

# CLUBINFO

44. Ausgabe

Kontaktadresse:  
**Club 80**  
**Hartmut Obermann**  
Wilhelm-Baur-Straße 8  
Postfach \_\_\_\_\_  
76135 Karlsruhe  
Tel.: 0721/85 40 68  
BTX: 0721/ \_\_\_\_\_  
Anrufbeantworter &  
FAX: 0721/85 40 68

# Inhaltsverzeichnis

## Autor & Seite

### Clubinternes

Protokoll der Hauptversammlung vom 14.05.94	1 - 2
	Gerald Schröder
Neues vom Vorstand	3 - 4
	Hartmut Obermann
10 Jahre Club 80 -eine kleine Club-Historie	5 - 6
Termine	6
	Jens Neueder
Mitgliedvorstellung	7
	Michael Kalb

### Software

Grafik des Genie IIIs unter Holte CP/M+	9 - 19
	Volker Dose, Egbert Schröder
Assembler-Tip	21
Textumsetzung die N + erste	22 - 28
Nachtrag und Erweiterung zum PCXPUT.C	29 - 33
Nachtrag und Erweiterung zum PCXSHOW.C	34 - 39
	Alexander Schmid

### Club 80 Börse

Suche ...	20
	Michael Vogl

### Sonstiges

Model-1 Emulator für Dosen	40 - 42
	Alexander Schmid

### Die letzten Seiten

Impressum	43
Schluß	44
	Redaktion
Mitgliederadressenliste	am INFO-Ende
Fragebogen DFÜ	am INFO-Ende
Bilder vom Club-Treffen 94	am INFO-Ende

Protokoll der Hauptversammlung vom 14. Mai 1994  
von Gerald Schröder

Vom 12. bis 15. Mai trafen sich die Aktiven des Club 80 und einige Gäste zur Hauptversammlung 1994, diesmal organisiert vom Alexander Schmid und daher in Kornach stattfindend.

Anwesend waren vom Club 80 (teilweise traten die Mitglieder erst während des Treffens dem Club bei):

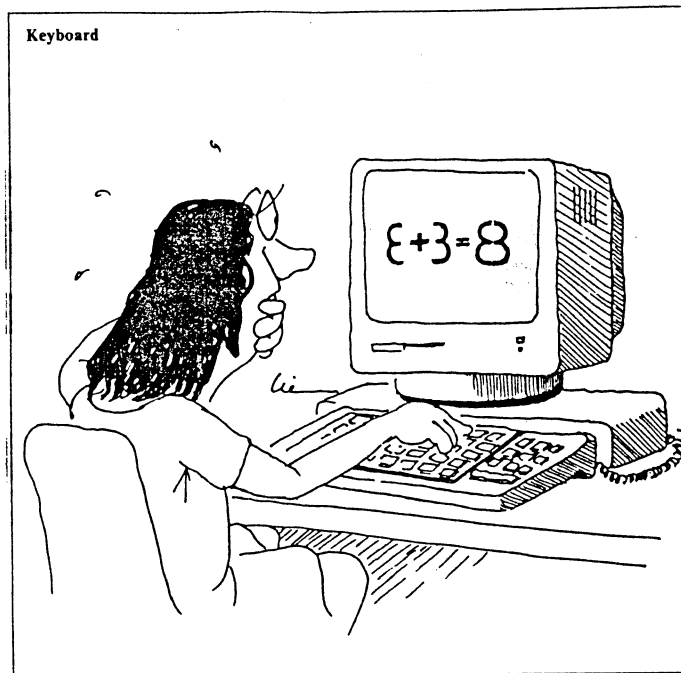
Helmut Bernhardt  
Günther W. Braun  
Fritz Chwolka  
Manfred Held  
Michael Kalb  
Jürgen Kemmer  
Jens Neueder  
Hartmut Obermann  
Jürgen Peters  
Alexander Schmid  
Rainer Schmitz  
Uwe Schoberth  
Gerald Schröder  
Ulrich Staimer.

Als Gäste durften wir begrüßen: Helmut Jungkuz  
Andreas Kleespies  
Tilmann Reh  
Claudio Romanazzi.

Es ließ sich kein TRS-80 oder kompatibler Rechner mehr blicken, dafür überwogen die CP/M-Kisten gegenüber den MS-DOSen, die nur wegen der Farbe und der Lautstärke auffielen.

Im (kurzen) offiziellen Teil wurden folgende Themen abgehandelt:

1. Die Clubbibliothek, aus der seit mindestens einem Jahr nichts mehr ausgeliehen wurde, wird von Kurt Müller zu Rainer Schmitz transferiert. Dieser wird sie ein Jahr lang hüten, eine aktuelle Bücherliste ins Info setzen und auf Ausleiher warten. Wenn niemand Bücher ausleiht, wird die Bibliothek aufgelöst und die Bücher werden meistbietend versteigert. Damit soll erreicht werden, daß die Bücher im Club verfügbar bleiben, da sie größtenteils nicht mehr zu kaufen sind, und trotzdem einer sinnvollen Nutzung zugeführt werden.
2. Hartmut machte den Vorschlag, einige Modems auf Clubkosten anzuschaffen und an die Mitglieder zu verleihen, damit diese mal in die weite Welt der freien Netze und Mailboxen schnuppern können. Vorher wird Hartmut herauszufinden versuchen, wie groß das Interesse der Mitglieder an der Ausleihe, an Schulungen und Installationshilfe ist. In diesem Zusammenhang hofft der Club, daß die Clubecke in der DISCOVERY-Mailbox von den DFÜlern auch mal schreibender- und nicht nur lesenderweise frequentiert wird.



3. Die Kassenprüfung von Rainer Schmitz und Uwe Schoberth förderte keine Unregelmäßigkeiten zu Tage, wofür wir uns wieder bei Jutta Obermann bedanken. Aufgrund des Kassenstands von 2459 DM und jährlichen Ausgaben von ca. 700 DM für das Info können wir weiterhin ohne Clubbeitrag auskommen, ausgenommen die Aufnahmegebühr.
4. Mangels neuer Bewerber mußte der alte Vorstand auf Clubkosten abgefüllt und dann wiedergewählt werden.
5. Jens Neueder klagte wieder über Artikelmangel. Daraufhin wurde nochmal bekräftigt, daß alle Themen interessant sind und abgedruckt werden, ausdrücklich auch Artikel zu IBM/MS-DOS. Allerdings wird der Club nicht seinen Namen und seine Ziele ändern, wie von Günther W. Braun vorgeschlagen, um nicht zu einem reinen MS-DOS-Club zu werden.
6. Weiterhin wollen wir wenigstens die Erinnerung an die alten Z80-Kisten wachhalten. Allerdings wurde angeregt, der inhaltlichen Verschiebung des Clubs durch ein neues Logo Rechnung zu tragen. Daher wird ein Wettbewerb zur Gestaltung eines neuen Club-Logos aufgelegt.

Gerald Schröder

## Neues vom Vorstand

Liebe CLUB 80-Mitglieder,

wie Ihr schon dem Protokoll der Jahreshauptversammlung entnehmen konntet, gibt es diesmal einiges neues vom Vorstand zu berichten. Also frisch ans Werk!

### 1. DFÜ im CLUB 80

Leider ist die CLUB 80-Ecke in Klaus Hermanns DISCOVERY-Mailbox nicht sehr gut besucht. Da es recht frustrierend ist festzustellen, daß schon wieder mal keine neuen Mails eingegangen sind, finde selbst ich mich dort nur noch äußerst selten ein.

Zunächst verwunderte mich das etwas, da unter Betreibern von MSDOSen die Datenfernübertragung inzwischen fast schon zur Selbstverständlichkeit geworden ist. Nach einigen Gesprächen mit Besitzern von CP/M- und Z80-Maschinen wurde die Ursache für die mangelnde Beteiligung an dieser modernen Form der Kommunikation sichtbar.

Während es für MSDOSen sehr einfach ist, ein passendes Modem zu beschaffen, haben CP/Mler da teilweise erhebliche Probleme. Dies liegt vor allem daran, daß sich Grundfragen des Modembetriebs für CP/M- und Z80-Rechner nicht pauschal beantworten lassen. Dies wiederum hängt mit der oftmals sehr unterschiedlichen Hardware der genutzten Rechner zusammen.

Viele dieser Fragen ließen sich klären, wenn man mal ein Modem zum Testen zur Verfügung hätte. Aber wer möchte schon gerne ein paar Hunderter in den Kauf eines Modems investieren, welches hinterher mit dem eigenen Rechner garnicht oder nicht richtig zusammenarbeitet? Sicherlich niemand!

Genau hier möchte der CLUB 80 seinen Mitgliedern zur Seite stehen. Dazu sollen einige Modems angeschafft werden, die sich die Mitglieder dann für einen begrenzten Zeitraum und gegen Hinterlegung des Kaufpreises als Verrechnungsscheck zum Testen ausleihen können. Zusätzlich soll der Ausleiher Hilfe bei der Installation und seinen ersten Gehversuchen im DFÜ-Dschungel bekommen.

Um festzustellen, ob für einen solchen Service tatsächlich Bedarf besteht, muß ich mal wieder eine Fragebogenaktion starten. Da das Erscheinen dieses Hefes gerade in die Urlaubszeit fällt, wird als Einsendeschluß für den Fragebogen der 31. Oktober festgelegt. Das sollte eigentlich reichen, um allen Interessenten die Möglichkeit zu geben, sich zu melden. Bei entsprechendem Bedarf wird dann entschieden, welche und wieviele Geräte angeschafft werden.

Spätestens zu Beginn nächsten Jahres soll die Aktion, entsprechendes Interesse vorausgesetzt, dann anlaufen. Ich würde mich über eine rege Beteiligung freuen!

### 2. CLUB 80-Logo

Bei der Hauptversammlung wurde angeregt, dem CLUB 80 doch ein etwas moderneres Logo zu verpassen. Dabei sollte vor allem die Tatsache nicht zu kurz kommen, daß der CLUB 80 inzwischen nicht mehr von den Tandy TRS 80-Rechner, sondern von CP/M-Maschinen sehr unterschiedlicher Bauart dominiert wird.

Da schon das bisher verwendete Logo des CLUB 80 im Rahmen eines Ideenwettbewerbs entstand, wollen wir auch diesmal so verfahren. Hiermit sind also alle CLUB 80-Mitglieder aufgerufen, ihre Kreativität spielen zu lassen. Die Einsendungen sollten folgende Bedingungen erfüllen:

1. Ausführung in schwarz/weiß,
2. nicht größer als DIN A5,
3. geistiges Eigentum des Einsenders,

Eure Vorschläge für das neue CLUB 80-Logo solltet ihr an meine Adresse (findet ihr im Impressum) schicken. Einsendeschluß ist der 31. Dezember 1994. Eure Einsendungen werden alle im ersten nach dem Einsendeschluß erscheinenden Info veröffentlicht. Die Entscheidung, welches das neue CLUB 80-Logo wird, trifft die nächste Hauptversammlung.

