

PC-NEWS

Das offizielle Mitteilungsblatt
des
PCC-TGM

(Personal Computer Club - Technologisches Gewerbe-
Museum)



Inhalt

Beitragsteil1
Berichte Software1
Lösungsverfahren zur numerischen Lösung linearer Gleichungssysteme1
MS-DOS-Treiber für fremde Diskettenformate	17
Proload	29
BIOS-Eprom im AT verändern	32
8-4-2-1	37
Erfahrungen mit Symphony 1.1	37
Programm SYSTEST	37
Berichte Hardware	39
Ein drittes Laufwerk an der Multi-I/O-Platine	39
Berichte, Allgemeines	39
HiSoft '89	39
Viren in Nippon	39
Auf der Hannover CeBit 1989	40
Clubteil	40
Fragen	41
TGM-Disketten	41

Näheres zu dieser Zeitschrift:

Bezugsbedingungen: Einzelheft öS 50,-, für Mitglieder des PCC-TGM im Mitgliedsbeitrag enthalten.

Impressum: Medieninhaber: PCC-TGM (Personal-Computer-Club-TGM), Wexstraße 21, Postfach 59, 1202 Wien.

Anrufbeantworter: (0222)/35 23 983 (Herr GINTHÖR); Mo, Fr: 9.00-12.00h

Mailbox: (0222)/602 10 36 (8-N-1)

BTX: 912213028

TELEBOX: RA2 FIALA.

Grundlegende Richtung: Auf Anwendungen im Unterricht bezogene Informationen über Personal-Computer-Systeme. Berichte über Veranstaltungen des Vereins. Beratung der Vereinsmitglieder gemäß den Statuten des PCC-TGM.

Layout und Satz: Walter Riemer, Rosengasse 9, 2102 Bisamberg

Druck: CA (Creditanstalt Bankverein), Wallensteinplatz, 1200 Wien

Erscheinungsort: Wien

Redaktion und für den Inhalt verantwortlich: Franz Fiala, Siccardsburggasse 4/1/22, 1100 Wien.

1. Allgemeines

Bei dem hier vorliegenden Text behandle ich zunächst möglichst kurz (und dadurch auch sehr ungenau) die Theorie der Lösung linearer Gleichungssysteme, um mich anschließend auf konkrete Lösungsverfahren, insbesondere das Gaußsche Eliminationsverfahren zu konzentrieren.

1.1. Notation

In diesem Text werden Matrizen durch Unterstreichen des jeweiligen Symbols gekennzeichnet z.B.:

$$\underline{A} = \text{Matrix A}$$

Die Unterscheidung zwischen eindimensionalen Matrizen (Vektoren) und zweidimensionalen Matrizen (Matrizen im eigentlichen Sinn) erfolgt dadurch, daß Vektoren durch Kleinbuchstaben bezeichnet werden.

Z.B.:

$$\underline{v} = \text{Vektor v}$$

Soll auf einzelne Elemente der Matrix hingewiesen werden, erfolgt das in Indexschreibweise mit Kleinbuchstaben z.B.:

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

Die hier vorgestellte Matrix ist eine $m \times n$ Matrix mit m Zeilen und n Spalten.

Eine derartige Matrix kann man sich nun wieder aufgespalten in ihre Zeilen und Spalten vorstellen, wobei hierdurch m Zeilen und n Spaltenvektoren definiert werden. Als Spaltenvektor wird im folgenden ein Vektor bezeichnet, der nur aus einer einzigen Spalte ($m \times 1$ Matrix) besteht z.B.:

$$\underline{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_m \end{bmatrix}$$

Als Zeilenvektor wird im folgenden ein Vektor bezeichnet, der nur aus einer einzigen Zeile ($1 \times n$ Matrix) besteht z.B.:

$$\underline{z} = \begin{bmatrix} z_1 & z_2 & z_3 & \dots & z_n \end{bmatrix}$$

Ein lineares Gleichungssystem läßt sich formal als die Matrizenmultiplikation einer Koeffizientenmatrix A mit einem Spaltenvektor x , welche als Ergebnis wieder einen Spaltenvektor b ergibt, darstellen:

$$\underline{A} * \underline{x} = \underline{b}$$

Die lineare Unabhängigkeit der Koeffizientenmatrix ist dann gegeben, wenn die Determinante der Koeffizientenmatrix ungleich Null ist.

1.2.2 Lösbarkeit von linearen Gleichungssystemen

Kehren wir nun wieder zurück zur Lösbarkeit von linearen Gleichungssystemen.

Es läßt sich hier zeigen, daß ein lineares Gleichungssystem nur dann eindeutig lösbar ist, wenn alle (n) Spalten- und Zeilenvektoren der Koeffizientenmatrix A linear unabhängig sind. Im Fall der Nichterfüllung dieser Bedingung ist das Gleichungssystem entweder nur nicht eindeutig lösbar (d.h. das Ergebnis ist wiederum ein geforderter Zusammenhang zwischen verschiedenen Variablen) oder nicht lösbar. Dies hängt hauptsächlich von der rechten Seite, dem Ergebnisvektor b ab.

2. Lösungsverfahren bei linearen Gleichungssystemen

2.1. Cramer'sche Regel

Die Cramer'sche Regel führt die Lösung eines linearen Gleichungssystems auf die Berechnung von Determinanten zurück:

$$x_i = \frac{1}{\text{Det } A} \begin{vmatrix} a_{11} & \dots & a_{1\ i-1} & b_1 & a_{1\ i+1} & \dots & a_{1n} \\ a_{21} & \dots & a_{2\ i-1} & b_2 & a_{2\ i+1} & \dots & a_{2n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \dots & a_{n\ i-1} & b_n & a_{n\ i+1} & \dots & a_{nn} \end{vmatrix}$$

Hierbei wird zur Berechnung der *i*-ten Variable jeweils die *i*-te Spalte der Koeffizientenmatrix durch die linke Seite ersetzt, und die Determinante der so erhaltenen Matrix durch die Determinante der ursprünglichen Koeffizientenmatrix dividiert.

Aus dieser Form ist auch (numerisch) leicht ersichtlich, daß die Determinante der Koeffizientenmatrix nicht Null sein darf, da hier sonst eine Division durch Null erfolgen würde.

Aufgrund des sehr hohen Aufwands der Berechnung von Determinanten, und der Notwendigkeit der Berechnung von $n + 1$ Determinanten für die Lösung eines Gleichungssystems mit *n* Variablen wird dieses Verfahren selten angewendet, und wird hier auch nicht weiter behandelt.

2.2 Gaußsches Eliminationsverfahren

Das Gaußsche Eliminationsverfahren beruht darauf, daß ein Gleichungssystem (nebenbei: wie auch die Determinante einer Matrix) seinen Wert nicht ändert, wenn zu (von) einer Zeile ein Vielfaches einer anderen Zeile addiert (subtrahiert) wird.

Dies wird beim Gaußschen Eliminationsverfahren dazu benützt der Reihe nach die einzelnen Variablen zu eliminieren, indem man eine Gleichung als Hauptgleichung benützt, und zu jeder der restlichen Gleichungen ein geeignetes Vielfaches dieser Gleichung addiert, sodaß in diesen eine Variable nicht mehr vorkommt.

Wir betrachten hierzu das ursprüngliche System,

$$\begin{array}{r} a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots + a_{1n} x_n = b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + \dots + a_{2n} x_n = b_2 \\ \vdots \\ a_{n1} x_1 + a_{n2} x_2 + a_{n3} x_3 + \dots + a_{nn} x_n = b_n \end{array}$$

welches sich nach dem ersten Schritt der Transformation mit der ersten Gleichung als Hauptgleichung zu

$$\begin{array}{r}
 a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots + a_{1n} x_n = b_1 \\
 a_{22}' x_2 + a_{23}' x_3 + \dots + a_{2n}' x_n = b_2' \\
 \vdots \\
 a_{n2}' x_n + a_{n3}' x_n + \dots + a_{nn}' x_n = b_n'
 \end{array}$$

ergibt.

Diese Transformation ist in dieser Form nur für $a_{11} \neq 0$ möglich, was aber keine ernsthafte Beschränkung darstellt, da durch Vertauschen zweier Zeilen, wodurch der Wert des Gleichungssystems ebenfalls nicht geändert wird, sicher eine Situation hergestellt werden kann, in der $a_{11}' \neq 0$ gilt; die Faktoren ergeben sich hierbei zu:

- a_{21} / a_{11} für die 2. Gleichung
- a_{31} / a_{11} für die 3. Gleichung
- a_{n1} / a_{11} für die n-te Gleichung

Das selbe Verfahren kann man nun (für $a_{22}' \neq 0$) mit der 2. Gleichung als Hauptgleichung zur Elimination der Variablen x_2 aus den Gleichungen 3 bis n verwenden.

Durch weitere Anwendung dieses Verfahrens kann man auf diese Art, wenn das System überhaupt eindeutig lösbar ist, das Gleichungssystem in die Form

$$\begin{array}{r}
 r_{11} x_1 + r_{12} x_2 + r_{13} x_3 + \dots + r_{1n} x_n = c_1 \\
 r_{22} x_2 + r_{23} x_3 + \dots + r_{2n} x_n = c_2 \\
 \vdots \\
 r_{nn} x_n = c_n
 \end{array}$$

überführen, aus dem man durch Lösen der letzten Gleichung $x_n (= c_n / r_{nn})$, mit dem Einsetzen von x_n in die $n-1$ - te Gleichung x_{n-1} und so weiter bis x_1 erhalten kann.

Ist die Überführung des Gleichungssystems in die obengenannte Form nicht möglich, so ist das Gleichungssystem nicht eindeutig lösbar, d.h. singular. Dieser Fall soll aus den weiteren Betrachtungen ausgeklammert werden, da dessen Behandlung einen wesentlich höheren Aufwand erfordern würde.

Man erhält dadurch ein neues, dem ursprünglichen Gleichungssystem aber äquivalentes Gleichungssystem, für welches ich folgende Notation benützen möchte:

$$\underline{R} * \underline{x} = \underline{c}$$

Dieses Verfahren ist gleichwertig mit der Multiplikation der Koeffizientenmatrix A sowie der rechten Seite b mit einer unteren Dreiecksmatrix L von links (beachte: die Matrizenmultiplikation ist nicht kommutativ!). Die untere Dreiecksmatrix (eine Matrix, bei der sämtliche Elemente oberhalb der Hauptdiagonale gleich Null sind) ergibt sich dabei aus der Multiplikation mehrerer (n) unterer Dreiecksmatrizen:

$$\underline{L} = \underline{L}_n * \dots * \underline{L}_2 * \underline{L}_1$$

Die Matrizen \underline{L}_m haben dabei in der Hauptdiagonale nur Einsen stehen. Die Elemente unterhalb der Hauptdiagonale sind alle gleich Null, außer jenen in der m-ten Spalte, wo die Koeffizienten der Elimination

- a_{m+1m} / a_{mm}
- a_{m+2m} / a_{mm}
- a_{nm} / a_{mm}

stehen.

Dadurch ergibt sich z.B. die Matrix \underline{L}_1 zu:

$$\underline{L}_1 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -a_{21} / a_{11} & 1 & 0 & \dots & 0 \\ -a_{31} / a_{11} & 0 & 1 & \dots & 0 \\ -a_{41} / a_{11} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1} / a_{11} & 0 & 0 & \dots & 1 \end{bmatrix}$$

Die Matrizen \underline{R} und \underline{c} ergeben sich hierbei durch:

$$\begin{aligned} \underline{R} &= \underline{L} * \underline{A} \\ \underline{c} &= \underline{L} * \underline{b} \end{aligned}$$

Diese Darstellung ist deshalb sehr günstig, weil sie die Berechnung eines Gleichungssystems für eine neue rechte Seite ermöglicht, ohne daß dafür die gesamte Elimination nochmals durchgeführt werden muß. Ich werde später darauf noch zurückkommen.

3. Numerik des Gaußschen Eliminationsverfahrens

Das Gaußsche Eliminationsverfahren, dessen Grundlagen im bisherigen Text bereits behandelt wurden ist grundsätzlich sehr einfach zu programmieren. Man findet dabei für eine einfache Version durchaus mit der Anwendung des bis jetzt Gesagten sein Auslangen.

Für höhere Ansprüche kann man das Verfahren aber noch verbessern.
Zunächst besteht hier die Möglichkeit die Genauigkeit des Verfahrens zu erhöhen:

3.1 Pivotisierung des Gleichungssystems

Hierzu möchte ich zunächst kurz auf die Zahlendarstellung im Computer eingehen.

Bei modernen Programmiersprachen wird in der Regel zwischen Integer (= ganzzahligen) und Real (= gebrochenen) Zahlen unterschieden.

Die Speicherung von Integerzahlen führt in der Regel kaum zu Problemen, da man sich hierbei auf einen Zahlenbereich beschränkt, der durch die verwendete Wortlänge exakt beschrieben werden kann.

Z.B. Integer = 16 Bit Worte ==>
Zahlenbereich: $-2^{(n-1)} = -32768 \leq Z \leq 32767 = 2^{(n-1)} - 1$

Hierbei wird jede mögliche Zahl des Zahlenbereiches durch genau eine Bitkombination beschrieben, wodurch keine Fehler auftreten können.

Anders ist die Situation bei Realzahlen: Hier muß ein Dezimalbruch dargestellt werden, der im Extremfall nie endet (z.B. π , $1/3$, ...). Derartige Zahlen können nicht exakt dargestellt werden, sodaß man zur Verarbeitung im Computer nur die ersten n Stellen des Dezimalbruches verwendet (n ist gleich der Anzahl der gültigen Stellen). Um trotzdem große und kleine Zahlen darstellen zu können verwendet man hier eine Darstellung mit Mantisse und Exponent. Die Mantisse sind die eigentlichen Stellen der Zahl, der Stellenwert der gesamten Zahl ergibt sich durch Multiplikation mit 10^{Exponent} .

z.B.: Mantisse = 0,1234; Exponent = 2: Zahl = $0,1234 * 10^2 = 12,34$

Eine Addition wird dabei so durchgeführt, daß die beiden Zahlen zunächst durch Verschieben der Stellen der kleineren Zahl auf den Exponenten der größeren Zahl gebracht und anschließend addiert werden. Hierbei kann es aber vorkommen, daß die eine Zahl soweit verschoben werden muß, daß keine gültige Stelle mehr für die Addition verfügbar ist, sodaß das Ergebnis so aussieht, als wenn Null addiert worden wäre.

z.B.:

a) Addition von $0,1234 * 10^0 + 0,5678 * 10^2$ in einem Computer mit 5 gültigen Stellen; die führende Null wird hierbei nicht mitgezählt:

1. Schritt: Auf gleichen Exponenten bringen:

Hierbei werden beide Zahlen auf den Exponenten der größeren Zahl gebracht.

$$\begin{aligned} \Rightarrow 0,12340 * 10^0 &\rightarrow 0,00123 * 10^2 \\ 0,56780 * 10^2 &\rightarrow 0,5678 * 10^2 \end{aligned}$$

Hierbei ging bereits die letzte gültige Ziffer des ersten Summanden verloren.

2. Schritt: Ziffernweise Addition

$$\begin{array}{r} \Rightarrow 0,00123 * 10^2 \\ + 0,56780 * 10^2 \\ \hline 0,56903 * 10^2 \end{array}$$

b) Addition von $0,1234 * 10^0 + 0,5678 * 10^8$ in einem Computer mit 5 gültigen Stellen; die führende Null wird hierbei nicht mitgezählt:

1. Schritt: Auf gleichen Exponenten bringen:

Hierbei werden beide Zahlen auf den Exponenten der größeren Zahl gebracht.

$$\begin{aligned} \Rightarrow 0,1234 * 10^0 &\rightarrow 0,00000 * 10^8 \\ 0,5678 * 10^8 &\rightarrow 0,56780 * 10^8 \end{aligned}$$

Hierbei ging die gesamte Information des ersten Summanden verloren.

2. Schritt: Ziffernweise Addition

Dieser Schritt ist in diesem Fall eine Addition von 0 zu $0,5678 * 10^8$

Wie bereits in der Theorie erläutert, wird beim Gaußschen Eliminationsverfahren das A_{ij}/A_{ii} - fache einer Zeile (mit i: Nummer der zu eliminierenden Variablen, j: Nummer der Zeile, in der eliminiert werden soll) von der j - ten Zeile abgezogen. Wird dieser Faktor sehr groß, so "überdeckt" die Hauptgleichung die anderen Gleichungen. Im diesem Fall wird die Gleichung, in der eliminiert werden soll für den Rechner mit seiner begrenzten Darstellungsmöglichkeit nur mehr ein Vielfaches der Hauptgleichung sein, was die eindeutige Lösung des Gleichungssystems verhindern würde.

Dieser Faktor wird nun (betragsmäßig) genau dann sehr groß, wenn A_{ii} betragsmäßig sehr klein wird.

Dies ist auch die Möglichkeit zu einer Abhilfe: Wenn als A_{ii} das (betragsmäßig) größte aller A_{ij} mit j von i bis Anzahl der Gleichungen gewählt wird, werden die Faktoren klein gehalten (betragsmäßig kleiner oder gleich 1).

Die Elimination der Variablen wird daher so ablaufen, daß man vor der Elimination einer Variablen das betragsmäßig größte Element der Koeffizientenmatrix in der Spalte der betrachteten Variablen unter und auf der Hauptdiagonale sucht, diese Gleichung zur Hauptgleichung erklärt, und mit Hilfe dieser Gleichung die Variable in den anderen Gleichungen eliminiert.

Für die Verarbeitung im Rechner, bei der man die zu eliminierenden Variablen in der Regel in Form einer Schleife von 1 bis Anzahl der Variablen eliminieren wird, ergibt sich daher die Notwendigkeit, das Gleichungssystem umzustellen.

Hier besteht die überlegungsmäßig einfachste Möglichkeit die i -te Zeile mit der Zeile, in der das betragsmäßig größte Element gefunden worden ist, auszutauschen, indem die entsprechenden Elemente der Koeffizientenmatrix A und des Spaltenvektors b kopiert werden. Aufgrund der hohen Anzahl von Zuordnungsvorgängen ist diese Methode aber relativ langsam.

Eine zweite Möglichkeit ist die Verwendung eines Steuervektors.

3.1.1 Pivotisierung mittels Steuervektor

Hierbei bedient man sich folgender Struktur:

```

s1 ->  a11  a12  ....  a1n  b1
s2 ->  a21  a22  ....  a2n  b2
:      :    :      :      :
sn ->  an1  an2  ....  ann  bn

```

Man definiert einen Steuervektor s , der ebensoviele Zeilen hat wie die Koeffizientenmatrix A . Bei der Adressierung der Zeilen der Koeffizientenmatrix A sowie des Spaltenvektors b adressiert man nun die zeilen nicht direkt, sondern über den Steuervektor:

z.B. $A(s(i),j)$, wenn man die das j -te Element der i -ten Zeile adressieren will; ebenso beim Spaltenvektor b :
 $b(s(i))$, wenn man die i -te Zeile des Spaltenvektors b adressieren will.

Vor Beginn der Elimination ist dafür zu sorgen, daß das i -te Element des Steuervektors den Wert i enthält, wodurch gewährleistet ist, daß wirklich die erste Zeile adressiert wird, wenn $A(s(1),j)$ adressiert wird.

Wenn nun aber im Zuge einer Pivotsuche die gewünschte Hauptgleichung z.B. die 5. Gleichung statt der 3. ist, so reicht ein Vertauschen von $s(5)$ mit $s(3)$ um den Zeilentauch durchzuführen. Es darf trotzdem nicht direkt, wie in diesem Beispiel naheliegend zu sein scheint

```

s(5.) := 3 und
s(3.) := 5

```

durchgeführt werden, da es ja möglich ist, daß durch eine frühere Vertauschung $s(5.)$ gar nicht mehr 5 enthält (desgleichen mit $s(3.)$)!

3.2 Verbesserung des Ergebnisses

Jener Ergebnisvektor x , den man nach der Durchführung des Gaußschen Eliminationsverfahrens und anschließender Berechnung der einzelnen Koordinaten erhält ist zwangsläufig nicht genau.

Es besteht jetzt die Möglichkeit, durch Einsetzen des Ergebnisvektors x in die ursprüngliche (!!!!) Gleichung einen Fehler für die linke Seite zu bestimmen.

$$\text{diff} = b - A * x_e$$

x_e : errechneter Ergebnisvektor

Diese Differenz ist nun aber noch kein Maß für den Fehler der Ergebnisvektors. Aufgrund der Distributivitätseigenschaften der Matrizenmultiplikation gilt aber:

$$\underline{A} * \underline{x} - \underline{A} * \underline{x_e} = \underline{b} - \underline{A} * \underline{x_e} = \underline{diff}$$

woraus folgt:

$$\underline{A} * (\underline{x} - \underline{x_e}) = \underline{diff} \text{ oder}$$

$$\underline{A} * \underline{Delta_x} = \underline{diff}$$

x: exakter Ergebnisvektor

Delta_x: Fehler, um den der errechnete Ergebnisvektor zu klein ist.

Daraus folgt nun aber, daß durch Lösung des Gleichungssystems

$$\underline{A} * \underline{Delta_x} = \underline{diff}$$

eine (wiederum nicht exakte) Lösung für den Fehler berechnet werden kann, der bei der Lösung des Gleichungssystems begangen wurde.

Indem dieser Fehler zur ursprünglichen Lösung addiert wird, erhält man eine genauere Lösung des ursprünglichen Systems. Diese Iterationen sind aber nur in einem beschränkten Ausmaß sinnvoll, da ja die Genauigkeit auch durch die Faktoren, mit denen die Elimination durchgeführt wurde beeinflusst wird, deren Genauigkeit aber durch diese Methode nicht gesteigert werden kann.

Zur Berechnung des Fehlervektors Delta_x gibt es nun die einfache, aber aufwendige, Möglichkeit die gesamte Elimination nochmals durchzuführen.

Schneller ist es, wenn man unter Ausnützung einiger Eigenschaften der Elimination die Berechnung für eine neue linke Seite einfacher durchführt. Der Schlüssel hierzu liegt in der Möglichkeit, die Elimination als Multiplikation sowohl der Koeffizientenmatrix A als auch des Spaltenvektors b mit einer unteren Dreiecksmatrix L von links zu interpretieren:

$$\begin{array}{l} \underline{R} = \underline{L} * \underline{A} = \underline{L_n} * \dots * \underline{L_2} * \underline{L_1} * \underline{A} \\ \underline{c} = \underline{L} * \underline{b} = \underline{L_n} * \dots * \underline{L_2} * \underline{L_1} * \underline{b} \end{array}$$

wodurch sich das Gleichungssystem, wie bei der Theorie bereits erläutert als

$$\underline{R} * \underline{x} = \underline{c}$$

darstellt.

Die Matrix R haben wir hierbei bereits einmal berechnet, es ist dies eine obere Dreiecksmatrix, die sich bei der Elimination von A ergibt.

Dadurch reicht es, die Matrizenmultiplikation von L mit b und anschließend die Berechnung der xi aus dem reduzierten System durchzuführen.

Dieses Matrizenmultiplikation kann mit der Kenntnis aller Lm ebenso gut wie mit der Kenntnis der unteren Dreiecksmatrix L durchgeführt werden. Wenn man alle Matrizen Lm in einem Feld speichert ergibt sich eine untere Dreiecksmatrix:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -a_{21}/a_{11} & 1 & 0 & \dots & 0 \\ -a_{31}/a_{11} & -a_{32}'/a_{21}' & 1 & \dots & 0 \\ -a_{41}/a_{11} & -a_{42}'/a_{21}' & -a_{43}''/a_{31}'' & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1}/a_{11} & -a_{n2}'/a_{21}' & -a_{n3}''/a_{31}'' & \dots & 1 \end{bmatrix}$$

Die "interessanten" Bereiche (jene, in denen eine nicht von vornherein bekannte Information steht) dieser Matrix sind angenehmerweise genau jene, die bei der reduzierten Koeffizientenmatrix Null werden, sodaß jene Speicherplätze der Koeffizientenmatrix, die zu Null werden, zur Speicherung eben jener Teile dieser Matrix verwendet werden können.

In etwas weniger mathematischer Form kann man auch sagen, daß man die Quotienten, mit denen man eine Variable aus einer Gleichung eliminiert hat, am Platz des Koeffizienten für diese Variable in der entsprechenden Gleichung abspeichert, da der Koeffizient nach der Elimination Null wird, was aufgrund der Struktur des Lösungsverfahrens bekannt ist.

Hierdurch ergibt sich nach Durchführung der gesamten Reduktion mit Speicherung der Quotienten folgender Zustand der Matrix:

$$A' = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21}/a_{11} & a_{22}' & a_{23}' & \dots & a_{2n}' \\ a_{31}/a_{11} & a_{32}'/a_{21}' & a_{33}'' & \dots & a_{3n}'' \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1}/a_{11} & a_{n2}'/a_{21}' & a_{n3}''/a_{31}'' & \dots & a_{nn}''' \end{bmatrix}$$

Bei der Berechnung der c_n wird nun sinnvollerweise wieder das Feld für die b_n doppelt verwendet, wodurch sich als eine Möglichkeit zur Berechnung der c_n folgende Methode ergibt, wobei nicht mehr zwischen b_n und c_n unterschieden wird, da nur ein Array vorhanden ist.

```

I := 1 to Dim - 1
  J := I + 1 to Dim
    b (.J.) := b (.J.) - A (.J,I.) * b (.I.)
  
