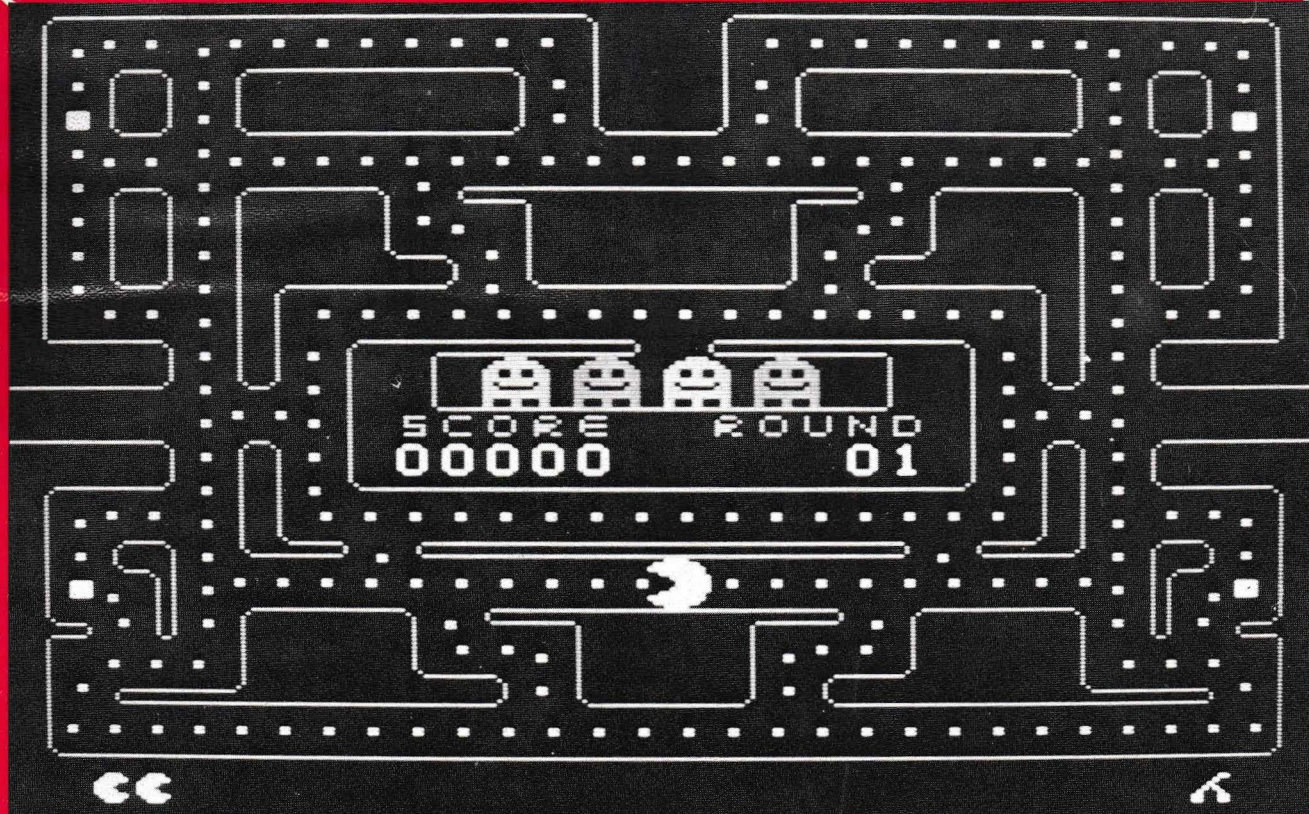




5/1984

öS 70,-  
DM 11,-  
sfr 11,-

# -99 JOURNAL



## SOFTWARE-ÜBERSICHT:

Neue Superspiele in Maschinensprache  
TRICKS IN EXTENDED BASIC  
SPACE-KILLER-Arcadespiel  
SPRITES FÜR MINIMEM

**Disketten  
kopiergeschützt**

[www.ti99inc.it](http://www.ti99inc.it)





## Editorial

Nachdem Sie nun doch schon eine Weile nichts von uns gehört haben, möchte ich mit ein paar Worten die Verzögerung der nun vorliegenden Nr.5 erklären.

Nachdem TEXAS INSTRUMENTS die Produktion des TI99 eingestellt hat, war am Markt, wider Erwarten, ein spürbarer Trend in Richtung Ausbau der bestehenden 99er-Anlagen, bzw. sogar Neukauf der eingestellten Konsole feststellbar. In kurzer Zeit war keine Peripherie mehr erhältlich, ebenso gab es nicht einmal mehr die fast schon zur Grundausstattung gehörenden EXTENDED BASIC Module. So ergab sich (soweit mir erinnerlich) erstmals in der Elektronik-Branche das Phänomen, daß gebrauchte Teile des TI99 weit über dem Neupreis gehandelt wurden. Durch diesen Kaufboom stieg natürlich auch die Nachfrage nach Literatur und Programmen explosiv an. Nachdem das TI99-JOURNAL das einzige dieser Art im deutschen Sprachraum ist, lief so ziemlich alles bei uns zusammen. Man wollte Händler nachweise, Module, Schaltpläne und andere Informationen. Unsere Redaktion war mit einem Schlag zu einem 99er-Infocenter geworden. Doch damit hatte keiner gerechnet und so waren wir innerhalb kürzester Zeit unter einem Berg von Zuschriften und Anfragen nahe dem Ersticken. Unser Klub (Klubmitglied wird man automatisch ohne Mehrkosten durch ein "Jahresabo!") verdoppelte sich in nur zwei Monaten. Es fanden sich Mitglieder aus der Türkei, Griechenland, Luxemburg, Niederlande, Finnland, Jugoslawien, Ungarn, der BRD und natürlich aus Österreich.

Um dieser enormen Nachfrage gerecht zu werden, brachten wir eine **Anleitung für das MINIMEN-Modul**, eine **Programmsammlung für BASIC** und ein für **ASSEMBLER** heraus und "fütterten" außerdem ein CHIP-Spezial mit Programmen. Durch die anhaltende Nachfrage nach Peripherie fanden sich bald einige Firmen, die kompatible Erweiterungen anboten. Das TI99-Journal versuchte diese Drittanbieter zu lokalisieren und sie gesammelt in dieser Ausgabe des Journals als Bezugsquellennachweis in einer großen Übersicht anzuführen. Doch es liefen und laufen fast täglich neue Informationen ein, wodurch sich diese Übersicht, trotz der aus diesen Grund bedingten Verzögerung der Nr.5, erst im TI99-Journal Nr.6 finden wird. Die **Journale Nr.6 und Nr.7** werden deshalb in ca. 6-wöchigen Abständen nach der Nr.5 erscheinen. Weiters erscheinen weitere Prgr.sammlungen in **BASIC** und **ASSEMBLER**.

Weiters werden wir die Gründung von regionalen Klubs tatkräftig unterstützen. Dies soll uns entlasten und auch regional zur

Verbesserung des Info.austausches beitragen.

**Dringende Anfragen** richten Sie bitte **telefonisch** an uns, da wir kaum mehr mit dem Beantworten der Leserbriefe nachkommen.

Ihre **Post** richten Sie bitte an:  
Thomas M. Fiedler  
TI99-JOURNAL-KLUB  
Felberstraße 24/26  
A-1150 WIEN

Abschließend möchte ich Sie im Namen der Redaktion nochmals um Verständnis für die, für uns neue Situation bitten, und mich für Ihre Geduld bedanken.

(Der Herausgeber)

**RICHTIGSTELLUNG** (lt. Mediengesetz)  
Im TI99-Journal Nr.4 wurde durch eine Fehlinformation behauptet, TI hätte auch die Produktion von Taschenrechnern, sowie deren Neuentwicklung eingestellt. Diese Information ist unrichtig.

## Impressum

Medienverleger und -herausgeber:

Thomas M. Fiedler  
Felberstraße 24/26  
A-1150 WIEN  
Tel.: 0222/92 29 033  
(10-13, 19-20 Uhr)

Layout:

Werbe Werkstatt Wiesmüller

Hersteller:

Stockinger & Morsack  
Buch- und Offsetdruckerei  
1070 Wien, Kirchbergg. 33-35  
Tel.: 0222/93 23 16

Redaktion:

Alexander Berger  
Friedrich Brezina  
Thomas Fiedler  
Mag. Karl Hagenbuchner  
Heiner Martin  
Isabella Roch  
Christian Roch  
Dipl.-Ing. Dr. Peschetz  
Herwig Reidlinger  
Erich Reitingier  
Dipl.-Ing. Michael Weiland

Alle: 1150 Wien, Felberstr. 24/26

Verlags- und Herstellungsort:  
WIEN

Offenlegung § 25:

Medienverleger Thomas M. Fiedler,  
A-1150 WIEN, Felberstraße 24/26,  
zu 100 %.

Die Richtung der Zeitung ergibt sich aus den Meinungen der Redakteure und ist sachlich informativ.

Anzeigentarif:

Für die Anzeigenpreise gilt derzeit die Anzeigenpreisliste 1/84.

Für Kleinanzeigen gelten folgende

Sätze:

Private : 6S 30,-/Zeile inkl.  
aller Abgaben

Gewerbl.: 6S 46,-/Zeile exkl.  
aller Abgaben

Eine Zeile besteht aus 32 Zeichen  
einschl. Satzzeichen und  
Zwischenräumen).

Chiffregebühr: 6S 30,- exkl. MWST

Erscheinungsweise:

5 Journale/Jahr

Bezugspreis:

Klubmitgliedschaft inkl.

Jahresabo:

Inland: 6S 420,- (inkl. Versand)

Ausland: BRD 6S 462,-

Schweiz " 462,-

sonstige Länder auf Anfrage

(Es werden nur **spesenfreie**

Zahlungen berücksichtigt!)

