPROM eingeblendet
OUT	(02H),3	sowohl beim Lesen als auch beim Schreiben liegt im Bereich 0-32K RAM vor
OUT	(02H),5	und danach
OUT	(02H),4	Zurücksetzen des Interface der PC-XT-Tastatur

Die Ausgabe der Werte 6-15 kann für weitere Konfigurations-Einstellungen benutzt werden. D0 gibt dabei jeweils den Pegel einer entsprechenden Steuerleitung wieder (Q3-Q7 des 74LS259-Latch) und D1-D3 geben vor, welche Steuerleitung geschaltet werden soll.

Die Portadressen, unter denen das Terminal für den Host-Computer erreichbar sein soll, lassen sich über Jumper in Schritten von 4 beliebig einstellen. Die Basisadresse X+0 ist das Status-Register. D0 des Status-Registers zeigt an, ob das Terminal ein weiteres Byte annehmen kann:

HEFT
31
Juli
1990

26

27

D0=0 : das Terminal kann ein weiteres Byte annehmen
 =1 : es darf noch kein weiteres Byte an das Terminal ausgegeben werden

D7 gibt an, ob das Terminal bei der Ausgabe von Daten an den Host ein neues Byte bereitgestellt hat:

D7=0 : es steht kein neues Byte des Terminals bereit
 =1 : das Terminal hat ein neues Byte ausgegeben

Je nach Zustand der Bits 0 und 7 des Status-Ports kann die CPU an den Daten-Port X+1 ein weiteres Byte ausgeben bzw. von dort ein weiteres Byte abholen.

Für den Z80 des Terminals liegen die Adressen der Ports des Interface fest (00H = Status-Register; 01H = Daten-Register). Durch Lesen des Status-Registers kann der Terminal-Z80 feststellen, ob er Daten entgegennehmen kann/muß bzw. ausgeben kann/muß:

D0=0 : es kann/muß ein weiteres Byte an den Host ausgegeben werden (der Host hat das letzte Byte abgeholt)
 =1 : es darf noch kein weiteres Byte an den Host ausgegeben werden
 D7=0 : es liegt kein neues Byte vom Host vor
 =1 : der Host hat ein neues Byte ausgegeben, das abzuholen ist

Zur Vereinfachung der Kommunikation zwischen Host und Terminal wird am Terminal-Z80 ein Interrupt erzeugt, wenn der Host ein neues Datum ausgegeben hat.

Als Tastatur kann wahlweise eine 8Bit-Parallel-ASCII-Tastatur oder eine serielle PC-XT-Tastatur verwendet werden. Beide Interfaces sind bei der bislang bestehenden Prototypen-Karte vorhanden und können durch Jumper selektiert werden. Wenn von der Tastatur ein Byte gekommen ist, wird am Terminal-Z80 ein /NMI ausgelöst.

Und das sollte die Software unterstützen:

In allen vorgesehenen Betriebsmodi des Terminals sind Umschaltbefehle möglich, die in einen jeweils anderen Modus umschalten oder sonstige Spezialaufgaben erledigen:

- 1) Einschalten des TVI 950 Textmodus (Grundeinstellung)
- 2) Einschalten des Tektronix 4014 Grafikmodus
- 3) Umschalten zwischen Local Mode und OnLine-Betrieb
- 4) Übernehmen eines Maschinenprogramms vom Host; Der Befehl enthält als Parameter Ladeadresse, Länge und Call-Adr.
- 5) Umschalten zwischen Video- und Drucker-Ausgabe
- 6) Übernahme eines Initialisierungsstrings und der Bit-Grafik-Einleitung für den jeweils an der Hercules angeschlossenen Drucker
- 7) Hardcopy des Bildschirms (sowohl Text als auch Grafik, was gerade eingestellt ist)

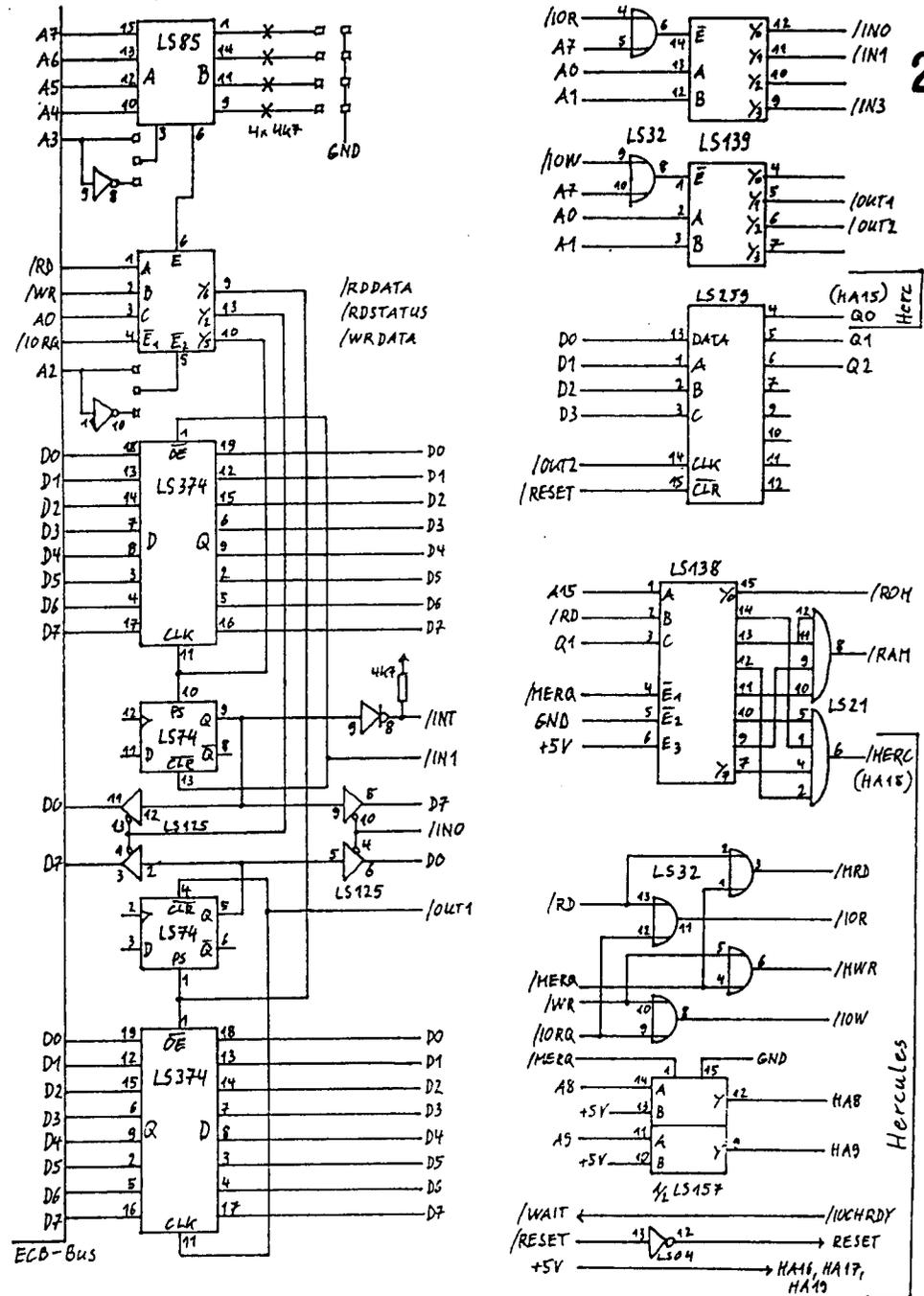
Die ersten 3 Punkte machen schon ein recht ansehnliches Terminal aus. Die Übernahme und optionale Ausführung von Maschinenprogrammen hat haupt

sächlich in der Entwicklungsphase der Software ihre Bedeutung. Außerdem erlaubt sie auch später noch den Mißbrauch der Z80-Power des Terminals für Anwenderzwecke.