Selbstverständlich soll sich das Mitmachen bei dem Ideenwettbewerb lohnen. Nach dem Motto "Ohne Preis keinen Fleiß", wird hiermit ein postzugelassenes 2400 Baud-Modem als Preis für den Gewinner des Wettbewerbs ausgesetzt. Dann zeichnet mal schön!

### 3. CLUB 80-Bibliothek

Die Büchersammlung des CLUB 80 ist leider eines unserer Sorgenkinder. Kurt Müller, der die Bibliothek seit einigen Jahren unter seinen Fittichen hat, klagt über zu geringe Ausleihraten. Im letzten Jahr wurde nicht ein einziges Buch entliehen.

An der Attraktivität der Bücher kann es eigentlich nicht liegen. Teilweise handelt es sich schon um richtige Raritäten, die auf dem freien Markt schon lange nicht mehr erhältlich und entsprechend gefragt sind.

Kurt Müller wird die Bibliothek in den nächsten Tagen an Rainer Schmitz abgeben. Dieser wird versuchen, das Interesse an der Bibliothek durch die eine oder andere Buchvorstellung nochmals anzuregen. Wenn nach dieser Aktion immer noch keine Bücher ausgeliehen werden, werden sie zur Versteigerung freigegeben.

Letzteres würde mir sehr leid tun, da damit eine Einrichtung, die der CLUB 80 für seine Mitglieder betreibt, unwiederbringlich verloren ginge.

### 4. Info-Beiträge

Ein anderes Sorgenkind des CLUB 80-Vorstands ist das Info, von dem Ihr inzwischen immerhin die vierundvierzigste Ausgabe in der Hand haltet. Vielleicht wird es statt eines Infos demnächst nur noch Vorstandsroundschreiben geben. Das wird dann der Fall sein, wenn die einzigen Beiträge, die zur Veröffentlichung kommen, die sind, in denen Jens oder ich Euch zu mehr Mitarbeit auffordern.

Wenn das Info stirbt, wird es keinen CLUB 80 mehr geben! Und wenn Ihr nicht aktiv am Info mitarbeitet, wird es bald sterben!

In der Hoffnung, demnächst mal ein bißchen was von Euch zu lesen,



## 10 Jahre Club 80 - eine kleine Club-Historie von Gerald Schröder

Hartmut stellte im Vorfeld der diesjährigen Hauptversammlung fest, daß der Club 80 in diesem Jahr sein 10-jähriges Jubiläum feiert. Also wurde ich beauftragt, die Clubhistorie anhand der alten Infos und unseres teilweise lückenhaften Gedächtnisses zu rekonstruieren.

Der Club 80 wurde im März 1984 von Günther Wagner gegründet. Nach einigen Veröffentlichungen der Adresse in Zeitschriften wie der CHIP strömten die Mitglieder scharenweise dem Club zu. Mitglied Nr. 2 war Hartmut Obermann, Nr. 3 Manfred Held, als Nr. 4 traf Jens Neueder ein, die Nr. 9 bildete Gerald Schröder. Von dem ersten Dutzend Mitglieder (Stand Juli 84) sind nur noch diese vier im Club.

Anfang 1985 war der Club auf 45 Mitglieder gewachsen, der Höchststand mit 70 Mitgliedern war nach ca. 3 Jahren erreicht. Durch Kartelleichen und mangels Clubbeitrag konnte dieser Mitgliederstand gehalten werden, bis Hartmut im letzten Jahr den FzdAM verschickte, der den "realen" Mitgliederstand von 51 Mitglieder offenbarte.

Das Clubinfo erblickte schon 1984 das Licht der Welt. Bis zur Nr. 6 übernahm Günther Wagner selbst die Redaktion, aber mit der wachsenden Anzahl von Artikeln, insbesondere von Hartmut Obermann, mußte ein hauptamtlicher Chefredakteur angeworben werden und fand sich mit Jens Neueder, der seit März 1985 (!), also seit 9 Jahren bzw. 38 Infos und dutzenden von Sonderinfos das Zusammenstellen des Infos, das Kopieren und das Versenden übernimmt!

An dieser Stelle sollten wir auch unsere inoffizielle Kassenwartin Jutta Obermann erwähnen, die schon so lange die Finanzen des Clubs in Ordnung hält, daß niemand mehr weiß, wann sie eigentlich damit angefangen hat.

Die Vorsitzenden des Clubs wechselten anfangs recht regelmäßig, in den letzten Jahren trat allerdings eine gewisse Beruhigung bei der Bildung des Vorstands ein. Den Anfang machte der Clubgründer Günther Wagner, der auf dem ersten Clubtreffen 1985 in Reinheim, bei dem 21 der 40 Mitglieder anwesend waren (!), für eine zweite Amtszeit bestätigt wurde. Beim Clubtreffen 1986 in Holzhausen übernahm Peter Stevens den Vorsitz, mußte aber mitten in seiner Amtszeit durch seinen Stellvertreter Hartmut Obermann abgelöst werden.

Hartmut wurde dann 1987 in Alsfeld von 16 Mitgliedern als 1. Vorsitzender eingesetzt. 1988 und 1989 trafen sich jeweils 21 Leute in Idstein, wobei die Youngsters Gerald Schröder und Alexander Schmid an die Spitze des Clubs gesetzt wurden. 1990 in Weilburg tauchte Fritz Chwolka auf, der sofort Mitglied, damit 18 Teilnehmer des Treffens und 1. Vorsitzender des Clubs wurde (es wird kolportiert, daß Fritz' Bereitschaft zur Übernahme des Vorsizes durch Helmut Bernhards Frage an Fritz "Darf ich Dir ein Bier ausgeben?" und der positiv ausfallenden Antwort erreicht wurde).

1991 fiel das Clubtreffen mangels Beteiligung aus, da aber Gerald Schröder schon seine Bahnkarte gekauft hatte und sich daher bitterlich bei Hartmut beschwerte, trafen sich diese beiden doch - und es tauchten noch 8 andere Mitglieder auf.

Im nächsten Jahr war der Tiefststand der Clubaktivitäten erreicht - statt Club- wurde nur ein Vorstandstreffen abgehalten. Fritz blieb mangels anderer Bewerber an der Spitze des Clubs.

1993 kamen in Burgau-Röfingen 9 Mitglieder zusammen und Hartmut übernahm wieder die Geschicke des Clubs. 1994 konnten wir wieder sensationelle 14 Teilnehmer plus 4 Gäste verzeichnen und Hartmut behielt seinen Posten.

Als Kuriosität am Rande sollten wir die Eintragung des Clubs ins Vereinsregister vermerken: Seit dem ersten Clubtreffen wurde diese Eintragung dauernd vorbereitet, durch Beschlüsse und mit ständig revidierten Satzungen untermauert, dann aber fallengelassen, nachdem festgestellt, daß zur Eintragung der gesamte Vorstand zu einem Ort und einer Zeit präsent sein müßte, was in endlicher Zeit nicht zu erwarten ist.

Außerdem wurde auf jedem Treffen die Öffnung des Clubs in Richtung anderer Rechner, insbesondere IBM/MS-DOS, diskutiert und verworfen, da dies unserer "Tradition" nicht entsprechen würde. Faktisch gehen zwar die meisten Mitglieder fremd, aber alle denken mehr oder weniger wehmütig an die guten alten Zeiten und fühlen sich dadurch zusammengehörig.

Gerald Schröder

*Redaktionsschluß für das nächste Clubinfo ist der  
01. September 1994.*

Ich war auf dem Clupstreffen in Erlabrück und war so davon begeistert das ich mich spontan im Clup anmeldete. Hartmut hat mich dann auch noch mich im Info etwas vorzustellen. Dieses werde ich sofort erledigen. Nun ich bin 31 Jahre jung, Verheiratet, 186 cm Groß, mein Markenzeichen ist ein Oberlippenbart und Dunkelbraune Haare. Mit einem Computer kam ich 1985 erstmals in kontakt und diesen habe ich auch heute noch. Es ist ein CPC 464 mit Disketten Laufwerk auf diesem spielt nun mein Großer Sohn Patrick (6) und auch der kleine Christjan (3) versucht hin und wieder mal sein Glück an ihm. Mit CPM kam ich aber erst vor sechs Jahren in behrührung als ich mir einen CPC 6128 zulegte. Etwas später hatte ich das große glück das Alexander Schmid nach Kronach/Gehülz zog (er wohnt etwa 10 Autominuten von mir entfernt) und mich mit dem Z3Plus in behrührung brachte. Nach anfänlichen Schwierigkeiten die mir Alexander aber beseitigte läut es nun ziehmlich ohne Schwierigkeiten. Es ist halt nicht einfach ohne Festplatten laufwerk es voll auszunutzen. Tja nun würde ich noch sagen das ich seit einem halben Jahr einen Joyce 9512 von Steimer Ulrich bekam auf dem ich nun herum hämmere. Ich bin natürlich immer an Sott. und Hardware für Joyce und CPC interessiert auch wenn jemand eine Bastelanleitung hat wie man ein Festplattenlaufwerk an den CPC oder Joyce anschliesen kann so würde mich dieses interessieren. Meine Configuration möchte ich doch nun auch noch nennen :

So da wäre ein CPC 464 von 85 mit DD1 3'' und Robotron 5.25 + 6128 Rom umschaltbar! natürlich mit Grünmonitor.	Nun noch ein CPC 6128 mit Dobbertin 512k Ram erweiterung Star 1c 24 10 Drucker Zwei 5.25 ein 3.5 Laufwerk Selbstbau Schnittstelle aus dem PC1 Modem Farbmonitor und Z3Plus System.	Ija jetzt wäre noch zu erwähuen der Joyce 9512 von Steimer Ulrich auf dem natürlich auch das Z3Plus läut er hat auch noch zwei 5.25 Laufwerke dabei.
--	--	--

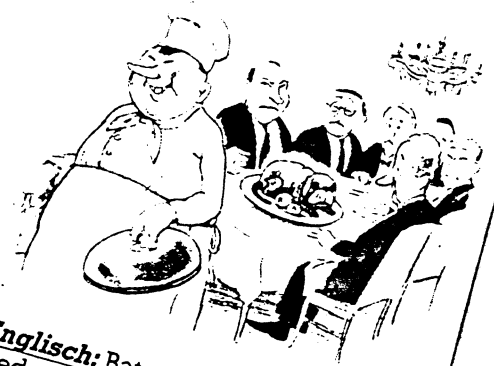
Wenn natürlich jemand von euch an Software für den CPC interessiert ist so kann er sich natürlich auch bei mir melden villeicht habe ich ja auch etwas interessantes für ihn.

Tja das solls fürs erstmalig gewesen sein bis dann sagt euch

Kalb M.

**Arabisch:** Hammam bila mojje.  
Eine Badeanstalt ohne Wasser.  
Müßiges Geplapper.

**Chinesisch:** Kuang yen hsia tzu.  
Ein Blinder mit strahlenden Augen.  
Einer, der mehr scheint, als er ist.  
Kou jou shang pu techeng hsi.  
Du kannst kein Hundefleisch auf einem Bankett servieren.  
Narren sind großen Unternehmungen nicht gewachsen.