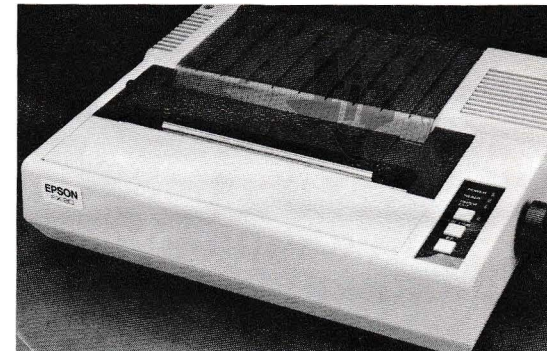
Bankverbindung:

PSK-Wien 6.333.544

Das TI99-JOURNAL wird mit dem komfortablen Matrixdrucker/Plotter mit über 50 Schriftbildern, dem

**EPSON FX-80**

gedruckt. Der EPSON FX-80, ein Drucker der mittleren Preisklasse, vereint Funktionalität und Design zu einer harmonischen Einheit.



Das TI99-JOURNAL wird mit dem erstklassigen Textsystem von TI, dem **TI-WRITER-MODUL** erstellt. Dieses Software-Paket macht aus dem TI99 eine Textverarbeitungsanlage, die jedem Vergleich standhält. Informationen bietet Ihnen der Fachhandel.

## Inhalt

- 5 **ASSEMBLER:** Das Leben ist ein Spiel
- 13 **PREIBRATSEL** - Software zu gewinnen
- 14 **DISKETTENVERWALTUNG** (+ 32K-RAM-Card)
- 15 **EXTENDED BASIC** noch besser (Teil 1)
- 16 **SUPERTIP:** TIPS f. Peripheriebesitzer
- 17 **SPACE-KILLER:** "ARCADE-Prgr." - BASIC
- 19 **HARDWARE:** Rekorderkabel im Selbstbau
- 20 **HARDWARE:** PLATINE f. V24-JOYSTICKPORT
- 20 **SPRITES** IN BASIC - für MINIMEN-Modul
- 21 **Der PRAXIERTE TIPS:** TI-Joystick-Umbau
- 21 **L E S E R E C K E** - Tips u. Anfragen
- 23 **BASIC-KURS** (letzter Teil)
- 24 **DISKETTEN-SCHUTZ** - DER SUPERTIP
- 24 **H I N W E I S E** -
- 25 **VIDEOTHEK:** Videoverwaltung in XBASIC
- 27 **NEUE LITERATUR** - 99-JOURNAL-SPEZIALS
- 29 **PRAXENTATION** von ASSEMBLER SOFTWARE
- 30 **FORTH AM TI99** - 1. Erfahrungsbericht
- 31 **NEUE HARDWARE:** 128K-Karte aus USA

# Multiplan

```

M*M
U***U
L*****L
T*****T
I****M****I
P*****U*****P
L*****L*****L
A*****T*****A
N**** MULTIPLAN ****N
A*****P*****A
L*****L*****L
P*****A*****P
I****N****I
T*****T
L*****L
U***U
M*M

```

## TI 99 Journal Software-Test

Wenn wir uns diesmal das **Multiplan**-Modul näher ansehen, dann genügt es keineswegs nur TI99/4A-spezifische Besonderheiten und Ausprägungen dieses Softwarepakets zu beschreiben und zu erklären. Bei einer Standardsoftware dieser Art muß man doch etwas weiter ausholen, um nicht ein zu vereinfachtes, und damit unvollständiges Bild zu schaffen. Was heißt eigentlich Standardsoftware?

Von allen möglichen Definitionen, die in diversen Lexika zu finden sind, scheint mir doch jene Auffassung am leichtesten verständlich, wonach man dann von Standardsoftware spricht, wenn allgemein zu bewältigende Arbeiten, Entscheidungen oder Prozesse von gleichartigen (standardisierten) computergesteuerten Methoden (Vorgängen) unterstützt oder sogar ersetzt werden. Am bekanntesten sind wohl eindeutig die verschiedenen Textverarbeitungen (wie z.B. Wordstar oder TI-Writer). Doch nicht weniger wichtig - wenn auch nur in der geschäftlichen Anwendung weit verbreitet - sind die vielen Datenanalyseprogramme zur Planung und Entscheidungshilfe für zahllose Problemkreise in jedem Betrieb. Sei es Kostenrechnung oder Budgetierung, Inventur oder Break-Even Analyse, überall hat man das Problem, enorme Datenmengen zu verarbeiten, auszuwerten und eventuelle Veränderungen rechtzeitig zu erkennen. Innerhalb dieser Gruppe hat sich in den letzten 10 Jahren eine Art riesige Matrix als Darstellungsform sehr bewährt. Dazu gehört unser Multiplan (aber auch andere bekannte Standardsoftwarepakete wie Visicalc u.v.m.). Ein großer Vorteil liegt dabei im Modellcharakter derartiger "Programmsysteme". Mögliche

Veränderungen lassen sich bei gegebener mathematischer oder logischer Beziehung simulieren, und damit sichtbar machen. Doch davon später. Wir haben es also mit einem elektronischen Arbeitsblatt zu tun auf dem jede noch so komplexe Kalkulation, Berechnung oder schematische Zahlenkette eingetragen werden kann. Nun gibt es Befehle mit denen man Beziehungen zwischen einzelnen Zellen (= Einheiten) auf dem Blatt herstellen kann. Verschiedene Kosten werden zum Beispiel addiert, und von der Summe wird der Mehrwertsteuersatz berechnet. Nehmen wir dies als Ausgangspunkt an. Alle Kosten zusammen führen immer zu einer bestimmten Summe bzw. MWST. Verändere ich nun in unserem Modell einen Kostenbetrag, dann rechnet Multiplan automatisch - nach unserer Definition von vorher - die neue Endsumme und den neuen Steuersatz aus. Das Modell bilden also die Platzhalter für die Kosten und eine frei wählbare Formel (z.B.: Summe). Nun kann der Anwender die Werte variieren, und erhält immer das verlangte Ergebnis. Bereits dieses kleine Beispiel zeigt die riesigen Vorteile solcher Modelle. Stellen wir uns ein kompliziertes Ablaufschema eines Industriebetriebes vor, bei dem am Ende der erwartete Ertrag steht. Hypothetisch kann man nun bestimmte Faktoren ändern, und die Auswirkungen beim Ergebnis ablesen. Selbst sehr komplexe Beziehungen werden damit transparent. Soweit der kurze Ausflug in die Theorie derartiger Planungsprogramme. Das Multiplan Modul selbst stammt nicht von Texas Instruments, sondern wie branchenüblich bei vielen Softwarepaketen von einer Drittfirma, die ihr Programm für die unterschiedlichsten Computer kompatibel macht. Im besonderen Fall ist es die Firma Microsoft, die Multiplan mit geschützter Bezeichnung vertreibt. Das Paket beinhaltet wie der TI-Writer ein Modul, eine Diskette mit dem Betriebsprogramm, ein Handbuch in englischer Sprache, sowie eine Einschubleiste für die geänderten FC/N- und CTRL-Befehle und eine blaßorange Quick Reference Card. Die Mindestkonfiguration zum Betrieb von Multiplan ist somit: Konsole + RAM Exp. + mind. ein Diskettenlaufwerk (besser sind aber zwei) (...+ Schnittstelle und Drucker, die zwar nicht notwendig aber unersetzbar sind!). Nachdem sich der vorsichtige User von der Diskette eine Backupcopy gemacht hat, startet man das System am Besten nach der "Trial & Error" Methode und schaut sich dieses elektronische Arbeitsblatt einmal an. Der erste Eindruck ist verwirrend, zumal der altgewohnte Cursor zu einem Rechteck mutiert. Oben am Bildschirm stehen in weißer Schrift auf blauem Grund die Spaltennummern 1,2,3,4... und links am Rand die Zeilen. "Aha", und sogleich denkt man an eine Matrix. Von dieser Matrix sieht man natürlich nur einen

Ausschnitt (Fenster), doch wenn man mit den Pfeiltasten den "Cinemascope-Cursor" nach rechts bewegt dann läuft der Spaltenindex mit. Man kann sich also innerhalb dieses Feldes frei bewegen. Doch noch ist fast der ganze Bildschirm leer!

Nur ganz unten stehen fünf unbewegliche Zeilen, die mit **COMMAND** betitelt sind. Wie beim TI-Writer oben, so sind hier unten die möglichen Befehle angeführt. Sie werden ganz einfach dadurch eingeleitet, indem man den ersten Buchstaben drückt, oder ENTER bei dem jeweiligen Befehlswort auf dem der zweite Cursor sitzt. Diesen Cursor bewegt man mit der SPACE Taste.

Es sei gesagt, daß jeder der in Extended Basic bewandert ist ohne Anleitung diese ersten Eindrücke erwerben kann. Als Rat kann man dem interessierten Anfänger immer wieder nur sagen, keine falsche Scheu vor unbekannten Menüs neuer Programme zu haben.

Und schon liest man den Befehl **HELP**, der auch gleich ausgelöst wird. Derartige programminterne Anleitungen bei Fehlern oder Problemen sind bei großen Computern immer vorhanden, leider aber Home Computer Besitzern weitgehend unbekannt. Die Diskette schnurrt, und prompt werden ganze Absätze von erklärenden Worten auf dem Bildschirm sichtbar. Es handelt sich um eine Art Bibliothek auf Diskette, der für jeden Befehl eine steckbriefartige Kurzerklärung entnommen werden kann.

**ALPHA** heißt also der Befehl für das Beschreiben der Matrix mit Worten, und **VALUE** bewirkt das Analoge für Zahlen. Wer aufmerksam ist, der merkt weiters das links unten in R/C (Reihe/Spalte)-Schreibweise die jeweilige Cursorposition auf der Matrix angezeigt wird. Rechts unten schließlich sieht man in Prozent den jeweiligen noch freien Speicherplatz auf der Matrix.

Bevor wir nun ein kleines Modell erstellen, möchte ich kurz auf die anderen Befehle und die Formeln eingehen.