```

4. Weitere Anwendungen des Gaußschen Eliminationsverfahrens

4.1 Berechnung von Determinanten

Für die Berechnung von Determinanten mit Hilfe von Computern ist der wesentlichste Satz jener, daß bei einer oberen oder unteren Dreiecksmatrix der Wert der Determinante gleich dem Produkt der Elemente der Hauptdiagonale ist.

Das Gaußsche Eliminationsverfahren ist nun ein Verfahren, welches eine Matrix in eine obere Dreiecksmatrix umwandelt, sodaß (im wesentlichen) nur mehr die Elemente der Hauptdiagonale miteinander multipliziert werden müssen.

Das Vorzeichen der Determinante ist nach dieser Maßnahme aber noch nicht unbedingt korrekt, da eine Determinante bei einer Vertauschung von Zeilen oder Spalten ihr Vorzeichen wechselt, das heißt, es ist notwendig das Produkt der Elemente der Hauptdiagonale mit $(-1)^{\text{Anzahl der Vertauschungen}}$ zu multiplizieren.

4.2 Inversion von Matrizen

Matrizeninversion kann am einfachsten durchgeführt werden, indem man n Gleichungssysteme in n Variablen löst:

$$\underline{A} * \underline{X} = \underline{E}$$

\underline{X} ist hierbei eine n x n Matrix, die Inverse zu \underline{A} . Die linke Seite ist in diesem Fall ebenfalls eine Matrix, die Einheitsmatrix, bei der alle Elemente außer den Elementen der Hauptdiagonale 0 sind; Letztere sind 1.

$$\underline{E} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Diese Gleichungssysteme kann man auf 2 Arten relativ effizient lösen:

Die erste Möglichkeit wäre die spaltenweise Bestimmung der Inversen \underline{X} indem in ein Gleichungssystem

$$\underline{A} * \underline{x} = \underline{b}$$

der Reihe nach als rechte Seite die Spalten der Einheitsmatrix eingesetzt werden. Die m-te Spalte der Inversen ist hierbei die Lösung des obigen Gleichungssystems mit der m-ten Spalte der Einheitsmatrix (alle Elemente gleich 0; nur das m-te gleich 1) als rechter Seite. Hierbei müßte nur einmal das gesamte Eliminationsverfahren abgearbeitet werden, danach könnte man das Verfahren zur Berechnung eines Gleichungssystems mit geänderter rechter Seite verwenden.

Dieses Verfahren bietet sich insbesondere an, wenn man auf ein vorhandenes System von Unterprogrammen zur Durchführung des Gaußschen Eliminationsverfahrens zurückgreifen kann.

Die zweite Möglichkeit ist wahrscheinlich etwas schneller, benötigt aber doch einige spezielle Unterroutinen:

Hierbei wird das gesamte System

$$\underline{A} * \underline{X} = \underline{E}$$

betrachtet. Multipliziert man dieses System von links mit einer unteren Dreiecksmatrix \underline{L} vom selben Aufbau wie beim Gaußschen Eliminationsverfahren, so erhält man folgendes System

$$\underline{R} * \underline{X} = \underline{L},$$

wobei die Matrix \underline{R} die in eine obere Dreiecksmatrix umgerechnete Koeffizientenmatrix ist. Hierbei kann man die Spalten der Inversen \underline{X} erhalten, indem man die Rückrechnung (!) des Gaußschen Eliminationsverfahrens für ein Gleichungssystem durchführt, dessen rechte Seite die entsprechende Spalte der unteren Dreiecksmatrix \underline{L} ist. Die Dreiecksmatrix \underline{L} müßte hierbei aber explizit berechnet werden.

5. Hinweise zur Programmierung

Es empfiehlt sich bei der Programmierung eines derartigen Programmes, eine Routine zu schreiben, die einem die Gestalt des aktuellen Gleichungssystems zeigt.

Im Falle der Verwendung der Version mit Steuervektor bedeutet dies, daß sowohl die Information vorhanden sein sollte, welche Zeile des ursprünglichen Systems eine ausgedruckte Zeile ist, als auch welche Zeile des aktuellen Systems sie darstellt, oder, was die selbe Information beinhaltet, wo die Quotienten enden und die eigentliche Koeffizientenmatrix beginnt.

Bei der Durchführung des Eliminationsverfahrens ist weiter darauf zu achten, daß eine Division durch Null verhindert (abgefangen) wird, wobei die Abfrage besser nicht auf Null erfolgen sollte, sondern eine, vom Problem abhängige Schwelle gesetzt werden sollte, unter der angenommen wird, daß dieses Element in Wirklichkeit (bei unendlich genauer Rechnung) Null wäre. In diesem Fall ist das Gleichungssystem nicht eindeutig lösbar.

6. Quellennachweis

- (1) H.J. Dirschmid; Mathematische Grundlagen der Elektrotechnik
2. Auflage 1987;
Verlag Vieweg
- (2) Mathematik A2 für Elektrotechnik
Vorlesung 101.695 Sommersemester 1987/88 Prof. Dirschmid
Technische Universität Wien
- (3) Programmiertechnik für Elektrotechniker
Vorlesung 356.895 Wintersemester 1988/89 Prof. Barth
Technische Universität Wien

GAUSS.SYS

(* Dieses File definiert alle nötigen Typen für das Programm Gauss *)

```
Const MaxN = 20;      (* Maximalanzahl der Gleichungen und Variablen *)
      Minimum = 1E-30; (* Grenze für die Erkennung einer singulären *)
                        (* Matrix *)
```

```
Type Koeff = Array (.1..MaxN, 1..MaxN.) of Real;
                        (* Typ für die Speicherung der Koeffizienten *)
      Spalte = Array (.1..MaxN.) of Real;
                        (* Typ zur Speicherung der rechten Seite und *)
                        (* des Ergebnisvektors *)
      ISpalte = Array (.1..MaxN.) of Integer;
                        (* Typ zur Speicherung des Steuervektors *)
```

GAUSS.SBR

(* Dieses Unterprogrammpaket berechnet ein lineares Gleichungssystem der Form

Ax = B,

A ... Matrix n x n (n: Dimension des Systems)

B ... Ergebnisvektor (Matrix n x 1)

x ... Variablenvektor (Matrix n x 1) *)

Procedure Gauss_Druck (DA: Koeff; DB: Spalte; DSteuer: ISpalte; DDimension: Integer);

(* Procedure zum Drucken eines gestaffelten Systems, wobei zwischen den Quotienten und der eigentlichen Matrix ein | gezeichnet wird *)

```
Var I, J:      Integer; (* Zählervariablen *)
      Dummy:   Char;   (* Dummy Zeichen zum Warten auf die Fortsetzung *)
```

```
Begin (* Gauss_Druck *)
  Writeln;
  Writeln;
  For I := 1 to DDimension do
    Begin For J := 1 to DDimension do
      Begin If DSteuer (.I.) = J then Write (' | ')
            else Write (' ');
            Write (DA (.I.,J.):7:1);
            End;
            (* Schreiben der Quotienten und der gestaffelten Matrix *)
            Writeln (' = ', DB (.I.):7:1);
            (* Schreiben des entsprechenden Elements, rechte Seite *)
          End;
        Writeln;
        Writeln ('Weiter mit Return');
        Read (Dummy);
        (* nach dem Einlesen Fortsetzung des Programms *)
      End; (* Gauss_Druck *)
```

Procedure Gauss_Init (Var ISteuer: ISpalte; IDimension: Integer);

(* Initialisierung des Steuervektors *)

```
Var I:      Integer; (* Zählervariable *)
```

```
Begin (* Gauss_init *)
  For I := 1 to IDimension do
    ISteuer (.I.) := I;
  End; (* Gauss_Init *)
```

Procedure Gauss_neue_Rechte (LA: Koeff; Var LB: Spalte; LSteuer: ISpalte; LDimension: Integer);

(* Diese Procedure berechnet die zur gestaffelten Matrix passende rechte Seite aus der ursprünglichen rechten Seite des Systems *)

```
Var I, J:      Integer; (* Zählervariablen *)
```

```
Begin (* Gauss_neue_Rechte *)
  For I := 1 to LDimension - 1 do
    For J := I + 1 to LDimension do
      LB (.LSteuer (.J.)) :=
        LB (.LSteuer (.J.)) - LA (.LSteuer (.J.), I.)
        * LB (.LSteuer (.I.));
    End;
  End; (* Gauss_neue_Rechte *)
```

```
Procedure Gauss_Elim (Var GA:Koeff; Var GB: Spalte; Var GSteuer: ISpalte; GDimension: Integer; Var Nicht_Singulaer: Boolean);
```

```
(* Diese Procedure rechnet die Koeffizientenmatrix GA um in die gestaffelte Koeffizientenmatrix, wobei der Steuervektor verändert wird, und in den zu 0 gemachten Koeffizienten der Quotient Q gespeichert wird *)
```

```
Var I, J: Integer; (* Zählervariablen *)
    Max_Betrag: Real; (* maximaler Betrag, *)
                (* der von Abs_Max gefunden wurde *)
```

```
Function Abs_Max (GAA: Koeff; Var GASTeuer: ISpalte; GADimension: Integer; GAZeile: Integer): Real;
```

```
(* Diese Function bestimmt das betragsmäßig größte Element der Koeffizientenmatrix GAA in der Spalte GAZeile unter der Hauptdiagonale unter Beachtung des Steuervektors GASTeuer, und tauscht die Zeile, in der das betragsmäßig größte Element gefunden wurde durch Veränderung des Steuervektors in die Zeile GAZeile. Der Rückgabewert von Abs_Max ist der größte gefundene Betrag *)
```

```
Var I: Integer; (* Zählervariable *)
    Hilf: Integer; (* Hilfsvariable zum Zeilentausch *)
    Max: Real; (* größtes bisher gefundenes Element *)
    Max_Zeile: Integer; (* Zeile, in der Max gefunden wurde *)
```

```
Begin (* Abs_Max *)
    Max := ABS (GAA (.GASTeuer (.GAZeile.),GAZeile.));
    Max_Zeile := GAZeile;
    For I := GAZeile + 1 to GADimension do
        If ABS (GAA (.GASTeuer (.I.),GAZeile.)) > Max then
            Begin Max := ABS (GAA (.GASTeuer (.I.),GAZeile.));
                Max_Zeile := I;
            End; (* Suche Zeile mit absolut größtem Element *)
        If Max_Zeile <> GAZeile then (* vertausche die Zeilen *)
            (* GAZeile und Max_Zeile *)
            Begin Hilf := GASTeuer (.GAZeile.);
                GASTeuer (.GAZeile.) := GASTeuer (.Max_Zeile.);
                GASTeuer (.Max_Zeile.) := Hilf;
            End;
        Abs_Max := Max;
    End; (* Abs_Max *)
```

```
Procedure Gauss_Single (Var GSA: Koeff; GSSteuer: ISpalte; GSDimension: Integer; GSZeile: Integer);
```

```
(* Diese Procedure macht einen Schritt einer Gauß-Transformation, d.h. sie transformiert die Koeffizientenmatrix derart, daß in der Spalte GSZeile unter der Hauptdiagonale nur mehr Nullen stehen würden. Tatsächlich werden in jene Elemente aber die bei der Berechnung verwendeten Quotienten geschrieben *)
```

```
Var I, J: Integer; (* Zählervariablen *)
    Q: Real; (* Quotient *)
```

```
Begin (* Gauss_Single *)
    For I := GSZeile + 1 to GSDimension do
        (* I zählt die Zeilen *)
        Begin Q := GSA (.GSSteuer (.I.),GSZeile.) /
            GSA (.GSSteuer (.GSZeile.),GSZeile.);
            GSA (.GSSteuer (.I.),GSZeile.) := Q;
            (* Koeffizientenmatrix wird gleichzeitig zum Speichern der Quotienten verwendet *)
            For J := GSZeile + 1 to GSDimension do
                (* J zählt die Spalten bei der Umrechnung einer Zeile in die gestaffelte Form *)
                GSA (.GSSteuer (.I.),J.) :=
                    GSA (.GSSteuer (.I.),J.) -
                    Q * GSA (.GSSteuer (.GSZeile.),J.);
            End;
        End; (* Gauss_Single *)
```

```

Begin (* Gauss_Elim *)
  Nicht_Singulaer := true;
  For I := 1 to GDimension do
    If Nicht_Singulaer then
      Begin Max_Betrag := Abs_Max (GA,GSteuer, GDimension, I);
        (* suche absolut größtes Element in der Spalte I
           unterhalb der Hauptdiagonale *)
        (* und tausche die dazugehörige Zeile in die Zeile I *)
        If not (Max_Betrag < Minimum) then
          Gauss_Single (GA,GSteuer, GDimension, I)
            (* führe einen Schritt der Transformation aus
              *)
          else Nicht_Singulaer := false;
        End;
      Gauss_neue_Rechte (GA, GB, GSteuer, GDimension);
    End; (* Gauss_Elim *)

Procedure Gauss_Rueck (RA: Koeff; RB: Spalte; RSteuer: ISpalte; RDimension: Integer; Var OK: Boolean; Var RX: Spalte);

(* Diese Procedure erledigt die Rückrechnung von einer gestaffelten Matrix
   mit passender linker Seite auf den Variablenvektor x *)

Var I, J:      Integer;
    Hilf:      Real; (* Hilfsvariable zum Berechnen des Divisors *)

Begin (* Gauss_Rueck *)
  OK := true;
  For I := RDimension downto 1 do
    (* I zählt die Zeilen = Elemente des Ergebnisvektors *)
    Begin Hilf := RB (.RSteuer (.I.));
      For J := RDimension downto I + 1 do
        (* J zählt die Spalten bei der
           Rücktransformation eines Elements
           des Ergebnisvektors *)
        Hilf := Hilf - RX (.J.) * RA (.RSteuer (.I.),J.);
        If (Abs (RA (.RSteuer (.I.),I.) > Minimum) then
          RX (.I.) := Hilf / RA (.RSteuer (.I.),I.)
            (* Verhinderung einer Division durch 0 *)
          else OK := false;
        End;
      End;
    End; (* Gauss_Rueck *)

Procedure Gauss_Defekt (RkA: Koeff; RkB: Spalte; RkSteuer: ISpalte; RkDimension: Integer; RkX: Spalte; Var RkD: Spalte);

(* Gauss_Defekt berechnet die Differenz zwischen dem Vektor B
   der Angabe und dem Vektor B, der bei einem Einsetzen
   des berechneten Variablenvektors in die ursprüngliche Matrix A
   entstehen würde *)

Var I, J:      Integer; (* Zählervariablen *)

Begin (* Gauss_Defekt *)
  For I := 1 to RkDimension do
    Begin RkD (.I.) := RkB (.I.);
      For J := 1 to RkDimension do
        RkD (.I.) := RkD (.I.) - RkA (.I.,J.) * RkX (.J.);
      End;
    End;
  End; (* Gauss_Defekt *)

LINGL.PAS

Program Lingl; (* Lösen eines linearen Gleichungssystems *)

(*$I GAUSS.SYS *) (* Definition der Typen für Gauss *)

Var Koeffmat: Koeff; (* Koeffizientenmatrix für Gauss *)
    Rechte: Spalte; (* rechte Seite für Gauss *)
    Koeffmat2: Koeff; (* original Koeffizientenmatrix
                       für Defektberechnung *)
    Rechte2: Spalte; (* original rechte Seite
                      für Defektberechnung *)
    Steuer: ISpalte; (* Steuervektor *)
    Dim: Integer; (* Dimension des Gleichungssystems *)
    X: Spalte; (* Ergebnisvektor des Gleichungssystems *)
    Differenz: Spalte; (* Defekt des Gleichungssystems *)
    DeltaX: Spalte; (* Fehler des Ergebnisvektors *)
    OK: Boolean; (* Hilfsvariable, die anzeigt,
                  ob die letzten Operationen
                  ordnungsgemäß abliefen *)
    I: Integer; (* Zählervariable zum allgemeinen Gebrauch *)
    Dummy: Char; (* Dummychar zum Bremsen des Programms *)

```

```
(*I GAUSS.SBR *)           (* Prozeduren und Funktionen für Gauss *)

Procedure Eingabe (Var Matrix: Koeff;Var LSpalte: Spalte; Var Nbr: Integer; Var EingabeOK: Boolean);

(* Procedure zum Einlesen des Gleichungssystems

Die Eingabe kann wahlweise von einem Textfile oder der Tastatur erfolgen.

Das Textfile muß dabei alle Informationene beinhalten, die sonst vom Benutzer
abgefragt würden:

o) Dimension des Gleichungssystems
o) Koeffizientenmatrix
o) rechte Seite

Aufbau des Textfiles:

Dimension
a11  a12  b1
a21  a22  b2

*)

Var Eing_File: Text;      (* File, von dem gelesen werden kann *)
    Eing_Filenam: String (.11.);
                        (* Name des Eingabefiles *)
    I, J: Integer;      (* Zählervariablen *)
    Eing_Quelle: Char;   (* Single Char für div. Abfragen *)
    E_File: Boolean;    (* true, wenn vom File gelesen werden soll *)

Begin (* Eingabe *)
    EingabeOK := true;
    Write ('Eingabe von File oder von Tastatur (F/T): ');
    Readln (Eing_Quelle);
    If (Eing_Quelle in (.'F','f'.)) then E_File := true
        else E_File := false;
    If E_File then Begin Repeat
        Write ('Eingabefilename: ');
        Readln (Eing_Filenam);
        Writeln;
        Assign (Eing_File, Eing_Filenam);
        (*$I-$) Reset (Eing_File);(*$I+*)
        Until (IOResult = 0);
        Readln (Eing_File,Dim);
        If Dim > MaxN then
            Begin
                Writeln ('Dimension des Gleichungssystems
                    zu groß');
                EingabeOK := false;
                Delay (10000);
            End;
        else Begin Repeat Write
            ('Dimension des Gleichungssystems:');
            Readln (Nbr);
            Writeln;
            Until Nbr <= MaxN;
        End;
    For I := 1 to Dim do
        Begin For J := 1 to Dim do
            If E_File then Read (Eing_File,Matrix (.I,J.))
            else Begin Write ('A [',I:1,',',J:1,'] = ');
                Readln (Matrix (.I,J.));
            End;
            If E_File then Readln (Eing_File, LSpalte (.I.))
            else Begin Write ('B [',I:1,'] = ');
                Readln (LSpalte (.I.));
                Writeln;
            End;
        End;
    If E_File then Close (Eing_File);
End; (* Eingabe *)

Begin (* Lingl *)
    ClrScr;
    Eingabe (Koeffmat, Rechte, Dim, OK);
    ClrScr;
    If OK then
        Begin Koeffmat2 := Koeffmat;
            Rechte2 := Rechte;
            (* sichern der ursprünglichen Daten für spätere *)
            (* Verwendung *)
            Gauss_Init (Steuer,Dim);
            Writeln ('Eingegebenes Gleichungssystem');
            Gauss_Druck (Koeffmat, Rechte, Steuer, Dim);
            Writeln;
        End;
    End;
End;
```

```

(* Ausgabe der ursprünglichen Matrix *)
Gauss_Elim (Koeffmat, Rechte, Steuer, Dim, OK);
(* berechnen eines ersten Ergebnisses *)
If OK then
(* nur, wenn Elimination ordnungsgemäß *)
Begin
  Writeln ('Gleichungssystem nach der Elimination:');
  Gauss_Druck (Koeffmat, Rechte, Steuer, Dim);
  (* Rücktransformation des Systems *)
  Gauss_Rueck
    (Koeffmat, Rechte, Steuer, Dim, OK, X);
  Writeln;
  Writeln
    ('Ergebnis unmittelbar nach Gleichungslösung');
  Writeln;
  Writeln ('Ergebnisvektor:');
  Writeln;
  For I := 1 to Dim do
    Writeln ('X',I:1,' = ',X (.I.));
    (* Berechnung des Defekts mit ursprünglicher *)
    (* Koeffizientenmatrix *)
    Gauss_Defekt
      (Koeffmat2, Rechte2, Steuer, Dim, X, Differenz);
    (* Ausgabe des Defekts *)
    Writeln;
    Writeln ('Defekt:');
    Writeln;
    For I := 1 to Dim do
      Writeln ('Defekt',I:1,' = ',Differenz (.I.));
      (* Aufbau eines neuen Gleichungssystems mit Fehlern *)
      (* als rechter Seite und Berechnung desselben *)
      (* Elimination nicht mehr nötig, da sich die *)
      (* Koeffizientenmatrix nicht geändert hat *)
      Gauss_neue_Rechte (Koeffmat, Differenz, Steuer, Dim);
      (* Ruecktransformation -> DeltaX *)
      Gauss_Rueck
        (Koeffmat, Differenz, Steuer, Dim, OK, DeltaX);
      Writeln;
      Writeln ('Fehler des Ergebnisvektors:');
      Writeln;
      For I := 1 to Dim do
        Begin
          Writeln ('DeltaX',I:1,' = ',DeltaX (.I.));
          X(.I.) := X(.I.) + DeltaX (.I.);
        End;
        Writeln;
        Writeln;
        Writeln ('Weiter mit Return');
        Read (Dummy);
        Writeln ('Ergebnis nach 1 Iteration:');
        Writeln;
        For I := 1 to Dim do Writeln ('X',I:1,' = ',X (.I.));
        Gauss_Defekt
          (Koeffmat2, Rechte2, Steuer, Dim, X, Differenz);
        Writeln;
        Writeln ('Defekt:');
        Writeln;
        For I := 1 to Dim do
          Writeln ('Defekt',I:1,' = ',Differenz (.I.));
          Gauss_neue_Rechte (Koeffmat, Differenz, Steuer, Dim);
          Gauss_Rueck
            (Koeffmat, Differenz, Steuer, Dim, OK, DeltaX);
          Writeln;
          Writeln ('Fehler des Ergebnisvektors');
          Writeln;
          For I := 1 to Dim do
            Writeln ('DeltaX',I:1,' = ',DeltaX (.I.));
          Writeln;
          Writeln ('Weiter mit Return');
          Read (Dummy);
        End
      else Begin Writeln;
              Writeln ('Fehler: singuläre Matrix');
              Writeln;
              Writeln ('Berechnung abgebrochen');
            End;
  End;
End;
End. (* Lingl *)

```

**MSDOS-Treiber für fremde
Diskettenformate**

(Beantwortung einer Anfrage in den vorletzten PC-NEWS)

TGM_98 : UNIDRV.ASM, UNIDRV.SYS
Thomas Kopelent / BOKU

Dieser Treiber ist an und für sich als Universaltreiber konzipiert, der verschiedene DOS-Formate, die vom Standardformat abweichen, als auch das ATARI ST-Format für einen PC/AT zur Verfügung stellen. Er ist der Zeitschrift c't 11/1987 p.216 ff. entnommen. An dieser Stelle findet sich auch ein ausführlicher Artikel über die Problematik des Lesens von Fremdformaten. Wenn Ihr BIOS 3,5" Laufwerke nicht standardmäßig unterstützt, so müssen Sie folgende Zeile in Ihr CONFIG.SYS aufnehmen:

```
device = driver.sys /D:1 /F:2
```

wobei der Parameter D: das physikalische Laufwerk angibt.

Der Treiber UNIDRV.SYS wird zum Bearbeiten von ATARI ST-Disketten wie folgt eingebunden:

```
device = unidrv.sys 80 1 2
```

Die Bedeutung der Parameter:

erster (80) : Anzahl der Tracks auf diesem Medium
zweiter (1) : Physikalisches Laufwerk (muß ggf. mit /D:x übereinstimmen)

dritter (2) : Dieser Parameter ist nur zum Bearbeiten von ATARI ST-Disketten erforderlich.

'1' bedeutet einseitige Diskette, '2' zweiseitig.

Nach einem Re-Boot meldet sich der Treiber am Bildschirm (am Besten ist es, ein eventuell in AUTOEXEC.BAT enthaltenes 'CLS' zu entfernen). Er gibt selbstständig an, als welches logische Laufwerk er ATARI ST-Disketten ansprechen kann (Es handelt sich immer um den nächsten freien Laufwerksbuchstaben).

Sollen sowohl ein- als auch zweiseitige ATARI ST-Disketten bearbeitet werden, so sollte der Treiber zweimal (!) ins CONFIG.SYS eingetragen werden; einmal mit '1' und einmal mit '2' als dritten Parameter. Sie erhalten dann zwei logische Laufwerke für die verschiedenen Formate (ACHTUNG: nicht durcheinanderbringen!).

Nach erfolgreicher Installation läßt sich das Funktionieren des Treibers am Einfachsten mit dir x: (x steht für das neue logische Laufwerk) überprüfen.