**Englisch:** Bats in the belfry.  
Fledermäuse im Glockenturm.  
Verrückt.

**Französisch:** Se noyer dans un verre d'eau.  
In einem Glas Wasser ertrinken.  
Sich unnötige Schwierigkeiten machen.



Miriam Rech,  
Glan-Münchweiler

Grafik des Genie IIIs unter Holte CP/M+

Volker Dose, Egbert Schröder, Juni-August 1994

1. Grundlagen

Der zweite Beitrag zur kleinen Artikelserie über die Ansteuerung der Grafik des Genie IIIs unter CP/M zeigte die Möglichkeiten mit Turbo Pascal. Hier sind nun Möglichkeiten zur Programmierung in ANSI-C.

Thomas Holte selbst hat viele Utilities in C programmiert, nutzte dazu aber den MIC Compiler von Rose. Dieser ist zwar für die unterschiedlichsten Rechner noch zu kaufen, allerdings läßt Herr Rose mit sich nicht über den Preis reden und so bleibt dieser C-Compiler mit ca. 400,- in der heutigen CP/M Welt unbeachtet; schade. Holte hat mit der Überlassung der Source Codes des Genie 3s für den Club 80 auch die C-Routinen zur Graphik-Ansteuerung zur Verfügung gestellt. Deshalb fällt die Entscheidung neue Wege mit dem jetzt PD-Compiler von Hitech zu gehen leicht, zumal dieser den ANSI Standard unterstützt.

Die erste Arbeit, die man nun erledigen muß, ist die Erstellung einer Library für den C-Compiler aus den Holte'schen MIC Routinen. Die eigentliche Graphikfunktionalität ist ja im BIOS implementiert. Dem C-Compiler wird halt nur in den Funktionen gesagt, welche BIOS-Funktion zu nutzen ist und wie. Dazu einige Beispiele:

```

/*****
* CLOSEPL * LIBPLOT015 * Thomas Holte * 860401 *
*****/
*
* Version 1.0 by Thomas Holte.
*
* Closes graphic screen.
*/

closepl ()
{
    system (25, 0); /* disable output on graphic screen of phoenix Genie IIIs */
}

/*****
* POINT * LIBPLOT035 * Thomas Holte * 860401 *
*****/
*
* Version 1.0 by Thomas Holte.
*
* Plots point (x,y).
*/

point (x, y)
{
    int x, y;
    system (27, 0x11, 0, y, x); /* plots point on phoenix Genie IIIs */
}

/*****

```

```

* LINE * LIBPLOT025 * Thomas Holte * 860401 *
*****/
*
* Version 1.0 by Thomas Holte.
*
* Draws a line from (x1,y1) to (x2,y2).
*/

line (x1, y1, x2, y2)
{
    int x1, y1, x2, y2;
}

/* common line drawing routine */
/*
int dx, dy, px, py;
int x = x1;
int y = y1;
int p = 0;

if (x2 >= x1)
{
    px = x2 - x1 + 1;
    dx = 1;
}
else
{
    px = x1 - x2 + 1;
    dx = -1;
}

if (y2 >= y1)
{
    py = y2 - y1 + 1;
    dy = 1;
}
else
{
    py = y1 - y2 + 1;
    dy = -1;
}

point (x, y); /* wurde ja in POINT.C definiert ! */

if (py <= px)
{
    while (x != x2)
    {
        if ((p += py) >= px)
        {
            p -= px;
            y += dy;
        }

        point (x += dx, y);
    }
}
else
{
    while (y != y2)
    {

```

```

if ((p += px) >= py)
  ä
  p -= py;
  x += dx;
  ü
point (x, y += dy);
  ü
  */
/* line drawing routine on phoenix Genie IIIs */
system (29, 0x11, 0, &x1);
  ü
/*=====*/

Diese Library mit Namen HOLTE.LIB enthält die volle Grafikfunktiona-
lität des Genie 3s - selbstverständlich mit den Windows Optionen -
und ist auf einer der V0xx Disk's (für Z3PLUS User) oder direkt bei
uns erhältlich.

/*=====*/
/* Funktionsprototypen der Grafiklibrary für Genie IIIs mit Holte */
/* CP/M + HOLTE.LIB-File */
/*=====*/
/* Volker Dose, Egbert Schroerer */
/* function window not included */

void erase (void); /* loescht HRG Seite 0 */
void openpl (void); /* loescht HRG Seite 0 */
/* und schaltet an */
void closepl (void); /* schaltet HRG Seite 0 aus */
void point (int x, int y); /* setzt einen Punkt (639,449) */

/* zieht Linie (links,unten,rechts,oben) */
void line (int x1, int y1, int x2, int y2);

/* Kreisbogen (mittel_x,y,start_x,y,end_x,y */
void arc (int x, int y, int x0, int y0, int x1, int y1);

void circle (int x, int y, int r); /* Kreis (mittel_x,y,radius) */
void move (int x, int y); /* Grafikcursor nach (x,y) */
void cont (int x, int y); /* draw line from (X,Y) to (x,y) */
void unpoint (int x, int y); /* loesche einen Punkt (639,449) */

void unline (int x1, int y1, int x2, int y2); /* loescht eine Line */

/* loescht einen Kreis mittelpunkt_x,y radius */
void uncircle (int x, int y, int r);

/* loescht einen Kreisbogen mittelpunkt_y,y radius */
void unarc (int x, int y, int x0, int y0, int x1, int y1);

void uncont (int x, int y); /* undraw line from (X,Y) to (x,y) */
/*=====*/

```

Natürlich wird auch - wie bei der Programmierung unter Turbo Pascal - eine Möglichkeit gebraucht, um über den BIOS-Aufruf über BDOS-

Funktion 50 zu realisieren. BIOS3.C befindet sich in der HOLTE.LIB, BIOS3.H wird als Header File in das entsprechende Programm eingebunden werden (s.a. Disk V057). Zusätzlich sind die "system()" Funktionen der original Holte Library eingebunden, um die Spezialfunktionen des BIOS nutzen zu können.

```

/* BIOS-Aufruf fuer CP/M Plus ueber die BDOS-Funktion 50 */

/* CP/M-80 BIOS entry points */
#define BOOT 0 /* perform cold start initialization */
#define WBOOT 1 /* perform warm start initialization */
#define CONST 2 /* check for console input char ready */
#define CONIN 3 /* read console character in */
#define CONOUT 4 /* write console character out */
#define LIST 5 /* write list character out */
#define AUXOUT 6 /* write auxiliary output character */
#define AUXIN 7 /* read auxiliary input character */
#define HOME 8 /* move to track 00 on selected disk */
#define SELDSK 9 /* select disk drive */
#define SETTRK 10 /* set track number */
#define SETSEC 11 /* set sector number */
#define SETDMA 12 /* set DMA address */
#define READ 13 /* read specifidd sector */
#define WRITE 14 /* write specified sector */
#define LISTST 15 /* return list status */
#define SECTRN 16 /* translate logical to physical sector */
#define CONOST 17 /* return output status of console */
#define AUXIST 18 /* return input status of aux. port */
#define AUXOST 19 /* return output status of aux. port */
#define DEVTBL 20 /* return address of char. I/O table */
#define DEVINI 21 /* initialize char. I/O devices */
#define DRVTBL 22 /* return address of disk drive table */
#define MULTIO 23 /* set number of logically consecutive */
/* sectors to be read or written */
#define FLUSH 24 /* force physical buffer flushing for */
/* user-supported deblocking */
#define MOVE 25 /* memory to memory move */
#define TIME 26 /* time set/get signal */
#define SELMEM 27 /* select bank of memory */
#define SETBNK 28 /* specify bank for DMA operation */
#define XMOVE 29 /* set bank when a buffer is in a bank */
/* other than 0 or 1 */
#define USERF 30 /* reserved for system implementor */

#ifdef uchar
#define uchar unsigned char
#endif

/* function prototypes */
unsigned bios(uchar nummer, ...);
unsigned bioshl(uchar nummer, ...);
/*=====*/

```

Um kein Plagiat zu begehen: An BIOS3.\* war Alexander Schmid nicht ganz unbeteiligt.



## 2. Programmierung der Grafik

Die Voraussetzungen sind nun implementiert. Es folgt ein kleines Beispielprogramm, das einige Grafikspielereien erzeugt.

Das Programm GRATEST zeigt dabei alle Möglichkeiten der Grafik. Die Punkte werden nur normiert gesetzt bzw. gelöscht, das schafft Kompatibilität zwischen Rechnern mit verschiedenen Zeichensätzen (z.B. FONT14 und FONT12) und Kreise werden immer normiert gezeichnet.

```

/* GRATEST.C Beispielprogramm mit den
   derzeitigen Grafikmöglichkeiten */

#include <stdio.h>
#include <time.h>
#include <grafik.h>

int main(void);
int rand();

int zaehler, radius, x0, y0, x1, y1;
int max_x,max_y,max_int;

#define sub 0x1a

/* draws a box (lower_left to upper_right) */
void box(int xb0, int yb0, int xb1, int yb1)
{
    line(xb0,yb0,xb1,yb0);      /* links unten nach rechts unten */
    line(xb1,yb0,xb1,yb1);     /* rechts unten nach rechts oben */
    line(xb1,yb1,xb0,yb1);     /* rechts oben nach links oben */
    line(xb0,yb1,xb0,yb0);     /* links oben nach links unten */
}

/* undraws a box (lower_left to upper_right) */
void unbox(int xb0, int yb0, int xb1, int yb1)
{
    unline(xb0,yb0,xb1,yb0);    /* links unten nach rechts unten */
    unline(xb1,yb0,xb1,yb1);    /* rechts unten nach rechts oben */
    unline(xb1,yb1,xb0,yb1);    /* rechts oben nach links oben */
    unline(xb0,yb1,xb0,yb0);    /* links oben nach links unten */
}

int main(void)
{
    x0 = 0; y0 = 0; x1 = 100; y1 = 100;
    putchar(sub);
    puts("Hallo Welt, wie geht's denn so ? ");

    openpl();
    line(x0,y0,x1,y1);

```

```

line(100,100,620,400);
circle(300,140,30);
box(30,30,240,100);
box(240,100,630,300);
box(250,110,620,290);
box(260,120,610,280);
box(270,130,600,270);

erase();
for (radius =5; radius <200; radius +=2)
circle(320,220,radius);

for (radius = 5; radius < 200; radius +=2)
uncircle(320,220,radius);

move (330,280);
srand (2204);
for (zaehler = 0; zaehler <120; zaehler ++)
{
    x0 = rand() / 52;
    y0 = rand() / 80;
    cont(x0,y0);
}

move(330,280);
srand(2204); /
for (zaehler = 0; zaehler <121; zaehler ++)
{
    x0 = rand() / 52;
    y0 = rand() / 80;
    uncont(x0,y0);
}

srand(2204);
for (zaehler =0; zaehler < 52; zaehler ++)
{
    x0 = rand() / 52; x1 = rand() / 52;
    y0 = rand() / 80; y1 = rand() / 80;
    box(x0,y0,x1,y1);
}

srand(2204);
for (zaehler =0; zaehler < 52; zaehler ++)
{
    x0 = rand() / 52; x1 = rand() / 52;
    y0 = rand() / 80; y1 = rand() / 80;
    unbox(x0,y0,x1,y1);
}

srand(2204);
for (zaehler = 0; zaehler < 80; -zaehler ++)
{
    x0 = rand() / 52; y0 = rand() / 80;
    radius = ( rand() / 1000 ) + 5;
    circle(x0, y0, radius);
}

srand(2204);