Der Punkt 5 erschlägt mit dem Terminal auch gleich eine Drucker-Schnittstelle. Die Punkte 6 und 7 drängen sich bei der gemeinsamen Lage von Video- und Drucker-Interface geradezu auf.

Als Grundstock an Treiberroutinen müssen die Initialisierung des Textbildschirms und des Grafikbildschirms sowie ein Tastatur-Treiber vorhanden sein. Für die beiden Video-Modi muß je ein Interpreter für die im jeweiligen Modus gültigen Control-Codes und Escape-Sequenzen existieren, der auch immer die oben angeführten globalen Umschaltbefehle mit berücksichtigt.

28



	1,0:	Superscript	nicht möglich
	1,1:	normale Schrift (default)	möglich
D6=1:		Breitschrift	nicht möglich
D7		ist immer low auszugeben	
ESC ')')		Invertieren einschalten	
ESC '(')		Invertieren ausschalten	
ESC 'n')		Bildschirm eingeschaltet	
ESC 'o')		Bildschirm dunkel	
ESC '\$')		Strichgrafik ein	Multiplan-kompatible
ESC '%')		Strichgrafik aus	Zeichen anstelle der Großbuschstaben

ESC ESC 'T' Tektronix-Grundmodus einschalten

Tektronix-Control-Codes

^M	ODH	Alphamodus ein, Cursor an den Anfang der Zeile der letzten Grafikposition setzen
^_	1FH	Alphamodus ein, Cursor auf die letzten Graphik-Koordinaten setzen
^ö	1CH	Punktmodus ein
^Ü	1DH	Vektormodus ein; der erste Vektor wird nicht gezeichnet, die Koordinaten werden als neuer Anfangspunkt übernommen
^G	07H	Sonderfunktion (nach ^Ü): der Vektor wird doch gezeichnet
^^	1EH	Inkrementalmodus ein
^Q	11H	lösche mit Schwarz (Hintergrundfarbe)
^R	12H	zeichne invers (Vordergrund: schw<->weiß)
^S	13H	zeichne mit weiß (->Vordergrundfarbe)

die letzten 3 Steuercodes sind nicht Bestandteil des Tektronix-Befehlssatzes sondern Erweiterungen der GRIP

Tektronix-Escape-Sequenzen

ESC '^L'	Clear Screen, Alphamodus ein, Cursor home
ESC '^1'	Bidschirm um 1 Zeile hochscrollen
ESC '^2'	Bidschirm um 1 Spalte nach rechts scrollen
ESC '^3'	Bidschirm um 1 Zeile runterscrollen
ESC '^4'	Bidschirm um 1 Spalte nach links scrollen

die letzte Grafik-Koordinate scrollt mit, die Position relativ zum Nullpunkt ändert sich also

ESC '^G' x	Zeichenattribute für Alphamodus setzen
	Attribute sind die gleichen wie bei TVI 950
ESC '^~'	Vektortyp := durchgehende Linie (default)
ESC '^a'	Vektortyp := gepunktete Linie
ESC '^b'	Vektortyp := Strichpunkt-Linie
ESC '^c'	Vektortyp := kurz gestichelte Linie
ESC '^d'	Vektortyp := lang gestichelte Linie
ESC '^F'	Fülle Fläche, in der der Grafik-Cursor steht mit dem Muster des Zeichens 7FH

Der Vektormodus (mit ^U einschalten)

Bei der Hercules beträgt die Grafikauflösung 720(x) * 348(y) Punkte. Der Ursprung (0,0) liegt unten links. Eine Koordinate wird durch 2 10Bit-Werte angegeben, die als 4 Bytes mit 5 signifikanten Bits (D0-D4) an das Terminal übertragen werden. Welche Funktion das jeweilige Byte hat, wird durch D5-D7 codiert:

0,0,1	obere 5 Bit der Y-Koordinate
0,1,1	untere 5 Bit der Y-Koordinate
0,0,1	obere 5 Bit der X-Koordinate
0,1,0	untere 5 Bit der X-Koordinate

Die Übertragung erfolgt in der Reihenfolge: Yhigh, Ylow, Xhigh, Xlow
 Folgende Formate zur Koordinaten-Übergabe sind auch erlaubt:

Yhigh Xlow	für die nicht übergebenen Anteile
Ylow Xhigh Xlow	der Koordinaten werden die zuletzt
Ylow Xlow	übergebenen Werte wieder verwandt
Xlow	

Mit dem Empfang des Xlow-Bytes ist die Übertragung eines Koordinaten-Wertepaares abgeschlossen, das Terminal zeichnet dann einen Vektor vom letzten Koordinatenpunkt zum neu angegebenen Punkt.

Der Punktmodus

funktioniert wie der Vektormodus, es wird aber keine Linie sondern nur der Endpunkt gezeichnet. Lösch- und Invertier-Befehle wirken auch hier.

Der Inkrementalmodus

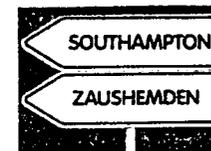
zeichnet von der letzten Grafik-Koordinate in 1-Punkt-Schritten weiter; die Werte (X,Y) des Grafik-Cursors laufen mit. Die Richtung wird durch die folgenden Zeichen vorgegeben:

F	D	E	'F' nach oben links
			'D' nach oben
B	*	A	'E' nach oben rechts
			'B' nach links
J	H	I	'A' nach rechts u.s.w.