Viel Glück !

```
*****
;*
;*      Universeller PC-/MSDOS-Diskettentreiber
;*
;* Programmname      : unidrv.asm
;* Ersteller         : Martin Ernst
;* Erweitert        : Peter Köhlermann
;* Erstellungsdatum  : 11.03.86
;* Letzte Änderung am : 16.08.87
;* Letzter Test am   : 02.09.87
;* Versionsnummer    : 3.0
;* Revisionsnummer   : 6.0
;*
;*****
```

```
page      72,132
title     PC-/MSDOS-Universal-Treiber
```

```
false     equ 0
true      equ not false
v20       equ false ; für Rechner mit 8088/86 od. 80286
;v20      equ true  ; für Rechner mit V20/30
```

```
lengt     equ 13h
```

```
.186
ausgabe   macro    message
           mov dx, offset message
           mov ah, 9
           int 21h
           endm
```

; diese Definition ist der Einsprung in die Monitorfunktion des c't 86

```
monitoreprom segment at 0f000h
               org 0e000h
monitor       label far
monitoreprom ends
```

```
code      segment para public
assume    cs:code,ds:code,es:code,ss:code
```

```
dskdev:  dw  -1,-1
          dw  0000100000000000b ; Bit 11 vom Attributwort gesetzt
          dw  strategy
          dw  dskint
drvmax   db  1

ptrsav   dd  0 ; hier wird der Zeiger auf die Anorderungs-
          ; kopfzeile gesichert
          db  3 dup (?)
```

```
dsktbl:  dw  dskinit ; "Init Device-Treiber"
          dw  mediac ; "Check Media Type"
          dw  getbpb ; "Get BIOS Parameter Block de
          ; selektierten "Media Type"
          dw  cmderr ; "I/O-Control Input (nicht implementiert)
          dw  dskred ; "Read Data"
          dw  busexit ; "Non Destructive Read Data" (n.i.)
          dw  exit ; "Input Status"
          dw  exit ; "Input Flush Buffer"
          dw  dskwrt ; "Write Data"
          dw  dskwrv ; "Write Data" mit Verify-Funktion
          dw  exit ; "Output Status"
          dw  exit ; "Output Flush Buffer"
          dw  exit ; "I/O-Control Output"
          dw  exit ; "open"
          dw  exit ; "close"
          dw  exit ; "Removable Media" --> gleich nach "exit",
          ; kein "busy", da Diskette
```

```
iodat    struc
cmdlen   db  ? ; Länge der Tabelle
unit     db  ? ; Einheiten-Code
cmd       db  ? ; Befehls-Code
status   dw  ? ; Status der Operation
          db  8 dup (?)
media    db  ? ; Media Deskriptor Byte
trans    dd  ? ; Transfer Adresse
count    dw  ? ; Anzahl der Blöcke oder charakter
          ; die übertragen werden sollen
start    dw  ? ; erster Block, der transferiert werden soll
drive    db  ? ; zugeordnete Drive-Bezeichnung
iodat    ends
```

```
bpbs     struc
          db  13 dup (?)
bpb1     db  ? ; Struktur für die Rücklieferung von Werten
bpb2     dw  ? ; über die Anforderungskopfzeile
          dw  ?
bpb3     dw  ?
          dw  ?
bpbs     ends
```

```
dbp      struc
seccsize dw  ? ; Sektorgröße (512 bytes)
alloc    db  ? ; Anzahl Sektoren pro Cluster
ressec   dw  ? ; Anzahl reservierter Sektoren
fats     db  ? ; Anzahl der FATS
maxdir   dw  ? ; Anzahl der Directory-Einträge
sectors  dw  ? ; Gesamtanzahl der Sektoren
mediaid  db  ? ; Media Descriptor Byte
fatsec   dw  ? ; Anzahl der Sektoren pro FAT
sectrk   dw  ? ; Anzahl der Sektoren pro Track
          ; (nicht Zylinder!)
koepfe   dw  ? ; Anzahl der Köpfe (bzw. Seiten bei Floppy)
hidden   dw  0 ; versteckte (hidden) Sektoren
dbp      ends
```

; IBM PS/2, 720 KByte, 80 Spuren

```

ibmps    dbp <512,2,1,2,112,1440,0f9h,3,9,2>

; Philips YES
philips  dbp <512,2,1,2,176,1440,0feh,3,9,2>

; DEC Rainbow (MS-PCDOS), 80 Spuren
decrain  dbp <512,1,20,2,96,800,0fah,3,10,1>

; ct86, 80 Spuren, einseitig
ddrive2  dbp <512,1,1,2,112,720,0fch,4,9,1>

; ct86, 80 Spuren, doppelseitig
ddrive1  dbp <512,2,1,2,224,1440,0fdh,4,9,2>

; Atari ST (default: Doppelseitig, 80 Spuren)
atari    dbp <512,2,1,2,112,1440,0f7h,5,9,2>

inittab1: dw  ibmps.secsiz

```

```

;=====

```

```

stratp   proc far
; Registerbelegung beim Aufruf:
; bx = Offset der Anforderungszeile
; es = Segment der Anforderungszeile

strategy:
    mov word ptr cs:[ptrsav],bx    ; Adresse der Zeile wird
    mov word ptr cs:[ptrsav+2],es ; gesichert
    ret
stratp   endp

```

```

;=====

```

```

diskint  proc far
dskint:
    if v20
        pusha
    else
        push ax
        push cx
        push dx
        push di
        push bp
        push bx          ; erst alle Register retten
        push si
    endif
    push ds
    push es
    mov si,offset dsktbl; Tabelle der Befehle
    lds bx,cs:[ptrsav] ; in BX der Offset der Anforderungskopfzeile
    mov al,[bx.unit]   ; AL: welche Einheit
    mov ah,[bx.media] ; Media Descriptor Byte
    mov cx,[bx.count] ; Anzahl Bytes oder Blöcke
    mov dx,[bx.start] ; Startadresse des ersten Bytes od. Blocks
    xchg di,ax         ; AX sichern, da nun die Adresse der Befehls-
    mov al,[bx.cmd]    ; routine berechnet wird. Dazu muß der
    xor ah,ah          ; Befehlscode zweimal auf die Startadresse
    add si,ax           ; der Tabelle (Wert in SI) addiert werden.
    add si,ax
    cmp al,18          ; Funktion größer als 18
    ja cmderr          ; ja --> Fehler
    xchg ax,di         ; altes AX zurück
    les di,[bx.trans] ; DI enthält nun den Offset, ES das
                    ; Segment der Transferadresse
    push cs            ; Datensegment gleich dem Code-Segment
    pop ds
    jmp word ptr [si] ; und ab zur Befehlsroutine

```

```

;-----

```

```

busexit:mov  ah,00000011b ; wenn das entsprechende Device "busy" ist,
            jmp  short exit1 ; dann hierher und weiter zum gemeinsamen
                        ; Ausgang

```

```

; Fehlercodes :
; 0 = schreibgeschützt
; 1 = unbekannte Einheit
; 2 = Laufwerk nicht bereit
; 3 = unbekannter Befehl in der Anforderungskopfzeile
; 4 = CRC-Fehler
; 5 = falsche Laufwerksanforderungszeilenlänge
; 6 = Seek-Fehler
; 7 = unbekanntes Media deskriptor Byte

```

```

;      8 = Sektor nicht gefunden
;      9 = Drucker hat kein Papier mehr
;     10 = Fehler beim Schreiben
;     11 = Fehler beim Lesen
;     12 = allgemeiner Fehler

cmderr:  mov  al,3          ; unbekannter Befehl --> Sprung hierher

errexit: mov  ah,10000001b ; Bits für "fertig" und "Fehler" setzen
         stc
         jmp short exit1   ; und weg

exit:    mov  ah,00000001b
exit1:   lds  bx,cs:[ptrsav]
         mov  [bx.status],ax ; in der Anforderungskopfzeile den
                           ; Status melden

set_double:
         cmp  atflg,1      ; ist es ein AT?
         jnz  ok2         ; nein, dann weg ohne Double Stepping an
         mov  bx,40h
         mov  es,bx
         mov  bx,90h
         add  bl,drivesel
         mov  dl,es:[bx]
         and  dl,0f0h      ; sonst das Double-Stepping aktivieren
         cmp  dl,50h      ; da wir es für 720 KByte Disketten in HD-
         jne  ok2         ; Laufwerk abgeschaltet hatten
         mov  dl,61h
         mov  es:[bx],dl

ok2:
         pop  es
         pop  ds
         if  v20
         popa
         else
         pop  si
         pop  bx
         pop  bp
         pop  di
         pop  dx
         pop  cx          ; alle Register zurück vom BIOS-Stack
         pop  ax
         endif
         ret              ; zurück zum DOS

diskint  endp

;=====

localer  proc near

dskwrv:
dskwrt:  mov  verify,1     ; wenn Verify ON

         call setup      ; überprüfen, ob angeforderte Sektoren
         jc   is_error   ; außerhalb der Grenzen der Diskette
         call diskwrt    ; wenn ok, dann Sektoren schreiben
         jc   is_error   ; bei Schreibfehler weg
         cmp  verify,0   ; ohne Verify sofort zum Ausgang, sonst
         jz   exit

         mov  verify,0   ; Kontroll-Lesen
         lds  bx,cs:[ptrsav] ; dazu die benötigten Parameter wieder
         mov  al,[bx.unit] ; holen
         mov  ah,[bx.media]
         mov  cx,[bx.count]
         mov  dx,[bx.start]
         les  di,[bx.trans]
         push cs
         pop  ds

dskred:  call setup      ; genauso wie beim Schreiben
         jc   is_error
         call diskrd
         jnc  exit

is_error:
         call derror    ; Fehlercode des INT 13 in den des Treibers
         jmp  short errexit ; wandeln und Fehler-Ausgang benutzen

;-----

getbpb:

```

```

mov ah,al          ; bei PCDOS wird es schon über Pointer
                   ; in DI geliefert
mov al,es:[di]     ; Media Byte holen, Unit Code jetzt in AH
push ax
push bx
xchg ah,al
call getpara       ; den richtigen DPB ermitteln
mov si,bx
pop bx
pop ax

getbp4:
get_bp5: lds bx,[ptrsav] ; und beide Werte in der
mov [bx.bpb1],al ; Anforderungszeile sichern
mov [bx.bpb3],si
mov [bx.bpb3+2],cs
xor al,al          ; kein Fehler
jmp exit          ; weg

;-----
mediac: xor di,di    ; 1: not changed, 0: don't know, -1: changed

mov al,ibmifcflg  ; wenn keine IFC-Karte,
                  ; dann nicht überprüfen

or al,al
jz media1
cmp al,1          ; die alte FDC braucht auch kein Media Check
jz media1
inc di
mov al,24h        ; nur bei IFC über callf monitor
call romcall      ; auf "motor aus" testen
or al,al
jz mediaexit

media1: xor di,di    ; sonst eine Null, da nicht bekannt

mediaexit:
lds bx,[ptrsav]   ; in Anforderungszeile den Status sichern
mov word ptr [bx].trans,di
mov al,0          ; kein Fehler
jmp exit

;-----
setup: xor si,si

call getpara      ; hole den zum Media-Byte gehörenden DBP

xchg bx,di        ; nach DI mit dem pointer
mov al,byte ptr [di].koepfe
mov ah,byte ptr [di].sectrk
imul ah          ; Anzahl der Sektoren pro Spur
mov [maxsec],al  ; für später (Überlauf in nächsten Track)
push si
mov si,cx         ; CX = Anzahl zu lesende Sektoren
add si,dx         ; DX = Startsektor
cmp si,word ptr [di].sectors
pop si            ; größer als die gesamte Diskette
jbe inrange      ; Sektoren hat? Nein --> Fehler = 8
mov al,8         ; --> Sektor nicht gefunden
stc
ret

inrange:
mov [secnt],cx   ; Anzahl zu lesende Sektoren sichern
add dx,si
mov si,bx        ; BX enthält Offset Transferadresse
and bx,0fh       ; der Offset soll so klein wie möglich sein
mov [dmaadr],bx ; darum wird entsprechend das Segment
mov ax,dx        ; in DX erhöht
xor dx,dx
mov bl,[maxsec] ; Nummer des logischen Sektors wird durch
xor bh,bh        ; Maximalzahl Sektoren pro Track geteilt
div bx           ; das ergibt den relativen Sektor
inc dl           ; innerhalb des Tracks
mov [cursec],dl
mov [curtrk],ax ; und als Quotient den Track-Wert
mov cl,4
clc
shr si,cl        ; hier jetzt die Segmentvergrößerung
mov cx,es
add cx,si

```

```

        mov [dmasegment],cx    ; Segmentadresse sichern für später

driveeq:
        mov cx,[curtrk]      ; sicher ist sicher
        clc                  ; kein Fehler
dirf:   ret

;-----

getpara:
        mov bx,offset ibmps   ; erster dbp-Eintrag
loopac: cmp ah,[bx].mediaid    ; ist es der richtige DBP ?
        jz sfou              ; ja dann weg
        add bx,leingt        ; sonst nächsten versuchen
        cmp bx,offset atari+leingt ; falls es nicht der letzte war
        jnz loopac

sfou1:  mov bx,offset atari    ; keiner stimmt, dann nehmen wir atari
sfou:   ret

;-----

diskrd:
        call reset_double    ; double stepping eventuell abschalten
rdlp:   call preset          ; Werte für Track, Sektor, Kopf, Anzahl der
        cmp ibmifcflg,2     ; SEKTOREN NUR BEI ifc-Karte mit uebercallf
        jnz ueberintles     ; sonst den INT 13 benutzen
        mov al,16h          ; Code des Callf-Monitors für Lesen
        call uebercallf
        jmp short rdlp2

ueberintles:
        mov ah,2            ; Lesefunktionsnummer des INT 13H
        int 13h
rdlp2:  jc errorr           ; in AH steht error-Code

        dec [seccnt]        ; nächsten Sektor
        jnz rdlp
        clc                  ; kein Fehler
        ret

errorr: stc                  ; bei Fehler hierher
        ret

diskwrt:
        call reset_double
wrlp2:  call preset          ; wie rdlp, nur eben schreiben
        cmp ibmifcflg,2
        jnz ueberintsch
        mov al,18h
        call uebercallf
        jmp short wrlp2

ueberintsch:
        mov ah,3
        int 13h

wrlp2:  jc errorr
        dec [seccnt]
        jnz wrlp2
        clc
        ret

; Double Stepping für die Bearbeitung normaler 80-Spur-Disketten mit 720
; KByte Kapazität in Multifunktionslaufwerken nur in Ats abschalten.

reset_double:
        cmp atflg,1         ; auf AT
        jz ok1              ; nein, dann weg
        ret

ok1:    mov bx,40h          ; sonst bei Adresse 40:90 oder 91
        mov es,bx
        mov bx,90h
        add bl,drivesel
        and byte ptr es:[bx],0dfh ; maskiert das doublestep-Bit aus
        ret

preset: mov bx,[dmaadr]     ; für den INT 13h müssen die Register belegt
        ; werden
        mov es,[dmasegment] ; Segment laden
        mov al,[cursec]
        cmp al,[maxsec]

```

```

jbe gotsec1      ; zu lesender Sektor im nächsten Track ?
inc [curtrk]    ; ja, dann eben track erhöhen
mov al,1        ; relativer Sektor wird 1
mov [cursec],al
mov cl,al
xor dh,dh      ; Kopf iost dann logischerweise 0
jmp short wegvon

gotsec1: push bx      ; kein Überlauf
mov bx,word ptr [di].sectrk
mov [secpertrak],bl ; dann den Kopf berechnen, indem man
xor dh,dh      ; solange sectrk abzieht, bis es richtig ist
loopinc:cmp al,bl
jbe nohead
inc dh        ; Kopf immer um eins erhöhen (damit können
sub al,bl    ; wir auch Harddisks bedienen)
jmp short loopinc

nohead: mov cl,al      ; richtiger Sektor und richtiger Kopf
pop bx      ; DMA-Adresse wieder vom Stack

; ist es eine Diskette vom DEC Rainbow ?

mov ax,offset decrain ; bei den Floppy-Disks müssen wir noch
cmp ax,di            ; überprüfen, ob es sich eventuell um eine
jnz nodec            ; DEC-Rainbow-Diskette handelt (findet man
                    ; anhand des BPB heraus)

; erst prüfen, ob Track 0 oder 1, denn da hat der DEC keinen Skew
; (diese Angabe im BIOS-Listing des DEC-Rainbow erwies sich leider als
; falsch, trotzdem bleibt der Code drin, falls nochmal eine Änderung
; zumachen ist)

; xor ax,ax
; cmp ax,[curtrk]
; jb nodec

; jetzt Skew für DEC Rainbow ändern

istdec: xor ch,ch
push si      ; aus der Tabelle die richtige Sektornummer
mov si,cx
mov cl,scewtable[si] ; holen
pop si

wegvon:
nodec: mov ax,word ptr [di].secsize
add [dmaadr],ax
inc [cursec] ; schon den nächsten Sektor einstellen
mov ax,[curtrk]
mov ch,al
and ah,03h  ; und in CL (Sektor) zusätzlich die beiden
ror ah,1    ; oberen Bits als Track benutzen
ror ah,1
or cl,ah    ; oberes Byte der Track-Nummer
mov dl,[drivesel]
mov al,1
ret

-----

derror: lds bx,cs:[ptrsav]
mov [bx.count],0
push cs    ; Fehlermeldung des INT 13h in Meldung
pop ds    ; für DOS umsetzen
cmp ibmifcflg,0 ; nur beim PC oder AT kann der Floppy-
jnz no_pc ; Controller zurückgesetzt werden
push ax
xor ax,ax  ; Reset auf Controller
cwd
int 13h
pop ax

no_pc: test ah,80h ; Zeitüberschreitung (time out)
jz de1
mov al,2
ret

de1: test ah,40h ; Spur nicht gefunden (Seek-Error)
jz de2
mov al,6
ret

de2: test ah,3 ; schreibgeschützt

```

```

    jz  de3
    xor  al,al
    ret

de3:   test ah,10h      ; CRC-Error
        jz  de4
        mov al,4
        ret

de4:   test ah,2      ; Sektor nicht gefunden
        jz  de5
        mov al,8
        ret

de5:   test ah,4      ; Bad Adress Mark
        jz  de6
        mov al,8
        ret

de6:   or  ah,ah      ; überhaupt ein Fehler aufgetreten ?
        jz  dee
        mov al,12     ; dann ist es ein allgemeiner Fehler

dee:   ret

```

; Skewtable für den DEC-Rainbow

```

scewtable  db  0,1,3,5,7,9,2,4,6,8,10

verify     db  0

drivesel   db  0
cursec     db  0
curtrk     dw  0
dmaadr     dw  0
dmasegment dw  0

maxsec     db  0      ; Anzahl Sektoren pro Zylinder

seccnt     dw  0
secpetrak  db  0

atflg      db  0      ; 0 = PC/c't86, 1 = AT

ibmifcflg db  0      ; 0 = PC/AT , 1 = c't86-FDC, 2 = c't86-IFC

; ab hier transienter Code, wenn PC oder AT

```

uebercallf:

```

    push ax          ; INT 13h - Aufruf in einem Callf-Monitor
    push cx          ; umsetzen
    push dx
    push bx
    mov  cx,es
    mov  al,14h      ; Setze Segment
    call romcall
    pop  cx
    mov  al,0ch      ; Setze Offset
    call romcall
    pop  ax
    push ax
    inc  al
    mov  ah,[secpetrak]
    mov  cx,ax
    mov  al,10h      ; setze Laufwerk

```

ifc_or:

```

    or  cl,al        ; für HD (AT)-Format
    call romcall     ; mit IFC-Karte auf c't86
    pop  dx
    mov  cl,dh
    mov  ch,0
    mov  al,[secpetrak]
    mov  ah,0        ; Kopf gibt es nicht, nur relativen Sektor
    mul  cl          ; innerhalb einer Spur, daher mit Kopfnummer
    pop  cx          ; multiplizieren
    push cx
    mov  ch,0
    add  cx,ax
    mov  al,12h      ; setze Sektor
    call romcall
    pop  cx
    mov  cl,ch

```

```

        mov ch,0
        mov al,0eh ; setze Spur
        call romcall
        pop ax ; lesen oder schreiben
        call romcall
        mov ah,al
        or al,al
        jz noerr
        stc
        ret
noerr:  clc
        ret

; callf-Monitoraufruf
romcall: push bx
        push di
        call monitor
        pop di
        pop bx
        ret

; ab hier transienter Code, wenn c't86
dskinit: mov es,dx
        mov bx,cx ; es:bx zeigt auf die Zeile im Config.sys
        push bx
        push es
        mov ax,0f000h
        mov es,ax
        mov bx,0fffh ; erst mal schauen ob es ein PC oder ein
        mov al,es:[bx]; c't86 ist, denn beim c't können wir mit
        ; callf-Monitor und ifc arbeiten
        cmp al,06h ; die letzte Version des c't86 Monitors
        ja istibmpc ; wird hoffentlich nur 6 sein
istct86: mov al,22h
        call romcall ; ist eine ifc-Karte vorhanden ?
        test al,20h ; wenn ja, dann auf jeden fall über callf
        jnz mitifckarte
        mov ibmifcflg,1 ; sonst nur alte fdc-Karte
        jmp short weiterinit

istibmpc:
        mov ibmifcflg,0
        cmp al,0fch ; ist es ein IBM PC/AT ??
        jnz weiterinit
        mov atflg,1 ; AT !!!
        jmp short weiterinit

mitifckarte:
        mov ibmifcflg,2 ; IBMIFCFLG auf entsprechenden Wert setzen
weiterinit:
        pop es
        pop bx
        ausgabe startmes ; signon Message ausgeben
        call scan_for_para ; Treibernamen übergehen
        mov al,es:[bx] ; sind parameter angegeben ?
        or al,al
        jnz node ; wenn ja, dann die Werte analysieren
        jmp default_init ; sonst voreingestellte Werte benutzen

node:   mov al,es:[bx]
        inc bx ; auf erstes Zeichen nach Treibernamen
        cmp al,'4' ; 40 oder 80 Track
        jz track40
        cmp al,'8'
        jz track80
        jmp errordata

track40: mov al,es:[bx]
        cmp al,'0' ; die Null von 40 oder 80 überprüfen
        jz ok11
        jmp errordata

ok11:   mov tr4080,0 ; Bei 40 Spuren braucht das Doublestepping
        mov atflg,0 ; nicht abgeschaltet werden
        mov dx,offset viertrack
        jmp short driveermit; text ausgeben

track80: mov al,es:[bx]
        cmp al,'0'
        jz ok22
        jmp errordata ; keine 0 dann Fehler

```

```

ok22:    mov tr4080,1
         ausgabe achttrack

driveermit:
         inc bx
         inc bx
         mov al,es:[bx]    ; physikalisches Laufwerk ermitteln
         cmp al,'0'       ; muß zwischen 0 und 3 liegen
         jl  errordata
         cmp al,'3'
         jg  errordata
         sub al,'0'
store:   mov drivesel,al
         cmp al,1         ; nur bei 0 und 1 kann man das Double-Stepping
         jle hdlauf      ; ausschalten, sonst nicht
         mov atflg,0

hdlauf:  add al,'A'        ; für die Textausgabe umwandeln
         mov physic,al
         cmp tr4080,1
         jnz move_them   ; bei 40 Spur-Formaten müssen wir DBPs
         inc bx          ; übertragen
         inc bx
         mov al,es:[bx]; bei 80 Spuren erst im dritten Parameter
         ; nachsehen, welcher zu übertragen ist
         and al,0dfh     ; Uppercase wandeln
         cmp al,'A'
         jz  atlauf      ; als normales HD-Laufwerk im c't86 benutzen
         cmp al,'I'
         jz  ibmpsin     ; als IBMP 720KByte-Format (PS/2)
         cmp al,'S'
         jz  siemin      ; mit Siemens PCD-Format
         cmp al,'T'
         jz  nixin       ; mit Nixdorf PWS-Format
         cmp al,'O'
         jz  olliin      ; mit Olivetti M24-Format
         cmp al,'N'
         jnz ibmpsin1    ; und wenn's keiner war,
         ; dann ist's IBM PS/2

nixin:   mov dx,offset nixtext
         mov si,offset nixdorf
         jmp short move_bpb ; Text ausgeben DBP übertragen

errordata:
         ausgabe fehlertxt

default_init:
         mov tr4080,0    ; bei Fehler oder ohne Parameter ist es 40
         mov drivesel,1 ; Spuren auf Laufwerk B:
         mov physic,'B'
         mov dx,offset deftxt
         jmp short move_bpb

atlauf:  mov dx,offset at_text
         mov si,offset hd
         jmp short move_bpb

siemin:  mov dx,offset siemtext
         mov si,offset siemens
         jmp short move_bpb

olliin:  mov dx,offset ollitext
         mov si,offset olivet
         jmp short move_bpb ; Text ausgeben und DPB übertragen

ibmpsin1:
         dec bx
         dec bx

ibmpsin:  mov dx,offset ibmpstext

move_them:
         mov si,offset ibmps

move_bpb:
         mov ah,9
         int 21h
         push es
         push ds
         pop  es
         mov di,offset ibmps
         cmp si,offset hd ; bei den 80 Spur-Formaten muß nur ein DBP
         jnz normal      ; übertragen werden, bei den HD-Formaten
         mov cx,lengt * 3 ; sind es aber drei
         jmp short move