```

```

for (zaehler = 0; zaehler < 80; zaehler ++)
  ä
    x0 = rand() / 52; y0 = rand() / 80;
    radius = ( rand() / 1000 ) + 5;
    uncircle(x0, y0, radius);
  ü

srand(2204);
for (zaehler = 0; zaehler < 1500; zaehler ++)
  ä
    x0 = rand() / 52; y0 = rand() / 80;
    point(x0,y0);
  ü

srand(2204);
for (zaehler = 0; zaehler < 1500; zaehler ++)
  ä
    x0 = rand() / 52; y0 = rand() / 80;
    unpoint(x0,y0);
  ü

puts("Fertig, Taste um zurueck ins Betriebssystem !!");
getch();

closepl();

ü
/*=====*/

Jetzt sollten die Grundlagen klar sein, um ein komplexeres C-
Programm vorzustellen, den PCX - Reader.

/* ----- */
/* dispcx.c - Read & Display a *.PCX graphics file in Microsoft's */
/* Quick C in EGA and in Borland's Turbo C 2.0 and equivalents with */
/* all supported BGI graphics modes (VGA/EGA/CGA/Herc). */
/* ----- */
/* Original by           : M. CLYNES   07-24-89 */
/* Borland Compiler Enhancements by : J. Braatz  02-16-92 */
/* Adapted to Club-80 Terminal by   : A. Schmid  06-07-93 */
/* adapted to Genie IIIs by         : E. Schroeer 18-11-93 */
/* ----- */
/* WARNING Only with LIBZ and HOLTE !!! */

#include <stdio.h>
#include <conio.h>
#include <grafik.h>

#define YMAX 449 /* Genie IIIs xmax=639,ymax=449 */
#define XMAX 639

char inversfl='N';
int xoff=0,yoff=0;
int eof_flag=0;

struct pcxheaderä
char manufact; /* 0 Hersteller-ID */
char version; /* 1 Versionsnummer */
char encode; /* 2 */

```

```

char bpp; /* 3 */
int xmin; /* 4, 5 Bildgroesse */
int ymin; /* 6, 7 " */
int xmax; /* 8, 9 " */
int ymax; /* 10,11 " */
int hres; /* 12,13 Aufloesung der Hardware */
int vres; /* 14,15 " " " */
char colormapA480; /* 16-63 */
char reserved; /* 64 momentan reserviert */
char ebene; /* 65 Zahl der Farbebenen */
int bpline; /* 66,67 Byte pro Zeile je Ebene */
int paletinfo; /* 68,69 Art des Bildes */
char blankAS80; /* derzeit keine Informationen */
ü header;

/*=====*/
/* global decoded data char decodebuffA2000; */
/*=====*/
/* Decode a line of .PCX */
/* This procedure reads one encoded line from the image file */
/* 0 = valid data stored */
/* EOF = out of data in file */

decodepcx(decodebuff, bytesline, fp)
char *decodebuff; /* where to place data */
int bytesline; /* # of bytes per line */
FILE *fp; /* image file handle */
  ä
    int data;
    int cnt;

    while(bytesline > 0) ä
      cnt = 1;
      if(EOF == (data = getc(fp))) return(EOF); /* retrieve a data byte */
      if(0xc0 == (0xc0 & data)) ä
        cnt = 0x3f & data; /* get repeat count */
        if(EOF == (data = getc(fp))) return(EOF); /* get real data */
      ü

      while(cnt-->0) ä /* expand data into buffer */
        if(data==0x1A) eof_flag=1; else eof_flag=0;
        if(eof_flag && data==0x1A) data=0;
        *decodebuff = data;
        ++decodebuff;
        --bytesline;
      ü

      return(0); /* file read status */
    ü
  /* decodepcx() end */

/*=====*/
/* read .PCX header into structure */
/*=====*/
readheader(fp)
FILE *fp;
  ä
    int count;

```

```

char *loader;

loader = (char *) &header;

for(count = 0; count < 128; ++count)ä
    *loader = (char)fgetc(fp);
    ++loader;
    ü

/* Note: For Borland Compiler users, the following will need to be
/* changed to outtext() if you want to see the header information on
/* Hercules monitors (with the old herc.bgi driver) */

printf("Encoding type %d\n",header.encode);
printf("Window X-min = %4d, Y-min = %4d\n",header.xmin, header.ymin);
printf("Window X-max = %4d, Y-max = %4d\n",header.xmax, header.ymax);
printf("Hres = %4d, Vres = %4d\n",header.hres, header.vres);
printf("# bytes per/scan line %4d  \n\n",header.bpline);
printf("Bild (i)nvers oder (n)ormal einlesen ? ");
inversfl=getch() & 0x5f;

return(header.bpline);
ü

/*=====*/
/*          pixel line dump          */
/*=====*/
/* DUMP A RASTER LINE TO DISPLAY      */
/* pixeldump((pointer buffer), (# of BYTES), (Y line on screen to dump)) */
/*=====*/

pixeldump(rdbuff, bytes, line)
char *rdbuff;
int line, bytes;
ä
unsigned x;
unsigned char byte, loop;

bytes = (bytes * 8);

/* Sehr grosses Bild in X-Richtung komprimieren */

if (header.xmax-header.xmin+xoff>YMAX*2)ä
for(x = xoff; x < (bytes/2)+xoff; ++rdbuff)ä
    if (inversfl=='I')
        byte = β(*rdbuff);
    else
        byte = *rdbuff;
    for (loop = 0x20; loop > 0; loop >>= 1)ä
        if (loop & byte) point(x, YMAX-line);
        loop >>= 1;
        if (loop & byte) point(x, YMAX-line);
        ++x;
    ü
elseä
for(x = xoff; x < bytes+xoff; ++rdbuff)ä

```

```

        if (inversfl=='I')
            byte = β(*rdbuff);
        else
            byte = *rdbuff;
        for (loop = 0x80; loop > 0; loop >>= 1)ä
            if (loop & byte) point(x, YMAX-line);
            ++x;
        ü
    ü
    /* pixeldump() End */

/*===== MAIN() =====*/

int main( argc, argv )
int argc;
char *argvA20;
ä
int line, status, bytesline;
char decodebuffA2000,ch;
FILE *fp;

z3vinit();

argc = argc; /* Nuke's Compiler warning */

cls();

if((fp = fopen(argvA10,"rb")) == NULL)ä
    printf("ERROR: FILE '%s' CANNOT BE OPENED.\n",argvA20);
    exit(0);
    ü
bytesline = readheader(fp);
printf("\n\nOffset aus dem Header nehmen ?");
ch=getch() & 0xDF;
if((ch=='J') öö (ch=='Y') öö (ch==0x0D))ä
    xoff=header.xmin;
    yoff=header.ymin;
    ü
elseä
    printf("\n X-Offset: "); scanf("%d",&xoff);
    printf("\n Y-Offset: "); scanf("%d",&yoff);
    ü
line = yoff;
status = 0;

printf("\n\nBildschirm vor dem Laden loeschen ? ");
if((ch=getch() & 0xDF)=='J') erase();
openpl();
cls();

while(status != EOF)ä /* do EXTRA READS to shrink vertical */
    if ((header.ymax-header.ymin+yoff)>YMAX*2)ä /* sehr grosses Bild */
        status = decodepcx(decodebuff, bytesline, fp);
        status = decodepcx(decodebuff, bytesline, fp);
        status = decodepcx(decodebuff, bytesline, fp);
        status = decodepcx(decodebuff, bytesline, fp);

```



Datum : 11-07-94 22:32 (PRIVAT) (EMPFANGEN) (GESENDET)  
Von : Alexander Schmid bei [77.7700740]  
An : Jens Neudecker bei [77.7700740]

## Assembler-Tip

Wenn innerhalb von Programmen Pufferbereiche vorkommen, mache ich es während der Entwicklung eigentlich immer so, daß ich im COM-File einen Text zur Identifizierung einsetze, was dann z.B. so aussieht:

```
0100 C3 45 02 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0110 42 75 66 66 65 72 20 31 11 11 11 11 11 11 11 Buffer 1.....  
0120 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 .....  
0130 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 .....  
0140 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 .....  
0150 42 75 66 66 65 72 20 32 22 22 22 22 22 22 22 Buffer 2".....  
0160 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 ".....  
0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

So sieht man nachher mit einem Debugger o.ä., ob man z.B. einen Stackbereich viel zu groß gemacht hat, oder ob man vielleicht einen anderen Puffer überschreibt. Im Quelltext steht dazu dann irgendwo:

```
buf1: defb 'Buffer 1' ; Markierungstext  
      defs 56,11h ; Rest des Puffers füllen  
buf2: defb 'Buffer 2'  
      defs 24,22h
```

Um aber nicht immer lange die Zeichen für die restlichen Füllbytes zählen zu müssen, habe ich mir folgendes kleines Macro gebastelt:

```
resbuf macro length,text,fill  
  local x1,x2,x3  
x1: defb text ; Text einsetzen  
x2: equ $-x1 ; Textlänge  
x3: equ length-x2 ; verbleibende Bytes  
     defs x3,fill ; Füllbytes einsetzen  
endm
```

Man gibt nur die Gesamtgröße des Puffers, den Text und das Füllbyte an, der Rest geht automatisch. In einem Programm sieht das dann so aus:

```
org 100h  
jp start  
ds 13  
buf1: resbuf 40h,'Buffer 1',11h  
buf2: resbuf 20h,'Buffer 2',22h  
.  
.  
.
```

Viele ältere Assembler erlauben es nicht, das Füllbyte bei DEFS frei zu wählen, dann muß man das Macro und den Aufruf entsprechend ändern.

Alexander Schmid

Datum : 11-07-94 22:33 (PRIVAT) (EMPFANGEN) (GESENDET)  
Von : Alexander Schmid bei [77.7700740]  
An : Jens Neudecker bei [77.7700740]

## Textumsetzung die N + erste

Es gibt ja schon diverse Programme, um die Umlaute zwischen CP/M und DOS umzusetzen und ich hatte das auch schon mal in einem Programm zusammengefaßt, aber das Programm war bis jetzt auf MSDOS-Umlaute beschränkt. Als ich mal wieder bei Hartmut war, hatte er die Idee, das Programm auch für Windows anzupassen und so kann man jetzt folgende Umwandlungen machen:

```
CP/M ASCII --> MSDOS Text  
CP/M ASCII --> MSDOS Wordstar  
MSDOS Text --> CP/M ASCII  
MSDOS Wordstar --> CP/M ASCII  
MSDOS Text --> CP/M ae, ue, oe  
MSDOS Wordstar --> CP/M ae, ue, oe  
MSDOS Text --> Windows  
Windows --> MSDOS Text  
CP/M ASCII --> Windows  
Windows --> CP/M ASCII
```

Bei den CP/M-Texten wird zusätzlich das 7. Bit gelöscht, sodaß auch Wordstar-Texte vom CP/M ohne Probleme umgesetzt werden können. Wer das Listing nicht abtippen will, kann es in der Discovery per DFÉ holen, oder von mir auf Floppy bekommen.