**BLANK:** Dieser Befehl löscht die Zelle (Position) auf der der Cursor gerade ruht.

**COPY:** Damit lassen sich Wörter, Zahlen oder Formeln von einer Zelle (Englisch: Cell) auf andere übertragen. (z.B. Linien für Tabellen, etc.)

**DEL:** Löscht wie BLANK, aber kürzt dabei Zeilen oder Spalten heraus.

**EDIT:** Erlaubt das Einsetzen einer Formel (Beziehung) beginnend ab Position des Cursors.

Z.B.: Will man drei Werte (Kosten) addieren, dann tippt man "SUM(" ein, rückt danach den Cursor zum ersten Wert, tippt einen Doppelpunkt, und bewegt nun den Cursor zum Ende der Kolonne, von der man die Summe bilden möchte. Das Drücken der ENTER-Taste bewirkt nun, daß der Computer für alle Werte zwischen Anfangs- und Endpunkt immer die Summe bildet und das Ergebnis dorthin setzt, von wo man den Befehl EDIT begonnen hat. →



**FOR:** Dieser Befehl erlaubt das Editieren der Größe der Zellen. (z.B. statt 8 Zeichen 16 Zeichen pro Zelle). Weiters kann man für Zahlen die allgemeine Schreibweise auf der Matrix bestimmen. (z.B. für alle numerischen Werte \$- Schreibweise mit zwei Kommastellen für Cent.)

**GO:** Erlaubt den Cursor an eine bestimmte Stelle zu setzen.

**INS:** Wie beim Basic analog zu Del.

**LOCK:** Schützt spezifizierte Zellen vor nachträglicher Veränderung.

**MOVE:** Ermöglicht das Verschieben von Zellen.

**NAME:** Damit kann eine Referenz zu einer Zelle über einen Namen errichtet werden.

**OPT:** Damit kann die automatische Durchrechnung der ganzen Matrix bei einer Änderung mit "NO" unterbunden werden. D.h. Zeitersparnis bei der Erstellung neuer Modelle.

**PRINT:** Druckt die Matrix aus. Entweder mit Werten, oder mit verwendeten Formeln.

**QUIT:** Damit verläßt man das Programm.

**SORT:** Ermöglicht das Umgruppieren der Matrix.

**TRANS:** Dieser Befehl bezieht sich auf die Diskette als Speicher. Bei zwei Laufwerken muß unter Option "DSK2" spezifiziert werden.

**WINDOW:** Damit kann der sichtbare Matrixbereich frei aus beliebigen Teilen der Matrix zusammengestellt werden.

**XTERN:** Dieser Befehl bindet inaktive Datenlisten in ein Modell ein.

Alle diese Befehle werden in Untermenues ausgeführt, und ermöglichen sehr komplexe Anwendungen, deren Erklärung hier zu weit gehen würde.

Der weitläufigste Befehl ist **EDIT**, bei dem, wie schon erwähnt, Formeln erstellt werden.

Denn innerhalb dieses Befehls liegt ein weiterer formaler Katalog von Multiplan begründet: Das Format der Formeln, und der darin verwendeten Funktionen.

Dazu aber nur einige Beispiele als Veranschaulichung, denn dieses Kapitel füllt schließlich fast alle 240 Seiten des Skriptums.

Man unterscheidet bei den Formeln solche, die sich auf eine ganze Gruppe von Zellen beziehen, und zwischen mathematischen, logischen und textbezogenen Funktionen.

Zur ersten Gruppe gehören Formeln wie etwa:

**AVERAGE(List):** Gibt für eine definierte Anzahl von Zellen (List) mit numerischem Inhalt das arithm. Mittel wieder.

**MAX(List):** Wird definiert wie **AVERAGE(List)** und beinhaltet den größten numerischen Wert.

**MIN(List):** Kleinster Wert.

**SUM(List):** Gibt die Summe aller Werte der Liste wieder.

Die zweite Gruppe von Funktionen, die in Formeln verwendet werden können, beinhaltet neben vom BASIC her bekannten mathematischen Operationen (wie **ABS(N)**, **COS(N)**, **EXP(N)**...) sehr interessante logische Funktionen:

**IF(Logical,Then Value,Else Value)**

entspricht dem Basicbefehl **IF/THEN/ELSE**, hat aber für Basic unbekannte Ausprägungen direkt innerhalb des Befehls. So z.B.:

**IF(R(-1)C)=0,"Gewinn","Verlust")**

Ist der numerische Wert der Zeile über dieser Zelle kleiner Null, dann schreibt Multiplan darunter das Wort "Verlust", andernfalls "Gewinn".

**ISERROR(Value):** Das Ergebnis ist "-1" wenn der Wert (Value) keinen Fehler aufweist, bzw. "1" wenn ein Fehler auftritt. (z.B. Kollision mit alphanumerischen Werten).

Dazu kommen natürlich die gängigen BASIC Symbole wie +, -, \*, /, %, &. Etwaige Fehler werden immer mit dem Präfix "E" unmittelbar in der Fehlerzelle angezeigt. (z.B. **#NUM!** für Overflow, oder **#NAME?** bei nicht definierten Namen.)

Dies nur als Abriß einer Fülle von

Bilde die Summe der Werte, die in derselben "Spalte" stehen, beginnend zehn Zeilen weiter oben bis einschließlich zwei Zeilen weiter oben. Der Doppelpunkt bildet mit den beiden Platzhaltern ein Intervall. Man muß jedoch nicht die kompliziert aussehenden Koordinaten selbst eingeben, sondern nur mit dem Cursor die jeweilige Stelle markieren. Multiplan schreibt die Formel dann automatisch an. Genau so kann man z.B. beliebige Werte der Matrix mit einer Konstanten multiplizieren (siehe Mietkosten). Man fährt nach dem Befehl **EDIT** an der gewünschten Stelle den Cursor zu jener Zahl, die man zur Berechnung verwenden will. Dann drückt man z.B. **\*3** und darauf **ENTER**, und schon ist die neue Formel fertig.

Man gewöhnt sich sehr schnell an das

MDAT: Februar									
GESAMTKOSTEN 8870									
Kosten			Kosten			Kosten			
Nahrung:	Miete:		Strom/Gas:		Gesamtkosten		pro Student: Februar		
.....	.....		.....		.....		-----		
Milch	120	Zins	1500	Strom	4600	A:		3280	
Brot	300	Betr. Kosten	390	Gas-		B:		2940	
Butter	100			Herd	590	C:		2650	
Fleisch	700								
Gemuese	100					Gesamtkosten		8870	
Kaffee	50								
Teigwaren	30								
Gefrierkost	200								
Sonstiges	190								
-----			-----			-----		-----	
Gesamt:	1790		1890		5190				
Anteil		Teilung:							
pro Student		3:2:1							
A:	605	3/6	A:	945	A:	1730			
B:	580	2/6	B:	630	B:	1730			
C:	605	1/6	C:	315	C:	1730			
-----			-----			-----		-----	
Kontrolle:	1790		1890		5190				

Funktionen, die man bei der Anwendung bald durchschaut. Das ganze Multiplan ist also eigentlich eine eigene Programmiersprache, die man lernen muß, bei der einem aber BASIC Kenntnisse weiterhelfen!

Einmal erstellte Modelle speichert man einfach auf Diskette, und kann sie wie Programme später laden und mit ihnen arbeiten.

Wie man so ein Modell erstellt, und die Formeln für bestimmte Probleme konstruiert, das ist so frei wählbar wie die Programmierweise in TI-BASIC! Jeder entwickelt dabei im Rahmen des Systems seine persönlichen Eigenarten!

Als Veranschaulichung diene folgendes kleine Modell einer Haushaltskostenberechnung von drei Studenten: A,B und C

Sie teilen sich die Miete im Verhältnis 3:2:1, da die Zimmer verschieden groß sind. Lebensmittel und Strom/Gas werden gedrittelt.

Student B trinkt aber keinen Kaffee. Diese Angaben bilden die Ausgangsbeziehungen, die wir nun in die Matrix übertragen (siehe Abb).

Beide Tabellenausdrucke zeigen dasselbe Modell: einmal mit den errechneten Werten und das andere Mal mit den verwendeten Formeln.

Die drei Kostenkolonnen (Nahrung, Miete und Strom) werden zunächst aufaddiert, und dann auf die einzelnen Studenten verteilt, wobei die vorher willkürlich getroffenen Annahmen berücksichtigt wurden.

Die Klammerausdrücke wie **R(-1)C** beziehen sich auf Werte die in die Formel als Rechengröße eingehen.

**SUM(R(-10)C;R(-2)C)** bedeutet also:

Gesamtumsatz: Plan/Ist - Ver		
		Plan
		005
Umsatz		20000
Kosten		
Herstellung	2700	
Vertrieb	2500	
Sonstige Kosten	2500	
Summe Kosten	18450	
Gewinn vor Steuern	1550	
Gewinn in %	7.75	

COMMAND: Blank Copy Del Edit For  
Go Help Ins Look Move Name Prt Print  
Quit Sort Trans Value Window Xtern  
Select option or type command Xtern  
RCL1 Umsatz 94% Umsatz

Betriebssystem von Multiplan.

Der oben erwähnte Modellcharakter von Multiplan erlaubt nun, einzelne Werte zu variieren (z.B. die Betriebskosten), und Sekunden später hat Multiplan alle Größen an die neuen Werte angepasst.

Bezüglich der Anwendung sind der Phantasie keine Grenzen gesetzt. Je mehr man sich mit Multiplan befaßt, desto schneller kann man Probleme in ein Modellschema einfließen lassen.

Aufgrund der besprochenen Möglichkeiten, und des relativ niedrigen Preises (ca.2500.-) sollte Multiplan für Interessenten von Informations-Management eine Art Pflichtsoftware werden. Wer selbst mit Multiplan arbeitet, kann gerne seine Erfahrungen und eventuelle Tips an uns zur Veröffentlichung weiterleiten.