```

```

normal:
    mov word ptr ds:ifc_or,9090h ; nop nop
    cmp tr4080,0
    mov cx,lemt
    jnz move ; falls 80 track,
                ; sonst überschreibe die
    mov si,offset lsdri1 ; ersten zwei 80 track bpb
                ; mit 40 track bpb
                ; 2 BPBs zu kopieren
move:
    add cx,cx
    cld
    rep movsb
    pop es
    cmp tr4080,0 ; bei 40 Spur Format gibt es keinen vierten
    jz no_atari ; Parameter für Atari
    call scan_for_para
    mov al,es:[bx] ; Atari-Parameter '1' oder '2' (-seitig)
    cmp al,'1' ; nur 1 wird ausgewertet, sonst DS
    mov atariflg,al
    mov dx,offset atarissmes
    mov si,offset atariss
    jz atari1
    mov dx,offset ataridsmes
    mov si,offset atari
atari1:
    mov di,offset atari
    push es
    push ds
    pop es
    mov cx,lemt ; Atari-Format übertragen und
    rep movsb
    pop es
    mov ah,9 ; Text ausgeben
    int 21h

no_atari:
    mov ah,30h
    int 21h ; get Version
    lds bx,[ptrsav]
    cmp al,3 ; wird erst ab Version 3.x unterstützt
    push ds
    jb below_3x
    mov dl,[bx.drive] ; sonst kann das logische Laufwerk nicht
    add dl,'A' ; ermittelt werden
    mov byte ptr cs:laufwerk,dl
    push cs
    pop ds
    ausgabe logical

below_3x:
    mov ax,offset dskinit
    cmp ibmifcflg,0 ; bei PC oder AT ab Dskinit transienter Code
    pop ds
    jnz here_end ; im c't86 erst ab uebercallf
    mov ax,offset uebercallf

here_end:
    mov word ptr [bx+0eh],ax
    mov word ptr [bx+10h],cs ; gib 'end of driver' an's DOS
    push cs
    pop ds
    mov al,drvmax ; Anzahl Untereinheiten des Treibers
    xor ah,ah
    mov si,offset inittab1 ; Standard-DBP einstellen
    jmp get_bp5

scan_for_para:
    push ds
    push es
    pop ds
    mov si,bx
s1:
    lodsb
    cmp al,' ' ; überspringe voraneilende Leerzeichen
    jz s1
s2:
    lodsb
    cmp al,0
    jz s3
    cmp al,' ' ; überspringe den Treibernamen
    jnz s2
s3:
    lodsb
    cmp al,' ' ; überspringe nachfolgende Leerzeichen
    jz s3
    dec si
    mov bx,si
    pop ds
    ret

```

```

; HD-Laufwerk
hd      dbp <512,1,1,2,224,2400,0f9h,7,15,2>
        dbp <512,2,1,2,224,2720,0fdh,7,17,2>
        dbp <1024,2,1,2,224,1440,0feh,7,9,2>

; IBMPC40Track, 9 Sektoern, doppelseitig
lsdriv1 dbp <512,2,1,2,112,720,0fdh,2,9,2>

; IBMPC 40 Track 9 Sektoren einseitig
lsdriv2 dbp <512,1,1,2,64,360,0fch,2,9,1>

; Olivetti M24
olivet  dbp <512,2,1,2,144,1440,0f9h,3,9,2>

; Siemens PC-D
siemens dbp <512,4,1,2,144,1440,0f9h,2,9,2>

; Nixdorf PWS oder Softec
nixdorf dbp <512,2,1,2,144,1600,0f9h,3,10,2>

; Atari ST doppelseitig (80 Spur 3,5 Zoll)
atarids dbp <512,2,1,2,112,1440,0f7h,5,9,2>

; Atari ST einseitig (80 Spur 3,5 Zoll)
atariss dbp <512,2,1,2,112,720,0f7h,5,9,1>

atariflg db 0
tr4080  db 0 ; 40 oder 80 Track

deftxt  db 'Default-Werte: ',0dh,0ah
        db '40-Track-Drive B:',0dh,0ah,'$'
fehlertxt db 'Fehler in Parameterfeld',0dh,0ah,'$'

viertrack db '40-Track-Drive eingestellt',0dh,0ah,'$'
achttrack db '80-Track-Drive eingestellt',0dh,0ah
db 'Bei Media-Byte F9h: ', '$'

ollitext db 'Olivetti M24',0dh,0ah,'$'
siemtext db 'Siemens PC-D',0dh,0ah,'$'
nixtext  db 'Nixdorf/Softec',0dh,0ah,'$'
ibmpstext db 'IBM 80-Spur',0dh,0ah,'$'
at_text  db 'AT-Laufwerk',0dh,0ah,'$'

startmes db 0dh,0ah,"Universeller c't-Disk-Treiber"
db 'V 3.6 (16.08.87)',0dh,0ah,'$'

logical db 'auf physik. Laufwerk '
physic  db ' '
        db ': als Laufwerk '
laufwerk db 'd'
        db ': installiert',0ah,0dh,'$'

ataridsmes db 'Bei Media-Byte F7h: Atari ST doppelseitig',0dh,0ah,'$'
atarissmes db 'Bei Media-Byte F7h: Atari ST einseitig',0dh,0ah,'$'

localer endp
code ends

if1
%out end of pass 1
endif

if2
%out end of pass 2
endif

end

```

PROLOAD

¹¹⁹⁹⁷
TGM_98 : PROLOAD.ASM, PROLOAD.EXE
 Scharl, Schächle / TGM-5AN

1. Aufgabe

Proload dient dazu, andere speicherresidente Programme, wie Treiber für Netzwerk, Drucker, Maus, usw. oder andere Utilities an einer beliebigen Speicherposition zu installieren. Die zu installierenden Programme und die Speicherposition sind frei wählbar.

2. Funktion

Proload stützt sich auf die Verwaltung des Speicherplatzes von MSDOS und nutzt deren Organisation. Um die Arbeitsweise dieses Programms zu verstehen, möchte ich eine kurze Einführung in die Speicherorganisation von MSDOS geben.

2.1 Speicherverwaltung von MSDOS

Damit das Betriebssystem feststellen kann, an welcher Position ein Programm im Speicher installiert werden kann, wird der Speicher in mehrere einzelne Blöcke unterteilt. Diese Blöcke nennt man Memory Control Blöcke (MCB). Am Beginn jedes MCBs befindet sich Information über die Position des nächsten Blocks und über die Länge des zur Verfügung gehaltenen Blockbereichs. Die einzelnen MCBs stehen hintereinander im Speicher und kennzeichnen die Platzreservierung für bereits installierte Programme. Um das Ende der Blockkette zu kennzeichnen, befindet sich an erster Position des letzten Blocks ein "Z". Die dazwischenliegenden Blöcke werden mit "M" bezeichnet.

Die Organisation dieser Blöcke kann aus Bild 5 entnommen werden.

Wird nun ein neues Programm geladen und ausgeführt, so sucht das Betriebssystem zuerst die Kette der einzelnen Blöcke ab.

Wird der Block mit dem "Z" gefunden, so kann aus dem Eintrag der Länge (Offset 3) die nächste freie Speicheradresse ermittelt werden. An dieser Position wird das Programm von DOS installiert und ausgeführt.

2.2 Arbeitsweise des Programms Proload

Proload nutzt diese Organisation bzw. die Blockaufteilung aus. Nach der Abfrage von Programmnamen, Programmadresse und Programmlänge wird als erstes die Entfernung der eingegebenen Programmadresse und der momentanen Programmadresse (von Proload) ermittelt.

Danach wird von der momentanen Adresse bis zur Adresse des zu installierenden Programmes ein Block markiert. Der Block enthält an erster Position ein "Z" und ist somit für DOS der letzte in der Kette befindliche Block. Weiters wird die Blocklänge und weitere für DOS notwendige Daten in den Blockhead geschrieben. Danach erfolgt ein Interrupt 21 (4B), und das zu installierende Programm wird über DOS gestartet.

Das Betriebssystem findet nun an letzter Stelle der Kette den soeben selbst generierten Block und ladet das Programm an das Ende dieses Blocks, welches nun der eingegebenen Speicheradresse entspricht. Nach dem Verlassen des installierten Programms wird der Block wieder gelöscht und der Block davor durch ein "Z" am Beginn als Schlußglied gekennzeichnet.

Proload wird anschließend beendet, ohne daß das speicherresident installierte Programm zerstört wird.

Die genaue Arbeitsweise kann aus dem nachfolgenden Listing entnommen werden.

Aufbau eines Memory-Control-Blocks (MCB) im Speicher

Addr.	Inhalt	Typ
+00h	ID ("Z" = letzter MCB, "M" = es folgen weitere)	1 BYTE
+01h	Segmentadresse des zugehörigen PSP	1 WORD
+03h	Anzahl der Paragraphen im allokierten Speicherbereich	1 WORD
+05h	ungenutzt	11 BYTE
+10h	der allokierte Speicherbereich	x PARAG.

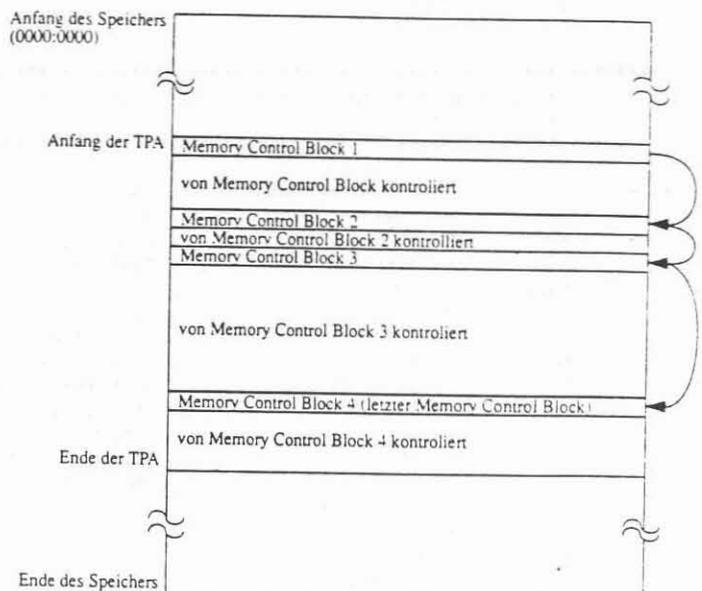
Länge: 16 + die Größe des allokierten Speicherbereichs

```

TITLE    ProLoad.COM 11-25-88      [11-26-88]

.RADIX   16
;
;INITIAL VALUES :  CS:IP    0000:0100
;                  SS:SP    0000:FFFF
Start    SEGMENT
ASSUME   DS:Start,SS:Start,CS:Start,ES:Start

                ORG     100H
    
```



```

Seg0 EQU $-100h
ParBlock EQU $+240h
RegSave EQU $+280h
ProLen EQU $+1EFFh
ProAdr EQU $-9100h

```

```
Begin: jmp NamInp
```

```

ErrMes: DB 0a,'Falscher Pfad oder Dateiname !!$'
Header1: DB '*****'
DB '*** Programm Loading Software ***'
DB '*****'
DB 0a
DB 'Dieses Programm installiert Ihnen ein'
DB 'Speicher Resistentes Programm an einer'
DB 'von Ihnen gewünschten Speicheradresse'
DB 0a
DB 'Programmname : $'
Header2: DB 0a,'Speicheradresse : $'
Header3: DB 0a,'Programmlänge : $'
MemLeng: DB 05,00,00,00,00,00,00
MemAdr: DB 05,00,00,00,00,00,00
ProNam: DB 20
DB 1f dup (?)

```

```

;*****
;* Ausgabe der Meldetexte und Abfrage des Programmnamens, *
;* der Programmlänge und der Speicherposition *
;*****
NamInp: mov ah,09 ;Ausgabe der Startmeldung
mov dx,offset Header1 ;und Aufforderung
int 21 ;zur
mov ax,offset ProNam
mov dx,ax ;Eingabe der Programmnamens
mov ah,0a
int 21
mov bx,offset ProNam+1 ;Rücksetzen des eingegebenen
mov al,[bx] ;ENTER ( 00h )
mov ah,00 ;auf 00h
mov bx,offset ProNam+2
adc bx,ax
mov ax,0000
mov [bx],ax
AdrInp: mov ah,09 ;Aufforderung
mov dx,offset Header2
int 21 ;zur
mov ax,offset MemAdr
mov dx,ax ;Eingabe der Adresse
mov ah,0a
int 21
LengInp: mov ah,09 ;Aufforderung
mov dx,offset Header3
int 21 ;zur
mov ax,offset MemLeng
mov dx,ax ;Eingabe der Progr.
mov ah,0a ;Länge
int 21
jmp Main

```

```

;*****
;* Verschieben der Memory Control Blöcke und installieren *
;* des Programms *
;*****

```

```

Main: mov bx,offset MemLeng+1 ;Text Programmlänge
call ASctoWrd ;in Word umwandeln
mov dx,bx ;Programmlänge nach dx
mov bx,offset MemAdr+1 ;Text Adresse
call ASctoWrd ;in Word umwandeln
MOV AX,CS ;Code Segment um 1
DEC AX ;verkleinern um ins
MOV DS,AX ;Datensegment Übertrag.
MOV BYTE PTR DS:Seg0,4Dh ;Markieren eines MCB
PUSH DS:Seg0+3 ;ENDE durch "M"
mov cx,bx ;Errechnen der Länge bis
SUB CX,AX ;gewünschter Progr. Beginn
SUB CX,1 ;und in MCB Pos 3 schreiben
MOV DS:Seg0+3,CX
mov ax,bx
MOV DS,AX
MOV BYTE PTR DS:Seg0,5A ;Markieren des bisherigen
MOV WORD PTR DS:Seg0+1,0 ;MCB ENDE als ANFANG durch

```

```

MOV     WORD PTR DS:Seg0+3,dx ;austauschen des "M" durch
MOV     AX,CS                  ;"Z"
MOV     DS,AX                  ;zurücksetzen des Datensegm.
MOV     RegSave+2,SS          ;Sichern der Register
MOV     RegSave+4,DX
MOV     RegSave+6,DS
MOV     RegSave+8,BX
MOV     RegSave+0Ah,ES
MOV     RegSave,SP
MOV     AX,CS
MOV     WORD PTR ParBlock,0    ;festhalten des alten
MOV     WORD PTR ParBlock+2,80 ;Parameter Block
MOV     ParBlock+4,AX
MOV     WORD PTR ParBlock+6,5C
MOV     ParBlock+8,AX
MOV     WORD PTR ParBlock+0Ah,6C
MOV     ParBlock+0Ch,AX
MOV     AX,CS
MOV     DS,AX
MOV     ES,AX
MOV     DX,OFFSET ProNam+2     ;Laden und ausführen
MOV     BX,OFFSET ParBlock     ;des Programms
MOV     AX,4B00
INT     21
jc      Error                  ;Sprung bei Fehler
Close:  MOV     SP,RegSave
        MOV     SS,RegSave+2    ;rückhohlen der Register
        MOV     DX,RegSave+4
        MOV     DS,RegSave+6
        MOV     BX,RegSave+8
        MOV     ES,RegSave+0Ah
        MOV     AX,CS
        DEC     AX
        MOV     DS,AX
        MOV     BYTE PTR DS:Seg0,5A ;Rücksetzen des MCB
        POP     DS:Seg0+3
Error:  INT     20              ;Programmübergabe an DOS
        mov     ah,09           ;Ausgeben des Error Textes
        mov     dx,offset ErrMes
        int     21
        jmp     Close

```

```

;*****
;*      Wandelt den Hexcode von ASCII Zeichen in die jeweiligen *
;*      um und macht somit aus vier folgenden Bytes ein Word *
;*****
;*      Eing:  BX = Adresse des Hexcodes - 1 *
;*      Ausg:  BX = Word mit ASCII Zeichen *
;*****

```

```

ASCtoWrd PROC NEAR
        push    dx              ;Register sichern
        push    cx
        mov     cx,04          ;Schleifenzähler
Loop1:  adc     bx,1
        mov     ax,[bx]
        cmp     al,01000000b    ;Testen ob Ziffer oder
        js     Number          ;Buchstabe
Alpha:  adc     ax,09h          ;Hex Code L-Byte = ASCII
        and     ax,00001111b    ;löschen des H-Byte
        push   ax              ;Ablegen auf Stack
        loop   Loop1
        jmp    SPtoWord
Number: sub     ax,30h          ;Hex Code L-Byte = ASCII
        and     ax,00001111b    ;löschen des H-Byte
        push   ax              ;Ablegen auf Stack
        loop   Loop1
SPtoWord:mov    dx,03          ;Schleifenzähler
        mov     cl,0           ;Positionsregister
        pop     bx             ;erstes Byte vom Stack holen
Loop2:  adc     cl,4
        pop     si             ;nächstes Byte vom Stack
        shl     si,cl          ;um 4 Bit nach links schieben
        or     bx,si          ;mit Rest verknüpfen
        sub     dx,1           ;Schleifenzähler um 1 kleiner
        jnz    Loop2
        pop     cx             ;zurückholen der Register
        pop     dx
        ret
ASCtoWrd ENDP

Start   ENDS

END     Begin

```

BIOS-EPROM im AT verändern

TGM_88 : SPLIT.*, 386_1.*, XLIT.*, LF.TBL
Franz FIALA / TGM

Dieser Beitrag richtet sich an alle, die sich an hardwarenahe Arbeiten am BIOS heranwagen wollen oder müssen. Die dazu notwendigen, eher 'handwerklichen' Fertigkeiten sind hier beschrieben.

Als Beispiel dient eine sehr häufige Tätigkeit beim AT-BIOS: das Erweitern der Festplattentabelle. Wenn Sie ein solches Vorhaben realisieren wollen, benötigen Sie ein EPROM-Programmiergerät, sowie ein EPROM-Löschgerät (oder eine Höhensonne, die man zwecks Schonung der Haut ohnehin nicht zum Bräunen verwenden sollte) und einige leere EPROMs (27128 oder 27256, je nach AT-Type).

Sie benötigen das Programm DEBUG des Betriebssystems und das für diesen Zweck geschriebene Programm SPLIT. Da nur sehr einfache Vorhaben auf Anhieb funktionieren, wurde auch die Möglichkeit vorbereitet, alle diese Arbeiten auch automatisch ausführen zu können.

Das AT-BIOS

Das AT-BIOS schreibt wichtige Festplattenparameter (Kopfzahl, Spurenzahl u.a.) in eine Tabelle im BIOS (genauer siehe PC-NEWS 1/87 S.53, Schlatter, AT-BIOS konfigurieren). Je nach BIOS gibt es etwa 30-50 eingetragene Festplattentypen. Mit ziemlicher Sicherheit ist Ihre neu angeschaffte Festplatte nicht dabei. Dafür bieten sich zwei Lösungen an: Entweder man verwendet eines der handelsüblichen (und bei vielen Festplatten gleich mitgelieferten) Festplatten-Utilities, wie SPEEDSTORE o.ä. oder die hier beschriebene etwas 'sportlichere' Methode der Veränderung der entsprechenden BIOS-Tabelle selbst. Die erwähnten Utilities haben den Nachteil, daß bei einer Aufteilung der Festplatte in mehrere Partitionen man einen besonderen Treiber in der Datei CONFIG.SYS geladen haben muß, der ca. 10k Speicherplatz beansprucht. Das Verändern der BIOS-Inhalte im XT ist einfach, da die Verarbeitungseinheit (Wortbreite) des XT und die Speicherbreite der EPROMs mit 8 bit gleich ist. Der AT, mit seinem 16 bit breiten Adressbus hat aber die Inhalte aller geraden und ungeraden Adressen in zwei verschiedenen EPROMs abgelegt. Daher muß bei der Anfertigung der beiden EPROMs eine Trennung des Codes in gerade und ungerade Adressen vorgenommen werden.

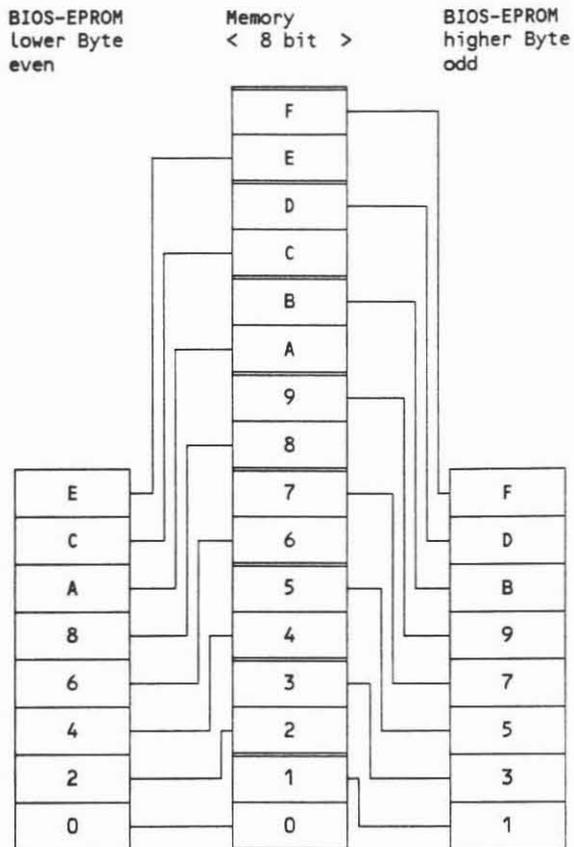
Oben in der Nebenspalte finden Sie die Speicheraufteilung auf gerade (even) und ungerade (odd) Adressen

Bei den PCs erfolgt die Zählung der Adressen bytewise, auch dann, wenn, wie beim AT, in 16-bit und beim 386-AT in 32-bit-Wortbreite auf den Speicher zugegriffen wird.

Verallgemeinert man dieses Prinzip für den AT-386, dann sollte man wegen des 32-Bit-breiten Datenbusses 4 EPROMs vorfinden. Hier hat man wahrscheinlich Platinenfläche sparen müssen, denn in diesem Adressbereich greift auch der 386er 16-bit-weise zu, es bleibt also bei dieser Aufteilung. (Neuerdings gibt es auch 386er, die das BIOS in 4 EPROMs aufteilen.)

Speichern des BIOS-Code in eine Datei

Zunächst ist es erforderlich, den BIOS-Code in einer Datei festzuhalten. Dazu benötigt man das Programm DEBUG des Betriebssystems.



Dasselbe noch einmal in Zahlen:

	----- A D R E S S E -----				
	real	segment: offset	relativ zu F8000	EPROM lo even	EPROM hi odd
BIOS-Anfang	F8000	F000:8000	0000	0000	----
	F8001	F000:8001	0001	----	0000
	F8002	F000:8002	0002	0001	----
	F8003	F000:8003	0003	----	0001
.....					
.....					
BIOS-Ende	FFFFE	F000:FFFE	7FFE	3FFF	----
	FFFFF	F000:FFFF	7FFF	----	3FFF

```
C:\>DEBUG
-mf000:8000 ffff 100
-n386_.08
-rcx
cx 0000
:8000
-rbx
bx 0000
:0
-w
Schreiben von 8000 Byte
-q
C:\>
```

Aufruf des DEBUGgers
Verschieben des BIOS-Bereichs
auf die Adresse 100h
der Name '386_' wurde gewählt, da das BIOS
eines AT-386 disassembliert werden sollte
das Suffix 08 deutet auf die 8-bit-Bus-
Version hin
die Länge des zu speichernden
Bereichs wird in den Registern
CX (offset) und BX (segment)
festgelegt

dient nur zur Kontrolle, daß BX=0

schreibt die Datei 386_.08
auf die Diskette

Modifikation des BIOS (Patch)

Jetzt ist es erforderlich die gewünschte Änderung anzubringen. Dazu fertigt man zunächst eine Kopie des Original-BIOS an (386_1.08) und lädt diese in den DEBUGger. Die hier folgenden Schritte sind nur beispielhaft angegeben.

Zunächst wird die Kopie des BIOS im DEBUGger modifiziert. Danach muß diese Datei wieder auf zwei Teile aufgespalten werden. Dazu wurde das Hilfsprogramm SPLIT geschrieben. (Beschreibung siehe später)

```
C:\>DEBUG 386_1.08      Laden des BIOS
-ehxxx                 Änderung der gewünschten Speicherzellen
...                   dabei muß beachtet werden, daß bei allen
...                   Adressen 100h addiert werden muß,
...                   da der DEBUGger die Datei auf den Offset
                        100h lädt
-e80ff                 jedenfalls sollte aber das höchste Byte
xx:00                 im BIOS auf Null gesetzt werden,
                        damit die spätere Prüfsummenbildung
                        einfacher wird
                        mit Änderungen zurückspeichern
-w
Schreiben von 8000 Byte
-q
C:\>SPLIT 2 386_1      Das Programm SPLIT teilt 386_1.08 in
                        zwei Dateien 386_1.160 und 386_1.161 auf
                        und liefert etwa folgende Meldung:
```

BINÄRDATEIEN für 8/16-Bit Bus SPLITEN/KOMBINIEREN

```
=====
386_08 = 386_160 + 386_161 : Dateinamen
32768(8000) = 16384( 4000) + 16384( 4000) : Byteanzahl
3840( f32) = 3970( fb4) + 4294967166(ffffff7e) : Prüfsummen
```

Die Zahlen sind dezimal (hexadezimal) angegeben.

Das Programm nimmt an, daß die aufzuspaltende Datei die Extension '08' hat und spaltet die Datei in zwei halb so große Dateien mit den Endungen '160' und '161' auf. Die Endungen 08 und 16 sind die Busbreite. Zusätzlich zu der Anzahl der verarbeiteten Bytes sind auch noch die Prüfsummen angegeben. Der Rechner verweigert nach der Modifikation bei falscher Prüfsumme den Dienst. Die Prüfsumme wird im letzten Byte des BIOS (f000:ffff) gespeichert. Man muß nochmals in den DEBUGger einsteigen und an der höchsten Stelle die Zahl (hier) 32h eintragen (32h ist die niederwertige Stelle der Prüfsumme des BIOS im o.a. Beispiel).

Es stellte sich bei den Versuchen bald heraus, daß es bei der ersten Änderung nicht blieb, es folgte eine zweite und... es war vorteilhafter den ganzen Vorgang zu automatisieren.