Durch '^P' '^Q' '^R' '^T' werden die Zeichenmodi eingestellt. Ein Modus ist so lange wirksam, bis ein neuer eingeschaltet wird. Nach dem Einschalten des Inkrementalmodus muß einer dieser Befehle folgen:

^~	Punkte nicht zeichnen, nur Grafik-Cursor bewegen
^P	Punkte zeichnen (weiß, wie ^S)
^Q	Punkte löschen (schwarz, wie ^Q)
^R	Punkte invertieren (wie ^R)

Helmut Bernhardt

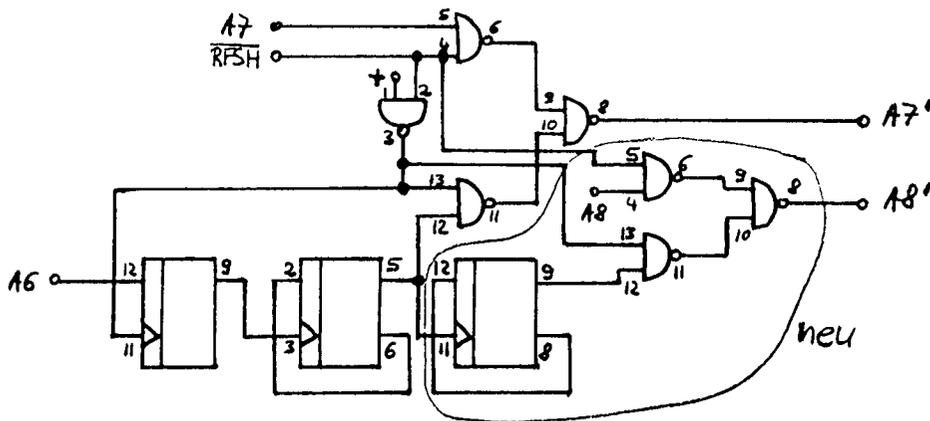


Klasse! Zum Beginn der Reisesaison haben sich die Engländer etwas einfallen lassen: Speziell für deutsche Touristen ist die Beschilderung jeweils unter der oft unverständlichen örtlichen Bezeichnung in deutscher Sprache ergänzt. Wir meinen: ein liebenswürdiger Sörwiß!

BÖRSE -- BÖRSE -- BÖRSE

1MB-Banker mit 1MB

Da Helmut's Banker ein ganzes Megabyte verwalten kann liegt es natürlich nahe, den Speicher auch soweit auszubauen. Die Möglichkeit mit vier übereinander gestapelten 256er RAMs funktioniert zwar und sieht phänomenal interessant aus, ist aber sicher nicht der Weisheit letzter Schluß. Einmal braucht man viel Platz nach oben und zweitens werden die Dinger ganz schön warm, entsprechend ist der Strom, der da verheizt wird. Inzwischen sind aber die Mega-Chips auch schon in bezahlbaren Regionen angelangt und damit löst sich fast alles in Wohlgefallen auf. Probleme bereiten nur noch die Größe der Chips und der Refresh. Da die neuen Käfer zwei Beine mehr haben als die alten und die Belegung von der der 256er etwas abweicht habe ich einen Platinenstreifen mit den 18-poligen Fassungen bestückt und die Signale einfach zu den alten Fassungen rübergefädelt. Geht einwandfrei und läuft auch mit 10MHz problemlos. Aber jetzt kommts. Für die 256er mußte der Refresh-Zähler bereits auf 8 Bit aufgebohrt werden und die Mega-Chips brauchen sogar einen 9-Bit-Refresh. Mit zwei weiteren ICs läßt sich aber auch dieses Problem ohne allzugroßen Aufwand lösen. Wenn man beim Umbauen mit dem Refresh anfängt, muß sich der Rechner hinterher noch ganz genauso verhalten wie vorher, da es die 256er RAMs überhaupt nicht stört, wenn sie mit einer 9-Bit-Adresse betan werden.



Alexander Schmid

H. G. Lütz

Heintemannstr. 147
4630 Boshum 1
Tel. 0234-707070

Alexander Schmid
St. Kajetanstr. 38/III
8000 München 80

Sehr geehrter Herr Schmid!

Vielen Dank für die Übersendung einer Club-Zeitschrift.

Nach Systemumstieg biete ich folgendes an:

- 1 TRS 80 Mod. 2
- 2 8-Zoll Lautwerke
- 1 Line Printer 3

Preis 333,- DM

Mit freudl. Gruß

Hann. Grotz C.E

In jedem Betriebssystem (und auch in manchen anderen Programmen) gibt es Treiber, die Peripherie-Geräte ansteuern. Z. B. verfügen die meisten Textverarbeitungsprogramme über eigene Drucker-Treiber, weil die des Betriebssystems ewig auf ausgeschaltete Drucker warten (oder das gar nicht bemerken), zu langsam sind, gewisse Zeichen verschlucken, kein Spooling ermöglichen usw.

Nun fragt sich natürlich: Wie programmiere ich einen Treiber? Die Frage läßt sich nicht so einfach beantworten, denn es kommt auf zwei Dinge an:

1. Um was für eine Art (Klasse) Peripheriegerät handelt es sich?
2. Welche Hardware zur Ansteuerung habe ich zur Verfügung?

Geräte-Klassen: Zeichen- und blockorientierte Geräte

Die erste Frage führt uns zu einer Unterscheidung in zwei Klassen: Die *zeichen-* und die *blockorientierten* Geräte.

Zeichenorientierte Geräte

Bei den zeichenorientierten Geräten wird immer nur ein Zeichen zur Zeit ein- oder ausgegeben. Darunter fallen Drucker, Bildschirme, Tastaturen und serielle Schnittstellen. Nach der Ausgabe eines Zeichens ist das Gerät meist eine gewisse Zeit mit der Verarbeitung beschäftigt, bevor es das nächste Zeichen empfangen kann; so lange müssen wir warten bzw. etwas anderes tun. Andererseits müssen wir dem Gerät nicht immer Zeichen liefern, wenn es (z. B. der Drucker) bereit ist. Bei der Eingabe rechnen wir nur sporadisch mit Zeichen, wobei es Probleme geben kann, wenn wir nicht rechtzeitig das Zeichen abholen (z. B. von der seriellen Schnittstelle). Dann überschreibt das nächste ankommende Zeichen das vorherige, was (falls bemerkt) dann Neu-Deutsch als „Overrun“ bezeichnet wird.

Oft wird noch weiter abstrahiert, indem wir sagen: Es gibt einen „Strom“ von Bytes („Byte Stream“) vom oder zum Gerät. Programmierer kennen das vom Datentyp „sequentieller File“, wobei darunter z. B. bei TURBO-PASCAL auch der Drucker fällt: Es kann immer nur sequentiell das nächste Byte gelesen oder geschrieben werden, eine Adressierung („Byte Nr. 10“) gibt es meistens nicht.

Blockorientierte Geräte

Bei den blockorientierten Geräten dagegen, zu denen vor allem Massenspeicher wie Floppy-Laufwerke gehören, tritt nie ein Zeichen allein auf, sondern es werden immer Blöcke übertragen. Meistens sind die Blocklängen festgelegt und typischerweise finden wir 128, 256, 512 oder 1024 Bytes je Block. Bei jedem Schreiben oder Lesen wird hier dem Gerät erstmal über ein Kommando mitgeteilt, daß jetzt geschrieben oder gelesen werden soll, und um welchen Block es sich handelt. D. h. hier findet eine Adressierung („Block Nr. 10“) statt!