Multiplan ist insgesamt ein sehr attraktives Softwarepaket, und eröffnet vor allem in wirtschaftlichen Belangen eine Anwendbarkeitsfülle, die man bei Home Computern dieser Preisklasse selten findet. (Alexander Berger)

DAS LEBEN IST EIN SPIEL

überleben. Wachstum und Tod sind die drei Begriffe, auf die sich jedes organische Leben und die anorganische Materie in unserem Weltall zurückführen lassen.

Aber welche Kriterien sind dafür maßgebend?

Wie entsteht Leben. Wie überlebt man, wie stirbt man?

Was hat das Ganze mit dem TI-99/4A zu tun?

"SIMULATION" ist ein gängiger Begriff für eine Programmiertechnik.

Dabei wird eine "initiale" Aufgabe definiert, die ein bestimmtes Problem beschreibt. Das Eintreffen von apriorie festgelegten Zuständen hat entscheidenden Einfluß auf die weitere Entwicklung des Problems.

Die erste und wohl berühmteste Aufgabe zu diesem Thema war ein Spiel, "LIFE", beschrieben von Martin GARDNER (in SCIENTIFIC AMERICA, Oktober 1970).

Das erste Programm hierzu schrieb John CONWAY (Gonville and Caius College, University of Cambridge, England).

CONWAY benutzte eine DEC PDP-7 Anlage mit einem GRAFIC TERMINAL.

Wir wollen das Spiel "LIFE" dazu benutzen, die wunderbaren graphischen Fähigkeiten des TI-99/4A in EXTENDED BASIC und TMS 9900 genauherzustellen.

Warum geht es beim Spiel "LIFE"?

Organismen existieren in der Form von Spielsteinen, die auf ein sehr großes Spielbrett gesetzt werden.

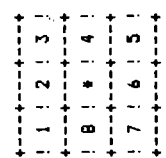
Ausgegangen wird von einer durch den Spieler gewählten initialen Population. Wir wollen diese Population "Startmuster" nennen.

Gemäß Abb. 1) besitzt jede Zelle auf dem Spielfeld 8 Nachbarfelder.

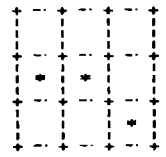
Einfache genetische Gesetze steuern das Überleben, Sterben oder die Geburt neuer Organismen.

CONWAY hat die genetischen Regeln folgendermaßen skizziert. Siehe auch Abb. 2):

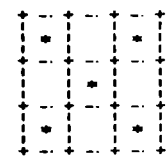
Umfelder einer Zelle



Zelle in Feld 0 überlebt



Zelle in Feld 0 stirbt



Zelle wird in Feld 0 geboren

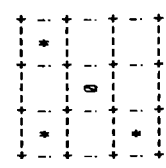


Abb. 1) GENETISCHE REGELN

Genetische Regeln

- i) Überleben Jede Zelle mit 2 oder 3 Nachbarn überlebt zur nächsten Generation.
  - ii) Tod Jede Zelle mit 4 oder mehr Nachbarn stirbt an Überpopulation. Jede Zelle mit nur einem oder keinem Nachbarn stirbt infolge Isolation.
  - iii) Geburt Jedes leere Feld, das von exakt 3 Nachbarzellen umgeben ist, ist eine Geburtszelle.
- Ganz wesentlich zur das Verständnis des Spieles ist es, daß alle Betrachtungen auf einmal auf alle Zellen einer Population angewendet werden müssen. Das heißt, alle Bedingungen treffen gleichzeitig auf alle Zellen einer Population zu.
- Was wir bei einem Spiel gewöhnlich einen Zug nennen, ist hier nicht mehr als ein Schritt weiter in der Lebensgeschichte unserer initialen Zellenkonfiguration.

Basierend auf den genannten genetischen Gesetzen haben wir die Entwicklung folgender "Organismen" beobachtet:

- i) Es gibt stabile Startmuster, hier treten keinerlei Veränderungen im Laufe der Generationen auf.
- ii) Es gibt nur ganz wenige Startmuster, für die der Organismus ohne jede Begrenzung wächst.
- iii) Es sind auch Startmuster möglich, für die der Organismus nach einer unterschiedlichen Zahl von Generationen von der Bildfläche verschwindet.
- iv) Dann haben wir auch noch Startmuster beobachtet, die eine Reihe von Generationen mutieren, aber dann einen stabilen Endzustand einnehmen, oder die in eine oszillierende Phase übertreten. (Abb. 2)

Häufig treten Symmetrien der Populationen nach einigen Generationen auf, auch wenn das Startmuster völlig zufällig gewählt wurde. Viele initiale Populationen sind nur von kurzer Lebensdauer, andere wiederum leben lange und bringen atemberaubend schöne Lebensformen (Muster) hervor.

In der Tat ist es aber sehr schwierig, trotz der sehr einfachen genetischen Regeln ein bestimmtes Resultat vorausszusagen.

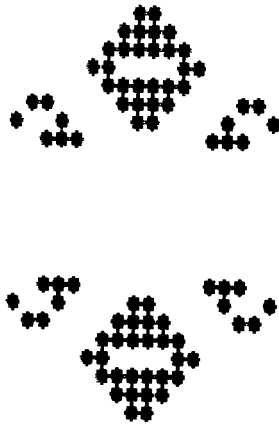
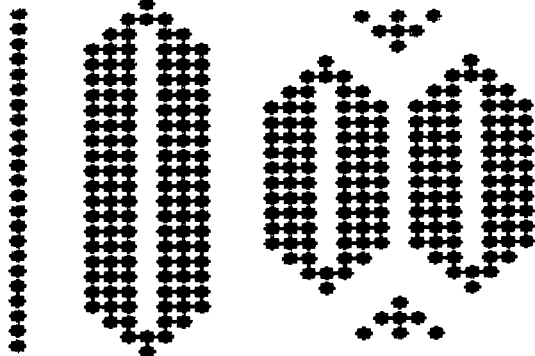


Abb. 2) Oszillierende Organismen

Langsames Wachstum in Extended Basic

Obzwar unser "LIFE" Algorithmus ökonomisch knapp gehalten ist, gehört eine gehörige Portion Geduld dazu, in XBASIC das Kommen und Gehen von Generationen von Organismen zu beobachten.

Wer allerdings eine Speichererweiterung besitzt, und sich der Mühe unterzieht, die vielen DATA Anweisungen des TMS 9900 Objekt Codes des Unterprogrammes "POPULATION" einzutippen, der wird auf vielfache Weise belohnt.

Gerätekonfiguration

11-99/4A	Konsole
	+ XBASIC Modul
	+ Kassettenrecorder
	optional
	+ Erweiterungsbox
	+ 32 K RAM Expansion

Programmbeschreibung "LIFE":

- 100 - 250 : Daten für die Haupt-Bildschirmmaske
- 340 : Wer keine Speichererweiterung besitzt und/oder den Object Code (1020 - 10610) nicht eingeben will, der muß hier ein REM einfügen.



```

360 - 600 : Startmuster (initiale Population)
390      : Der Bildschirm Hintergrund ist immer schwarz, die Vorder-
      : grundfarbe wird für jedes Beispiel zufällig neu be-
      : stimmt.
460 - 520 : Die Startpopulation wird zufällig vom Computer bestimmt.
540 - 590 : Die Startpopulation kann vom Spieler zufällig eingegeben
      : werden. Eine Zelle wird durch einen unternierten "*"
      : gekennzeichnet.
590      : Sobald der Computer in einer Eingabezeile ein "Q" ent-
      : deckt, bricht er die Zeichnentgenahme ab.
620 - 660 : Das Spiel "LIFE"
650      : Wer die reine XBASIC Version testen will, muß hier das
      : REM Statement entfernen.
660      : Aufruf des Maschinenprogrammes zur Generierung neuer
      : Populationen
      : (Nur wenn 32 k RAM Expansion angeschlossen ist!)
670 - 870 : Subprogramm POPULATION
730      : H(26,34) ..... Spaltenfeld
      : Darstellung der Zellen:
      :
      :   ABC 32 --> " " leere Zelle
      :   ASC 42 --> "*" lebende Zelle
      :
      : Um nicht wertvolle Rechenzeit durch das Lesen von Hanu-
      : bedingungen zu vermeiden, haben wir die interne Spiel-
      : feldmatrix um 2 zusätzliche Zeilen und Spalten er-
      : weitert. Diese werden generell mit 32 (leerfeld) be-
      : setzt.
      :
      : Da der Index nur die Suche von Zeilen (X=2 TO 33;
      : Y=2 TO 26) nur im Bereich der Bildschirm Matrix ordne
      : laßt, berühren sich alle Umfelder der betrachteten
      : Zelle immer innerhalb der Grenzen.
      : Dieser simple Index sortiert das Lesen von 4 Kande-
      : diungen (1 (<= X <= 32, 1 (<= Y <= 24) und somit wert-
      : volle Zeit!
740 - 770 : Immer zu Beginn der Generierung einer neuen Generation
      : wird die Population vom Bildschirm in das Feld H(,)
      : übertragen.
780 - 840 : Hauptalgorithmus
870 - 940 : C ist der Zähler, der die Anzahl der Zellen im Umfeld
      : der betrachteten Zelle festhält.
      : Er wird zur Optimierung der Geschwindigkeit auf direktem
      : Weg über Vergleiche bestimmt.