Automatische Änderung des BIOS

Eine nicht sehr häufig benutzte, aber manchmal sehr praktische, Möglichkeit automatischer Eingaben aus einer Datei, statt von der Konsole ist der Operator '<' (= Umlenkung der Eingabe von Datei oder Gerät) auf Betriebssystemebene. Man kann so den gesamten Dialog mit dem DEBUGger in einer Datei in aller Ruhe fehlerfrei vorformulieren und kleine Änderungen in dieser Datei editieren.

Ein Beispiel für eine solche Debugger-Sitzung finden Sie in der Datei 386_1.TXT und nebenstehend:

Die beiden ersten Zeilen zeigen nur ein Detail aus dem Speicher, sozusagen als Quittung, daß die richtige Datei geladen wurde. Danach beginnt der eigentliche Patch: es werden drei Bereichen mit je 16 Bytes (die Festplatten-Parameter) beginnend bei 6670, 6780 und 6790 geändert. Der

Patch auf 67a0 dient als Abschlußzeichen, der bei 70cb war ein Versuch das BIOS zum Lesen von Disketten mit 18 Sektoren zu bewegen (was allerdings nicht gelang). Zum Abschluß wird 0 auf die höchste Speicherzelle geschrieben.

Jetzt sollte man eigentlich nur mehr das Umlenkkommando anwenden müssen?

```
DEBUG 386_1 < 386_1.TXT
```

Leider nein! Diese Umlenkung verträgt die in der Textdatei enthaltenen Zeichenkombinationen Wagenrücklauf-Zeilenvorschub CR-LF nicht. Es muß daher zuerst das Zeichen Zeilenvorschub LF=0ah aus der Datei entfernt

werden. Mit dem Editor geht das gar nicht; dazu braucht man ein Utility. Ich verwendete das sehr universelle Public-Domain-Programm XLIT. XLIT substituiert jedes gewünschte Zeichen durch ein anderes oder auch durch nichts. Dazu benützt es eine Tafel, in der diese Entsprechungen eingetragen sind (Näheres siehe in der Dokumentation XLIT.DOC). In diesem Fall enthält diese Tabelle nur eine Eintragung. Die Tabelle erhält Namen LF.TBL (da sie das Zeichen LF eliminiert) und enthält nur eine Zeile:

```
h0a
```

Das zugehörige Kommando, welches die Substitution vornimmt:

```
XLIT <386_1.TXT >386_1.PAT LF.TBL
```

Die neue Datei 386_1.PAT enthält jetzt nur mehr Zeilen, die mit CR statt mit CR-LF abgeschlossen sind.

Nach diesen kleinen Hürden haben wir es geschafft: Wir haben einen einfachen Mechanismus, mit dem das Ändern der ERPOM-Inhalte problemlos möglich ist. Die zugehörige BAT-Datei heißt 386_1.BAT und enthält:

```
xlit < 386_1.txt > 386_1.pat lf.tbl
copy 386_08 386_1.08
debug 386_1.08 < 386_1.pat
split -2 386_1
```

Das Patchen des BIOS erfolgt durch Editieren der Datei 386_1.txt. und Aufruf der obigen Prozedur. Die Prüfsumme ist im allgemeinen ungleich Null. Die Differenz wird ausgerechnet und mittels Editor in der Datei 386_1.txt eingetragen (Bis jetzt stand an dieser Stelle der Wert Null). Und danach die BAT-Datei noch einmal aufrufen. Die Prüfsumme ist jetzt 0, wenn richtig gerechnet wurde.

Das Programm SPLIT (Compiliert mit ZORTECH-C-Compiler Vers. 3.0) beginnt auf der nächsten Seite.

```
d100
d6500
e65e0
00 22 05 07 00 00 ff ff 00 00 00 00 22 05 11
e6670
00 00 04 06 00 00 ff ff 00 00 00 00 00 04 11
e6780
00 32 01 02 00 00 ff ff 00 00 00 00 31 01 11
e6790
00 00 04 05 00 00 ff ff 00 00 00 00 00 04 11
e67a0
00
e70cb
12
e80ff
00
w
q
```

```
/* split.c */

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>

#define BUF_S 2048 /* Die Datei wird in 4k-Portionen bearbeitet */
#define BUF_O BUF_S/2

FILE *fs,*f0,*f1;
char BufferS [BUF_S];
char Buffer0 [BUF_O];
char Buffer1 [BUF_O];

char Dateis [50]; /* Dateiname ohne Pfadangabe */
char Datei0 [50]; /* Dateiname ohne Pfadangabe */
char Datei1 [50]; /* Dateiname ohne Pfadangabe */

void Namenbildung (char *Name)
{
    int i;

    if (Name[0]==0)
    {
        printf ("Kein Dateiname angegeben\n");
        exit (0);
    }
    if (strlen(Name)>8)
    {
        printf ("Dateiname zu lang\n");
        exit (0);
    }
    for (i=0;i<strlen(Name);i++)
        if (!( isalnum (Name[i]) || (Name[i]=='_') || (Name[i]=='-')) )
        {
            printf ("%c ist nicht erlaubt\n",Name[i]);
            exit (0);
        }
    strcpy (Dateis,Name);
    strcpy (Datei0,Name);
    strcpy (Datei1,Name);
    strcat (Dateis,".08");
    strcat (Datei0,".160");
    strcat (Datei1,".161");
}

void spalt ()
{
    unsigned gelesen,gelesen0,gelesen1,i;
    unsigned long anzahl,anzahl0,anzahl1;
    unsigned long summe,summe0,summe1;

    fs=fopen (Dateis,"rb");
    if (fs==0)
    {
        printf ("Datei %s nicht gefunden \n",Dateis);
        exit (0);
    }

    f0=fopen (Datei0,"wb");
    if (f0==0)
    {
        printf ("Datei %s kann nicht eröfnet werden \n",Datei0);
        exit (0);
    }

    f1=fopen (Datei1,"wb");
    if (f1==0)
    {
        printf ("Datei %s kann nicht eröfnet werden \n",Datei1);
        exit (0);
    }

    anzahl=0;anzahl0=0;anzahl1=0;
    summe=0;summe0=0;summe1=0;
}
```

```

while ((gelesen=(unsigned)fread (BufferS,1,BUF_S,fs))!=0) /* Jeweils 1 k wird aus der Datei gelesen */
{
    i=0;
    while (i<gelesen)
    {
        Buffer0[i/2]=BufferS[i];
        summe0=summe0+(unsigned long)BufferS[i];
        summe=summe+(unsigned long)BufferS[i];
        i++;
        Buffer1[i/2]=BufferS[i];
        summe1=summe1+(unsigned long)BufferS[i];
        summe=summe+(unsigned long)BufferS[i];
        i++;
    }
    gelesen0=(gelesen/2)+(((gelesen%2)==0)?0:1);
    gelesen1=(gelesen/2);
    anzahl=anzahl+(unsigned long)gelesen;
    anzahl0=anzahl0+(unsigned long)gelesen0;
    anzahl1=anzahl1+(unsigned long)gelesen1;
    fwrite (Buffer0,1,gelesen0,f0);
    fwrite (Buffer1,1,gelesen1,f1);
}
printf ("%17s = %17s + %17s : Dateinamen \n",DateiS,Datei0,Datei1);
printf ("%7lu(%8lx) = %7lu(%8lx) + %7lu(%8lx) : Byteanzahl dez(hex)\n",
        anzahl,anzahl,anzahl0,anzahl0,anzahl1,anzahl1);
printf ("%7lu(%8lx) = %7lu(%8lx) + %7lu(%8lx) : Prüfsummen dez(hex)\n",
        summe,summe,summe0,summe0,summe1,summe1);

if (fs) fclose (fs);
if (f0) fclose (f0);
if (f1) fclose (f1);
}

```

```

void kombi ()
{
    unsigned gelesen0,gelesen1,i;
    unsigned long anzahl,anzahl0,anzahl1;
    unsigned long summe,summe0,summe1;

    f0=fopen (Datei0,"rb");
    if (f0==0)
    {
        printf ("Datei %s nicht gefunden \n",Datei0);
        exit (0);
    }

    f1=fopen (Datei1,"rb");
    if (f1==0)
    {
        printf ("Datei %s nicht gefunden \n",Datei1);
        exit (0);
    }

    fs=fopen (DateiS,"wb");
    if (fs==0)
    {
        printf ("Datei %s kann nicht eröffnet werden \n",Datei0);
        exit (0);
    }

    anzahl=0;anzahl0=0;anzahl1=0;
    summe=0;summe0=0;summe1=0;
    while ((gelesen0=(unsigned)fread (Buffer0,1,BUF_0,f0))!=0)
    {
        gelesen1=fread (Buffer1,1,BUF_0,f1);

        if ( abs ((gelesen0-gelesen1)<2) )
        {
            i=0;
            while (i<gelesen0)
            {
                BufferS[2*i]=Buffer0[i];
                summe=summe+(unsigned long)Buffer0[i];
                summe0=summe0+(unsigned long)Buffer0[i];
                if (i<gelesen1)
                {
                    BufferS[2*i+1]=Buffer1[i];
                    summe=summe+(unsigned long)Buffer1[i];

```

```

        summe1=summe1+(unsigned long)Buffer1[i];
    }
    i++;
}
fwrite (BufferS,1,gelesen0+gelesen1,fs);
anzahl=anzahl+(unsigned long)gelesen0+(unsigned long)gelesen1;
anzahl0=anzahl0+(unsigned long)gelesen0;
anzahl1=anzahl1+(unsigned long)gelesen1;
}
else
{
    printf ("Dateien %s und %s sind verschieden lang!\n",Datei0,Datei1);
    exit (0);
}
}
printf ("%15s + %15s = %15s : Dateinamen\n",Datei0,Datei1,DateiS);
printf ("%7Lu(%6lx) + %7Lu(%6lx) = %7Lu(%6lx) : Byteanzahl dez(hex)\n",
        anzahl0,anzahl0,anzahl1,anzahl1,anzahl,anzahl);
printf ("%7Lu(%6lx) + %7Lu(%6lx) = %7Lu(%6lx) : Prüfsummen dez(hex)\n",
        summe0,summe0,summe1,summe1,summe,summe);

if (fs) fclose (fs);
if (f0) fclose (f0);
if (f1) fclose (f1);
}

void Hilfe ()
{
    printf ("\nAufspaltung einer Datei 'name.08' in zwei Dateien, \n");
    printf ("mit den geraden Adressen 'name.160' \n");
    printf ("und ungeraden Adressen 'name.161' \n");
    printf ("''ERPOM -2 name' : name.08 -> name.160 + name.161\n");
    printf ("\nVereinigung zweier Dateien \n");
    printf ("name.160' mit den geraden Adressen und \n");
    printf ("name.161' mit den ungeraden Adressen zu \n");
    printf ("einer Datei 'name.08' \n");
    printf ("''ERPOM -1 name' : name.160 + name.161 -> name.08\n");
}

void main (int argc,char *argv[])
{
    printf ("\nBINÄRDATEIEN für 8/16-Bit Bus SPLITEN/KOMBINIEREN\n");
    printf ("=====\n");

    switch (argc)
    {
        case 1:
            printf ("Zuwenig Argumente\n");
            goto Fehler;
        case 2:
            if ( (argv[1][0]!='-') && (toupper (argv[1][1])=='H') )
            {
                Hilfe ();
                break;
            }
            else
            {
                printf ("Falsche Option %s, zuwenig Argumente\n");
                goto Fehler;
            }
        case 3:
            if (argv[1][0]!='-') /* Option angegeben */
                switch (toupper (argv[1][1]))
                {
                    case 'H': Hilfe ();
                                break;
                    case '1': Namenbildung (argv[2]);
                                kombi ();
                                break;
                    case '2': Namenbildung (argv[2]);
                                spalt ();
                                break;
                    default: printf ("Falsche Option \n");
                                goto Fehler;
                }
            break;
        default:
            printf ("Zuviele Argumente\n");
            Fehler:
            printf ("EPROM -(h12) name    -h..Hilfe, -1 kombi, -2 spalt\n");
            break;;
    }
}
}

```

8-4-2-1

TGM_98 : REIHE421.BAS

Dipl.-Ing. R. Neubauer / TGM

Bezug: Wissenschaftliche Nachrichten Nr.79 - Jänner 1989, Seite 23 u. 24: Perfektion und Information im Mathematikunterricht 1.

"Der Lehrer läßt sich eine kleine natürliche Zahl $N = a_1$ nennen und die weiteren Glieder einer Folge nach der Vorschrift

$$a(n+1) = a(n) / 2 \text{ für gerades } a(n) \text{ bzw.}$$

$$a(n+1) = a(n) * 3 + 1 \text{ für ungerades } a(n)$$

berechnen. Für z.B. $N = 5$ ergibt sich die Folge

5, 16, 8, 4, 2, 1, 4, 2, 1,

für $N = 11$

11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1,

Obwohl das Einmünden in die Periodizität 4, 2, 1 bisher immer eintrat, ist die Frage nach der Allgemeingültigkeit des Ergebnisses noch offen.

Der Vorschlag, die Schüler mögen doch nach Ausnahmen suchen, wird sicher schon in den untersten Klassen zur Eigentätigkeit anspornen und dabei das Rechnen festigen."

Soweit aus o.a. Zeitschrift zitiert !

Das kleine BASIC - Programm REIHE421.BAS kann die Rechenarbeit ganz erheblich erleichtern:

```
50 INPUT A
70 IF A MOD 2=0 THEN PRINT A;
100 IF A MOD 2=0 THEN A=A/2
150 PRINT A;
200 IF A=0 THEN 450
210 IF A=1 THEN 450
230 IF A MOD 2=0 THEN 100
250 A=A*3+1:PRINT A;
270 IF A=1 THEN 450 ELSE 100
450 PRINT
500 INPUT A
600 IF A=0 THEN ENDE ELSE 70
```

Interessant sind die etwas längeren Ausgaben bei den Eingaben 27, 31,41, 47, 54, 55, 62, 63, 71, 73, 82, 83, 91, 94, 97, 103, 107, 108, 109, 110, 111, 124 u.125 u.s.w., aber auch die kürzeren Ausgaben bei Eingabe anderer Werte, welche wirklich alle mit der Periodizität 4, 2, 1 endigen. Beendet wird der Programmlauf durch Eingabe von 0.

Erfahrungen mit Symphony 1.1

G. Silberbauer / TGM

Symphony ist ein universell verwendbares Programmpaket zur Textverarbeitung, Tabellenkalkulation, als Datenbank, als Grafikpaket und für Kommunikation, alles unter einer Benutzeroberfläche.

Wenn Sie an Ihre Textverarbeitung hohe Anforderungen stellen und viele Schriftarten wünschen oder eine deutsche Rechtschreibhilfe etc., dann vergessen Sie Symphony - es gibt viel bessere Nur-Textverarbeitungs-Programme!

Doch für normale Briefe oder Formbriefe zusammen mit der immer bereiten Datenbank und gemischt mit Tabellenkalkulation (z.B. Rechnungswesen, Mahnwesen etc.) kommt man ganz gut mit Symphony aus.

Meine Überlegung war ursprünglich die, daß es einfacher sein müßte, ein einziges Programm gut zu lernen als verschiedene für verschiedene Zwecke. Das glaube ich auch jetzt noch. Leider gibt es bei uns noch sehr wenige Symphony-User, sodaß man leichter jemanden findet, der einem bei dBase 3+ hilft oder bei Word als bei Symphony. So muß man sich vieles selbst erarbeiten mit Hilfe von Büchern und dem Tutor.

Leider gibt es auch keine Kollegen, die wirklich verschiedene Programmpakete gleich gut beherrschen und objektiv zwischen den Paketen vergleichen können. So kann ich nur von eigenen Erfahrungen und von Bemerkungen anderer Kollegen zu meinen Problemen berichten.

Eine der Hauptstärken von Symphony ist die ausgereifte Tabellenkalkulation. Damit ist es einfach, komplexere Tabellen anzulegen, die sich selbst berechnen. Ich habe z.B. eine selbstgemachte "Umgebung", bestehend aus mehreren Fenstern, zur Berechnung von allem, was ich am 10. jedes Monats für das Finanzamt brauche: Lohnkonten wegen Lohnsteuer und Dienstgeberbeitrag, Rechnungsübersicht wegen Mehrwertsteuer, Vorsteuern von Kassa und Banken, Überstundenabrechnungen etc. Das läuft auf dem Turbo-XT mit 640 k, jedoch lade ich dieses File ca. 40 Sek. von der Harddisk - die Berechnung dauert eben. Ähnliche große Files habe ich als Kundenkartei (Datenbank) und als Rechnungsverzeichnis.

Die Datenbank arbeitet anders als in dBase: es wird immer die ganze Datenbank in den Rechner geladen. Da gibt es Grenzen mit 640 k, doch wenn einmal geladen ist (auch das dauert über 30 Sek. bei einer ausführlicheren 1200-Personen-Datenbank), geht alles schneller als in dBase. Man kann auch viel programmieren und hat dazu mächtige Befehle, kann Makros schreiben, automatisch ablaufen lassen beim Laden des Files oder mit Kurzbezeichnungen aufrufen oder mit einem Makro-Manager überall verwenden etc. - man kann immer noch dazulernen.

Programm SYSTEST

TGM_98 : SYSTEST.EXE, ILTEST.EXE, LESEN.EXE

Zitate aus den Hilfe-Menüs:

"Das Programm ermöglicht Ihnen den Test Ihrer Systemkomponenten. Folgende Tests sind möglich: Prozessor Festplatte Bildschirm Setup Information Vergleich. Wählen Sie die Komponente aus dem Hauptmenü durch Eintippen ihres ersten Buchstabens (P, F, B) oder benutzen Sie die Pfeiltasten und die Return-Taste."

"Wenn Sie Ihre Resultate in eine Datei speichern wollen, starten Sie das Programm einfach mit:

SYSTEST laufwerk:dateiname."

"Die Option Vergleich (V) erlaubt es Ihnen, Ihr Gerät mit verschiedenen Standards zu vergleichen: Dem 'klassischen' IBM PC/XT und den drei neuen IBM-PS/2-Systemen. Wählen Sie in diesem Submenü Ihr Vergleichssystem mit den Pfeiltasten aus und drücken Sie <RETURN>. Standardeinstellung ist: IBM PC/XT." Ein Beispiel für die Ausgaben von SYSTEST bei einem portablen AT mit Hercules-Monitor und 40MB Festplatte finden Sie umseitig.

SYSTEST-Ausgabebeispiel:**Prozessor**

Prozessortyp	Intel 80286
Datenbus	16 Bit
Taktfrequenz	12.05 MHz
Datendurchsatz	3820497 BPS
RAM Wait States	1

Leistungsindex verglichen mit IBM PC/XT

Benchmark 1	4.773 Sekunden
Leistungsindex 1	7.025
Benchmark 2	4.953 Sekunden
Leistungsindex 2	4.005

Festplatte

Bytes pro Sektor	512	FAT Sektoren	064
Sektoren/Speicherblock	004	Directory-Einträge	512
Sektoren/Spur	017	Verborgene Sektoren	017
Lese/Schreibköpfe	005	Reservierte Sektoren	001
Zylinder	976	Spuren	4880
FAT Kopien	002	Plattenkapazität (MB)	42.5

Spur zu Spur Zugriffszeit	5.12 Millisekunden
Durchschnittliche Zugriffszeit	25.00 Millisekunden
Maximale Zugriffszeit	40.00 Millisekunden

Interleave-Faktor 2

Datendurchsatz bester Fall	253592 Bytes / Sekunde
Datendurchsatz schlechtester Fall	94873 Bytes / Sekunde

Leistungsindex verglichen mit IBM PC/XT 3.057

Bildschirm

Video RAM Datendurchsatz	144073 Bytes/Sekunde
ROM BIOS Video Datendurchsatz	4060 Bytes/Sekunde
DOS Video Datendurchsatz	2375 Bytes/Sekunde
Video RAM Wait States	5

Leistungsindex verglichen mit IBM PC/XT

Video RAM Ausgabe	3.305
BIOS Video Ausgabe	2.719
DOS Video Ausgabe	4.264

Setup

Gerätetyp	IBM AT oder kompatibel
DOS Version	3.30

Gegenwärtige Installation

Bildschirmtyp	MONOCHROM
Game Adaptor	NICHT INSTALLIERT
Math. Koprozessor	NICHT INSTALLIERT
Systemspeicher	640 KB
Parallele Schnittstellen	1
Serielle Schnittstellen	1
Festplatten-Laufwerke	1
RAM-Erweiterung	384 KB
Festplattentyp Nr.	15
Diskettenlaufwerke	1
High-Density-Laufwerke	0
Low-Density-Laufwerke	1

Berichte Hardware

Ein drittes Laufwerk an der Multi-I/O-Platine.

Mag. Peter Schneeweis / HTBLA f. Mode u. Bekleidung

Ich habe einen XT und war kürzlich vor die Aufgabe gestellt, Daten von einem 3 1/2 Zoll Laufwerk (720 kb) zu übernehmen. Nachdem ich mir die Multi-I/O-Platine etwas näher angesehen hatte, ergab sich eine verblüffend einfache Lösung.

Im 8-D-Typ-Register 74SL173, bezeichnet als U12 sind alle Motor-On-Leitungen und die Drive-Select für alle vier Drives anliegend. Weiters sind 3 Nand-Glieder mit offenem Kollektor am Chip U2 vorhanden. Damit ist das Problem sDrähte und fertig.

Der Anschluß 16 des U12 ist mit den Anschlüssen 1,2 und 4 des U2 zu verbinden. Weiters ist der Anschluß 5 von U12 mit dem Anschluß 5 von U2 zu verbinden. Die Anschlüsse 3 und 6 von U2 sind jeweils mit Widerständen von 150 bis 320 Ohm an +5V zu legen. (offener Kollektor). Anschluß 3 von U2 ist mit Anschluß 1 des Steckers zu verbinden, Anschluß 6 von U2 mit Anschluß 5 des Steckers. Die Steckeranschlüsse 2,4 und 6 sind an Masse zu legen, der Anschluß 3 bleibt offen. Damit sind die Änderungen an der Platine fertig.

Am Verbindungskabel zu den Diskettendrives ist ein dritter Stecker zu montieren, wobei die Leitungen 11 bis 16 zu entfernen sind so durch das neue Verbindungskabel von dem 6-poligen Stecker zu ersetzen sind, daß der Steckeranschluß 16 mit Leitung 1, 15 mit 2 usw. bis 11 mit 6 zusammenkommen.

Berichte, Allgemeines

HiSoft '89

Vielleicht kann Ihnen diese Sonderausgabe des TREND-PROFIL-EXTRA von Gergely & Göschl helfen, einen Überblick über die Leistungsfähigkeit von Software geben.

Aus dem Inhalt:

Die zehn Gebote für PC-Anwender, Der Computermarkt gestern, heute, morgen, Kampf der Betriebssysteme, Buchhaltung, Glossar, Firmenverzeichnis

Getestet und verglichen wurden:

TEXTVERARBEITUNG: Comfotex, Enable, Euroscript, Framework II und III, GEM 1st Word Plus, PC Text 4, Writing Assistant, Lotus Manuscript, Microsoft Word, Microsoft Works, Multimate Advantage, Open Access II, Pharaoh, Q-One, Star Writer, Symphony, Tex-Ass Window Plus, Windows Write, Word Perfect, Wordstar 2000, Wordstar 4.0 Extra

DATENBANK: dBase III+, Enable, Excel, F&A, Foxbase plus, Framework II und III, GBase, Lotus 1-2-3, Microsoft Works, Open Access II, Paradox, Pharaoh, Rapid File, RBase, Symphony

TABELLENKALKULATION: Enable, Excel, Framework II und III, Lotus 1-2-3, Microsoft Works, Multiplan, Open Access II, Pharaoh, Quattro, Star Planer, SuperCalc 4, Symphony

INTEGRIERTE PROGRAMME: Enable, Excel, Framework II und III, Microsoft Works, Open Access II, Symphony

GRAFIK: Chart Master, GEM Graph, GEM Presentation Team, Graph Plus, Harvard Graphics, Microsoft Chart, Perspective, VCN Concorde

ZEICHENPROGRAMME: Freelance Plus, GEM Draw Plus, Windows Draw

MALPROGRAMME: GEM Paint, PC Paintbrush, Windows Paint

DESKTOP PUBLISHING: Byline, First Publisher, GEM Desktop Publisher, Page Maker Ventura Publisher

UTILITIES: DS Backup+ und DS Recover, Floppy Driver, Norton Commander, Norton Utilities, PC+Softlock und PC+Master, PC-Tools, PKarc, Safe-Guard, Sideways, Star Manager, Turbo Backup, Word for Word.

VIREN in Nippon

Ing. Alois Lippert

Abschrift aus der Sonderbeilage der Süddeutschen Zeitung Nr.260 vom Dienstag 10.November 1988 - übersetzter Artikel aus der japanischen Zeitung ASAHI SHIMBUN. Publiziert zwischen September-Oktober 1988.

"COMPUTER-VIREN" aufgetaucht

Zum ersten Mal ins landesinnere Netz eingedrungen

Laut Meldung vom 13.September ist ein "Computer-Virus", der sich von Computer zu Computer überträgt und dort gespeicherte Daten löscht oder verändert, in das größte Computer-Netz des Landes, PC-VAN, das vom NEC (Nippon Electric Company) betrieben wird, eingedrungen. Da dieser Virus das Kennwort des Benutzers, in dessen Computer er eingedrungen ist, stiehlt und da man davon ausgeht, daß schon etwa zehn Personen Schaden erlitten haben, ist man im PC-VAN-Büro dazu übergegangen, die ca. 45 000 Mitglieder über das Netz zu warnen. Während in Europa und den USA dieser neue "Virus" um sich greift und man sich vergeblich bemüht, ihn auszurotten, ist sein Eindringen in ein japanisches Benutzer-Netz eine Premiere.