```
(* Textumsetzungen zwischen CP/M und IBM *)  
(* A. Schmid, 5/93-6/94 *)  
(* compilierbar mit TP 3.0 - 6.0 unter CP/M und DOS *)
```

```
TYPE str15=STRING[15];
```

```
VAR infile,outfile,temp,mode:str15;  
     b,c:CHAR;  
     t,z:TEXT;
```

```
PROCEDURE snow_mode;  
BEGIN  
  WriteLn('Modi sind: 0 - CP/M ASCII -> IBM Text');  
  WriteLn(' 1 - CP/M ASCII -> IBM WS');  
  WriteLn(' 2 - IBM Text -> CP/M ASCII');  
  WriteLn(' 3 - IBM WS -> CP/M ASCII');  
  WriteLn(' 4 - IBM Text -> CP/M ae,ue,oe');  
  WriteLn(' 5 - IBM WS -> CP/M ae,ue,oe');  
  WriteLn(' 6 - IBM Text -> Windows Text');  
  WriteLn(' 7 - Windows Text -> IBM Text');  
  WriteLn(' 8 - CP/M ASCII -> Windows Text');  
  WriteLn(' 9 - Windows Text -> CP/M ASCII');  
  WriteLn;  
END;
```

```
PROCEDURE con0; (* CP/M ASCII file -> IBM text file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    c:=Chr(Ord(c) AND $7F);
    IF c IN [#31,'@','[','\','|','~'] THEN
    BEGIN
      CASE c OF
        #31 :b:=#45;
        #64 :b:=#21;
        #91 :b:=#142;
        #92 :b:=#153;
        #93 :b:=#154;
        #123:b:=#132;
        #124:b:=#148;
        #125:b:=#129;
        #126:b:=#225;
      END;
      Write(z,b); Write('.');
    END
    ELSE
      Write(z,c);
  END;
END;
```

```
PROCEDURE con1; (* CP/M ASCII file -> IBM WS file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    c:=Chr(Ord(c) AND $7F);
    IF c IN ['@','[','\','|','~'] THEN
    BEGIN
      Write(z,#27);
      CASE c OF
        #64 :b:=#21;
        #91 :b:=#142;
        #92 :b:=#153;
        #93 :b:=#154;
        #123:b:=#132;
        #124:b:=#148;
        #125:b:=#129;
        #126:b:=#225;
      END;
      Write('.');
      Write(z,b);
      Write(z,#28);
    END
    ELSE
      Write(z,c);
  END;
END;
```

```
PROCEDURE con2; (* IBM text file -> CP/M ASCII file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    CASE c OF
      #21 : BEGIN c:=#64; Write('.'); END;
      #142: BEGIN c:='['; Write('.'); END;
      #153: BEGIN c:='\'; Write('.'); END;
      #154: BEGIN c:']'; Write('.'); END;
      #132: BEGIN c:='{'; Write('.'); END;
      #148: BEGIN c:='|'; Write('.'); END;
      #129: BEGIN c:='}'; Write('.'); END;
      #225: BEGIN c:='~'; Write('.'); END;
    END;
    c:=Chr(Ord(c) AND $7F);
    Write(z,c);
  END;
END;
```

```
PROCEDURE con3; (* IBM WS file -> CP/M ASCII file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    IF (c=#27) OR (c=#155) THEN
    BEGIN
      Read(t,c);
      CASE c OF
        #21 : b:=#64;
        #142: b:='[';
        #153: b:='\';
        #154: b:']';
        #132: b:='{';
        #148: b:='|';
        #129: b:='}';
        #225: b:='~';
      END;
      Write('.');
      Write(z,b);
      Read(t,c);
    END
    ELSE
      BEGIN
        c:=Chr(Ord(c) AND $7F);
        Write(z,c);
      END;
  END;
END;
```

```
PROCEDURE con4; (* IBM text -> CP/M ae,oe,ue *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    CASE c OF
      #21 : BEGIN Write(z,#64); Write('.'); END;
      #142: BEGIN Write(z,'A'); Write(z,'e'); Write('.'); END;
      #153: BEGIN Write(z,'O'); Write(z,'e'); Write('.'); END;
      #154: BEGIN Write(z,'U'); Write(z,'e'); Write('.'); END;
      #132: BEGIN Write(z,'a'); Write(z,'e'); Write('.'); END;
      #148: BEGIN Write(z,'o'); Write(z,'e'); Write('.'); END;
      #129: BEGIN Write(z,'u'); Write(z,'e'); Write('.'); END;
      #225: BEGIN Write(z,'s'); Write(z,'s'); Write('.'); END;
    ELSE
      c:=Chr(Ord(c) AND $7F);
      Write(z,c);
    END;
  END;
END;
```

```

PROCEDURE con5; (* IBM WS -> CP/M ae,oe,ue *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    IF (c=#27) OR (c=#155) THEN
    BEGIN
      Read(t,c);
      CASE c OF
        #21 : BEGIN Write(z,#64);           END;
        #142: BEGIN Write(z,'A'); Write(z,'e'); END;
        #153: BEGIN Write(z,'O'); Write(z,'e'); END;
        #154: BEGIN Write(z,'U'); Write(z,'e'); END;
        #132: BEGIN Write(z,'a'); Write(z,'e'); END;
        #148: BEGIN Write(z,'o'); Write(z,'e'); END;
        #129: BEGIN Write(z,'u'); Write(z,'e'); END;
        #225: BEGIN Write(z,'s'); Write(z,'s'); END;
      END;
      Write('.');
      Read(t,c)
    END
    ELSE
    BEGIN
      c:=Chr(Ord(c) AND $7F);
      Write(z,c);
    END;
  END;
END;

```

```

PROCEDURE con6; (* IBM text file -> Windows text *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    CASE c OF
      #21 : BEGIN c:=#A7; Write ('. '); END;
      #142: BEGIN c:=#C4; Write ('. '); END;
      #153: BEGIN c:=#D6; Write ('. '); END;
      #154: BEGIN c:=#DC; Write ('. '); END;
      #132: BEGIN c:=#E4; Write ('. '); END;
      #148: BEGIN c:=#F6; Write ('. '); END;
      #129: BEGIN c:=#FC; Write ('. '); END;
      #225: BEGIN c:=#DF; Write ('. '); END;
    END;
    Write(z,c)
  END;
END;

```

```

PROCEDURE con7; (* Windows text -> IBM text file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    CASE c OF
      #A7: BEGIN c:=#21 ; Write ('. '); END;
      #C4: BEGIN c:=#142; Write ('. '); END;
      #D6: BEGIN c:=#153; Write ('. '); END;
      #DC: BEGIN c:=#154; Write ('. '); END;
      #E4: BEGIN c:=#132; Write ('. '); END;
      #F6: BEGIN c:=#148; Write ('. '); END;
      #FC: BEGIN c:=#129; Write ('. '); END;
      #DF: BEGIN c:=#225; Write ('. '); END;
    END;
    Write(z,c)
  END;
END;

```

```

PROCEDURE con8; (* CP/M ASCII file -> Windows text *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    c:=Chr(Ord(c) AND $7F);
    IF c IN [#31,'@','[\','\','\'],'{','|','}','~'] THEN
    BEGIN
      CASE c OF
        (* #31 :b:=#45; *)
        #64 :b:=#A7;
        #91 :b:=#C4;
        #92 :b:=#D6;
        #93 :b:=#DC;
        #123:b:=#E4;
        #124:b:=#F6;
        #125:b:=#FC;
        #126:b:=#DF;
      END;
      Write(z,b); Write('. ');
    END
    ELSE
    BEGIN
      Write(z,c);
    END;
  END;
END;

```

```

PROCEDURE con9; (* Windows text -> CP/M ASCII file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    CASE c OF
      #A7: BEGIN c:=#64; Write ('. '); END;
      #C4: BEGIN c:= '['; Write ('. '); END;
      #D6: BEGIN c:= '\'; Write ('. '); END;
      #DC: BEGIN c:= ']'; Write ('. '); END;
      #E4: BEGIN c:= '{'; Write ('. '); END;
      #F6: BEGIN c:= '|'; Write ('. '); END;
      #FC: BEGIN c:= '}'; Write ('. '); END;
      #DF: BEGIN c:= '~'; Write ('. '); END;
    END;
    c:=Chr(Ord(c) AND $7F);
    Write(z,c)
  END;
END;

```

```

BEGIN
  IF ParamCount=0 THEN
  BEGIN
    WriteLn('IBM und CP/M Textumsetzer V1.1 A.Schmid 6/94');
    WriteLn;
    WriteLn('Dieses Programm setzt IBM Texte in CP/M Texte um und umgekehrt');
    WriteLn;
    WriteLn('Syntax: CONVERT infile outfile modus');
    WriteLn;
    show mode;
    REPEAT
      Write('Name des Eingabefiles '); ReadLn(infile);
    UNTIL infile<>'';
    IF Pos('.',infile)=0 THEN
      temp:=infile+'.xxx'
    ELSE
      temp:=copy(infile,1,Pos('.',infile))+'.xxx';
    Write('Name des Ausgabefiles (ENTER=',temp,',) '); ReadLn(outfile);
    IF outfile='' then outfile:=temp;
  END;

```

```
REPEAT
  Write('Modus '); ReadLn(mode);
UNTIL (mode[1] in ['0'..'9']);
WriteLn;
END
ELSE
BEGIN
  infile:=ParamStr(1);
  outfile:='';
  mode:=ParamStr(2); mode:=mode[1];
  IF Not(mode[1] in ['0'..'9']) then
  BEGIN
    outfile:=ParamStr(2);
    mode:=ParamStr(3); mode:=mode[1];
  END;

  IF outfile='' THEN
  BEGIN
    mode:=ParamStr(2); mode:=mode[1];
    IF Pos('.',infile)=0 THEN
      outfile:=infile+'.xxx'
    ELSE
      outfile:=copy(infile,1,Pos('.',infile))+'.xxx';
    WriteLn;
  END;

  IF (mode='') OR (mode>'9') THEN
  BEGIN
    show_mode;
    REPEAT
      Write('Modus : ',#8,#8);
      ReadLn(mode); mode:=mode[1];
    UNTIL mode[1] IN ['0'..'9'];
  END;
END;

writeln('infile=',infile,' outfile=',outfile,' Modus=',mode);writeln;
```

```
Assign(t,infile); (*$I-*) Reset(t); (*$I+*)
IF IOResult<>0 THEN
BEGIN
  WriteLn('Datei ',infile,' konnte nicht geoeffnet werden');
  HALT;
END;
temp:='$$$.$$$';
Assign(z,temp); Rewrite(z);