```

```

100 DATA " *****"
110 DATA " *****"
120 DATA " **"
130 DATA " ** L I F E **"
140 DATA " **"
150 DATA " *****"
160 DATA " *****"
170 DATA " "
180 DATA " Videogame by @HPESOF"
190 DATA "
200 DATA "Nach einer Idee von J.CONWAY"
210 DATA "
220 DATA " 1983-12-09 - XBASIC"
230 DATA "
240 DATA "Startpopulation eingeben...."
250 DATA " Zufallsmuster (J/N)?",EOF
260 REM
270 REM
280 REM DEKLARATIONEN
290 REM *****
300 REM
310 CALL CHAR(42,"187E7EFFFF/E7E18")
320 RANDOMIZE
330 REM
340 CALL LIFEPUKE
350 REM
360 REM STARTPOPULATION
370 REM *****
380 REM
390 CALL CLEAR : CALL SCREEN(2): Y=3+INT(14*RND)
400 FOR X=0 TO 12 : CALL COLOR(X,Y,2): NEXT X
410 Y=2 : RESTORE 100
420 READ M$ : IF M$<>"EOF" THEN DISPLAY A$(Y,1):M$ : Y=Y+1 : GOTO
420
430 CALL KEY(0,T,ST): IF ST=0 THEN 430
440 ON 1+POS("JNn",CHR$(T,1))GOTO 430,490,490,570,570
450 REM
460 REM Startpopulation zufällig wählen
470 REM -----
480 REM
490 CALL CLEAR : Z=0
500 CALL SOUND(-50,2000+500*RND,15)
510 CALL HCHAR(8+8*RND,12+8*RND,42)
520 Z=Z+1 : IF Z<=20 THEN 500 ELSE 650
530 REM

```

LIFE (TEIL 2)

```

1080 DATA 136,1,134,1,169,63,160,60,170,56,167,53,167,50,170,47
1090 DATA 175,44,167,42,161,40,173,37,171,35,171,33,172,31,160,30
1100 DATA 165,28,172,26,5,159,191,223,255,227,1,3,133,5,146,6
1110 DATA 1,147,3,3,128,5,148,3,6,135,4,167,50,145,182,6
1120 DATA 4,133,6,147,191,6,133,3,146,184,6,4,133
1130 DATA 3,147,191,6,4,133,3,146,184,6,4,133,5,147,191,6
1140 DATA 4,140,4,146,184,6,4,133,5,147,191,6,1,159,0,0
1150 DATA 52,84,52,90,200,59,131,204,248,32,51,166,131,233,216,32
1160 DATA 51,166,131,206,3,0,2,4,91,2,224,48,0,216,32
1170 DATA 51,164,131,116,4,96,53,146,2,6,3,116,2,3,48,32
1180 DATA 220,224,51,167,6,6,22,252,3,0,0,2,15,0,24
1190 DATA 2,3,51,114,2,4,0,32,6,67,192,5,6,0,10,80
1200 DATA 160,4,6,0,4,32,32,40,212,193,6,3,6,4,22,245
1210 DATA 6,5,22,240,3,0,2,2,5,0,24,2,3,51,114
1220 DATA 4,198,2,12,0,16,6,76,23,0,195,67,163,108,51,148
1230 DATA 2,4,0,32,6,67,192,5,6,0,10,80,160,4,6,0
1240 DATA 152,29,51,168,22,248,5,134,16,246,152,19,51,168,22,10
1250 DATA 2,134,0,2,17,4,2,134,0,3,21,1,16,14,208,96
1260 DATA 51,167,16,5,2,134,0,3,22,0,208,96,51,168,3,0
1270 DATA 0,0,4,32,32,3,0,0,2,6,3,6,4,22,211
1280 DATA 6,5,22,206,3,0,0,4,32,32,28,3,0,0,2
1290 DATA 152,32,131,117,51,169,19,5,152,32,131,117,51,170,19,72
1300 DATA 16,241,2,12,0,14,6,160,51,172,2,44,0,2,2,1
1310 DATA 1,0,10,204,240,76,2,0,8,17,3,0,0,0,4,32
1320 DATA 32,32,3,0,0,2,16,32,131,206,131,206,22,141,2,6
1330 DATA 0,10,2,7,52,100,2,12,0,28,6,160,51,172,194,55
1340 DATA 5,136,194,108,51,192,222,9,214,32,48,19,6,22,243
1350 DATA 2,6,0,2,2,0,52,47,2,12,0,16,6,160,51,172
1360 DATA 194,108,51,248,222,9,214,32,48,19,2,40,0,13,6,6
1370 DATA 22,243,2,0,32,0,2,1,52,24,2,0,76,3,0
1380 DATA 0,0,4,32,32,36,6,160,52,120,32,0,4,96,52,172
1390 DATA 216,32,51,164,131,124,2,224,131,224,4,96,0,112,6,160
1400 DATA
1410 REM
1420 REM OBJECT CODE LOW MEMORY POKEN
1430 REM =====
1440 REM
1450 CALL INIT :: RESTORE 1020
1460 FOR A=13204 TO 13824 STEP 16
1470 FOR I=1 TO 16
1480 READ B(I)
1490 NEXT I
1500 CALL LOAD(A,B(1),B(2),B(3),B(4),B(5),B(6),B(7),B(8),B(9),B(10),B(11),B(12),B(13),B(14),B(15),B(16))
1510 NEXT A
1520 REM
1530 REM DEF TABLE SETZEN
1540 REM =====
1550 REM
1560 CALL LOAD(8196,63,248):: CALL LOAD(8274,48,0)

```

LIFE (TEIL 3)

```

540 REM Startpopulation eingeben
550 REM -----
560 REM
570 CALL CLEAR :: FOR Y=1 TO 24
580 ACCEPT AT(Y,1)VALIDATE(" Q",BEEP:MS)

LIFE (TEIL 1)
590 IF POS(MS,"Q",1)THEN DISPLAY AT(Y,1):"" :: GOTO 650
600 NEXT Y
610 REM
620 REM DAS LIFE SPIEL
630 REM *****
640 REM
650 REM CALL POPULATION :: GOTO 390
660 CALL LINK("POFUL"):: GOTO 390
670 REM
680 REM
690 REM WACHSTUMALGORITHMUS
700 REM *****
710 REM
720 SUB POPULATION
730 OPTION BASE 1 :: DIM H(26,34)
740 FOR Y=1 TO 26 :: FOR X=1 TO 34
750 IF Y=1 OR Y=26 OR X=1 OR X=34 THEN H(Y,X)=32 :: GOTO 770
760 CALL GCHAR(Y-1,X-1,CHARS):: H(Y,X)=CHARS
770 NEXT X :: NEXT Y
780 FOR Y=2 TO 25 :: FOR X=2 TO 33
790 C=-(H(Y-1,X-1)=42)-(H(Y-1,X)=42)-(H(Y-1,X+1)=42)
800 C=-(H(Y+1,X-1)=42)-(H(Y+1,X)=42)-(H(Y+1,X+1)=42)
810 IF H(Y,X)=42 THEN IF C<=1 OR C>=4 THEN H(Y,X)=32 :: GOTO 830 ELSE 860
820 IF C=3 THEN H(Y,X)=42 ELSE 860
830 CALL SOUND(50,1000+120*H(Y,X))
840 CALL HCHAR(Y-1,X-1,H(Y,X))
850 CALL SOUND(50,3000+500*H(Y,X),15)
860 NEXT X
870 CALL KEY(0,T,ST):: IF T=15 THEN 890 ELSE IF ST THEN 870
880 NEXT Y :: CALL COLOR(2,3+INT(14*H(Y,X))):: GOTO 740
890 SUBEND
900 REM
910 REM
920 REM TMS9900 OBJECT CODE VON "POPULATION"
930 REM *****
940 REM
950 SUB LIFEPOKE
960 REM
970 DIM B(16)
980 REM
990 REM FOKELISTE
1000 REM =====
1010 REM
1020 DATA 255,221,255,222,255,223,0,1,0,35,0,34,0,33,255,255
1030 DATA 0,0,1,128,138,255,15,0,2,13,111,229,59,96,131,192
1040 DATA 2,46,122,185,200,14,131,192,59,14,4,91,139,6,133,6
1050 DATA 143,15,138,5,133,5,128,5,140,4,135,4,131,4,128,4
1060 DATA 140,3,137,3,133,3,130,3,128,3,141,2,138,2,136,2
1070 DATA 134,2,132,2,130,2,128,2,142,1,140,1,139,1,137,1

```

**SCHNELLES WACHSTUM IN TMS 9900 ASSEMBLER**

Die kleine Maschinenprogramm Routine **"POPULTN"** demonstriert auf ganz überzeugende Art und Weise die graphischen und musikalischen Fähigkeiten des TI-99/4A. Und nicht zuletzt die phänomenale Exekutiongeschwindigkeit von **TMS 9900**.

Es wurde zur Vereinfachung des Programmes nur jener Teil in TMS 9900 geschrieben, der zeitkritisch ist. Alle Eingaberoutinen blieben in XASIC; und die Kommunikation zwischen XASIC und TMS 9900 ASSEMBLER funktioniert problemlos.

Der mehrseitige Source-Code "POPOLUTNS" ersetzt die Zeilen 730 bis 890 des X BASIC Programmes "LIFE".  
Welch ein Aufwand... und es erhebt sich die Frage, ob sich diese Mühe überhaupt lohnt.

Der jedoch die Generierung von Organismen im TMS 9900 Unterprogramm beobachtet, für den ist das keine Frage mehr.

Organismen kommen und vergehen, wechseln Gestalt, Form und Farbe, und das Ganze wird noch unterteilt mit rhythmischer Zupalmusik.

## Gerätekonfiguration für Programmentwicklung

TI-99/4A

- Konsole
- + Erweiterungsbox
- + 32 K RAM Expansion
- + Diskettensystem (mindestens 1 Lautwerk)
- + EDITOR/ASSEMBLER Programmpaket

Der Quellencode besteht aus folgenden Segmenten:

i) POPULINS: Enthält die Deklarationen und diverse Unterprogramme.

iii) POPULTNIS: Enthalt das Unterprogramm "POPUL" zur Generierung von Populationen.

Wir wollen nun ein paar interessante Prozeduren herausgreifen:

**BL RANDOM:** Generiert eine Zufallszahl. Wir werden dieser Routine in "GREVERSI" und "EGERBERUS" noch begegnen.

**Zufallsmusik:** Aus einer Liste von Sopran und Baß Noten werden zufällig Töne für Soundgenerator 1 und 2 gewählt und an die entsprechenden Stellen in der Soundliste RND\$ND gesetzt.