Beim Lesen oder Schreiben des Blocks haben wir es dann natürlich wieder mit einzelnen Bytes zu tun, die aber meist so schnell hintereinander gelesen/geschrieben werden müssen, daß die Zusammzufassung zu einem Block gerechtfertigt ist. Der Zeitfaktor ist hier ganz anders gelagert als im zeichenorientierten Fall: Beim Schreiben *muß* das nächste Zeichen geliefert werden, wenn das Gerät dies verlangt („Data Request“), sonst gehen Zeichen verloren („Data lost“). Ebenso beim Lesen: Wenn das Zeichen nicht schnell genug abgeholt wird, kommt es wie oben zu einem „Overrun“, wobei jetzt aber der ganze Block nicht mehr stimmt!

Soweit zu den beiden Klassen. Jetzt wird es etwas konkreter: Welche Möglichkeiten für die Ein-/Ausgabe gibt es?

Programmierung: Polling, Interrupt, DMA

Die sich uns bietenden Möglichkeiten zur Ein-/Ausgabe-Programmierung hängen vor allem von der vorhandenen Hardware ab. Oben habe ich immer nur von „Geräten“ gesprochen; meistens finden wir aber zwischen dem Hauptprozessor und dem Gerät noch einen oder mehrere Chips, die „Controller“ genannt werden. Diese Controller sind Spezial-Prozessoren, die dem Hauptprozessor einige Arbeit abnehmen. Drei unterschiedliche Controller-Typen spielen im Bereich „Ein-/Ausgabe“ eine Rolle:

- **Ein-/Ausgabe-Controller:** Das sind Chips, die sich nur um ein bestimmtes Gerät bzw. dessen Ansteuerung kümmern. Wichtig sind für uns Floppy-Controller und SIO-Controller („Serial Input/Output“, serielle Schnittstelle). Wenn ich im folgenden Teil von „Gerät“ spreche, meine ich meistens den zu diesem Gerät gehörenden Controller, denn für den Hauptprozessor sieht der Controller so aus, als wäre er selbst das Gerät.
- **Interrupt-Controller:** Manchmal hat ein Rechner oder ein Hauptprozessor intern (z. B. der HD64180) auch einen Interrupt-Controller. Dieser verwaltet für den Hauptprozessor die verschiedenen Interrupt-„Quellen“, wobei es sich meist um die Controller von Geräten handelt. Die Controller melden sich meistens über Interrupts, wenn eine Zustandsänderung auftritt, wenn sie also wieder bereit sind, ein Byte zu empfangen, oder eins senden möchten oder ein

Fehler von ihnen erkannt wurde. Der Interrupt-Controller entscheidet dann, was zu passieren hat. Normalerweise wird der Hauptprozessor bei der Abarbeitung seines normalen Programms unterbrochen. Wenn mehrere Interrupts gleichzeitig auftreten oder gerade ein vorhergehender noch bearbeitet wird, entscheidet der Interrupt-Controller, welcher Interrupt Vorrang hat. Er kann auch einige oder alle Interrupts sperren, so daß deren Auftreten den Hauptprozessor bei seiner Arbeit nicht stören.

- **DMA-Controller:** „DMA“ bedeutet „Direct Memory Access“. Ein DMA-Controller darf also direkt auf den Hauptspeicher (RAM) zugreifen, ohne den Hauptprozessor bemühen zu müssen. Der DMA-Controller liest und/oder schreibt selbst Bytes, während der Hauptprozessor weiter sein eigentliches Programm abarbeitet. Für DMA-Controller gibt es zwei Anwendungen: Verschieben von Speicherblöcken, Lesen oder Schreiben von Blöcken von/auf Geräte(n). Uns interessiert natürlich nur der letzte Fall. Besondere Zustandsänderungen (z. B. das Ende eine Übertragung) werden wieder an den Hauptprozessor gemeldet, evtl. über einen Interrupt-Controller.

Einerseits müssen also in unserem Rechner die nötigen Controller vorhanden sein, vor allem ein Interrupt- und ein DMA-Controller, andererseits muß die Architektur von Rechner und Betriebssystem darauf abgestimmt sein. So verfügen z. B. die Z80-Prozessoren von Haus aus über eine sehr mächtige Interrupt-Behandlung in Form des Interrupt-Modus 2 (IM 2), aber nur wenige Rechner bzw. Betriebssysteme nutzen dies aus (NEWDOS gar nicht und CP/M selten). Noch besser sind der HD64180 und der Z280, die beide auf dem Hauptprozessor-Chip einen Interrupt- und mehrere DMA-Controller beherbergen, was aber auch selten ausgenutzt wird.

Bei jeder Programmierung gibt es drei verschiedene Fälle zu beachten:

- **Eingabe bzw. Empfangen:** Bei der Eingabe bzw. dem Empfangen müssen die ankommenden Bytes rechtzeitig abgeholt werden, damit nachfolgende Bytes sie nicht überschreiben. Hier kommt es also auf schnelle Abholung an.
- **Ausgabe bzw. Senden:** Bei der Ausgabe bzw. dem Senden ist zu beachten, daß wir die Bytes nicht beliebig schnell ausgeben können. Meist braucht das Gerät (bzw. dessen Controller) nach jeder Ausgabe erst einmal eine „Denkpause“, bevor wir das nächste Byte senden können. Hier dürfen wir also nicht zu schnell sein.
- **Fehler:** Neben der normalen Ein-/Ausgabe können natürlich auch Fehler auftreten, wenn wir etwas falsch programmiert haben (Bytes zu langsam angeholt oder zu schnell ausgegeben) oder ein physikalischer Fehler auftrat (Diskette kaputt, Leitung unterbrochen, Checksummen-Fehler usw.). Wir müssen dauernd auf Fehler gefaßt sein und geeignet darauf reagieren. Der Rechner sollte nicht abstürzen (in einer Endlosschleife hängenbleiben); wenn möglich, sollten wir den Fehler beheben (nochmal versuchen); falls das nicht klappt, ist die Ein-/Ausgabe-Operation mit einer Meldung an den Benutzer abzubrechen.

Als Fallbeispiel für die verschiedenen Möglichkeiten werde ich die Ein-/Ausgabe von/auf eine serielle Schnittstelle nehmen. Sehr ähnlich sieht die Verbindung zum Drucker (wenn auch nur Ausgabe) oder der Tastatur (nur Eingabe) aus. Dabei werde ich keinen Z80-Code benutzen oder eine bestimmte Hardware (Adressen, Ports, Chips) voraussetzen, sondern alles in einer „Meta-“Sprache, einer Art Pascal, formulieren. Das muß dann jeder für sich auf seine Hardware und Lieblings-Sprache übersetzen.