Der sogenannte "Computer-Virus" ist eigentlich eine Art Programm.

Es hat diesen Namen erhalten, da es sich in die Computer anderer Leute einschleicht und vermehrt. Es ist jetzt zwei Wochen her, daß das Eindringen des Virus in das PC-VAN-Netz bemerkt wurde. Ein Mitglied hat eine merkwür-

dige Nachricht auf einer "Angebots-Tafel" des Netzes entdeckt und die Dienststelle benachrichtigt. Die darauf erfolgte Untersuchung ergab, daß es sich hier um eine neue bössartige Sorte handelte, welche die Kennwörter anderer stiehlt. Mit einem gestohlenen Kennwort ist es möglich, innerhalb des Netzes die Rolle des Besitzers zu übernehmen, zum Beispiel eigene Benützergebühren auf ihn zu übertragen oder unter seinem Namen Einkäufe zu tätigen (on-line shopping). Der Virus sei, so die Dienststelle, in einem Programm eingebaut, das der Täter per elektronischer Post an das Opfer geschickt habe. Wenn das Opfer ein solches Programm benutzt, schleicht sich der Virus in die Basis-Software ein. Wenn sich das Opfer dann mit dem PC-VAN-Host-Computer in Verbindung setzt, zeigt sich der Virus in seiner wirklichen Gestalt. Er verschlüsselt das Kennwort des Opfers und schreibt es in die "Angebots-Tafel" des PC-VAN-Netzes. Später sieht sich der Täter dann die Tafel an und kommt so an das Kennwort. So sieht der komplizierte Mechanismus aus. Der Täter, der den Virus geschickt hat, scheint schon von Anfang an gestohlene Benutzer-Nummern (user-ID) und Kennwörter benutzt zu haben; er konnte noch nicht bestimmt werden. Aus diesem Grunde hat das PC-VAN-Büro am 12. September damit begonnen, die Mitglieder davon in Kenntnis zu setzen, daß der Virus in die Tafel des Netzes eingedrungen ist und sie zu warnen. Gleichzeitig wurde, um die Ausweitung des Schadens festzustellen, zur Mitarbeit an einer Untersuchung aufgefordert. Was ist zu tun, um Schaden durch den Virus zu verhindern? Keine verdächtigen Programme benutzen, die per elektronischer Post kommen. Mitglieder haben bereits einen "Impfstoff" entwickelt, mit dessen Hilfe es möglich ist, festzustellen, ob der eigene Computer vom Virus infiziert ist oder nicht, und den das PC-VAN-Büro bereits über das Netz verteilt. Außerdem heißt es, daß auch bei einer "Infektion" nichts passieren, wenn man die Basis-Software in ihre ursprüngliche Lage bringt und das Kennwort ändert.

PS: Glaubhafte Quellen versichern uns, daß der vorstehend beschriebene Virus mit Sicherheit **nicht** wie hier abgebildet aussieht und daher leider auch nicht ganz leicht zu erkennen ist.



Auf der Hannover CeBit 1989

...mit Berücksichtigung des Themas Taiwan
Dipl. Ing. A. Zandomeneghi

Da es mir heuer möglich war, die 1036 km nach Hannover zur Messe zu bewältigen, möchte ich einige für Clubkollegen interessante Neuigkeiten zusammenfassen:

Was das Thema "Taiwan" betrifft, kann man hier sozusagen eine "Konsolidierungsphase" von v.a. größeren Firmen beobachten. Wie ich aus Gesprächen mit einigen Firmenvertretern erfahren habe, wird es in Hinkunft nicht mehr möglich sein, bei Firmen direkt in Taiwan zu bestellen, die schon in Österreich eine Vertretung haben. Dies gilt z.B. für "Qtronix" (Mäuse,

Bildschirme), von der ich in einem vergangenen Artikel berichtet habe. Die Vertretung wird in Hinkunft laut einem "offiziellen" Blatt der Firma in Österreich von "Herlango" übernommen. Für viele Clubmitglieder wohl keine sehr positive Nachricht! Ich werde aber in Hinkunft auf kleinere taiwanische Firmen ohne Vertretung hierzulande ausweichen.

Als Anlaufstelle für Fragen kann ich das "CETRA" (C_hina E_xternal T_rade D_velopment C_ouncil) empfehlen, das auch auf der CeBit vertreten war. Die Aufgabe dieser Institution ist es, eine Art Standort- und Industrierwerbung für das Land zu machen. Ein Exemplar der wirklich informativen Broschüre "Microcomputers in Taiwan, ROC 1989" wurde mir von einer freundlichen Chinesin am CETRA-Stand überlassen (Danke Fräulein Wu !), nachdem ich einen Fragebogen betreffend einen Vergleich der Fernost-Computerländer Taiwan, Japan, Hongkong und Singapur beantwortet hatte. Das Heft umfaßt etwa 100 Seiten und enthält eine komplette Aufstellung aller taiwanischen Hersteller, aufgeschlüsselt nach der Angebotspalette; Firmenprofile (Gründung der Firma, Anzahl der Beschäftigten etc.) sowie eine Beschreibung der angebotenen Hardware! Ich werde mein Exemplar dem Verein zum Zweck des Kopierens zur Verfügung stellen.

Da ich in den vergangenen Monaten von mehreren Clubmitgliedern in Bezug auf die neueste Preissituation in Taiwan, speziell betreffend 386er-Boards, angeschrieben worden bin, hier einige allgemeine Informationen; man kann die angebotenen 386 Board m.E. in vier Kategorien einteilen (Stand März '89):

Taktfrequenz	Preis (ungefähr)
) 386 mit 16 MHz	850 US\$
) 386 mit 20 MHz	1000 US\$
) 386 mit 25 MHz	1450 US\$
) 386 mit 25 MHz mit Cache-Speicher	1700 US\$

In allen genannten Preisen sind 1MB Speicherbausteine enthalten. Dazu müssen noch 20 % Mehrwertsteuer sowie die Transportkosten gerechnet werden. Die Musterpreise beziehen sich auf Angebote der Firma "Digicom", doch ist die Preissituation bei anderen taiwanischen Firmen, schon aufgrund der Konkurrenzsituation ähnlich. Durch die angespannte Lage auf dem Markt für Speicherbausteine ist aber der Preis für RAMs in Taiwan auch nicht viel niedriger als hierzulande.

Bleibt nur noch anzumerken, daß es ganz und gar unmöglich war, das riesige Angebot auf der Hannover Messe an einem Tag zu inspizieren, obwohl ich 8 Stunden unterwegs war. Doch kann man die bekannten Anbieter (Microsoft, IBM etc.) auch auf der Wiener "Ifabo" besuchen. Ich kann aus eigener Erfahrung sagen, daß in Österreich auf den Messen v.a. Stände mit Anbietern aus Fernost sowie aus Übersee fast gänzlich fehlen, wobei in Hannover allein die USA und Kanada auf der CeBit fast eine ganze Halle (!) für sich in Anspruch nehmen. Falls Clubmitglieder Interesse an diesen Adressen haben, mögen sie sich mit mir in Verbindung setzen.

CA, die Bank zum Erfolg.



Man kann nie früh genug beginnen, seinen Nachwuchs auf einen finanziellen Polster zu betten: Am besten gleich morgen mit Ihrer persönlichen Schilling-Aufwertung und einem Erfolgssparbuch.



CREDITANSTALT

Clubteil

FRAGEN

Frage:

Kennt jemand ein Grafikprogramm, das einen STAR-LC-10 Color-Drucker unterstützt? Kennt oder hat jemand DELUXE PAINT II PC?

Antworten an Mag. Walter Neidhart, Karlsdorf 15, 9851 LIESERBRÜCKE

Frage:

Suche das Programm STEN von Side&Soundstudio in der speziellen Fassung für Linkshänder. (Lern- und Trainingsprogramm für Steno-Schrift). Antworten erbeten an: Hans Schattauer, Magdalenenstraße 25/7, 1060 Wien

Frage:

Anfrage des RENIETS-VERLAG, Obkirchergasse 36, 1190 Wien, 32 72 74, Herr Anton Burger, Bitte bei Rückantwort Korrespondenzzeichen Bu/896.19 angeben!

Gesucht werden Spezialisten für die mathematische Textverarbeitung TEX, die in der Lage wären, reprofähige Blätter von ca. 150 handgeschriebenen A4-Seiten zu erstellen. Diese Vorlagen sind ein Lösungsheft für den Mathematikunterricht und enthalten mathematische Formeln. Die genauen Konditionen dieses Auftrags und die gesamte Auftragsabwicklung sind mit dem Verlag abzusprechen. TEX-Liebhaber bitte melden.

TGM-Disketten

TGM_98 : NEWS 2/89, Texte und Programme

N	TXT	136704	12.05.89	19.50	/ Texte
STANDARD	TBS	1024	12.05.89	19.51	/
GAUSS	SBR	8303	25.03.89	7.42	/ Lineares
GAUSS	SYS	780	25.03.89	7.43	/ Gleichungssystem
LINGL	DOC	830	10.05.89	0.21	/
LINGL	PAS	8762	25.03.89	7.54	/
SPLIT	C	6347	17.04.89	21.43	/
SPLIT	EXE	13828	17.04.89	21.44	/
386_1	BAT	100	22.04.89	14.49	/
386_1	PAT	258	22.04.89	15.03	/ EPROMs im AT
386_1	TXT	276	22.04.89	15.03	/ verändern
LF	TBL	5	18.04.89	9.14	/
XLIT	COM	1780	24.02.85	17.01	/
XLIT	DOC	7572	26.02.85	9.27	/
XLIT	TBL	13	15.02.88	12.07	/
REIHE421	BAS	262	12.05.89	19.52	/ 8-4-2-1
UNIDRV	ASM	28780	27.12.88	14.57	/ Fremdformate
UNIDRV	SYS	2005	27.12.88	15.00	/ im XT
PROLOAD	COM	937	6.02.89	16.50	/ PROLOAD
PROLOAD	ASM	8031	6.02.89	16.48	/
ILTEST	EXE	22635	26.02.87	12.07	/ Programm SYSTEST
LESEN	EXE	13450	1.08.87	13.00	/
SYSTEST	EXE	46809	1.08.87	13.00	/

TGM_99 : EPSON-Druckertreiber-1

Vor allem Treiber für die Drucker LQ-850 und LQ-1050 aber auch für andere 9- und 24-Nadel-Drucker.

EPSON	BAT	53	28.04.88	11.45
HILFE		490	30.06.88	13.40
READ	ME	828	28.04.88	15.08
ES	<DIR>		9.05.89	22.47
MM	<DIR>		9.05.89	22.47
PCTEXT	<DIR>		9.05.89	22.47
TWIN	<DIR>		9.05.89	22.47
WORD4	<DIR>		9.05.89	22.47
WP	<DIR>		9.05.89	22.47
WS	<DIR>		9.05.89	22.47

TGM_100 : EPSON-Druckertreiber-2

EPSON	BAT	53	28.04.88	11.45
HILFE		490	30.06.88	13.40
READ	ME	829	28.04.88	15.14
ACAD	<DIR>		9.05.89	22.48
CHART	<DIR>		9.05.89	22.48
FW	<DIR>		9.05.89	22.48
GEM	<DIR>		9.05.89	22.48
LOTUS	<DIR>		9.05.89	22.49
OA	<DIR>		9.05.89	22.49
WIN	<DIR>		9.05.89	22.49

TGM_101 : EPSON-Hardcopy

Ein residentes Bildschirmdruckprogramm für 9- oder 24-Nadel-Drucker mit und ohne Farbe, ähnlich GRAPHICS.COM des Betriebsystems, aber für CGA, HGA, MGA und EGA-Karte.

24	DOK	98532	29.10.87	15.49
9	DOK	98283	29.10.87	16.14
COPYFEST	BAT	403	9.11.87	10.48
INSTALL	EXE	39200	9.11.87	11.24
LIESMICH	BAT	1247	24.03.88	19.15
PRTINS	BAT	178	29.03.89	13.36
PRTSC24H	EXE	24827	15.10.87	1.10
PRTSC24L	EXE	24715	9.11.87	8.52
PRTSC9	EXE	25415	15.10.87	1.10

Vertrieb elektronischer Bauelemente und Geräte Gesellschaft mbH.
A-1120 Wien, Tichelgasse 10, Tel.: 0222/83 41 01-0, 87 14 00
Telefax: 87 20 07, Telex: 134606, DVR.NR.: 0449831



High Speed Festplattenset

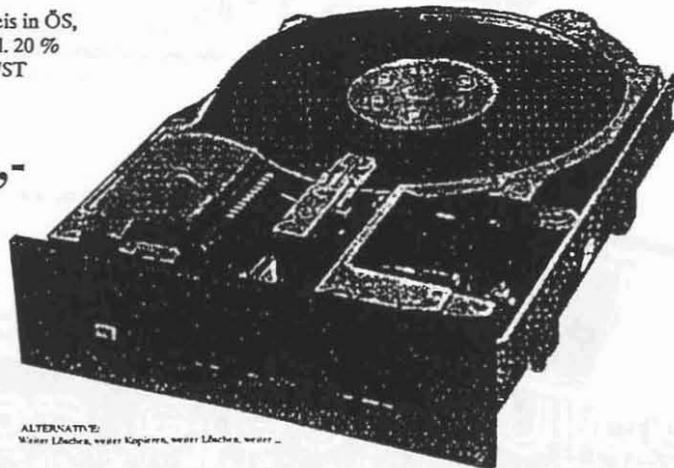
152 MByte

noch immer zu klein ?

bestehend aus:
Miniscribe ESDI Festplattenlaufwerk Model 3180 E,
halbe Bauhöhe (Slim Line),
formatierte Kapazität 152 MByte,
mittlere Zugriffszeit ca. 16 ms,
DTC 6280 kombinierter EDSI Hard- und Floppycontroller,
Interleave 1:1, Datentransferrate 850 KByte/sec

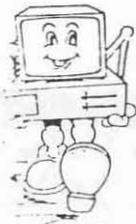
Preis in ÖS,
exkl. 20 %
MWST

23.950,-



ALTERNATIVE:
Wasser Lösliche, weisse Kopieren, weisse Lösliche, weisse ...

Bankverbindung: Die Erste Österr. Sparr-Casse-Bank Kto. 099-06657



Vertrieb elektronischer Bauelemente und Geräte Gesellschaft mbH.
A-1120 Wien, Tichelgasse 10, Tel.: 0222/83 41 01-0, 87 14 00
Telefax: 87 20 07, Telex: 134606, DVR.NR.: 0449831



Vergessen Sie alles, denn jetzt gibt's *GoScript*, den Postscript-Interpreter für Laser- und NADELdrucker

Oder wollen Sie
vielleicht das
Fünffache
für Postscript
ausgeben ?



Preise in ÖS,
exkl. 20 % MWST

Unterstützt
Expanded Memory und
arithmetischen Co-Prozessor

+++++

Erweiterte Text- und Grafikanmanipulationsmöglichkeiten für verschiedene Anwendungsprogramme wie z.B. Rotation - Skalierung - Spezialeffekte

+++++

Interaktiver Modus, um Postscript-Sprache zu entdecken, zu lernen und zu nutzen

System ?

PC-XT, AT,
386 oder kompatibel,
PC-DOS oder MS-DOS 3.xx,
640 KByte Hauptspeicher,
Festplattenlaufwerk

Optional ?

1 oder 2 MByte LimEms Expanded Memory, arithmetischer Co-Prozessor

Programme ?

Xerox Ventura Publisher,
Aldus PageMaker, Microsoft Word,
WordPerfect 5.0, Borland Quattro und viele andere

Drucker ?

HP-LaserJet Serie II und LaserJet plus, HP-DeskJet,
HP-PainJet, Canon LBP-SII, Epson FX und LQ-Serie,
NEC-Pinwriter, IBM ProPrinter, Fujitsu DL-Serie
und viele andere

GoScript

inkl. 12 Outline Fonts

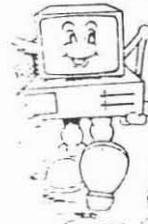
4.200,-

GoScript Plus

inkl. 35 Outline Fonts

8.500,-

Bankverbindung: Die Erste Österr. Sparr-Casse-Bank Kto. 099-06657



cebos electronics

Vertrieb elektronischer Bauelemente und Geräte Gesellschaft mbH.
A-1120 Wien, Tichtelgasse 10, Tel.: 0222/83 41 01-0, 87 14 00
Telefax: 87 20 07, Telex: 134606, DVR.NR.: 0449831



Bekennen Sie Farbe!
Und bekennen Sie sich zur Farbe!



Paintjet

Spezifikation:

Thermischer Tintenstrahldrucker,
durch sieben verschiedene Farben ergeben
sich tausende von Farbschattierungen

Max. Auflösung:

180 * 180 dpi

Druckgeschwindigkeit:

200 Zeichen pro Sekunde,
30 - 40 Sekunden für eine Textseite,
4 Minuten für eine ganzseitige Farbgrafik

Papierzuführung:

Endlos und Einzelblattpapier (Kopierpapier),
Bedrucken von Transparentfolien möglich

Schnittstellen:

wahlweise Centronic's, RS-232C, IEEE 488

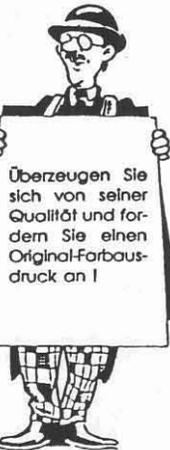
Pufferspeicher:

8 KByte

Kompatibel zu allen gängigen

Softwarepaketen, wie
z.B. Lotus 1-2-3,
Word, ACAD,
GEM ...

Flüsterleiser
Druckbetrieb



Überzeugen Sie
sich von seiner
Qualität und for-
dern Sie einen
Original-Farbaus-
druck an!



Hoffentlich
wird es
ihnen
früher nicht
zu bunt!



Preis:
in ÖS.
exkl. 20 % MWST

Ausführung:
Centronic's-
oder
RS-232C Schnittstelle

21.995,-

Bankverbindung: Die Erste Österr. Spar-Casse-Bank Kto: 098 06557



Um
14.450,-
kommen Sie sicher
nicht mehr über unser
Angebot hinweg,
noch dazu gibt's

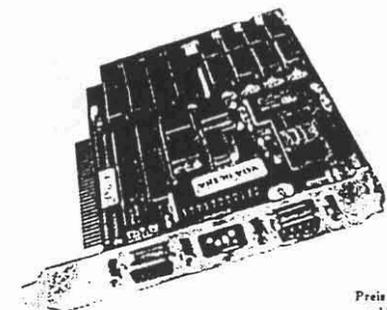
12 Monate
Garantie

Hitachi Multiscan Monitor

+ VGA-Ultra Grafikkarte

Monitor Spezifikation:

14" VGA/EGA/HGC/CGA kompatibel,
Auflösung: 800*560 (800*600),
Horizontaler Frequenzbereich: 15.5-35 KHz,
Vertikaler Frequenzbereich: 50-80 Hz,
Video-Bandbreite: 30 MHz,
Dot Pitch: 0.31 mm,
Entspiegelte Bildröhre,
RGB/TTL-Kabel, RGB/Analog-Kabel,
Dreh und Schwenkfuß



Preis in ÖS,
exkl. 20 %
MWST

VGA-Ultra Spezifikation:

512 KByte Video-RAM,
Hercules, CGA, EGA, MCGA/VGA und VGA-Mode,
Erweiterter Grafikmode: 640*350, 640*480, 800*600, 1024*768,
Text Mode: 60*80, 40*100 und 25,28,44 Zeilen * 132 Spalten,
Treibersoftware für Lotus 1-2-3, AutoCad, GEM und Windows,
Diagnose-Software, Hardware-Zoom, Pan und Windows

VGA ist ein eingetragenes Markenzeichen der
International Business Machine Corporation



BAT XT

HÄNDLERPREISLISTE

SYSTEME
und
Komponenten

Gültig ab 8.05.89

BAT X10CL

AT-Look
8088 CPU 10MHz Taktfrquenz
Sockel für 8087
256 KB Hauptspeicher (erweiterbar auf 640 KB)
1 Floppylaufwerk 360 KB
Parallele Schnittstelle
Deutsche MF-Tastatur 102 Key
Hercules kompatible Graphik-Display-Karte
12" Monochrom-Bildschirm

8,480.-

BAT X11CL

Wie X10CL jedoch mit
2 Floppylaufwerke 360 KB
512 KB Hauptspeicher

10,100.-

BAT X12CL

Wie X10CL jedoch mit
512 KB Hauptspeicher
Serielle Schnittstelle
Batteriegepufferte Uhr mit Datum
Festplattenkontroller für 2 HD
1 Festplatte 20MB 65ms

13,180.-

Aufrüstung von 512KB auf 640KB 990.-

Umrüstung auf andere Bildschirmsets siehe Seite 6

Technische Änderungen, Irrtümer und Druckfehler vorbehalten

Beachten Sie bitte unsere Liefer und Zahlungsbedingungen

ALLE PREISE EXCL. MWST.

Telefax 513 48 60/17

DVR 0477184 • Bankverbindung Z-Kto.-Nr. 685 028 300 Bank Winter & Co Kto.-Nr. 43 88 005



BAT AT

BAT M20CL

Standart AT Gehäuse
80286 CPU 12 MHz Taktfrequenz 0 Wait
Sockel für 80287
640KB Hauptspeicher (bis 4MB on Board erweiterbar)
Lim/EMS Funktion auf Systemplatine
Batteriegepufferte Uhr mit Datum
WD FD/HD Kontroller für 2FD und 2HD
1 Floppy LW 1,2MB 5,25"
Parallele Schnittstelle
2 Serielle Schnittstellen
220W Netzteil
Deutsche MF Tastatur 102 Key
Herkules kompatible Graphik-Display-Karte
12" Monochrom-Bildschirm

14,900.-

BAT M22CL

Gleiche Konfiguration wie M20CL
mit 20 MB Festplatte

17,690.-

BAT M24CL

Gleiche Konfiguration wie M20CL
mit 40 MB Festplatte

19,790.-

BAT M28CL

Gleiche Konfiguration wie M20CL
mit 80 MB Festplatte

23,280.-

Aufpreis für Towergehäuse 1,700.-

Aufpreise für Speichererweiterungen on Board
siehe Seite 6

Umrüstung auf andere Bildschirmsets siehe Seite 6

BAT M20SL

Slim Line Gehäuse
200W Netzteil
Freie Laufwerksplätze insgesamt
2x5,25" und 2x3,5"
Gleiche Komponenten wie M20CL

14,800.-

BAT M22SL

Gleiche Konfiguration wie M20SL
mit 20 MB Festplatte

17,590.-

BAT M24SL

Gleiche Konfiguration wie M20SL
mit 40 MB Festplatte

19,690.-

Aufpreis für Speichererweiterungen on Board
siehe Seite 6

Umrüstung auf andere Bildschirmsets siehe Seite 6



BAT XT

HÄNDLERPREISLISTE

SYSTEME
und
Komponenten

Gültig ab 8.05.89

BAT X10CL

AT-Look
8088 CPU 10MHz Taktfrquenz
Sockel für 8087
256 KB Hauptspeicher (erweiterbar auf 640 KB)
1 Floppylaufwerk 360 KB
Parallele Schnittstelle
Deutsche MF-Tastatur 102 Key
Hercules kompatible Graphik-Display-Karte
12" Monochrom-Bildschirm 8,480.-

BAT X11CL

Wie X10CL jedoch mit
2 Floppylaufwerke 360 KB
512 KB Hauptspeicher 10,100.-

BAT X12CL

Wie X10CL jedoch mit
512 KB Hauptspeicher
Serielle Schnittstelle
Batteriegepufferte Uhr mit Datum
Festplattenkontroller für 2 HD
1 Festplatte 20MB 65ms 13,180.-

Aufrüstung von 512KB auf 640KB 990.-

Umrüstung auf andere Bildschirmsets siehe Seite 6

Technische Änderungen, Irrtümer und Druckfehler vorbehalten

Beachten Sie bitte unsere Liefer und Zahlungsbedingungen

ALLE PREISE EXCL.MWST.