CALL LINK ("POPUL")

Enthält den Wachstums-Algorithmus.

Es wird zunächst der Inhalt des Bildschirm Speichers nach HPFIELD kopiert und beginnend von der höchsten Adresse abwärts die Wachstumsrate von CONWAY auf jede einzelne Zelle angewendet. Die entstehende neue Population wird sofort Zelle für Zelle in den Bildschirm Speicher zurücktransformiert.

Übabei kommt es zu einem blitzartigen Wechseln von Generationen, beginnend mit der rechten unteren Bildschirm Achse, alles untermauert mit rhythmischer Musik.

[illegible]



```
*
*
* TMS 9900 ASSEMBLER Version 1.1 (XBASIC MODULE)
*
* *
* Programname: "ePOPULTNS ("ePOPULTN")
*
* ENZERSDORF 1983-12-10
*
* Dieses Programm enthaelt den Wachstumsalgorithmus
* fuer das Simulationsbeispiel LIFE.
*
* Aufruf von XBASIC ueber:
* CALL LINK("POPUL")
*
* IDT 'POPULTN'
*
* *****
* * DEKLARATIONEN *
* *****
*
* ADRG >3000      (D 12288)   Wir generieren absoluten Code
                              fuer XBASIC Modul
* DEF POPUL
*
* EQUATES fuer XBASIC Modul
VSWBW EQU >2020                VDP Single Byte Write
*
* *****
*
* POPULTN (TEIL I)
*****
VMBW EQU >2024                 VDP Multiple Byte Write
VSWBR EQU >2028               VDP Single Byte Read
XKSCAN EQU >201C              Tasteneubernahme Routine
NEXT EQU >0070               BASIC Return Routine
*
STATUS EQU >837C             GPL Status Byte
GPLWS EQU >83E0              GPL Arbeitsregister
KEYBRD EQU >8374            Tastatur Nummer
KEY EQU >8375               ASCII Kode der gedruckten Taste
*
* ARBEITSREGISTER
*
* BSS 32                     Eigene Arbeitsregister
H EQU 3                      Adressen: Spielbreit
X EQU 4                      Spaltenkoordinate
Y EQU 5                      Zeilenkoordinate
COUNT EQU 6               Zellenzahler
*
* KONSTANTE, VARIABLE
*
HF FIELD BSS 26*34           Spielbreit (D 12320)
DATA -35,-34,-33,-1        Offset fuer 8 Untfelder
DATA 35,34,33,-1          einer Zeile
```

```
*      * DATA 0 Wortkonstante
*      *
BYT1   BYTE I Bytekonstante
BLANK   BYTE 128 Leere Zelle (ASCII 32+96)
CELL    BYTE 130 lebende Zelle (ASCII 42+96)
NOKEY   BYTE >FF keine Taste gedrueckt
BACK    BYTE 15 FCIN9 = BACK Taste
*      *
EVEN
*      *
*      *****
*      *****
**     **
** DIVERSE UNTERPROGRAMME **
**     **
*****
*****
RANDOM GENERATOR
*****
Format: BL @RANDOM
*      *
```

---

```
EPILOGUE (TEIL 2)
```

```
*      * Liefert eine Zufallszahl zwischen 0 und oberer
*      * Grenze - 1
*      *
Register: R12.....obere Grenze (06)
          R12.....Ergebnis (0..06-1)
          R13,R14...Hilfsregister
          >R3C0....Randomgeneratoradresse
*      *
RANDOM LJ R13,28645 Basiswert
MPY @>R3C0,R13 multiplizieren mit Inhalt von >R3C0
AI R14,31417 Konstante hinzu
MOV R14,@>R3C0 neuer Inhalt von >R3C0
MPY R14,R12 Division durch >FFFF -> Shift 16, dh.,
RT RT R12 ist gleichzeitig Resultat
*      *
ZUFALLSMUSIK
*****
TI6 EQU 6 Notenwert 1/16
I32 EQU TI6/2 Notenwert 1/32
L11 EQU >91 Lautstärken fuer
L12 EQU >92 Sound-Generator 1
L13 EQU >93
```

**POPULTM (TEIL 2)**

```
* * Liefert eine Zufallszahl zwischen 0 und oberer
* * Grenze - 1
* *
* * Register: R12.....obere Grenze (06)
* *           R12.....Ergebnis (0..06-1)
* *           R13,R14..Hilfsregister
* *           >83C0....Randomgeneratoradresse
* *
* * RANDOM L1 R13,28645 Basiswert
* * MPY @>83C0,R13 multiplizieren mit Inhalt von >83C0
* * AI R14,31417 Konstante hinzu
* * MOV R14,@>83C0 neuer Inhalt von >83C0
* * MPY R14,R12 Division durch >FFFF -> Shift 16, dh.
* * RT R12 ist gleichzeitig Resultat
* *
* *
* * ZUFALLSMUSIK
* * *****
* *
* * T16 EQU 6 Notenwert 1/16
* * T32 EQU T16/2 Notenwert 1/32
* *
* *
* * L11 EQU >91 Lautstärken fuer
* * L12 EQU >92 Sound-Generator 1
* * L13 EQU >93
```

L14	EQU	>94		
L1F	EQU	>9F		
*				
L24	EQU	>86	Lautstaerken fuer	
L25	EQU	>88	Sound-Generator 2	
L2F	EQU	>8F		
*				
RNDMSC	EQU	>2000	Sound Liste im VDP-RAM	
*				
SOPRAN	DATA	>8B06,>8506,>8F05,>8A05	Etwas mehr als 2 Oktaven	
	DATA	>8505,>8005,>8C04,>8704	Sound Generator 1	
	DATA	>8304,>8004,>8C03,>8903		
	DATA	>8503,>8203,>8003,>8D02	C41 bis D#61	
	DATA	>8A02,>8802,>8602,>8402		
	DATA	>8202,>8002,>8E01,>8C01		
	DATA	>8B01,>8901,>8E01,>8601		
*				
BASS	DATA	>A93F,>A03C,>AA38,>A735	Etwas mehr als 1 Oktave	
	DATA	>A732,>AA2F,>AF2C,>A72A	Sound Generator 2	
	DATA	>A128,>AD25,>A823,>A821		
	DATA	>AC1F,>A01E,>A51C,>AC1A	A01 bis C21	
*				
<u>OPCODES (TEIL 3)</u>				
*	RNDNSD	BYTE 5,L1F,L2F,>DF,>FF,>E3,>01	Sound Generatoren	
*			abstellen	
SND0	BYTE 3,0,0,L12,T16,1,L13,T32		Auftakt	
SND1	BYTE 3,0,0,L14,T32			
*				
SND2	BYTE 6,0,0,0,L11,L24,T16		Dieser Table enthaelt einen	
SND3	BYTE 4,0,0,L13,L2F,T16		Takt Zufallsmusik, wobei	
SND4	BYTE 6,0,0,0,L12,L25,T16		die Bytes "0" zunaechst als	
SND5	BYTE 4,0,0,L13,L2F,T16		Platzhalter dienen und spae-	
SND6	BYTE 4,0,0,L12,L25,T16		ter zufaellig mit Sopran	
SND7	BYTE 4,0,0,L13,L2F,T16		und Bassnoten gefuellt	
SND8	BYTE 4,0,0,L12,L25,T16		werden	
SND9	BYTE 4,0,0,L13,L2F,T16			
*				
	BYTE 1,L1F,0		Soundgenerator abstellen	
*				
SOUND	DATA SND0,SND1,SND2,SND3,SND4		Adressen der Akkorde	
	DATA SND5,SND6,SND7,SND8,SND9			
*				
*				
*				
*				
*	SOUND SUBROUTINE			
*	*****			
*				
*				
*				
*				
*				
*				
FORMAT:	BL	EMUSIC		
	DATA	SOUNDPINTER		