Polling

Die einfachste und am häufigsten benutzte Methode der Ein-/Ausgabe-Programmierung ist das „Polling“ (ein entsprechendes deutsches Wort fällt mir nicht ein). Dabei erledigt der Hauptprozessor, bei uns also der Z80, die Haupt-Arbeit und steht währenddessen für nichts anderes zur Verfügung. Er muß dauernd die Status-Leitung(en) des Geräts (bzw. Controllers) abfragen und warten, bis das Gerät zur Ein- oder Ausgabe bereit ist. Das entsprechende Programmstück sieht so aus:

```

REPEAT                                (* wiederhole *)
  WHILE No_RW_Request AND No_Error DO (* solange E/A-Anforderung u. *)
    Nothing                            (* kein Fehler: warten *)
  END_WHILE                            (* *)
  IF No_Error THEN                    (* falls kein Fehler: *)
    Read_Or_Write_Byte                (* lies/schreibe das nächste Byte *)
  ELSE                                  (* sonst: *)
    Try_To_Correct_Error              (* versuche, Fehler zu beheben *)
  END_IF                                (* *)
UNTIL End_Of_Block OR Fatal_Error     (* bis Blockende o. fataler Fehler *)
IF Fatal_Error THEN                  (* falls fataler Fehler: *)
  Show_Error_Message                 (* zeige Fehlermeldung *)
END_IF                                (* *)

```

Hier stellen sich gleich mehrere Probleme:

- **Senden oder Empfangen:** Das Gerät (hier: die serielle Schnittstelle) kann entweder ein Byte verlangen oder liefern. Beim Senden *können* wir ein Byte schreiben (bei einem zeichenorientierten Gerät) oder *müssen* es sogar (bei einem blockorientierten Gerät). Beim Empfangen hingegen *müssen* wir das Byte abholen, denn sonst wird es vom nächsten überschrieben. Also: Je nach Fall (Senden oder Empfangen) und Gerät (zeichen- oder blockorientiert) ist eine andere Reaktion angebracht. Aber vielleicht wollten wir *nur* senden, empfangen aber plötzlich ein Zeichen! Was machen wir dann damit? Eine Lösung wären zwei Buffer: ein Sende- und ein Empfangs-Buffer (taucht weiter unten auch wieder auf).
- **Timing:** Sobald der Request kommt, sollten wir sofort reagieren, sonst kann es zu Fehlern kommen. Also darf der Prozessor nichts anderes machen, denn sonst

überschreiten wir evtl. die kritische Zeit zwischen dem Request und dem Abholen oder Senden des Zeichens. Besonders fällt dies bei Floppy-Zugriffen ins Gewicht: Schon die falsche Anordnung der Befehle kann zu Timing-Problemen führen! Auf jeden Fall darf kein dann kein Interrupt auftreten, also werden sie bei Floppy-Zugriffen meist gesperrt. Deshalb verlieren interruptgesteuerte Uhren wie die des NEWDOS bei jedem Floppy-Zugriff einige Millisekunden.

- **Blockgröße:** Wann können wir die äußere Schleife verlassen? Bei blockorientierten Geräten ist das leicht: ein Block hat meist eine feste Länge. Bei zeichenorientierten Geräten wissen wir nur beim Senden, wieviele Zeichen zu übertragen sind, beim Empfangen aber (meistens) nicht. Bei einer seriellen Schnittstelle müssen wir eigentlich *immer* auf empfangene Zeichen warten und dürfen die Schleife nie verlassen! Wenn wir es doch tun, z. B. um ein Zeichen von der Tastatur zu holen oder auf dem Bildschirm anzuzeigen, verpassen wir (besonders bei hohen Übertragungsraten) evtl. das nächste empfangene Byte. Mögliche Lösungen: Niedrige Baudraten, Stop- („Bitte nichts mehr senden!“) und Start-Bytes („Ich bin wieder soweit!“), feste Blockgrößen auch bei zeichenorientierten Geräten (evtl. enthält das erste Byte die Blockgröße).

Interrupt

Wie Ihr seht, ist das Polling nicht das Gelbe vom Ei, jedenfalls nicht für zeichenorientierte Geräte. Ärgerlich ist vor allem, daß der Prozessor seine ganze Zeit in den Schleifen damit vertrödelt, auf das Gerät zu warten. Es wäre doch viel schöner, wenn der Prozessor seine normale Arbeit erledigt und dabei irgendwann von dem Gerät (oder Controller) unterbrochen wird.

Da haben wir schon unser Stichwort: „Unterbrechung“, auf Neu-Deutsch „Interrupt“. Der Prozessor fragt also nicht mehr dauernd den Status des Geräts ab, um auf eine Änderung zu warten, sondern das Gerät meldet selbständig eine Änderung des Zustands an den Prozessor. Dieser verzweigt dann *automatisch* in die entsprechende Routine, die eine angemessene Reaktion ausführt. Nach Beendigung der Routine wird das Programm bei dem Befehl fortgesetzt, den der Prozessor gerade bearbeiten wollte, als der Interrupt auftrat.

Ich zeige das Prinzip hier am Beispiel des Sendens:

```
PROCEDURE Send (Char: BYTE)
BEGIN
  IF Ready_To_Send THEN
    Send_It(Char)
  ELSE
    Put_Into_Buffer(Char)
  END_IF
END Send
```

Wir warten jetzt nicht mehr in einer Schleife, bis das Gerät bereit ist, sondern senden das Zeichen sofort, wenn das Gerät vorher durch einen Interrupt bekanntgegeben hat, daß es bereit ist. Ansonsten merken wir uns das Zeichen und führen unser Hauptprogramm fort.

```
PROCEDURE Send_Int
VAR Char: BYTE
BEGIN
  IF Char_In_Buffer THEN
    Get_From_Buffer(Char)
    Send_It(Char)
  ELSE
    Ready_To_Send := TRUE
  END_IF
END Send_Int
```

Wenn das Gerät dann irgendwann die Sende-Bereitschaft meldet, wird automatisch (!) diese Interrupt-Routine aufgerufen. Falls sich noch ein Zeichen im Buffer befindet, wird dieses gesendet. Ansonsten wird unser Flag Ready_To_Send gesetzt, damit bei einem folgenden Aufruf der Prozedur „Send“ (s.o.) das nächste Zeichen sofort gesendet wird.

Ähnlich sehen die Routinen zum Empfangen aus. Aber eine Frage stellt sich noch: Was passiert bei Fehlern? Auch bei einem Fehler wird eine Interrupt-Routine aufgerufen, aber wahrscheinlich wartet das Programm gar nicht auf eine Fehlermeldung, sondern werkelt irgendwo anders herum. Wir müssen uns also (fatale) Fehler merken und dem Programm mitteilen, wenn es dazu bereit ist. Das wäre entweder möglich, wenn „Send“ (s.o.) aufgerufen wird (Rückmeldung: „Es gab da mal einen Fehler!“) oder über eine spezielle Routine, die das Programm aufrufen kann, um aufgetretene Fehler oder andere Meldungen (z. B. „Alle Bytes wurden gesendet.“) abzufragen. Etwas ähnliches findet Ihr im nächsten Abschnitt bei der Routine Send_Status?.