Telefax 513 48 60/17

DVR 0477184 • Bankverbindung Z-Kto.-Nr. 685 028 300 Bank Winter & Co. Kto.-Nr. 43 88 005



BAT AT

BAT M20CL

Standart AT Gehäuse
80286 CPU 12 MHz Taktfrequenz 0 Wait
Sockel für 80287
640KB Hauptspeicher (bis 4MB on Board erweiterbar)
Lim/EMS Funktion auf Systemplatine
Batteriegepufferte Uhr mit Datum
WD FD/HD Kontroller für 2FD und 2HD
1 Floppy LW 1,2MB 5,25"
Parallele Schnittstelle
2 Serielle Schnittstellen
220W Netzteil
Deutsche MF Tastatur 102 Key
Herkules kompatible Graphik-Display-Karte
12" Monochrom-Bildschirm 14,900.-

BAT M22CL

Gleiche Konfiguration wie M20CL
mit 20 MB Festplatte 17,690.-

BAT M24CL

Gleiche Konfiguration wie M20CL
mit 40 MB Festplatte 19,790.-

BAT M28CL

Gleiche Konfiguration wie M20CL
mit 80 MB Festplatte 23,280.-

Aufpreis für Towergehäuse 1,700.-

Aufpreise für Speichererweiterungen on Board
siehe Seite 6

Umrüstung auf andere Bildschirmsets siehe Seite 6

BAT M20SL

Slim Line Gehäuse
200W Netzteil
Freie Laufwerksplätze insgesamt
2x5,25" und 2x3,5"
Gleiche Komponenten wie M20CL 14,800.-

BAT M22SL

Gleiche Konfiguration wie M20SL
mit 20 MB Festplatte 17,590.-

BAT M24SL

Gleiche Konfiguration wie M20SL
mit 40 MB Festplatte 19,690.-

Aufpreis für Speichererweiterungen on Board
siehe Seite 6

Umrüstung auf andere Bildschirmsets siehe Seite 6



BAT AT

BAT M20SLW

Slim Line Gehäuse Für Workstation
 200W Netzteil
 Freie Laufwerksplätze insgesamt
 2x3,5" FDD und 1x3,5" HDD
 Gleiche Komponenten wie M20CL jedoch
 statt 5,25" Floppy ist ein 3,5" Floppy
 eingebaut

14,800.-

BAT M22SLW

Gleiche Konfiguration wie M20SLW
 mit 20 MB 3,5" Festplatte

17,550.-

Aufpreis für Speichererweiterungen on Board
 siehe Seite 6

Umrüstung auf andere Bildschirmsets siehe Seite 6

BAT N20CL

Standart AT Gehäuse
 80286 CPU 16 MHz NEAT
 Sockel für 80287
 1 MB Hauptspeicher (bis 8 MB on Board erweiterbar)
 Batteriegepufferte Uhr mit Datum
 220W Netzteil
 WD FD/HD Kontroller für 2 FDD und 2 HDD
 1 Floppy LW 1,2 MB 5,25"
 Parallele Schnittstelle
 2 Serielle Schnittstellen
 Deutsche MF Tastatur 102 Key
 Herkules kompatible Graphik-Display-Karte
 12" Monochrom-Bildschirm

auf Anfrage

BAT N22CL

Gleiche Konfiguration wie N20CL
 mit 20 MB Festplatte

BAT N24CL

Gleiche Konfiguration wie N20CL
 mit 40 MB Festplatte

BAT N28CL

Gleiche Konfiguration wie N20CL
 mit 80 MB Festplatte

Aufpreis für Tower Gehäuse

1,700.-

Aufpreis für Speichererweiterung on Board
 siehe Seite 6

Umrüstung auf andere Bildschirmsets siehe Seite 6



HAUPTSPEICHERERWEITERUNGEN für 286-12 Systeme

640KB auf 1MB	1,160.-
1MB auf 2MB	1,400.-
2MB auf 4MB	4,900.-

HAUPTSPEICHERERWEITERUNGEN für 286-16/386 Systeme

1MB auf 2MB	1,750.-
2MB auf 4MB	5,390.-
4MB auf 8MB	10,750.-

AUFPREISE für BILDSCHIRMSETS

14" Grün	580.-
14" Amber	580.-
14" Paper White	460.-
14" EGA Set 640x350	4,970.-
14" PGA Set 800x600	8,990.-
14" VGA Set	7,890.-

Alle Systeme werden in unserer Werkstatt assembliert und getestet.
 Selbstverständlich werden unsere Systeme auch mit anderen Komponenten, Festplatten, Floppy Drives und Bildschirmsets ausgerüstet.

Für diesbezügliche Fragen steht Ihnen gerne unser Hr. Resch oder Ihr zuständiger Verkäufer zur Verfügung



BAT 386

BAT H30CL	Standart AT Gehäuse 80386 CPU 20 MHz 0 Wait 64 KB Cache Memory 1 MB Hauptspeicher (bis 8 MB on Board erweiterbar) Sockel für 80387 Batteriegepufferte Uhr mit Datum 220W Netzteil WD FD/HD Kontroller für 2 FDD und 2 HDD 1 Floppy LW 1,2 MB 5,25" Parallele Schnittstelle 2 Serielle Schnittstellen Deutsche MF Tastatur 102 Key Herkules kompatible Graphik-Display-Karte 12" Monochrom-Bildschirm	33,160.-
BAT H32CL	Gleiche Konfiguration wie H30CL mit 20 MB Festplatte	35,950.-
BAT H34CL	Gleiche Konfiguration wie H30CL mit 40 MB Festplatte	38,050.-
BAT H38CL	Gleiche Konfiguration wie H30CL mit 80 MB Festplatte	41,540.-
	Aufpreis für Tower Gehäuse	1,700.-
	Aufpreis für Speichererweiterung on Board siehe Seite 6	
	Umrüstung auf andere Bildschirmsets siehe Seite 6	
BAT VH30CL	Gleiche Ausstattung wie H30CL jedoch 80386 Motherboard mit 25 MHz 0 Wait 64 KB Cache Memory	43,160.-
BAT VH32CL	Gleiche Konfiguration wie VH30CL mit 20MB Festplatte	45,950.-
BAT VH34CL	Gleiche Konfiguration wie VH30CL mit 40 MB Festplatte	48,050.-
BAT VH38CL	Gleiche Konfiguration wie VH30CL mit 80 MB Festplatte	51,540.-
	Aufpreis für Tower Gehäuse	1,700.-



KOMPONENTENPREISLISTE

Floppy Kontroller		
	XT für 2 LW	340.-
	AT für 2 LW	690.-
Festplatten Kontroller		
	XT für 2 HD	810.-
	AT für 2 HD	1,630.-
Dual Kontroller		
	WD 1003 MFM für 2 FD und 2 HD	1,850.-
	WD 1003 RLL für 2 FD und 2 HD	2,250.-
	WD 1003 MFM mit 8k Buffer	2,470.-
	NCL 5455-00 MFM 4k Buffer 1:1 Interleave 5 Mbit Übertragungsrate	1,980.-
	NCL 5455-10 MFM 16k Cache Mem 1:1 Interleave 5 Mbit Übertragungsrate	2,200.-
SCSI und ESDI Kontroller auf Anfrage		
Floppy Drives		
	Teac 360KB 5,25"	1,170.-
	Teac 1,2MB 5,25"	1,350.-
	Teac 720KB 3,5"	1,340.-
	Teac 1,44MB 3,5"	1,400.-
Festplatten		
	Seagate ST225 5,25" MFM 20MB 65ms	2,790.-
	Kaloc KL320 3,5" MFM 20MB 50ms	2,990.-
	Seagate ST125 3,5" MFM 20MB 40ms	3,190.-
	Seagate ST251 5,25" MFM 40MB 40ms	4,890.-
	Seagate ST251-1 5,25" MFM 40MB 28ms	5,490.-
	Seagate ST151 3,5" MFM 40MB 40ms	6,350.-
	Seagate ST157 3,5" RLL 49MB 40ms	6,180.-
	Seagate ST277 5,25" RLL 65MB 40ms	6,050.-
	Seagate ST4096 5,25" MFM 80MB 28ms Full-high	8,380.-

SCSI und ESDI Festplatten auf Anfrage

Wir führen auch Maxtor, Imprimis und Micropolis Festplatten



Monitore	
Monitore Monochrom	
12" Grün	1,170.-
14" Grün	1,750.-
14" Amber	1,750.-
14" Paper White	1,630.-
14" Philips VGA Mono	2,050.-
Monitore EGA	
Microware	5,750.-
Philips 9043	4,350.-
Philips 9CM053	4,940.-
Monitore Multisync	
Philips 8CM875	8,170.-
Monitore VGA	
Philips 9CM082	5,980.-
Graphikkarten	
Herkules kompatible Karte	530.-
CGA	
EGA 640x350	2,250.-
PGA 800x600	2,520.-
VGA 8 Bit	3,760.-
VGA 16 BIT	4,220.-
VGA Philips Video Seven	4,150.-
VGA - Set Philips 9CM082 + VGA Karte	9,590.-
Sämtliche EIZO Karten und Monitore auf Anfrage	
I/O Karten	
Serielle Karte XT/AT	340.-
Parallele Karte XT/AT	340.-
XT Multi I/O (ser,par,FDC,Uhr)	680.-
AT Multi I/O (2x ser,par)	580.-

DATAPRINT GesmbH

PC-line

Netto-Preise excl. Mwst.

PCLAT LAPTOP kompatibles Personalcomputer- System 640 KB RAM Speicher, 80286 CPU mit 12 MHz, 1 Stk. 3,5" 1,44 MB Floppy Laufwerk, 40 MB Harddisk, Plasma Display, 1 Stk. Ser. 1 Stk. Par. Schnittstelle, RGB Ausgang, Tastatur mit 84 Tasten, Gewicht 6,4 kg	ö S 38,500.--
PCLXT 8088 kompatibles Personalcomputer- System 640 KB RAM Speicher, CPU Geschwindigkeit umschaltbar zw. 4,77 und 10 MHz, 1 Floppy Laufwerk 360KB, 20 MB Harddisk, Hercules- oder Colorgrafik u. Printerinterface, 150 W Netzteil	ö S 11,150.--
detto jedoch mit 30 MB RLL Harddisk	ö S 12,850.--
PCLXT 8088 MINI Personalcomputer- System 2 Stk. 5,25" und 2 Stk. 3,5" Slim Line Einschübe von außen zugänglich, 5 Stk. Busstecker auf Motherboard sonstige Daten wie oben	ö S 11,600.--
PCLAT 286 kompatibles Personalcomputer- System mit 1 MB RAM Speicher, (bis 4 MB auf Haupt- platine erweiterbar) CPU 8/12 MHz 0 waitstate umschaltbar, 1.2 MB Floppy, Harddiskcontroller, Hercules- oder Colorgrafik, V24 u. par. Printer Interface.	ö S 15,680.--
PCLAT 286 MINI kompatibles Personalcomputer- System 2 Stk. 5,25" und 2 Stk. 3,5" Slim Line Einschübe von außen zugänglich, 5 Stk. Busstecker auf Motherboard sonstige Daten wie oben	ö S 16,130.--
PCLAT 286 NEAT kompatibles Personalcomputer- System technische Daten wie PCLAT 286 jedoch mit CPU 12/16 MHz. 0 waitstate, bis 8 MB erweiterbar 1 MB RAM Erweiterungsmodul SIP	ö S 18,100.-- ö S 4,550.--
PCLAT 386 kompatibles Personalcomputer- System mit 2 MB RAM Speicher, bis 16 MB erweiterbar CPU 16 MHz, 1,2 MB Floppy, Harddiskcontroller, Hercules- oder Colorgrafik, 2 Stk. V24 u. 2 Stk. Parallel Interface.	ö S 40,300.--
Aufpreis für Ausführung im Towergehäuse	ö S 1,900.--
Standard Tastatur mit 84 Tasten	ö S 700.--
Extended Tastatur mit 101 Tasten	ö S 900.--

Seite 2

Optionen:

20 MB Harddisk für XT od. AT 65 msec Zugriffszeit	ö S 3,400.--
40 MB Harddisk für AT 28 msec Zugriffszeit	ö S 8,300.--
71 MB Harddisk für AT 28 msec Zugriffszeit	ö S 9,500.--
105 MB RLL Harddisk für AT 28 msec Zugriffszeit	ö S 10,500.--
80 MB Harddisk für AT 28 msec Zugriffszeit	ö S 11,900.--
je zusätzliches MB RAM Speicher für AT	ö S 4,900.--
40 MB Ministreamer intern	ö S 6,600.--
45/60 MB Streamer intern	ö S 9,250.--
45/60 MB Streamer extern	ö S 10,500.--
80287 Coprozessor 10 MHz	ö S 5,950.--
zusätzliches serielles Interface für AT	ö S 665.--
Floppy Laufwerk 360Kb für XT od. AT	ö S 1,450.--
Floppy Laufwerk 1,2Mb für XT od. AT	ö S 1,990.--
Floppy Laufwerk 3,5 " 720 KB	ö S 1,650.--
Floppy Laufwerk 3,5 " 1,44 MB	ö S 2,250.--
V24, Uhr u. zus. par. Interface für XT	ö S 850.--
Maus System optische Maus, Serieller Anschluß, incl. Netzgerät	ö S 1,750.--
V24 Kabel Buchse auf Stifte (gerade verbunden)	ö S 480.--
Paralleles Printerkabel	ö S 225.--
MS-DOS version 3.2 Eng.	ö S 950.--
MS-DOS version 3.3 Deutsch	ö S 1,550.--
EGA Enhanced Graphics Adapter 16 Modes Auflösung bis 800 x 600 Punkte	ö S 3,185.--
VGA "Video Graphics Array" Karte	ö S 3,920.--

Netzwerkkomponenten

Novell- Netzwerk Software für 8 User	ö S 28,900.--
Novell- Netzwerk Software für 100 User	ö S 57,900.--
Boot ROM	ö S 1,420.--
Netzwerkadapter für PCLAT 286/386	ö S 10,650.--
Ethernet Datenkabel per Meter	ö S 46.--
Ethernet Stecker	ö S 48.--
Ethernet Abschlußwiderstand	ö S 118.--

Philips Monitore:

Monochrom 12 Zoll

BM7913 Flatsquare-Anti-Reflex-Röhre grün	ö S 1,450.--
BM7923 Flatsquare-Anti-Reflex-Röhre bernstein	ö S 1,550.--

Monochrom Monitore 14 Zoll Pro Line

Preise incl. drehbaren Standfuß

7BM713 Flatsquare-Anti-Reflex-Röhre grün	ö S 1,750.--
7BM723 Flatsquare-Anti-Reflex-Röhre bernstein	ö S 1,850.--
7BM743 Flatsquare-Anti-Reflex-Röhre PaperWeiß	ö S 1,950.--

Dreh- und schwenkbarer Standfuß für

Monochrom Monitor BM97nn	ö S 290.--
--------------------------	------------

Philips Farbmonitor 14 Zoll

CM8833 Auflösung: 600 x 285	ö S 4,350.--
-----------------------------	--------------

EGA Monitorkarte

ö S 3,185.--

Philips EGA Monitore 14 Zoll Pro Line

9CM053 Auflösung: 700 Punkte x 350 Zeilen	ö S 5,500.--
9CM073 Auflösung: 800 Punkte x 350 Zeilen	ö S 6,600.--
8CM875 Multisync Monitor 800 Punkte x 560 Zeilen	ö S 8,300.--

VGA Set

VGA Monitorkarte + VGA Monochrom Monitor	ö S 5,800.--
VGA Monitorkarte + 9CM082 VGA Farbmonitor	ö S 9,300.--
VGA Monitorkarte + 9CM875 Multisync Monitor	ö S 12,300.--

VISTA 19" monochrom Monitor Subsystem

Auflösung 1600 x 1280 Punkte Bildwiederholffrequenz 67 Hz incl. Controller, Softwaretreiber für Autocad, GEM, Microsoft Windows und Ventura	ö S 28,900.--
--	---------------

19" Hochauflösender Farbmonitor mit 1024 x 1280 Punkten	ö S 43,000.--
Monitorkarte für 19" Farbmonitor	ö S 35,000.--

Diese Preisliste ist ab 1. Jan. 1989 gültig und setzt alle bisherigen außer Kraft. Irrtümer sowie Preisänderungen vorbehalten.

DATA PRINT

HÄNDLER-PREISLISTE
Matrixdrucker

star

LC-10
LC-10C

Ein neuer in Form und Ausdruck.

Der neue LC-10 ist ein typisches Star-Modell; erstklassig in der Schrift, enorm in den Druckmöglichkeiten, einfach in der Bedienung und mit Extras, die nichts extra kosten. Er wurde für alle die Anwender entwickelt - ob zu Hause, in der Schule oder im Büro - die ein preiswertes Gerät suchen, aber auf Leistung und Komfort nicht verzichten können. Mit den 4 eingebauten Schriftarten (in Schönschrift und Kursiv verfügbar), den über Tastenfeld wählbaren Funktionen, den 144 Zeichen pro Sekunde im EDV- sowie den 36 Zeichen pro Sekunde im Schönschrift-Modus (NLO) ist der LC-10 den meisten 9-Nadeldruckern überlegen, in der Papierhandhabung ist er einsame Spitze. Als erster Drucker seiner Klasse bietet er serienmäßig eine Papier-Park-Funktion, die das Bedrucken von Einzelblättern ermöglicht, ohne daß Zugtraktor und Endlospapier entfernt werden müssen. Der LC-10 verfügt über drei Emulationen und läßt sich mit allen gängigen Computern in Verbindung bringen. Speziell an den Commodore 64 und 128 angepaßt, gibt es den LC-10 als Version LC-10 C.

Technische Daten:

9 Nadel Druckkopf
4 kByte Druckspeicher
8-bit parallel Interface
Commodore seriell Interface
Bidirektionale Druckrichtung
120 Zeichen/Sek. in EDV- Qualität
30 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Grafik- Drucker,
IBM Proprinter II Emulation
Zeichenmatrix 9 x 9 in EDV- Qualität
Zeichenmatrix 18 x 23 in Brief- Qualität
Schriftarten Courier, Sanserif, Orator
Papier-Park-Funktion

Matrixdrucker STAR LC - 10

80 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive:
4 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

öS 2.950.--

Matrixdrucker STAR LC - 10 C

80 Zeichen pro Zeile,
Commodore 64/128
Inklusive:
4 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Papiereinzug
Commodore seriell Schnittstelle

öS 2.950.--

Optionen und Zubehör:

Einzelblatteinzug	öS 960.--
Farbbandkassette	öS 77.--
Paralleles IBM Datenkabel	öS 245.--

Dieses Preisblatt setzt alle bisherigen außer Kraft.
Alle Preise verstehen sich inklusive Mehrwertsteuer.

Weitere Information über dieses Produkt erhalten
Sie bei uns oder einem unserer Fachhändler.

DATAPRINT GesmbH. 3400 Klosterneuburg Agnesstrasse 35 Tel.: 02243/7565

DATA PRINT

HÄNDLER-PREISLISTE
Matrixdrucker

star

LC-10CL
LC-10C CL

In Farbe nochmal so schön.

Der LC-10 Color glänzt durch Farbvielfalt. Bilder, Grafiken und Texte bringt er in 7 verschiedenen Farben auf Endlospapier oder Einzelblätter - in Rot, Grün, Blau, Gelb, Violett, Orange und Schwarz. Der LC-10 Color arbeitet aber auch als ganz normaler Schwarz-/Weiß-Drucker. In diesem Fall wird einfach das Farbband gegen ein normales, schwarzes Band ausgetauscht. Mehr zu bieten, als die meisten 9-Nadeldrucker, hat der LC-10 Color auch bei den Schriften und Zeichenbreiten. Eingebaut sind Courier, Sanserif, Orator1 und Orator 2, die in der Schönschriftqualität in den Breiten Pica, Elite und proportional sowie in kursiv gedruckt werden können. Außerdem gibt es serienmäßig eine Papier-Park-Funktion, die das Bedrucken von Einzelblättern ermöglicht, ohne daß Schubtraktor und Endlospapier entfernt werden müssen. Speziell an den Commodore 64/128 mit DIN-Zeichensatz angepaßt, gibt es dieses Multitalent auch als LC-10 C Color.

Matrixdrucker STAR LC - 10 Color

80 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive:
4 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

öS 3.650.--

Matrixdrucker STAR LC - 10 C Color

80 Zeichen pro Zeile,
Commodore 64/128
Inklusive:
4 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Papiereinzug
Commodore seriell Schnittstelle

öS 3.650.--

Technische Daten:

9 Nadel Druckkopf
4 kByte Druckspeicher
8-bit parallel Interface
Commodore seriell Interface
Bidirektionale Druckrichtung
120 Zeichen/Sek. in EDV- Qualität
30 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Grafik- Drucker,
IBM Proprinter II Emulation
Zeichenmatrix 9 x 9 in EDV- Qualität
Zeichenmatrix 18 x 23 in Brief- Qualität
Schriftarten Courier, Sanserif, Orator
Papier-Park-Funktion

Optionen und Zubehör:

Einzelblatteinzug	öS 960.--
Farbbandkassette (Schwarz)	öS 77.--
Farbbandkassette (4 Färbig)	öS 154.--
Paralleles IBM Datenkabel	öS 245.--

Dieses Preisblatt setzt alle bisherigen außer Kraft.
Alle Preise verstehen sich inklusive Mehrwertsteuer.

Weitere Information über dieses Produkt erhalten
Sie bei uns oder einem unserer Fachhändler.

DATAPRINT GesmbH. 3400 Klosterneuburg Agnesstrasse 35 Tel.: 02243/7565

In Preis/Leistung ein starkes Stück.

Der neue LC 24-10 ist ein typisches STAR-Modell: erstklassig in der Schrift, enorm in den Druckmöglichkeiten, und mit Extras, die nichts extra kosten. Er wurde für alle die Anwender entwickelt (ob zu Hause oder im Büro) die ein preiswertes Gerät Suchen, aber auf Leistung und Komfort nicht verzichten können. Im Papermanagement ist er einsame Spitze. Als erster Drucker seiner Klasse bietet er serienmässig eine Papier-Park-Funktion die das Bedrucken von Einzelblättern ermöglicht, ohne das das Endlospapier entfernt werden muß. Mit den vier eingebauten Schriftarten (in Korrespondenzqualität und Kursiv verfügbar), den über Tastenfeld wählbaren Funktionen, den 170 Zeichen pro Sekunde im EDV sowie den 56 Zeichen pro Sekunde im NLQ-Modus ist der Drucker den meisten 24-Nadeldruckern seiner Preisklasse überlegen.

Eine weitere angenehme Neuerung ist die "Leise-Taste" mit der, der LC24-10 beispielsweise zum Telefonieren leiser gestellt werden kann. Der LC 24-10 wird mit Paralleler Schnittstelle geliefert.

Technische Daten :

24 Nadel Druckkopf
7 kByte Druckspeicher
8-bit parallel Interface
Bidirektionale Druckrichtung
170 Zeichen/Sek. in EDV- Qualität
56 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Proprinter II Emulation
Zeichenmatrix 9 x 9 in EDV- Qualität
Zeichenmatrix 24 x 23 in Brief- Qualität
Schriftarten Courier,
Orator, Prestige, Script
kursiv, Shadow, Outline, proportional,
Papier-Park-Funktion

Weitere Information über dieses Produkt erhalten Sie bei uns oder einem unserer Fachhändler.

Matrixdrucker STAR LC24-10

80 Zeichen pro Zelle,
IBM Proprinter - ESC/Pumschaltbar
Inklusive :
7 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

öS 4.800.--

Optionen und Zubehör :

Einzelblatteinzug	öS 960.--
Farbbandkassette	öS 133.--
Paralleles IBM Datenkabel	öS 245.--
RAM Cartridges	öS 1.120.--
Font Cartridges: Letter Gothic, OCR-B, Bllpp0	öS 980.--

Dieses Preisblatt setzt alle bisherigen außer Kraft.
Alle Preise verstehen sich exklusive Mehrwertsteuer.

Kleiner Preis, große Leistung.

Wer fix ein Listing oder eine große Tabelle ausdrucken will, braucht nur den Draft-Mode des NX-15 anzuwählen. Schon legt dieser mit 120 Zeichen pro Sekunde los. Im NLQ-Mode, auch Schönschreibmodus genannt, bringt es der NX-15 bei einer Zeichenauflösung von 23 x 18 Punkten auf 30 Zeichen pro Sekunde. Damit drucken Sie Briefe ganz schön schnell. Wer ständig die Schriftarten und Papierformate wechseln muß, wird die Vorzüge des übersichtlich gestalteten Bedientableaus zu schätzen wissen. Wer seine Firmenbriefbogen und Formulare bedrucken will, setzt den NX-15 mit Einzelblatt-Magazin ein. Die Papierkassette faßt bis zu 100 Blatt. Ob es sich dabei um Hoch- oder Querformat handelt, ist unerheblich. Genauso flexibel zeigt sich der NX-15 bei Endlospapier. Von 4 bis 15,5 Zoll bedruckt er alles, was ihm in Papierform angeboten wird.

Technische Daten :

9 Nadel Druckkopf
4 kByte Druckspeicher
8-bit parallel Interface
Bidirektionale Druckrichtung
120 Zeichen/Sek. in EDV- Qualität
30 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Grafik- Drucker,
IBM Proprinter Emulation
Zeichenmatrix 9 x 11 in EDV- Qualität
Zeichenmatrix 18 x 23 in Brief- Qualität

Weitere Information über dieses Produkt erhalten Sie bei uns oder einem unserer Fachhändler.

Matrixdrucker STAR NX - 15

136 Zeichen pro Zelle,
IBM - Epson umschaltbar
Inklusive :
4 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

öS 4.810.--

Optionen und Zubehör :

Seriellles RS232 C Interface (current loop)	öS 1.300.--
Paralleler 16 kByte Puffer	öS 1.800.--
Paralleler 512 kByte Puffer	öS 3.850.--
Einzelblatteinzug	öS 2.300.--
Farbbandkassette	öS 154.--
Farbbandnachfüllung	öS 77.--
Paralleles IBM Datenkabel	öS 245.--
Seriellles Datenkabel	öS 483.--

Dieses Preisblatt setzt alle bisherigen außer Kraft.
Alle Preise verstehen sich exklusive Mehrwertsteuer.

Alles drin und alles dran.