<pre> * WACHSTUMSALGORITHMUS * *   Bildschirm in das Spielfeld kopieren * LIFE  LIM1 0      VDP-Interrupt unterdruecken       LI  Y,24      24 Zeilen       LI  H,25*34+HF1ELD Startoffset in HF1ELD       SCRLPY LI  X,32       DECI H       SCRLPX MOV  Y,R0       DEC R0       SLA R0,5       A  X,R0       DEC R0       BLWP EVSBR       ASCII Code ist nun in R1 (HByte) </pre>	<pre> JLT DIE      Ja --&gt; stirb an Isolation CI  COUNT,3  COUNT &gt;= 4? JBT DIE      Ja --&gt; stirb an Ueberpopulation JMP NEXTX    Naechste Spalte  * DIE  MOVB @BLANK,R1      Diese Zelle stirbt       JMP SHOW            Anzeigen  *       Bedingung fuer Geburt einer neuen Zelle  * BIRTH CI  COUNT,3      3 Nachbarzellen vorhanden?       JNE NEXTX          Nein --&gt; naechste Spalte </pre>	<pre>       neues Leben entsteht        MOVB @CELL,R1       LIM1 0       BLWP @VSWB       LIM1 2       NEXTX DEC  H      VDP-Interrupt unterdruecken             DEC  X      Zelle anzeigen             JNE LIFEX    VDP-Interrupt wieder zulassen             DEC  Y             JNE LIFEX </pre>
<pre> * *   Ueber Leben und Tod einer Zelle entscheiden *       LI  Y,24      24 Zeilen       LI  H,25*34+HF1ELD Startoffset in HF1ELD       LIFEX LI  X,32       DECI H       MOV  Y,R0       DEC R0       SLA R0,5       A  X,R0       DEC R0 </pre>	<pre>       Zellenzaehler loeschen       LI  R12,16       INCNT DEC R12       JNC  DEATH       MOV  H,R13       A  @YX(R12),R13       CB  *R13,@CELL       JNE INCNT       INC COUNT       JMP  INCNT </pre>	<pre>       VDP Interrupt unterdruecken       BLWP @KSCAN       LIM1 2       CB  @KEY,@NOKEY       JEQ CCOLOR       CB  @KEY,@BACK       JEQ @SCRTN       JMP DELAY </pre>
<pre> * *   Anzahl der Zellen in den Umfeldern bestimmen *       CLR COUNT       LI  R12,16       INCNT DEC R12       JNC  DEATH       MOV  H,R13       A  @YX(R12),R13       CB  *R13,@CELL       JNE INCNT       INC COUNT       JMP  INCNT </pre>	<pre>       Zellenzaehler loeschen       LI  R12,16       INCNT DEC R12       JNC  DEATH       MOV  H,R13       A  @YX(R12),R13       CB  *R13,@CELL       JNE INCNT       INC COUNT       JMP  INCNT </pre>	<pre>       VDP Interrupt unterdruecken       Tastatur ueberpruefen       LIM1 0       BLWP @KSCAN       LIM1 2       JEQ @KEY,@NOKEY       CB  @COLOR       CB  @KEY,@BACK       JEQ @SCRTN       JMP DELAY </pre>
<pre> * *   Bedingungen fuer den Tod einer Zelle *       CB  *H,@CELL       JNE BIRTH       CI  COUNT,2 </pre>	<pre>       Obere Grenze fuer Farbe       BL  @RANDOM       AI  R12,2       LI  R1,&gt;0100       SLA R12,12       SDCB R12,R1       LI  R0,&gt;20*40*17       LIM1 0       BLWP @VSWB       LIM1 2 </pre>	<pre>       Zufallsgenerator       Farbenkode 2..15 fuer Vordergrund       Hintergrundfarbe ist Schwarz       Vordergrundfarbe in HNibble von R1       Vordergrundfarbe uebertragen       Adresse des Farbenbytes im COLOR Table       Farbe von "*" aendern </pre>
<pre> * *   Zufallsmusik festlegen *       MOVB @&gt;83CE,@&gt;83CE       JNE LIFE </pre>	<pre>       Musik abgespielt?       Nein, sie spielt noch </pre>	<pre>       Sorpran zufaellig festlegen </pre>



[illegible]

# TI 99 Journal Rätsel

Zuerst möchten wir uns für die große Beteiligung am letzten Preisausschreiben, das sicherlich nicht besonders leicht war, bedanken. Nachdem uns zahlreiche Leser aus Deutschland über einen zu knapp gesetzten Einsendeschluß berichteten, haben wir den Einsendeschluß dieses Preisausschreibens um drei Wochen verlängert.

Wer also wieder oder zum ersten Mal  
mitspielen möchte, dem winken mit  
etwas Glück folgende Preise:

Preis: 10 SUPERPROGRAMME

Preis: 7 SUPERPROGRAMME

Preis: 5 SUPERPROGRAMME

-8. Preis: je 3 SUPERPROGRAMME

\_\_\_\_\_

**Alle Programme in ARCADE-Qualität!**

if Kasette: für BASIC und XBASIC.

Doch schon wollen wir uns dem eigentlichen Rätsel zuwenden! Sie lösen wieder einmal die Fragen, und notieren sich die Zahlen (in der Klammer) bei den richtigen Antworten. Danach brauchen Sie nur mehr alle Zahlen zu addieren, und schon sind Sie im Besitz der **Lösungszahl**. (Für schwache Grübler gibt es am Ende noch ein paar Hinweise! (122))

Diese betreffende Endzahl schreiben Sie dann bitte deutlich auf eine Postkarte, und senden diese an:

Thomas Fiedler  
Kennwort: Rätsel  
Felberstraße 24/26  
A-1150 Wien

Einsendeschluß ist diesmal der 21. Juli. (Datum des Poststempels). Der Rechtsweg ist ausgeschlossen. Mitarbeiter, sowie deren Angehörige auch. Die Gewinner werden in der nächsten Ausgabe veröffentlicht.

1.) Wieviele verschiedene Zeichen kann man mit einem Byte maximal darstellen?

a) 16 (34)  
b) 256 (99)  
c) 144 (69)

2.) Was ist ein "Chip"?

a) Spielgeld	(132)
b) Knabbergebäck	(163)
c) Halbleiterelement	(111)

3.) "ON" verhält sich zu "OFF", so  
wie "RUN" zu

a) STOP	(5421)
b) GOTO	(6732)
c) UNTRACE	(3499)

4.) Was versteht man unter einem "Word Processor"?

a) Übersetzungsprogramm (777)  
b) Textverarbeitung (888)  
c) Telekommunikation (346)

5.) "Munch Man" verhält sich zu "Pac Man", so wie "Visicalc" zu

a) TI-Writer	(44)
b) Multiplan	(43)
c) Cobol	(23)

6.) Worin liegen die grundsätzlichen Vorteile von Typendruckern?

a) Graphiktauglichkeit	(7007)
b) Schriftartenvielfalt	(9213)
c) Schriftqualität	(8621)

7.) Was versteht ein Computerexperte unter dem Begriff "Winchester"?

- a) Ein Repetiergewehr (97)
- b) Eine Stadt in Texas (18)
- c) Einen Festplattenspeicher (50)

8.) Was bedeutet die Abkürzung "TP"?

- a) Test Programm (21)
- b) Texas Produkt (37)
- c) Thermal Printer (10)

9.) Wie lautet das Kürzel für Texas?

a) TX (71)  
b) TE (82)  
c) TA (93)

10.) BTX überträgt die Daten eines Benutzers mittels:

a) Funk (119)  
b) Fernschreiber (321)  
c) Telefon (170)

11.) Geisterfahrer in einem fremden Datennetz nennt man;

- a) Knacker (19)
- b) Hacker (20)
- c) Hoonos (17)

12.) Wie lautet die USA-Fan -  
Kurzbezeichnung für den TI99/4A?

a) 4a (55)  
b) 99'er (47)  
c) TI99 (42)

Als Hinweis für eventuelle Sackgassen sei erwähnt, daß die Lösung (Endergebnis) fünfstellig und symmetrisch ist, und die erste und letzte Ziffer gleich sind!!

Viel Vergnügen jedenfalls beim Addieren wünscht Ihnen wie immer Alexander Berger.

Die Gewinner der Preise vom letzten Mal sind:

#### MODULE:

Gottfried STEHLIK  
Gallitzinstr. 13/9/6  
A-1160 WIEN

Manuela METKA  
Ramsauer Str. 92  
A-4020 LINZ

Ines M. KONDR  
Vorgartenstr. 107/9/16  
A-1200 WIEN

#### Kugelschreiber mit Quarzuhr:

Reinhard KERN, A-3100 St. PÖLTEN  
Dkfm. Elisabeth REISCH, A-1130 WIEN  
Konrad ROM, A-1230 WIEN  
Ewald ECKEL, A-1160 WIEN  
Manfred MENZL, A-1210 WIEN

WIR GRATULIEREN RECHT HERZLICH!

(Alexander Berger)

Nicht gewonnen, jedoch mit einem interessanten Spiel auf Kassette, als Trostpreis, bedacht wurde Hr. Nietzschmann aus A-4020 Linz, von dem wir folgendes Schreiben auf Postkarte erhielten:

"Seit 12/83 bin ich TI99 Besitzer, also ein Newcomer; da brachte mir mein Sohn Ihr TI99-Journal nach Hause. Wir waren beide sehr angetan davon. ... Begeistert war ich von dem DATA-Sort, da ich ähnliches schon längere Zeit suche. ... Ich habe inzwischen 14 Jahrgg. meiner Hobbyzeitschrift archiviert. Gesuchtes ist in 1/2 - 1 Min. zur Stelle. Nun zu Ihrem Weihnachtsrätsel: Wie soll ich Laie diese Fragen....beantworten? Leichter gings schon, als mir in den Sinn kam, bei den Antwortzahlen könnte es sich um den ASCII-Code handeln. Fast wollte ich, um aus den 59.049 Möglichkeiten (3 hoch 10) mein TOTO-Prögr. bemühen, da hätte ich jedoch 177 Min. den laufenden Monitor beobachten müssen. Da es sich um die 35.496. Möglichkeit handelt, hätte ich also 106'24" lang schauen müssen; mit KW-Rätselverstand dauerte es nur einige Minuten; einige weitere, um unter Verwendung Ihres Hilfsprogrammes die Bestätigung zu erhalten, mit den Worten:

FROHES FEST

(T.F.)

## Diskettenverwaltung

Diese Programm wurde geschrieben, um jedes beliebige Programm auf einer beliebigen Diskette anhand des Diskettenkataloges auswählen und starten zu können. Nun hatte die ganze Sache einen kleinen Haken: Unter Extended-Basic kann man zwar z.B. diese Programmzeile schreiben:

```
100 RUN "DSK1.TEST"
Dann wird das Programm TEST von der Diskette geladen und gestartet. Leider versteht der TI die folgende Anweisung aber nicht:
100 A$="DSK1.TEST"
110 RUN A$
```

Somit konnte rein mit Extended Basic kein universelles Menueprogramm geschrieben werden. Also muss die entsprechende Programmzeile jeweils ins RAM gepoked werden. Dazu ist natürlich eine Speichererweiterung notwendig.

Zu Beginn des Programms wird erst einmal nach der Laufwerknummer gefragt. Besitzer eines Laufwerkes geben hier immer 1 ein. Dann wird gemäß den Angaben im Handbuch zum Diskettenkontrolller der ganze Inhalt in Variablenfelder gelesen und die ersten 20 Titel auf dem Bildschirm ausgegeben (Zeile 190-340). Die Zeilen 350 bis 480 beinhalten die Auswahl des gewünschten Programms. Der Cursor wird mit den Funktionstasten auf- bzw. abwärts auf den gewünschten Titel bewegt. Sind mehr als 20 Titel auf der Diskette, so wird der Cursor einfach oben bzw. unten über den Rand bewegt, dann werden die weiteren Titel angezeigt. Wird ein Titel mit der ENTER-Taste bestätigt, wird dieses Programm geladen und gestartet. Wird eine Datei ausgewählt erscheint am unteren Bildrand der Hinweis "Kein Programm, Neue Eingabe".