DMA

Über DMA-Controller habt Ihr ja schon weiter oben etwas erfahren. Sie eignen sich vor allem für blockorientierte Geräte: Ein Block von Bytes, dessen Länge normalerweise bekannt ist, soll von einem bestimmten Speicherbereich aus an ein Gerät ausgegeben bzw. von einem Gerät in einen bestimmten Speicherbereich gelesen werden. Währenddessen soll der Hauptprozessor weiter werkeln.

```
PROCEDURE Send_Block (Adress: ADDRESS; So_sieht_die_Routine_zum_Senden_eines_Blocks_aus.Die_Variable_End_of_Send_wird_TRUE_sobald_der_Block_vollkommen_übertragen_wurde,_Error dagegen, wenn ein Fehler_auftrat.
                      Length: INTEGER)
BEGIN
  End_of_Send := FALSE
  Error       := FALSE
  Start_DMA (Adress, Length)
END Send_Block
```

```

PROCEDURE DMA_Int
BEGIN
  End_of_Send := TRUE
END DMA_Int

```

```

PROCEDURE Error_Int
BEGIN
  Stop_DMA
  Error := TRUE
END Error_Int

```

TYPE Status = (Busy, Ready, Error)

```

PROCEDURE Send_Status? : Status
BEGIN
  IF Error THEN RETURN Error
  ELSIF End_of_Send THEN RETURN Ready
  ELSE RETURN Busy
  END_IF
END Send_Status

```

Natürlich muß der Hauptprozessor irgendwie mitbekommen, wie er die Variablen zu besetzen hat. Das wird normalerweise wieder mit Interrupts erledigt. Die beiden Routinen DMA_Int und Error_Int werden von den Interrupts gestartet, wenn eine Übertragung (mit oder ohne Fehler) beendet wurde.

Unser Hauptprogramm kann die Vollzugs- oder Fehler-Meldung über die Routine Send_Status? abholen, wobei es folgende Möglichkeiten gibt: Die Übertragung läuft noch („Busy“), sie ist fertig („Ready“) oder sie wurde aufgrund eines Fehlers abgebrochen („Error“).

Schlußwort

In diesem Artikel habe ich Euch ein bißchen darüber berichtet, was *allgemein* bei der Ein-/Ausgabe-Programmierung möglich ist. Naturgemäß konnte ich dabei nicht sehr tief in die Materie einsteigen, z. B. fehlt noch die Buffer-Verwaltung, der Zugriff aus den Anwendungsprogrammen heraus und die Einbindung in das Betriebssystem.

Interessant wird es erst, wenn Ihr vor einem konkreten Problem steht und selber programmieren (oder vorher noch die Hardware dafür stricken) müßt. Dann erst könnt Ihr anhand der Möglichkeiten (Hardware, Betriebssystem), des Geräts (zeichen-/blockorientiert, Daten-Durchsatz...) und Eurer Anwendung entscheiden, was sinnvoll ist. Ihr werdet dann feststellen, daß vieles bei Euch von vornherein nicht möglich ist, weil Euer Rechner oder das Betriebssystem oder die Anwendungsprogramme es nicht unterstützen. C'est la vie!

PS: Vielen Dank an Hans-Günther Hartmann für die Anregungen und das Korrekturlesen. Und an das Informatik-Rechenzentrum HH für den Laser-Druck per LaTeX.



von Gerald Schröder

43

Ihr habt es ja alle schon gelesen: Der Z80 ist "out". "In" sind dagegen der Z180 (bzw. HD64180) und der Z280. Ich möchte hier nun keine genauen Beschreibungen der Prozessoren abliefern. Für den 180er gab's das schon (Sonderheft HD64180), für den 280er dürfte es bald kommen. Aber es stellen sich folgende Fragen:

Was sind die großen Verbesserungen an den beiden Neuen?

Lohnt sich das Umrüsten?

Wenn ja: für wen?

180

Die erste Frage ist für den 180er leicht beantwortet: Man nehme ein komplettes Z80-Board mit: SIO-Controller (serielle Schnittstellen), Timer/Counter, Banking-Logik, DMA- und Interrupt-Controller (jedesmal mindestens ein Chip). Das packe man auf einen Chip, der etwas größer ist als vorher der Z80 und einige Pins mehr hat, und schon hat man einen 180.

Wie Ihr seht: wenn Ihr einen ordentlich aufgebauten Z80-Rechner mit allem drum und dran habt, braucht Ihr kaum einen 180. Er ist vielleicht etwas schneller (aber auch der Z80 kommt bis zu 12 Mhz, wenn ich Alexander glauben darf), hat einige kleine Befehle mehr, die aber kein Programm benutzt, hat eine "Multi-Prozessor-Schnittstelle", die aber kein Board und keine Software unterstützt. Also: Für Euch ist das nichts.

Aber für alle anderen ist das Ding ein Sprung um eine Größenklasse nach vorn. Da kommt Helmut mit seinem HD64180-Adapter, der den Z80 ersetzt, ganz recht. (Helmut wird mich wegen der Schleichwerbung wieder steinigen, aber von Kiel bis Hamburg muß er schon eine ganz schöne Wurfkraft haben...)

Was bringt's Euch? Nun, unter NEWDOS so gut wie nichts, aber unter CP/M wird es interessant. Dort lassen sich eine Menge der neuen "Fietschers" (Heikendorf-Friesisch für "Features") ausnutzen, besonders wenn jemand CP/M-Plus fahren möchte, denn dann ist das Banking des 180er gefragt. Das ist haargenau auf CP/M-Plus zugeschnitten und wartet nur darauf, damit zusammengebracht zu werden. Allerdings legt der TRS-80 seinen Besitzern tausend Steine in den Weg, doch jeder "normale" Rechner verträgt sich ganz gut damit.

280

Den Z280 kenne ich bisher noch nicht "persönlich", aber sein Datenbuch ist doch sehr informativ. Erstmal kann er ähnlich viel wie der 180, hat aber nur eine serielle Schnittstelle (der HD64180 deren zwei sowie eine spezielle). Was kann er mehr?

- bis zu 16 MB Adreßraum (HD64180: 512 Kb oder 1 Mb)
- virtueller Adreßraum mit Pages (statt Banking)
- unheimlich viele neue Befehle
- Cache (256 Byte) und 3-stufige Pipeline
- Daten und Befehle können voneinander getrennt werden
- das Betriebssystem kann vom Anwenderprogramm getrennt werden
- Co-Prozessoren könnten eingesetzt werden (wenn es sie gibt)

44

Einige von Euch werden jetzt sagen: "Wau, ein ultramoderner Super-High-Tech-Prozessor." Das sind die Freaks, die eh' mehr in der Hardware und dem Betriebssystem stecken und ansonsten ihre Verwaltung lieber auf Karteikarten machen sowie ihre Briefe per Hand schreiben (hallo, Helmut!).