Die beiden Stars ND-10 und ND-15 fallen durch viele gute Seiten ins Auge. Neben dem Design sind das vor allem die Schriftqualitäten und die Druckgeschwindigkeiten. Im EDV-Modus machen 180 Zeichen pro Sekunde jede Menge sauberen Druck, im Schönschreib-Modus lassen 45 gestochene scharfe Zeichen pro Sekunde Schriftstücke repräsentativ aussehen. Die Drucker unterscheiden sich nur durch ihre Breite. Der ND-10 verarbeitet Papierbreiten bis zu 25 cm, der ND-15 bis zu 38 cm. Besonderheiten wie Funktionssteuerung über Tastenfeld, Textspeicher, Papiereinzug, Hervorheben von Überschriften, steckbare Schnittstellen, IBM-Emulation und ESC/P-Code sind selbstverständlich. Die Drucker lassen sich mit allen gängigen Computern in Verbindung bringen und zeichnen sich durch hohe Lebensdauer aus.

Matrixdrucker STAR ND - 10

80 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive :
12,6 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

GS 4.950.--

Matrixdrucker STAR ND - 15

136 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive :
12,6 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Papiereinzug
Centronics 8-bit parallel Schnittstelle

GS 6.250.--

Technische Daten :

9 Nadel Druckkopf
12,6 kByte Druckspeicher
8-bit parallel Interface
Bidirektionale Druckrichtung
180 Zeichen/Sek. in EDV- Qualität
45 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Grafik- Drucker,
IBM Proprinter Emulation
Zeichenmatrix 9 x 11 in EDV- Qualität
Zeichenmatrix 18 x 23 in Brief- Qualität

Optionen und Zubehör :

Seriell RS232 C Interface (current loop) GS 1.350.--
Paralleler 16 kByte Puffer GS 1.800.--
Paralleler 512 kByte Puffer GS 3.850.--
Einzelblatteinzug (ND-10) GS 1.100.--
Einzelblatteinzug (ND-15) GS 2.300.--
Farbbandkassette (ND-10) GS 154.--
Farbbandkassette (ND-15) GS 171.--
Farbbandnachfüllung GS 77.--
Paralleles IBM Datenkabel GS 245.--
Seriell Datenkabel GS 483.--

Weitere Information über dieses Produkt erhalten Sie bei uns oder einem unserer Fachhändler.

Dieses Preisblatt setzt alle bisherigen außer Kraft. Alle Preise verstehen sich exklusive Mehrwertsteuer.

Zwei Schnelle für alle Fälle.

In den meisten Fällen ist nicht der teure Spezialist gefragt, sondern der gute Handwerker. Wir haben deshalb zwei Nadeldrucker entwickelt, die was können und trotzdem kostengünstig sind. Der eine heißt NR-10 und ist für alle Drucksachen bis 25 cm Breite gedacht, der andere heißt NR-15 und bewältigt Papiere bis zu 38 cm. Mit 240 Zeichen pro Sekunde machen die beiden jeder EDV einen ordentlichen Druck. Rundschriften, Formulare, Protokolle und Listen sehen nicht nur gut aus, sie sind auch ruckzuck draußen. Ähnliches gilt für die Geschäftspost. Der eingebaute NLQ-Modus schafft 60 saubere Zeichen pro Sekunde. Schubtraktor und Friktionswalze gehören zur Standardausrüstung. Endlospapier und Einzelblätter unterschiedlicher Breiten können damit vorwärts und rückwärts transportiert werden. Der halbautomatische Papiereinzug erleichtert das Einlegen und die Abrißkante macht von der Perforation unabhängig. Den automatischen Einzelblatteinzug mit einem 100 - Blatt Magazin gibt es als Option.

Matrixdrucker STAR NR - 10

80 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive :
12,6 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

GS 5.950.--

Matrixdrucker STAR NR - 15

136 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive :
12,6 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Papiereinzug
Centronics 8-bit parallel Schnittstelle

GS 7.330.--

Technische Daten :

9 Nadel Druckkopf
12,6 kByte Druckspeicher
8-bit parallel Interface
Bidirektionale Druckrichtung
240 Zeichen/Sek. in EDV- Qualität
60 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Grafik- Drucker,
IBM Proprinter Emulation
Zeichenmatrix 9 x 11 in EDV- Qualität
Zeichenmatrix 18 x 23 in Brief- Qualität

Optionen und Zubehör :

Seriell RS232 C Interface (current loop) GS 1.300.--
Paralleler 16 kByte Puffer GS 1.800.--
Paralleler 512 kByte Puffer GS 3.850.--
Einzelblatteinzug (NR-10) GS 1.100.--
Einzelblatteinzug (NR-15) GS 2.300.--
Farbbandkassette (NR-10) GS 154.--
Farbbandkassette (NR-15) GS 171.--
Farbbandnachfüllung GS 77.--
Paralleles IBM Datenkabel GS 245.--
Seriell Datenkabel GS 483.--

Weitere Information über dieses Produkt erhalten Sie bei uns oder einem unserer Fachhändler.

Dieses Preisblatt setzt alle bisherigen außer Kraft. Alle Preise verstehen sich exklusive Mehrwertsteuer.

Schnell, schön und vielseitig.

Der NB24-10/15 ist ein 24 - Nadel Matrixdrucker, der die Vorzüge eines Matrixdruckers und einer Schreibmaschine auf elegante Art und Weise vereint. Im EDV-Mode bringt er es auf 216 saubere Zeichen pro Sekunde, im Korrespondenz-Mode ist er mit 72 Zeichen pro Sekunde schneller als die meisten Typenraddrucker, steht ihnen im Druckbild aber nichts nach. Viele Funktionen lassen sich über das Bedienungsfeld per Fingerdruck steuern, so daß auch Benutzer ohne Programmiererfahrung problemlos mit ihm umgehen können. Neben 4 ASCII- stehen noch 13 Internationale Zeichensätze zur Verfügung. Prestige Font, in neun verschiedenen Schreibdichten sowie Pica, Elite und Proportionalschrift sind eingebaut, ebenso Superscript, Subscript und Italics. Courier, Orator, Letter Gothic, Prestige Italic, Courier Italic, Script können mittels Font-Cartridges (Steckmodule) zusätzlich eingesetzt werden. Für die professionelle Textverarbeitung steht ein automatischer Einzelblatteinzug mit zwei Papiermagazinen, die jeweils 100 Blatt fassen, zur Verfügung.

Technische Daten :

24 Nadel Druckkopf
8 kByte Druckspeicher
8-bit parallel Interface
Bidirektionale Druckrichtung
216 Zeichen/Sek. in EDV- Qualität
72 Zeichen/Sek. in Brief- Qualität
ESC/P, IBM Grafik- Drucker,
IBM Proprinter Emulation
Zeichenmatrix 24 x 9 in EDV- Qualität
Zeichenmatrix 24 x 31 in Brief- Qualität

Matrixdrucker STAR NB 24 - 10

80 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive :
8 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Einzelblatteinzug
Centronics 8-bit parallel Schnittstelle

öS 6.435.--

Matrixdrucker STAR NB 24 - 15

136 Zeichen pro Zeile,
IBM - Epson umschaltbar
Inklusive :
8 kByte Druckspeicher
Endlospapier Schubtraktor
Halbautomatischer Papiereinzug
Centronics 8-bit parallel Schnittstelle

öS 9.250.--

Optionen und Zubehör :

Seriellles RS232 C Interface (current loop) öS 1.350.--
Paralleler 16 kByte Puffer öS 1.800.--
Paralleler 512 kByte Puffer öS 3.850.--
Font Cartridges :
Courier, Courier Italic,
Prestige Italic, Orator
Letter Gothic, Script Je öS 760.--
Einzelblatteinzug (NB24-10) öS 1.100.--
Einzelblatteinzug Einschacht (NB24-15) öS 3.300.--
Nachrüstung auf Doppelschacht (NB24-15) öS 2.450.--
Farbbandkassette (NB24-10) öS 154.--
Farbbandkassette (NB24-15) öS 171.--
Farbbandnachfüllung öS 91.--
Paralleles IBM Datenkabel öS 245.--
Seriellles Datenkabel öS 483.--

Dieses Preisblatt setzt alle bisherigen außer Kraft.
Alle Preise verstehen sich exklusive Mehrwertsteuer.

Weitere Information über dieses Produkt erhalten
Sie bei uns oder einem unserer Fachhändler.

DATAPRINT GesmbH. 3400 Klosterneuburg Agnesstrasse 35 Tel.: 02243/7565

Drucker ohne Fehl und Nadel.

Die Vorteile der Laserdruck-Technologie wie Satzqualität, Schnelligkeit, Geräuscharmut und Gestaltungsmöglichkeiten sind unbestritten und für viele Computer-Anwender Motivation sich nach einem geeigneten Laser-Drucker umzuschauen. Drei Dinge sollten kaufentscheidend sein : ausgereifte Technik, hoher Bedienungskomfort und niedrige Gesamtkosten. Im STAR Laserprinter 8 finden Sie neben den Vorteilen der Laserdruck-Technologie auch diese Punkte optimal verwirklicht. Er überzeugt durch Perfektion und Wirtschaftlichkeit, aber auch durch Können. Von Haus aus mit drei Standard-Schnittstellen und vier Emulationen ausgestattet, läßt er sich an alle gängigen Computer anschließen und beherrscht die Steuercodes der meisten Laser-, Typenrad- und Matrixdrucker. Vorhandene Programme können weiter verwendet und zahlreiche Softwareangebote genutzt werden. Vorteilhaft ist auch die hohe Speicherkapazität von 1 MByte RAM. In Verbindung mit Download haben alle notwendigen Schriftarten und weitere Bitimage Data für Grafiken Platz ohne daß es zu Speicherproblemen kommt.

Technische Daten :

Laser Beam Druckmethode
300 x 300 Punkte pro Zoll
8-bit parallel Interface
Seriellles RS-232C und RS-422 Interface
8 Seiten pro Minute
Epson EX-800, Diablo, IBM Proprinter,
HP Laser Jet Plus Emulation
Schrifttypen Courier, Prestige Elite
TMS Roman und Line Printer
1 MByte Druckspeicher
A4 Papierkassette mit 200 Blatt
Hoch und Querformat zu bedrucken
Tonerkassette für ca. 4000 Blatt

Weitere Information über dieses Produkt erhalten
Sie bei uns oder einem unserer Fachhändler.

STAR Laserprinter 8

300 x 300 Punkte pro Zoll
4 Emulationen, 4 Schrifttypen
Inklusive :
1 MByte Druckspeicher
Papierkassette A4 für 200 Blatt
Parallel und Serielle Interface
Tonerkassette

öS 24.800.--

Optionen und Zubehör :

1 MByte Speichererweiterung öS 4.340.--
Tonerkassette öS 1.365.--
Seriellles Datenkabel öS 483.--
Paralleles IBM Datenkabel öS 245.--
Font Cartridges:
Courier (FC1), Helvet (FC2),
Helvet/Tms Roman (FC3),
Tms/Roman (FC4),
Prestige Elite (FC5),
Presentation (FC6),
Letter Gothic/Bar Code/OCR-A/B (FC7) öS 1.750.--

Dieses Preisblatt setzt alle bisherigen außer Kraft.
Alle Preise verstehen sich exklusive Mehrwertsteuer.

DATAPRINT GesmbH. 3400 Klosterneuburg Agnesstrasse 35 Tel.: 02243/7565

10.06.89

P R E I S L I S T E G R O S S H A N D E L Nr. 1/89
gültig ab 1.5.89
CLUBBRABATT -10% BEREITS BERÜCKSICHTIGT
ALLE PREISE INCL. 20% MEHRWERTSTEUER

Best.Nr.	Artikel	Preis incl. MWSt
** Microcomputer IBM-compatibel		
0402	Microcomp. Profi-16-AT 512k	24645.60
0403	Microcomp. Profi-32-AT 20MHz	45360.00
0404	Microcomputer LapTop-AT-286	46105.20
0409	Portable XT	21581.64
0410	Portable AT, 20MB, 640x400	33458.40
0411	Portable 386 mit Harddisk 20MB	53978.40
0499	Microcomp. XT Economy 128k	8629.20
0500	Microcomp. Profi-16 XT / 1FDD	11858.40
** Display-Karten		
0503	Graphikkarte Hercules 6.0	1293.84
0512	Farbgraphikkarte für IBM	969.84
0517	SEGA-Karte (800X600)	3758.40
0575	Hercules- und Farbgrafikkarte	1728.00
0588	EVGA-Karte	5378.40
0608	EGA-Aufrüstg. 800x600 Portable	3888.00
** Speicher-Karten		
0522	2 MB-Karte mit 0k für XT	1922.40
0523	2 MB-Karte mit 0K für AT	1922.40
0526	2MB-Multif.Karte m. 0k f. AT	2149.20
0578	8MB-Karte mit 0MB für 386	2484.00
0597	Speichererw. für Inboard 1MB	14569.20
** Interface-Karten		
0504	Multi-I/O-Karte für IBM	1404.00
0520	AD/DA Karte	2041.20
0524	I/O-Plus-Karte für XT	1010.88
0525	IEEE-488 Interface für IBM	4320.00
0537	Multi-Serielle Karte 4-fach	2527.20
0538	Multi-Serielle Karte 8-fach	4298.40
0542	Experimentierkarte	397.44
0551	Parallel-I/O-Karte 8255	1401.84
0572	Druckerkarte Centronics	540.00
0573	Seriell-/Parallel-Karte für AT	1077.84
0576	Serielle Karte für XT/AT	540.00
0592	Inboard 386/PC mit 1MB RAM	18338.40
0601	Game-Port für XT und AT	321.84
** Floppylaufwerke und -Zubehör		
0501	Floppy-Disk-Laufwerk 360k	1944.00
0516	Floppy-Contr. für XT 360k/720k	830.52
0527	Floppy-Disk-Laufwerk 1,2MB	2905.20
0536	Floppy-Contr.1,2MB/1,44MB/720k	961.20
0583	Floppy-Laufwerk 3 1/2", 720k	2373.84
0584	Floppy-Laufwerk 3 1/2", 1,44MB	3229.20
0607	Floppy Laufw. ext. f. LapTop	5378.40

** Harddisks und -Zubehör		
0533	Harddisk-Controller	1725.84
0534	Harddisk 20 MB	4860.00
0535	Harddisk 40 MB/40ms	8100.00
0571	Harddisk 80MB/25ms	12960.00
0574	Harddisk-/Floppycontroller AT	3229.20
** Streamer und -Zubehör		
0544	Streamer 20 MB für XT oder AT	8618.40
0545	Streamer 40 MB für AT	10778.40
** Nadeldrucker und -Zubehör		
3101	Drucker SP-180AI	3591.00
3102	Drucker SL-80IP(24 Nadeln)	5751.00
3103	Drucker SP-1600AI (für IBM)	4042.44
3104	A3-Drucker SL-130AI(24-Nadeln)	16092.00
3105	Drucker NEC Pinwriter P6+	15109.20
3106	Drucker NEC Pinwriter P7+	18338.40
3107	Einzelblatteinzug für P6/P7	5378.40
3110	Farbband für Drucker SP1600AI	162.00
3111	Farbband für Drucker P6/P7	270.00
3115	Einzelblatteinzug SP1600/SL80	2268.00
** Laserdrucker und -Zubehör		
3100	Laserdrucker L-2060	35640.00
3112	Tonerbehälter für L-2060	432.00
3113	Fixierer-Einheit f. 10.000 Bl.	1712.88
3114	Trommel+Fixierer f. 20.000 Bl.	3425.76
3117	Schriftkassette für L-2060	3002.40
** Eingabemedien		
0513	Strichcode-Leser für IBM	4570.56
0528	Lightpen für IBM	1922.40
0530	Maus seriell Logitech für IBM	2149.20
0531	Grafiktablett	10000.80
0549	Joy-Stick	321.84
0552	Maus seriell GM-6-Plus+Dr.Halo	861.84
0567	Akustikkoppler"dataphon" 300Bd	2999.16
0581	Handy-Scanner	4741.20
0589	Scanner SS-300(+)	23760.00
** Kabel		
3010	Monitorkabel schwarz/weiß	43.20
3011	Monitorkabel RGB	554.58
3012	Druckerkabel für IBM parallel	172.80
3013	Seriellles Kabel für IBM 1,8m	172.80
3014	Kabel Centronix für Dataswitch	237.60
3015	Seriellles Kabel M/M	172.80
3016	Serieller Adapter AT-XT	194.40
3018	Seriellles Kabel für IBM 6m	432.00
3019	Drucker-Verlängerungskab. 1,8m	237.60
3020	Tastatur-Verlängerungskabel	172.80
3021	Monitor-Verlängerungskabel	172.80
3022	Seriellles Kabel 1,8m F/F	172.80
** Monitore		
3001	Monitor Philips 14"	2484.00
3003	Monitor NEC Multisync XL 20"	48600.00
3004	Monitor Philips 12", BAS	1512.00
3005	Monitor Philips Color CM 8852	5378.40
3006	Monitor Philips 12" BM7913/23	1944.00
3007	EGA-Monitor NEC Multisync II	13320.00
3008	EGA-Monitor Philips Autosync	11124.00
3009	EGA-Monitor Philips (dualsync)	7415.28
3018	Monitor Philips VGA 3CM9809	8343.00

** Zubehör		
0502	Motherboard Profi-16XT 0k	2808.00
0505	MF-Tastatur (101 Tasten)	1317.60
0511	Netzteil XT, AT, 386	1425.60
0540	Data-Switch parallel	466.56
0541	Data-Switch seriell	466.56
0548	Schwenkfuß für Monitor	315.36
0554	Unterbrechungsfreie Stromvers.	8618.40
0556	RS-232-Mini-Tester	194.40
0557	Jumper-Box RS-232	194.40
0558	Seriell-Parallel-Umsetzer	1077.84
0559	Parallel-Seriell-Umsetzer	1077.84
0565	Dataswitch autom. 4-f. parall.	1674.00
0566	Diskettenbox für 100 Disks	162.00
0566	Dataswitch autom. 4-f. seriell	2041.20
0570	Motherboard Profi-16-AT 0k	5281.20
0586	Serieller Zwischenstecker F/F	172.80
0587	Nullmodem-Stecker M/F	172.80
0591	Werkzeugsatz	1728.00
0599	Dataswitch 512k mit 256k	5832.00
0603	Akku für LapTop 286	1404.00
0609	Neatboard für AT mit 20MHz	12960.00
0610	Towergehäuse	3780.00

** Sonderkarten		
0532	Copy-Card für 360k-Laufwerke	1077.84
0539	(E)EPROM-Programmer bis 1MBit	4298.40
0553	IC-Tester-Karte m. ext. Sockel	2805.84
0555	PAL-Programmer für IBM	5559.84
0580	Löschgerät für EPROMS	2743.20
0598	Modem 300/1200 Baud	3218.40
0600	Copycard für alle Laufwerke	3758.40
0602	EPROM-Programmer bis 512kBit	2678.40
0605	Video-Digitizer VGA Color	21578.40
0606	Universal-Programmer+Tester	9687.60

** Coprozessoren, Speicher-IC's		
0509	8087 CO-Prozessor für 10MHz	3553.20
0510	Material für 2. RS-232	378.00
0515	80287 Co-Prozessor für 12 MHz	4298.40
0546	Speicher-IC 64kBit = 8kByte	59.40
0547	Speicher-IC 256kBit =32 kByte	172.80
0577	80387 CO-Prozessor 20 MHz	8496.00
0590	Speicher-IC 1MBit	313.20

** Disketten, Papier etc.		
2001	Diskette DS/DD Redstone	12.00
2002	Diskette DS/HD "sunnyline"	34.56
2003	Diskette 3,5" DS/DD für 720k	37.80
2004	Diskette 3,5" DS/HD für 1,44MB	96.12
2500	1000 Blatt Druckerpapier	172.80

** Netzwerk und -Zubehör		
0560	Netzwerkkarte dLINK 2-draht	3499.20
0561	Ethernetkabel für dLINK 1m	27.00
0561	Programm für dLINK	1188.00
0562	Bus-Repeater für dLINK	2668.68
0563	Boot-ROM für dLINK	240.00
0564	LAN-Smart-Programm für dLINK	4212.00
0579	dLINK-ETHERNET-Karte	5162.40
0593	dLINK-ETHERNET für PS-2	7322.40

** Software

4001	MS-DOS 3.3 m. deutschem Manual	2138.40
4002	Betriebssystem MOS386 (5 User)	14256.00
4003	Texterkennungs-Programm SS-300	11880.00
4004	Business Total	32378.40
4005	Business Total Netzwerkversion	43659.00
4006	Business Total Demoverision	1080.00
4007	Business Total Schulung 2 Tage	7560.00
4009	MS-DOS 4.01	3024.00

in allerletzter Minute ... in allerletzter Minute ... in allerletzter Minute

*Scriptum TURBO PASCAL Version 3./4 in Buchform
Anweisungssammlung mit Beispielen
von Peter PFENICHER
um ö⁸ 100.- bei Abholung im Clubsekretariat*

in letzter minute..in letzter minute..in letzter minute..in letzter

Herr Hönninger sucht eine Festplatte 20-40MB (auch gebraucht), wenn möglich mit Controller.

Adresse: Herr Hönninger
Schloßgegend 29
3204 Kirchberg/Pielach

Gesucht: PC-XT mit Drucker und Monitor.
Ankaufsumme öS 10000.- (auch Einzelteile)
Angebote bitte an den PCC-TGM Tel.:35-23-983

in letzter minute..in letzter minute..in letzter minute..in letzter

CA, die Bank zum Erfolg.



Teenager zum Erfolg. Wenn Sie wissen wollen, wie sich die kleinen und größeren Freuden des Lebens verwirklichen lassen, sehen Sie einfach in Ihrem Sparsbuch Junge CA nach. Dort lesen Sie von Ihrem Erfolg, wie er im Buche steht.



CREDITANSTALT

Absender:

P C C - T G M
Wexstraße 21
Postfach 59
1202 W I E N

P.b.b.
Verlagspostamt
1200 WIEN

Num: 77 Lfd:207

Dipl.-Ing. Franz FIALA
Siccardsburggasse 4/1/22
1100 Wien

An den Zusteller:
Wenn unzustellbar, zurück an den Absender

An den Empfänger:
Bei Doppelsendungen, zurück an den Absender

VORSICHT



HOCHSPANNUNG Computer
Hard und Software

Verkauf : 1040 Wien Lambrechtg. 16

Tel. 56-52-40

Zentrale: 1040 Wien Große Neug. 32

PCC-TGM Preisliste gültig ab 6.4.1989

Alle Preise inkl.MWSt., netto Kassa

Tornado PC 4000

CPU 8088-2 (4.77/8 MHz),
512 KB Ram, ein 5.25" Laufwerk 360 kB,
eine Harddisk 20 MB, Batteriegepufferte
Echtzeituhr, parallele und serielle
Schnittstelle, Gameport, Grafikkarte von CGA
auf Hercules umschaltbar

Art.Nr. 4304 13.474.-

Tornado PC 5000

wie Tornado PC 4000, aber 40 MB Harddisk

Art.Nr. 4305 16.374.-

Tornado AT Turbo 286

CPU 80286 (8/12 MHz), 0-Wait-States, Sockel
für 80287, batteriegepufferte Echtzeituhr,
640 kB Ram (erweiterbar auf 4 MB on Board),
EMS-Funktion, ein 5.25"-1.2 MB Diskettenlaufwerk,
Multi I/O Karte mit serieller und paralleler
Schnittstelle (aufrüstbar auf zweite serielle),
Kombinierter Harddisk- und Floppycontroller,
erweiterte AT-Tastatur mit 102 Tasten

mit CGA Karte Art.Nr. 4307 17.209.-

mit MGP Karte Art.Nr. 4306 17.209.-

Aufpreis für EGA-Version (640x480) 2.190.-

Aufpreis für PGA-Version (800x600) 3.190.-

Aufpreis für VGA-Version (920x480) 4.190.-

Aufpreis für Tower-Gehäuse 3.690.-

Festplatten-Diskettenlaufwerke

YE-DATA 5.25", 360 KB Laufwerk	1431.-
YE-DATA 5.25", 1.2 MB Laufwerk	2331.-
TEAC 3.5", 720 KB Laufwerk	2511.-
TEAC 3.5", 1.44 MB Laufwerk	2241.-
Einbaukit für 3.5" Laufwerke	490.-

Seagate Festplatten

ST-125, 3.5", 20 MB, ca.40 ms	4590.-
ST-225, 5.25", 20 MB, ca.65 ms	3590.-
ST-251, 5.25", 40 MB, ca.40 ms	5990.-
ST-251/1, 5.25", 40 MB, ca.28 ms	6990.-
ST-4096, 5.25", 80 MB, ca. 28 ms	10.990.-
Einbaukit für 3.5" Festplatten	490.-

Div. Erweiterungskarten

XT-Game (Joystick) Karte	441.-
XT-Printer Karte	441.-
XT-RS 232 Karte	531.-
XT-Uhren Karte	441.-
XT-Multi I/O Plus II Karte	1071.-
AT-Multi I/O Karte	891.-
XT-Harddiskcontroller	1071.-
AT-Harddisk- und Floppycontroller	2691.-
Modem Karte "Smart Link 1200"	2691.-

Drucker je nach Fabrikat	abzüglich	3 - 10 %
Monitore	abzüglich	10 %
Erweiterungen und Zubehör	abzüglich	10 %
Bücher Markt & Technik Verlag	abzüglich	20 %
alle anderen Verlage	abzüglich	10 %
Software	abzüglich	10 %

Gilt nicht für Sonderangebote !