CASE mode[1] OF
  '0': con0;
  '1': con1;
  '2': con2;
  '3': con3;
  '4': con4;
  '5': con5;
  '6': con6;
  '7': con7;
  '8': con8;
  '9': con9;
END;

Close(t);
Close(z);

IF infile=outfile THEN
BEGIN
  Assign(t,infile); Erase(t);
END;

Assign(t,outfile); (*$I-*) Reset(t); (*$I+*)
WHILE IOResult=0 DO
BEGIN
  WriteLn;
  Write('Ausgabedatei existiert schon, ueberschreiben (J/N) ? ');
  ReadLn(mode); mode:=UpCase(mode[1]);
  IF (mode<>'Y') AND (mode<>'J') THEN
  BEGIN
    Write('Neuer Name: ');
    ReadLn(outfile)
  END
  ELSE
    Erase(t); (* altes outfile loeschen *)
  Assign(t,outfile); (*$I-*) Reset(t); (*$I+*)
END;
Assign(t,temp); Rename(t,outfile);
END.
Alexander Schmid
```



Datum : 11-07-94 22:34 (PRIVAT) (EMPFANGEN) (GESENDET)  
 Von : Alexander Schmid bei [77:7700740]  
 An : Jens Neueder bei [77:7700740]

## Nachtrag und Erweiterung zum PCXPUT.C

Inzwischen habe ich es auch zu einer DOSe gebracht, wenn auch nur zu einem 386sx16, der normalerweise als reiner DFÉ-Knecht dient. So hatte ich aber die Gelegenheit, die mit PCXPUT erzeugten Bildchen auf der DOSe anzusehen und das Erstaunen war groß: geht nicht. Warum nicht? Die DOSe wollte mehr Informationen, als PCXPUT in den Header geschrieben hat. Im einzelnen sind das die Hersteller-ID, die Versionsnummer und die Farbtabelle. PCXSHOW braucht die nicht und daher bin ich erst jetzt drauf gekommen. Manche Programme m-gen es auch nicht, wenn eine Zeile mitten im Byte auf-rt und es ist sinnvoller, das Fenster in 8-Punkt-Schritten zu bewegen. Um nicht unn-tig Verwirrung zu stiften, kommt hier nochmal das mberarbeitete Listing

```

/*****
/* PUTPCX.C
/*
/* Gets pixel data from the screen and writes a PCX format file */
/* Alexander Schmid 9/93 - 7/94
*****/

```

```

#include <stdio.h>
#include <conio.h>
#include <grafmod2.h> /* computer specific graphic driver */

```

```

struct pcxheader{
  uchar manufact; /* 0 Hersteller-ID */
  uchar version; /* 1 Versionsnummer */
  uchar encode; /* 2 */
  uchar bpp; /* 3 */
  int xmin; /* 4, 5 Bildgroesse */
  int ymin; /* 6, 7 " */
  int xmax; /* 8, 9 " */
  int ymax; /* 10,11 " */
  int hres; /* 12,13 Aufloesung der Hardware */
  int vres; /* 14,15 " " " */
}

```

```

uchar colormap[48]; /* 16-63 */
uchar reserved; /* 64 momentan reserviert */
uchar ebene; /* 65 Zahl der Farbebenen */
int bpline; /* 66,67 Byte pro Zeile je Ebene */
int paletinfo; /* 68,69 Art des Bildes */
uchar blank[58]; /* derzeit keine Informationen */
} header;

```

```

uchar colormap[48]={0,0,0,255,255,255,0,170,0,0,170,170,170,
0,0,170,0,170,170,170,0,170,170,170,85,85,
85,85,85,255,85,255,85,85,255,255,255,85,
85,255,85,255,255,255,85,255,255};

```

```
int bytcnt=0;
```

```

/*=====
/* write .PCX header into file
/*=====

```

```

writeheader(fp)
FILE *fp;
{
  int count;
  char *loader; /* Zeiger auf ein character-array */

  loader = (char*) &header; /* LOADER zeigt auf den Header im Speicher */

  for(count = 0; count < 128; ++count){ /* 128 Bytes schreiben */

    putc(*loader,fp);
    ++loader; /* Zeiger eins weiter */
  }
}

```

```

/*=====
/* read pixel from screen
/*=====
unsigned char readscreen(x,y)
int x,y;
{
  return point(x,YMAX-y);
}

```

```

/*-----*/
/*          mark area to write          */
/*-----*/
area(x1,y1,x2,y2)
int *x1,*y1,*x2,*y2;
{
    int xa1,ya1,xa2,ya2;
    char ch;

    hires();                /* Grafik einblenden */
    color=2;                /* Punkt invertieren */
    box(*x1,YMAX-*y1,*x2,YMAX-*y2); /* Ausschnitt markieren */
    while((ch=getch())!=0x0D){ /* auf ENTER warten */
        xa1=*x1; ya1=*y1; xa2=*x2; ya2=*y2; /* Werte merken */
        switch(ch){
            case 'd': if(*x2<=XMAX-8) (*x2)+=8; break; /* rechter Rand -> */
            case 's': if(*x2>*x1+16) (*x2)-=8; break; /* rechter Rand <- */
            case 'x': if(*y2<=YMAX-8) (*y2)+=8; break; /* unterer Rand v */
            case 'e': if(*y2>*y1+16) (*y2)-=8; break; /* unterer Rand ^ */
            case 'D': if(*x1<*x2-16) (*x1)+=8; break; /* linker Rand -> */
            case 'S': if(*x1>=8) (*x1)-=8; break; /* linker Rand <- */
            case 'X': if(*y1<*y2-16) (*y1)+=8; break; /* oberer Rand v */
            case 'E': if(*y1>=8) (*y1)-=8; break; /* oberer Rand ^ */
        }
        box(xa1,YMAX-ya1,xa2,YMAX-ya2); /* alten Rand loeschen */
        box(*x1,YMAX-*y1,*x2,YMAX-*y2); /* neuen Rand ziehen */
    }
    box(*x1,YMAX-*y1,*x2,YMAX-*y2); /* Markierung wieder loeschen */
    textmode(); /* Grafik abschalten */
}

/*-----*/
/*          do run length encoding      */
/*-----*/
compress(arr1,bytes,fp)
unsigned char arr1[];
int bytes;
FILE *fp;
{
    unsigned char arr2[150],ch;
    int repeat,ausgabe=0,eingabe=0;

    while(eingabe<bytes){ /* Zahl der Eingangsbytes */
        ch=arr1[eingabe]; /* Byte aus dem Eingangsarray holen */
        ++eingabe; /* Eingabezeiger eins weiter */
        repeat=1; /* Wiederholungszahler */
        while((ch==arr1[eingabe]) && (repeat<0x3F)){ /* Byte wiederholt sich */
            ++repeat; ++eingabe; /* Zaehler erhoehen */
        }
        if(repeat>1){ /* wenn es Wiederholungen gab */
            arr2[ausgabe]=(char)repeat + 0xC0; /* Zaehlbyte zusammensetzen */
            ++ausgabe; /* Ausgabezaehler erhoehen */
        }
        else{
            if((ch & 0xC0) == 0xC0){ /* Datenbyte mit gesetztem Bit 6 und 7 */
                arr2[ausgabe]=0xC1; /* Zaehlbyte = 1 schreiben */
                ++ausgabe; /* Ausgabezaehler erhoehen */
            }
        }
        arr2[ausgabe]=ch; /* Byte in Ausgabearray schreiben */
        ++ausgabe; /* Ausgabezaehler erhoehen */
    }
    bytcnt += ausgabe; /* geschriebene Bytes zaehlen */
    for(eingabe=0; eingabe<ausgabe; ++eingabe) /* Ausgabearray auf Floppy */
        fputc(arr2[eingabe],fp); /* schreiben */
}

```

```

/*-----*/
/*          write .PCX data            */
/*-----*/
writepcx(x1,y1,x2,y2,fp)
int x1,y1,x2,y2;
FILE *fp;
{
    int x,y,bits;
    unsigned char array[150],byte,count;

    hires();                /* Grafik einblenden */
    for(y=y1; y<=y2; ++y){ /* vom oberen zum unteren Rand */
        count=0; /* Zaehler fuer den Zeilenpuffer */
        for(x=x1; x<=x2; x+=8){ /* vom linken zum rechten Rand */
            byte=0;
            for(bits=0; bits<8; ++bits){ /* Byte zusammenbasteln */
                byte |= readscreen(x+bits,y)<<(7-bits);
            }
            array[count]=byte;
            ++count;
        }
        compress(array,(x2-x1+1)/8,fp); /* Zeile komprimieren und auf */
    } /* Floppy schreiben */
    textmode(); /* Grafik abschalten */
    for(x=0; x<128-bytcnt%128; ++x) /* Rest des letzten Sektors mit */
        fputc(0x1a,fp); /* 1Ah fuellen */
}

/*-----MAIN()-----*/

main( argc, argv )
int argc;
char *argv[];
{
    FILE *fp;
    int x1,y1,x2,y2,xoff=0,yoff=0;
    int a;
    char ch;

    argc = argc;

    if((fp = fopen(argv[1],"rb")) != NULL){
        printf("ERROR: FILE '%s' DOES ALREADY EXIST.\n",argv[1]);
        exit(0);
    }

    if((fp = fopen(argv[1],"wb")) == NULL){
        printf("ERROR: CAN NOT OPEN FILE '%s' \n",argv[1]);
        exit(0);
    }
    x1=0; y1=0; x2=199; y2=199;
    do{
        area(&x1,&y1,&x2,&y2);
        printf("Die Koordinaten sind %d/%d %d/%d\n",x1,y1,x2,y2);
    }while((ch=getch())!=0x0D); /* Korrektur mit RETURN moeglich */

    printf("\nX-/Y-Offsets so lassen ? ");
    if(((ch=getch()) & 0xDF) == 'J') || (ch=='Y') || (ch==0x0D)){
        header.xmin=x1;
        header.ymin=y1;
        header.xmax=x2;
        header.ymax=y2;
    }
}

```