Der größte Teil dürfte sagen: "Was'n dat? Kenn ich nich'." Müßt Ihr auch nicht, denn: Für die Z80-Welt sind das alles Begriffe, die nie im Gespräch waren. Wer einen PC hat, kennt schon eher einige davon, ohne sich groß Gedanken darüber zu machen (dort zählen nur Mhz und Waitstates).

Eins sollte beiden Gruppen gesagt werden: Unter CP/M(-Plus) (geschweige denn NEWDOS) seht Ihr von den Leistungsmöglichkeiten des Z280 nur Bruchteile. Er ist dann wie ein Porsche-Motor im Käfer: Das Drumherum kann die PS nicht unterstützen und mehr als 120 hält die Kiste dann auch nicht aus. Jedenfalls hat noch nicht mal das "modernste" CP/M (3.0) irgendwelche Unterstützung dafür parat.

Nun aber die gute Nachricht: Für die Freaks unter Euch sei gesagt: Eine viel bessere Plattform zum Experimentieren könnt Ihr wohl kaum bekommen. Das Ding kann mehr als mancher IBM-Kompatibler, ist aber flexibler und offener. Für mich sieht das Ding aus wie ein UNIX-Prozessor. Aber: Wer hat schon von Z80-UNIX-Maschinen gehört? Ich nicht. Also bastelt mal schön!

Fazit

Für den normalen Z80-User, der endlich CP/M-Plus komfortabel genießen möchte (und dafür die BIOS-Anpassung selbst vornehmen kann), bietet sich der 180er an. Er ersetzt einfach eine Menge Chips, die auf einer normalen Platine kaum Platz haben und hat alle Möglichkeiten der Z80-Welt nebst einigen Schmankerln.

Für den Power-User (UNIX nennt es "Super-User") bietet sich der Z280 mit seinem Superuser-Modus geradezu unerschämt an. Natürlich kann er auch sein CP/M-Plus darauf fahren, aber das allein bringt es nicht die Bohne.

Eins sollte ich natürlich nicht vergessen: Wenn Ihr vorhabt, Eurem alten TRS-80 oder sonstigen Z80-Rechner endlich die letzte Ruhe zu gönnen, aber der Z80-Welt nicht ganz abhanden kommen wollt, macht's wie ich: Legt Euch einen HD64180- oder Z280-Rechner zu. Dann habt Ihr wenigstens einen Rechner, den nicht jeder hat. Und obendrein habt Ihr eine Menge Möglichkeiten zum Rumspielen in Hard und Soft, die PCs und STs nicht haben.

PS: Natürlich ist das alles meine persönliche Meinung und Ihr dürft alle ganz anderer Meinung sein.

Hallo Clubfreunde,



als neuer Bibliothekar liefere ich meinen "Einstand" gleich mit zwei Buchvorstellungen ab. Die Bücher wurden mir auf der diesjährigen Hauptversammlung übergeben. Beide sind aus dem VEB Verlag Technik in Berlin und behandeln das Thema Technische Informatik. Buch 1 beleuchtet das Thema "CP/M in der Praxis", der andere Band trägt den Titel "MS-DOS". Während der erste Band sicherlich viele aufhorchen läßt, dürfte der MS-DOS Band wohl mehr als Reizstoff empfunden werden. Immerhin wurde ja erneut bekräftigt, daß der Z80 und sein Umfeld weiterhin "Hahn im Korb" bleiben sollen. Aber gehen wir der Reihe nach vor.

Erst wollte ich die Inhaltsverzeichnisse abtippen, wegen deren Länge habe ich diese Idee dann aber sehr schnell fallen gelassen. Stattdessen also neben meinem Kommentar die Kurzbeschreibung von der Rückseite jedes Bandes:

CP/M in der Praxis

"Der Band vermittelt dem Anfänger bei der Arbeit mit einem 8-Bit-Personalcomputer das notwendige Einstiegswissen und ist dem fortgeschrittenen Anwender eines Fachgebietes ein ausreichendes Nachschlagewerk. Es werden das Betriebssystem CP/M mit seinen wesentlichen Systembestandteilen, das Dienstprogramm POWER und das Textverarbeitungssystem WordStar mit seiner Komponente Mailmerge beschrieben"

Meiner Ansicht nach, ist zum Thema CP/M im Ganzen zu wenig zu lesen. Eine umfangreiche Beschreibung des inneren Aufbaues vom CP/M sollte man besser nicht erwarten. Der Schwerpunkt liegt dafür eindeutig auf der Beschreibung der Dienstprogramme, deren Handhabung und Befehlssatz. In diesem Punkt wird dann auch nicht mit Informationen geizt. Wer zu den genannten Programmen keine Beschreibung greifbar hat oder einfach mehr zur Bedienung wissen will, wird mit einer Menge an Information bedient. Welche WordStar-Version als Grundlage diente, konnte ich beim Durchsehen nicht feststellen.

MS-DOS

"MS-DOS ist das international am meisten genutzte Betriebssystem für 16-Bit-PC. Das Buch enthält alle wichtigen Informationen für die System- und Anwendungsprogrammierung. Dazu gehören: Hardwareübersicht, BIOS- und DOS-Funktionen, Kommandos, Stapelverarbeitung, Programmierwerkzeuge und wichtige Anwendungsprogramme."

Was ich beim CP/M-Band aufgrund des Titels erwartet hatte, gibt es hier für MS-DOS wirklich satt. Jede Systemfunktion wird beschrieben, inklusive der Belegung der CPU-Register für den Aufruf. Ebenso wird die Struktur von MS-DOS, Datenformate und dergleichen mehr erklärt. Kurz - wer mehr als nur die Schreibweise von AUTOEXEC.BAT und CONFIG.SYS wissen will, wird bestens bedient.

Seviel vorerst zu den "Neuzugängen" und bis zur nächsten Wortmeldung

Tschüß

Handwritten signature

--- Termine --- Termine --- Termine ---
Nächster Redaktionsschluß Anfang November
Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!
Natürlich nehme ich auch größere Werke an!

--- Messen '90 ---
System 90 22.10. - 26.10.
electronica München 06.11. - 10.11.

Clubtreffen Süd
Am Sonntag zur Hobbytronik in Stuttgart währe eine gute Gelegenheit sich wieder einmal treffen.
Treffpunkt ist ab 10.00 Uhr (und dann alle zwei Stunden) im Eingangsbereich.
Vielleicht bietet sich die Möglichkeit Fahrgemeinschaften zu bilden. Specht
Buch bitte ab.

- 47 1. Vorsitzender Fritz CHWÖLKA
Saarstraße 34
5173 Aldenhoven
☎ 02464 / 8920
- 2. Vorsitzender Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 / 2602
- Hardwarekoordinator Andreas MAGNUS
Pommernstr. 4
4658 Gelsenkirchen
☎ 0209 / 870230
- NEWS-Diskotheke Oliver VÖLZ
Waldburgstr. 73
7000 Stuttgart 80
☎ 0711 / 7353817
- CP/M -Diskotheke Rüdiger SÖRENSEN
Thomas-Mann-Str. 3a
6500 Mainz 1
☎ 06131 / 32860
- Club-Bücherei Kurt MÜLLER
Sophie-Scholl-King 38
2054 Geesthacht
☎ 04152 / 70643
- Redaktion Jens NEUEDEK
Gschlachtenbretzingen
Rudolf-Then-Straße 32
7178 Michelbach /Bilz
☎ 0791 / 42877
- Autoren Die Redaktion bedankt sich bei
den im INHALTSVERZEICHNIS genannten
Autoren für die Mitarbeit an der
Club-INFO.
- Bankverbindung des CLUB 80
Postgirokonto
Sonderkonto CLUB 80
Übermann, H. 6209 Heidenrod
Konto-Nummer 496 071 - 605
Postgiroamt Frankfurt
BLZ 500 100 60

S C H L U S S

Hallo Club-80er,

nachdem nun scheinbar Frühjahrsmüdigkeit, Sommerloch und andere Unannehmlichkeiten überwunden sind, kamen auch einmal wieder ein paar INFO-Beiträge in meinem Briefkasten an.

Obwohl ich doch noch nicht alle Sparten unserer INFO belegen konnte, habe ich trotzdem mal wieder ein INFO fertiggestellt. Dazu kann ich Euch zusätzlich noch ein Sonderheft

TANDY und GENIE ...ein Hardwareüberblick

bieten.

Zur Zeit habe ich noch zwei Sonderhefte in der Mache. Ansonsten fehlen mir leider halt doch immer die Beiträge für das nächste INFO, um ein geregeltes Erscheinen zu sichern.

Als nächsten Redaktionsschluß habe ich mir nun Anfang November vorgemerkt. Es hängt natürlich auch von Eurer Mitarbeit ab, ob die pünktliche Herausgabe gewährleistet werden kann.

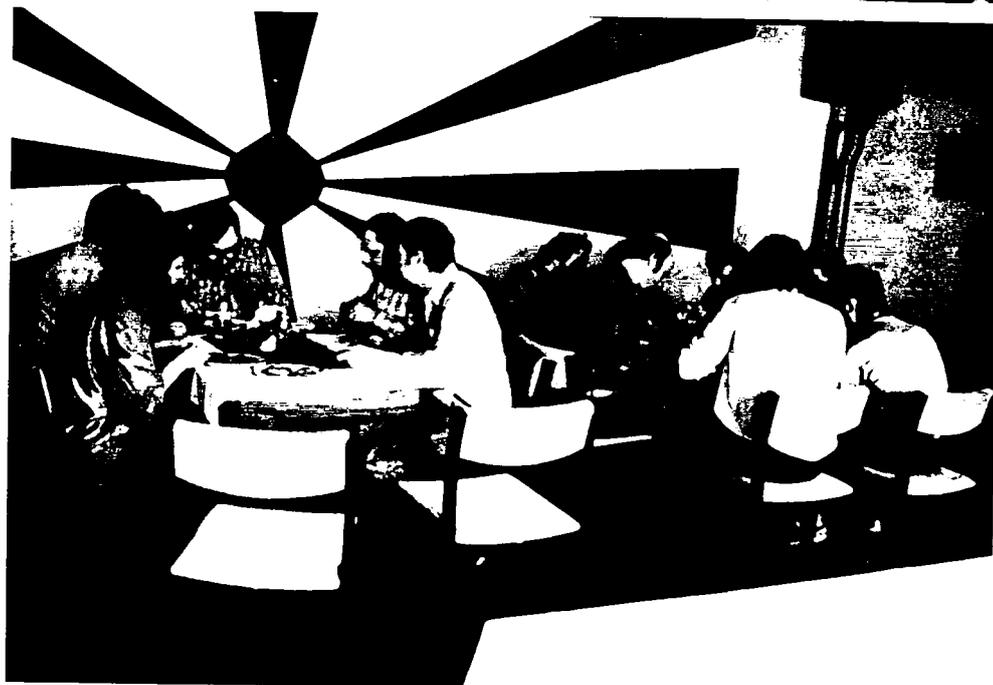
Bis demnächst ...

Es grüßt Euch Eure Redaktion

Jens

Eine Zensur oder Kontrolle der Infobeiträge erfolgt nicht.
Die Redaktion

Karlheinz kopiert
und kopiert ...



Hacken macht hungrig

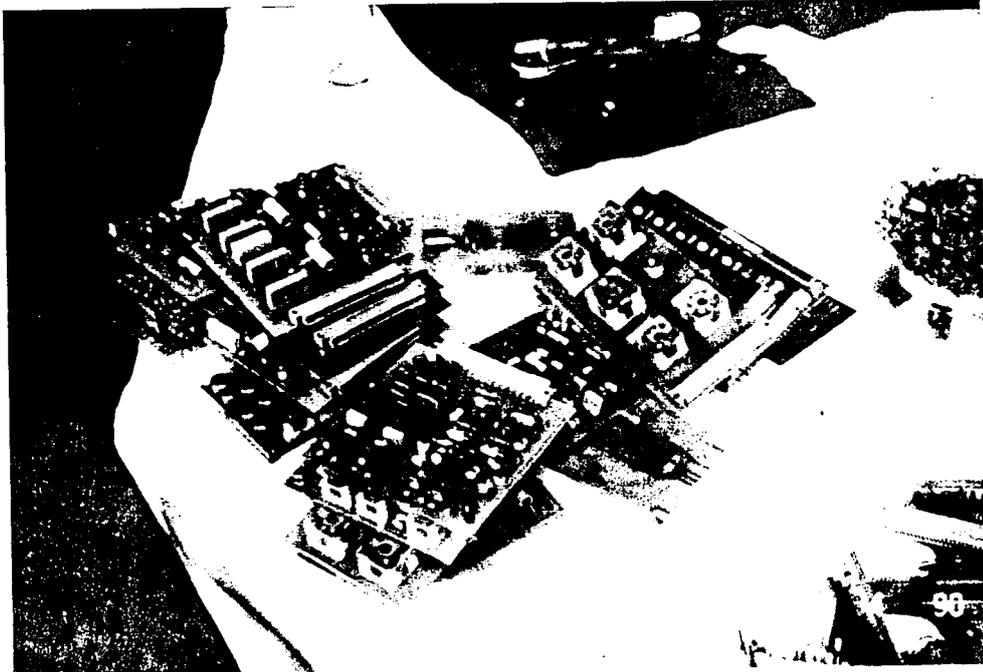
... und nachts
bei Frank
die Theorie
(mit Flasche:
der Hausherr)





Dasselbe Bild in West
wie Ost
(links:
Frank-Michael Schober)

Sinclair ZX-Spectrum
+ Technik = Computer



Beute von der Stasi