```

else{
printf("\n\n X-Offset: "); scanf("%d",&xoff);
printf(" Y-Offset: "); scanf("%d",&yoff);
header.xmin=xoff;
header.ymin=yoff;
header.xmax=x2-x1+xoff;
header.ymax=y2-y1+yoff;
}
header.manufact=10;
header.version=5;
header.encode=1;
header.bpp=1;
header.hres=XMAX;
header.vres=YMAX;
header.ebene=1;
header.bpline=(x2-x1+1)/8;

for(a=0; a<48; ++a)
header.colormap[a]=colormap[a];

writeheader(fp);
writepcx(x1,y1,x2,y2,fp);
fclose(fp);

exit(0);
}
/* End of main() */

```

Jetzt kann man seine Kreationen (hoffentlich) problemlos mit DOSen austauschen. Bei dem Programm, das ich auf der DOSe habe, wurde es auch ohne die Farbtabelle, also nur mit ID und Versionsnummer, gehen, aber die "fertigen" PCX-Bildchen haben auch immer einer Tabelle und sicher ist sicher. Wie die Werte in der Tabelle genau zustande gekommen sind, weiß ich nicht, aber alle meine schwarz/weiß Bilder haben dieselbe Tabelle, so falsch kann die also nicht sein. Die uchar in der Struktur bedeuten beim Hi-Tech-C unsigned char und ich habe die hier statt char eingesetzt, damit es mit dem Wert 255 keine Probleme gibt. Ihr k-nnt natrlich auch char lassen und ausprobieren, ob es damit bei Euch geht. Um ganz korrekt zu sein, wird auch noch der Rest des letzten Sektors mit 1Ah aufgefüllt.

Alexander Schmid

Datum : 11-07-94 22:36 (PRIVAT) (EMPFANGEN) (GESENDET)  
Von : Alexander Schmid bei (77:7700743)  
An : Jens Neusser bei (77:7700743)

### Nachtrag und Erweiterung zum PCXSHOW.C

PCXSHOW war auch noch nicht ganz wasserdicht, es ist z.B. davon ausgegangen, daß der unbenutzte Rest des letzten Sektors mit 1A gefüllt war und hat dabei nichtmal abgebrochen, sondern nur keinen Punkt gezeichnet. Abgebrochen wurde erst beim physikalischen Dateiende. PCXPUT hatte bis jetzt keine EOPs geschrieben und so war am unteren Rand der Bilder meistens ein Streifen mit MALL. Jetzt wird die Zeilenzahl, die ja im Header steht, ausgewertet und am echten Bildende abgebrochen. Auch wenn das PCX-File defekt ist, wird spätestens am Dateiende abgebrochen.

Eine andere Erweiterung ist die Erkennung eines anderen Formats. Das DOS-Programm PV, mit dem man Bilder u.a. zwischen zahlreichen Formaten konvertieren kann, besteht darauf, PCX-Files mit 8 Bit pro Pixel zu schreiben, was dann z.B. so aussieht: FF 00 FF 01 FF 01. Das wären hier 63 dunkle und danach 126 helle Punkte. Wahrscheinlich werden bei Farbbildern auch die anderen Bits benutzt, aber das alte PCXSHOW konnte damit nichts anfangen.

```

/* ----- */
/* dispcx.c - Read & Display a *.PCX graphics file in Microsoft's */
/* Quick C in EGA and in Borland's Turbo C 2.0 and equivalents with */
/* all supported BGI graphics modes (VGA/EGA/CGA/Herc). */
/* ----- */
/* Original by : M. CLYNES 07-24-89 */
/* Borland Compiler Enhancements by : J. Braatz 02-16-92 */
/* Adapted to Club-80 Terminal by : A. Schmid 06-07-93 - 07-11-94 */
/* ----- */

```

```

#include <stdio.h>
#include <conio.h>
#include <grafmod2.h>
char inversfl='N';
int xoff=0,yoff=0;
int linecnt;

```

```

struct pcxheader{
  char manufact; /* 0 Hersteller-ID */
  char version; /* 1 Versionsnummer */
  char encode; /* 2 */
  char bpp; /* 3 */
  int xmin; /* 4, 5 Bildgroesse */
  int ymin; /* 6, 7 " */
  int xmax; /* 8, 9 " */
  int ymax; /* 10,11 " */
  int hres; /* 12,13 Aufloesung der Hardware */
  int vres; /* 14,15 " " " " */
  char colormap[48]; /* 16-63 */
  char reserved; /* 64 momentan reserviert */
  char ebene; /* 65 Zahl der Farbebenen */
  int bpline; /* 66,67 Byte pro Zeile je Ebene */
  int paletteinfo; /* 68,69 Art des Bildes */
  char blank[58]; /* derzeit keine Informationen */
} header;

```

```

/*=====*/
/* gobal decoded data char decodebuff[200]; */
/*=====*/
/* Decode a line of .PCX */
/* This procedure reads one encoded line from the image file */
/* 0 = valid data stored */
/* EOF = out of data in file */

```

```

decodepcx(decodebuff, bytesline, fp)
char *decodebuff; /* where to place data */
int bytesline; /* # of bytes per line */
FILE *fp; /* image file handle */
{
  int data;
  int cnt;
  while(bytesline > 0 ){
    cnt = 1;
    if(EOF == (data = getc(fp))) return(EOF); /* retrieve a data byte */
    if(0xc0 == (0xc0 & data)){
      cnt = 0x3f & data; /* get repeat count */
      if(EOF == (data = getc(fp))) return(EOF); /* get real data */
    }
  }
}

```

```

while(cnt--){ /* expand data into buffer */
  *decodebuff = data;
  ++decodebuff;
  --bytesline;
}
return(0); /* file read status */
}
/* decodepcx() end */

```

```

/*=====*/
/* read .PCX header into structure */
/*=====*/

```

```

readheader(fp)
FILE *fp;
{
  int count;
  char *loader;

  loader = (char *) &header;

  for(count = 0; count < 128; ++count){
    *loader = (char)fgetc(fp);
    ++loader;
  }
}

```

```

/* Note: For Borland Compiler users, the following will need to be */
/* changed to outtext() if you want to see the header information on */
/* Hercules monitors (with the old herc.bgi driver) */

```

```

printf("Encoding type %d\n",header.encode);
printf("Window X-min = %4d, Y-min = %4d\n",header.xmin, header.ymin);
printf("Window X-max = %4d, Y-max = %4d\n",header.xmax, header.ymax);
printf("Hres = %4d, Vres = %4d\n",header.hres, header.vres);
printf("# bytes per/scan line %4d \n\n",header.bpline/header.bpp);

```

```

if(header.bpp!=1 && header.bpp!=8){
  printf("\n\nSorry ich kann nur Bilder mit 1 oder 8 Bit pro Pixel\n");
  printf("Dieses Bild hat %d Bit/Pixel\n",header.bpp);
  exit(0);
}

```

```

printf("Bild (i)nvers oder (n)ormal einlesen ? ");
inversfl=getch() & 0x5f;

return(header.bpline);
)

/*=====*/
/*          pixel line dump          */
/*=====*/
/* DUMP A RASTER LINE TO DISPLAY      */
/* pixeldump((pointer buffer), (# of BYTES), (Y line on screen to dump)) */
/*=====*/

pixeldump(rdbuf, bytes, line)
char *rdbuf;
int bytes, line;
{
    unsigned x;
    unsigned char byte, loop;
    int xxx;

    if(header.bpp==1) bytes = (bytes * 8);
    if (header.xmax-header.xmin+xoff>XMAX){ /* in X-Richtung komprimieren */
        for(x = xoff; x < (bytes/2)+xoff; ++rdbuf){
            if(header.bpp==1)
                byte = *rdbuf;
            else {
                byte=0;
                for (xxx=0; xxx<8; ++xxx) {
                    byte |= *rdbuf << 7-xxx;
                    ++rdbuf;
                }
            }
            if (inversfl=='I')
                byte = ~byte;

            for (loop = 0x80; loop > 0; loop >>= 1){
                if (loop & byte) plot(x, YMAX-line);
                loop >>= 1;
                if (loop & byte) plot(x, YMAX-line);
                ++x;
            }
        }
    }
    else{
        for(x = xoff; x < bytes+xoff; ++rdbuf){
            if(header.bpp==1)
                byte = *rdbuf;
            else {
                byte=0;
                for (xxx=0; xxx<8; ++xxx) {
                    byte |= *rdbuf << 7-xxx;
                    ++rdbuf;
                }
            }
            if (inversfl=='I')
                byte = ~byte;

            for (loop = 0x80; loop > 0; loop >>= 1){
                if (loop & byte) plot(x, YMAX-line);
                ++x;
            }
        }
    }
}
/* pixeldump() End */

```

```

/*===== MAIN() =====*/

int main( argc, argv )
int argc;
char *argv[2];
{
    int line, status, bytesline;
    static char decodebuff[5000],ch;
    FILE *fp;

    argc = argc; /* Nuke's Compiler warning */

    if((fp = fopen(argv[1],"rb")) == NULL){
        printf("ERROR: FILE '%s' CANNOT BE OPENED.\n",argv[1]);
        exit(0);
    }

    if (argv[2][0]=='1')
        color=1; /* je nach Bild kann setzen oder */
    else
        color=2; /* invertieren besser aussehen */

    bytesline = readheader(fp);
    printf("\n\nOffset aus dem Header nehmen ?");
    ch=getch() & 0xDF;
    if((ch=='J') || (ch=='Y') || (ch==0x0D)){
        xoff=header.xmin;
        yoff=header.ymin;
    }
    else{
        printf("\n X-Offset: "); scanf("%d",&xoff);
        printf("\n Y-Offset: "); scanf("%d",&yoff);
    }
    line = yoff;
    status = 0;

    printf("\n\nBildschirm vor dem Laden loeschen ? ");
    if((ch=getch() & 0xDF)=='J') clsh();
    hires();
}

```

```
linecnt = header.ymax - header.ymin;
while(status != EOF && linecnt>0){ /* do EXTRA READS to srink vertical */
  if ((header.ymax-header.ymin+yoff)>YMAX*2){ /* sehr grosses Bild */
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    --linecnt;
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    --linecnt;
  }
  if (header.ymax-header.ymin+yoff>YMAX){ /* grosses Bild */
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    --linecnt;
  }
  status = decodepcx(decodebuff, bytesline, fp);
  pixeldump(decodebuff, bytesline, line);
  --linecnt;
  ++line;
}
color=2;
grprint(1,1,"PRESS 'ENTER' to continue");
getch();
color=2;
grprint(1,1,"PRESS 'ENTER' to continue"); /* erase text */
textmode();
}
/* End of main() */
```

Ich hoffe, daß ich jetzt alle Schwachpunkte gefunden habe, aber soweit sieht's ganz gut aus.

Alexander Schmid

Datum : 24-08-94 22:36 (PRIVAT) (EMPFANGEN) (GESENDET)  
Von : Alexander Schmid bei [77:7700/740]  
An : Jens Neueder bei [77:7700/740]

## Model-1 Emulator für DOSen

Wer denkt nicht hin und wieder wehmütig an die gute alte Zeit, als man stundenlang Hex-Dumps abgetippt hat und noch niemand an Windows und ähnliche Entgleisungen gedacht hat ? Aber jetzt ist Abhilfe in Sicht. In der Discovery liegt seit einiger Zeit ein Model-1 Emulator und was soll ich sagen, der funktioniert sogar. Das File heißt MODEL1\_B.ZIP und enthält fast alles, was man braucht, um auf der DOSe einen richtigen Rechner zu emulieren. Aus Copyright-Gründen fehlen die ROMs und das DOS und das waren auch die größten Hürden, das Teil zum Laufen zu bringen. Ich hatte vor langer Zeit mal die original ROMs eines Model-1 ausgelesen, aber entweder ist da was schiefgegangen, oder ich habe die Reihenfolge vertauscht, jedenfalls lief er damit nicht. Dann habe ich die ROMs des Genie IIs genommen und damit funktioniert's jetzt einwandfrei. In der Anleitung steht, daß die vier "Laufwerke" des Emulators, die als Files auf der Festplatte verwaltet werden, 80 Track SS SD sind. Das stimmt auch im Prinzip, aber die "Bootfloppy" muß unbedingt 40 Track SS SD haben, sonst steht man im Wald. Die PDRIVE-Parameter erlauben nur eine einzige Einstellung, mit der man auch die Originalfloppy formatieren muß, die dann als Muster dienen soll:

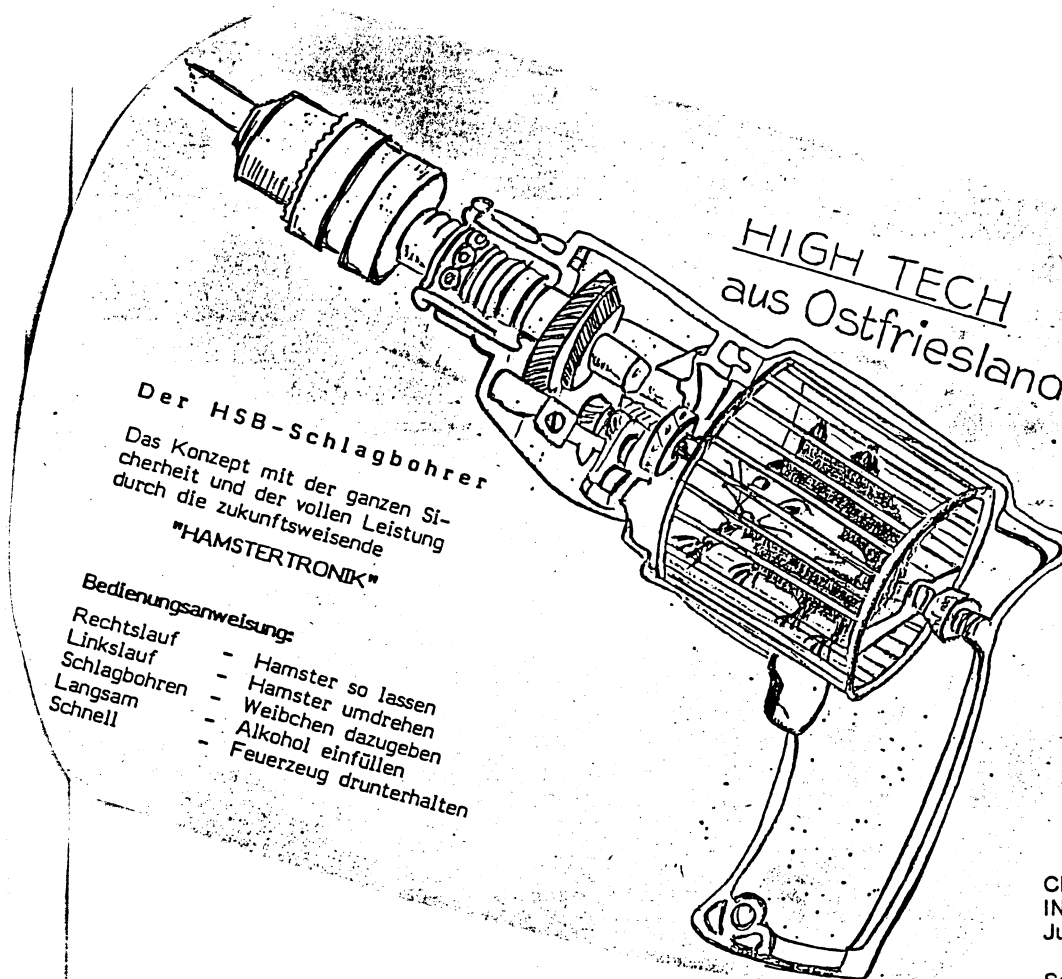
TI=AL TD=A SP=40 SEK=10 SWZ=0 EIB=2 SBIV=17 AEIV=2

Das sind die Namen vom GDOS, aber die Bedeutung für's NewDOS dürfte klar sein. Die "Floppy-Files" sind eine exakte, sektorweise 1:1 Kopie, von der Bootspur bis zu HIT und GAT. Bei den anderen Laufwerken kann man problemlos SP=80 eintragen. Soweit ich es sehe, ist der Emulator 100% kompatibel, er stellt ja auch nur die "Hardware" zur Verfügung und die Software ist original.

Sogar der FORMAT-Befehl merkt nicht, daß er beschummelt wird und läßt schön brav das Sternchen rechts oben blinken und auch die Dreitasten-Befehle (123, DFG und JKL) funktionieren. Ärger machen nur Programme, die Illegals benutzen (z.B. Zeus), die laufen nicht und Verursachen einen Warmstart. Daneben bietet der Emulator einige Funktionen, die auf dem Model 1 so natürlich nicht möglich waren, z.B. auf Tastendruck einen Monitor, der u.a. am Bildschirmrand alle Register und einen Teil des Speichers anzeigt. Auf meinem 386sx16 ist die Arbeitsgeschwindigkeit nicht gerade berauschend, aber auf einem 486er erreicht man durchaus die Originalgeschwindigkeit und etwas mehr. Ein Problem stellt die Übertragung von Programmen dar, es gibt leider kein Tool, mit dem man einzelne Files direkt von der DOSe in den Emulator übertragen kann. Der Emulator unterstützt zwar die serielle Schnittstelle, aber wer hatte die schon für das Model 1 ? Hartmut Obermann und ich hatten angefangen, in Anlehnung an die Anleitung einen Weg über die parallele Schnittstelle auszutüfteln, aber das ist noch nicht lauffähig. Ich habe die Bootfloppy komplett über zwei Ecken mit Hilfe eines CP/M-Programms gelesen und als File zur DOSe überspielt, aber das dürfte für die meisten nicht in Frage kommen. Da muß also noch was getan werden. Mein Genie hat zwar eine serielle Schnittstelle, aber ich habe für NewDos leider kein funktionsfähiges Terminalprogramm gefunden. Beim Kermit funktioniert zwar der Terminalmodus, aber der Filetransfer geht nicht. Hat vielleicht noch jemand ein Terminalprogramm für NewDos das XMODEM kann ? Wenn Bedarf besteht, kann ich mit meiner Methode aber ohne Probleme Floppies mit den wichtigsten Utilities und Programmen zusammenstellen. Am besten schickt Ihr mir dann eine NewDos-Floppy mit den Programmen und ich bastel Euch daraus Files, die der Emulator schluckt. Innerhalb des Emulators können die Files dann beliebig umkopiert werden und mit FORMAT kann man sich sovieler Leerfloppies basteln, wie auf die Festplatte passen. Wäre nett, wenn Ihr die Floppies mit den o.g. Parametern (mit SP=80) formatieren würdet und auch ein paar 3 1/2" Floppies für die DOSen-Files dazu tut.

Wenn man sich für 25 Dollar registrieren läßt, bekommt man eine neuere Version des Emulators, die mehr Funktionen hat und schneller ist und wenn sich genug Interessenten finden, könnten wir auch versuchen, einen Clubrabatt rauszuhandeln. Wenn einer mit der Installation gar nicht zurechtkommt, kann er sich ruhig melden. Die ROMs werden natürlich nur gegen eine amtlich beglaubigte Kopie der original Kaufrechnung rausgegeben ;-)))

Alexander Schmid



## Impressum

1. Vorsitzender: **Hartmut Obermann** Tel.: 07 21/ 85 40 68  
Wilhelm-Baur-Straße 8 BTX: 07 21/  
76135 Karlsruhe FAX: 07 21/ 85 40 68
2. Vorsitzender: **Gerald Schröder** Tel.: 0 41 05/ 26 02  
Am Schützenplatz 14  
21218 Seevetal
- Hardwarekoordinator: **Andreas Magnus** Tel.: 02 09/ 87 02 30  
Bismarckstraße 29  
45879 Gelsenkirchen
- NewDOS-Diskothekar: **Oliver Volz** Tel.: 07 11/ 74 40 51  
Am Ochsenwald 37A  
70565 Stuttgart (Rohrerhöhe)
- CP/M-Diskothekar: **Fritz Chwolka** Tel.: 0 24 64/ 89 20  
Saarstraße 34  
52457 Aldenhoven
- C-128-Diskothekar: **Günther W. Braun**  
Postfach 80 02 26  
81602 München
- Clubbücherei: **Kurt Müller** Tel.: 0 41 52/ 7 06 43  
Sophie-Scholl-Ring 3b  
21502 Geesthacht
- Redaktion: **Jens Neueder** Tel.: 07 91/ 4 28 77  
Geschlachtenbretzingen BTX: 07 91/ 44 47 22  
Rudolf-Then-Straße 32 FAX: 0 79 71/ 2 50 55  
74544 Michelbach / Bilz
- Bankverbindung: **Club 80** Postgiroamt Frankfurt  
Postgiro Sonderkonto CLUB 80 BLZ: 500 100 60  
Obermann H., 8870 Günzburg Kto.Nr.: 498 071 - 605

Autoren: Die Redaktion bedankt sich bei den im  
Inhaltsverzeichnis genannten Autoren  
für die Mitarbeit an der Club-INFO.  
Eine Zensur oder Kontrolle der INFO-Beiträge  
erfolgt nicht.

# Schluß

Hallo Club 80'er,

nachdem ich meinen Urlaub hinter mir habe, ist wieder einmal INFO-Time. Die letzte Hauptversammlung und die Softwareartikel brachten einiges an Infomaterial, so daß wir diesmal doppelt so dick sind wie beim letzten INFO. Ich hoffe, Ihr habt damit mal wieder genügend zum Lesen. Ich fand zum Beispiel die Club-Historie recht spannend. Was in zehn Jahren so alles passiert! Prima wäre auch, wenn in der Discovery etwas mehr Aktivität herrschen würde (nicht nur zu gunsten meines Arbeitgebers, Ihr unterstützt indirekt auch mich!). Die Infoartikel könnt Ihr, wie Ihr sicher alle wißt, auch dort bunkern. Ich hole sie mir spätestens vor dem jeweiligen aktuellen INFO dort ab.

Nichts desto trotz möchte ich, wie immer an dieser Stelle, wieder die Bitte um Futter für das INFO loswerden. Nach der Sommerpause habt Ihr doch sicher schon den einen oder anderen Artikel für mich bereit?

Als Beilage im INFO findet Ihr diesmal einen Fragebogen DFÜ im CLUB 80 und einige Impressionen von unserem 94er Clubtreffen. Die Bilder hat uns Tilmann Reh zur Verfügung gestellt. Vielleicht fühlt sich der ein oder andere dadurch animiert, beim nächsten Mal mit dabei zu sein.

Bis zum nächsten INFO  
grüßt Euch Euer







Clubtreffen '94