Die Zeilen 490 und 500 dienen der Errorbehandlung. In den Zeilen 510 bis 630 wird die entsprechende Programmzeile ins RAM gepoked, d.h. erst wird die Adresse der Zeile 1000 gesucht und dann die Programmzeile in einem String zusammengestellt (Zeile 590). Dieser wird dann an die richtige Stelle ins RAM geschrieben (Zeile 600-630). Die Zeile 1000 darf auf gar keinen Fall verändert werden, d.h. es muß sowohl die Nummer und auch die Zeilenlänge (Blanks) erhalten bleiben. Auch muß es die letzte Zeile im Programm sein. Das war's, ein universell anwendbares Diskettenmenue ist fertig, das am einfachsten unter dem Namen "LOAD" auf jede Diskette gespeichert wird, dann wird es jedesmal beim Anwählen von Extended Basic automatisch gestartet

Heiner Martin

Nachdruck und gewerbli. Nutzung nur mit ausdrücklicher Genehmigung des Verfassers.

#### DSKIMENUE

```
100 REM DISKETTENMENUE
110 REM DIESES PROGRAMM ERMO
EGLICHT DIE AUSWAHL UND DAS
STARTEN JEDES AUF DER DISKET
TE BEFINDLICHEN PROGRAMMS
120 REM X--BASIC UND SPEICHER
ERWEITERUNG ERFORDERLICH
130 REM COPYRIGHT 10/83 H. M
ARTIN/ULM
140 CALL INIT
150 ON ERROR 490
160 DISPLAY AT(8,1)ERASE ALL
:"LAUFWERKNUMMER EINGEBEN:"
170 ACCEPT AT(8,27)VALIDATE(
"123")SIZE(1):A$
180 A$="DSK"%A$%". "
190 DIM B$(127),C(127),D(127
),E(127),TYPE$(5)
200 TYPE$(1)="DIS/FIX"
210 TYPE$(2)="DIS/VAR"
220 TYPE$(3)="INT/FIX"
230 TYPE$(4)="INT/VAR"
240 TYPE$(5)="PROGRAMM"
250 OPEN #1:A$,INPUT ,RELATI
VE,INTERNAL
260 INPUT #1:C$,J,J,K
270 DISPLAY AT(1,1)ERASE ALL
:"DISKETTENNAME: ";C$
280 DISPLAY AT(2,1):"SEKTOR
FREI: ";K;"BELEGT: ";J-K
290 FOR X=1 TO 127
300 INPUT #1:B$(X),C(X),D(X
),E(X)
310 IF LEN(B$(X))=0 THEN 340
320 IF X>20 THEN 330 ELSE DI
SPLAY AT(X+3,1):STR$(X);TAB(
4);B$(X);TAB(15):STR$(D(X));
TAB(19):TYPE$(ABS(C(X))): I
F ABS(C(X))<5 THEN DISPLAY A
T(X+3,26):STR$(E(X))
330 NEXT X
340 CLOSE #1
350 Z,M=1
360 DISPLAY AT(Z+3,1):STR$(M
);TAB(4);B$(M);TAB(15):STR$(
D(M));TAB(19):TYPE$(ABS(C(M
))): IF ABS(C(M))<5 THEN DIS
PLAY AT(Z+3,26):STR$(E(M))
370 ACCEPT AT(Z+3,1)VALIDATE
(NUMERIC)SIZE(-3):M
380 CALL KEY(0,K,S): IF K=1
0 THEN 390 ELSE IF K=11 THEN
420 ELSE IF K=13 THEN 450 E
LSE 370
390 M=M+1 :: Z=Z+1 :: IF Z>2
0 THEN Z=1
400 IF M>X-1 THEN M=1 :: Z=1
410 DISPLAY AT(24,1):: GOTO
360
420 M=M-1 :: Z=Z-1 :: IF Z=0
THEN Z=20
430 IF M=0 THEN M=X-1 :: Z=2
0 :: IF Z>X-1 THEN Z=X-1
440 DISPLAY AT(24,1):: GOTO
360
450 ON ERROR 500
460 IF ABS(C(M))<5 THEN DISP
LAY AT(24,1):"KEIN PROGRAMM!
NEUE EINGABE." :: GOTO 360
470 DISPLAY AT(24,1):"LOADIN
G: ";B$(M)
480 GOTO 510
490 ON ERROR 500 :: CLOSE #1
:: RETURN 150
500 RETURN 150
510 CALL PEEK(-31952,J,I)
520 J=J*256+I-65536
530 CALL PEEK(J,I,0)
540 IF I*256+0<>1000 THEN 35
0
```

# XBasic noch besser

```
550 J=J+2
560 CALL PEEK(J,I,0)
570 J=I*256+0-65536-1
580 D$=A$&B$(M)
590 D$=CHR$(LEN(D$)+4)&CHR$(
169)&CHR$(199)&CHR$(LEN(D$))
&D$&CHR$(0)
600 FOR X=1 TO LEN(D$)
610 I=ASC(SEG$(D$,X,1))
620 CALL LOAD(J+X-1,I)
630 NEXT X
1000 REM "
```

## Fehlerberichtigung:

Bedauerlicherweise haben sich sowohl ins letzte Journal wie auch ins Assembler Spezial 1 kleinere Fehler eingeschlichen:

### SELBSTBAU-RS232-Schnittstelle:

(Journal 4/83, Seite 12): Test der Schaltung - es müssen die ca. +12V an Pin 3 des RS232-Steckers bei Verbinden von Pin 3 des Joysticksteckers mit Masse, nicht mit plus !, messbar sein. Zur Platine: der eingezeichnete Widerstand R4 ist an dieser Stelle durch eine Drahtbrücke zu ersetzen und in die Verbindungsleitung zum RS232-Stecker frei verdrahtet zu legen!

**CALL POKEV** für XBasic: (Assembler Spezial 1, Seite 10)

Im Source-Code ist nach der Zeile POKEV MOV \$ADR,R0 einzufügen: MOV B \$VAL,R1 (nächste Zeile dann: BLWP \$2020).

**JOYSTICKPORT-SOFTWARE** (Assembler Spezial 1, Seite 47ff): Den System Equates ist hinzuzufügen: FASTWS EQU PAD. auf Seite 52 ist beim Label RETPAR der Op-Code SLA R1,1 hinzuzufügen.

**K1. BIT-MAP-GRAFIK** (Assembler Spezial 1, Seite 8) Im Source-Code sind folgende Änderungen vorzunehmen:

```
7D5B PUNKT MOV R11,R10
CLR R0
LI R1,1
.
ANDI R0,R7
A R3,R7
S R0,R7
.
7D9E NXT4 DEC R0
JEQ WRITE
SRC R9,1
JMP NXT3
```

**BASICBEFEHLE** (Minimum Spezial, Seite 6) Das CALL PEEK Beispiel muß folgendermaßen lauten: CALL PEEK(-31888,A,B) PRINT A\*256+B

**VSBW** (Minimum Spezial, Seite 10) Im Demoprogramm ist die Anmerkung zu Label LOOP von VSBW auf VSBW zu verbessern.

**K1. BIT-MAP-GRAFIK** (Minimum Spezial, Seite 43) Korrekturen siehe oben!

Mit den folgenden acht "Geboten" können Sie nicht nur Speicherplatz (Bytes) sparen, sondern auch die Geschwindigkeit erhöhen:

1. Halten Sie die Namen der Variablen so kurz wie möglich. (z.B. A, B usw.)
2. Halten Sie die Anzahl der Variablen möglichst gering, und verwenden Sie sie sooft es geht.
3. Gewöhnliche Variable "laufen" rascher als ein- oder mehrdimensionale. (z.B. A ist rascher als A(2). Und A(2) ist rascher als A(2,2).)
4. Benutzen Sie die niedrigste Nummer für das Sprite, welches am meisten in Bewegung ist.
5. Versuchen Sie möglichst viele Charaktere in einem CALL CHAR-Statement zu definieren. z.B.: CALL CHAR(46,"FFFFFF",70,"10AFCE",42,A\$)
6. Reduzieren Sie die GOTO- und GOSUB-Statements auf ein Minimum.
7. Je weniger Bytes ein Statement hat, desto rascher läuft es.
8. Füllen Sie die Zeilen mit Mehrfach-Statements. Aber ACHTUNG! Dieser Rat ist vor allem für geübte Programmierer. Sollte ein Fehler auftreten, so wird die Zeilennummer ausgeworfen. Dieser Hinweis ist bei Zeilen mit mehreren Statements keine große Hilfe mehr.

## CALL JOYST

Die folgenden drei Programme zeigen, wie Sprites bewegt werden können. Das erste Programm bewegt das Sprite, wenn der Joystick betätigt wird. Das zweite demonstriert steigende Geschwindigkeit. Je länger die Sticks in eine Richtung gehalten werden, desto rascher wird die Geschwindigkeit. Das dritte Programm bewegt das Sprite jedesmal in eine neue Graphikzeile und/oder -spalte, wenn der Stick bewegt wird. Das Sprite kann mit dem Joystick 1 in alle 8 Richtungen gelenkt werden. (Beachten Sie bitte die Anmerkung nach dem dritten Programm.)

ACHTUNG: Die ALPHA-LOCK-TASTE darf nicht gedrückt sein.

### Programm\_1

```
100 CALL CLEAR :: CALL
SPRITE(1,42,1 6,100,100)
110 CALL JOYST(1,X,Y):: CALL
MOTION(1 1,-Y*4,X*4):: DISPLAY
AT(23,4):" Y="
Y," X="X,"-Y*4=":-Y*4,"X*4=":X*4 ::
GOTO 110
```

Zu 110: Wenn der Stick bewegt wird, wird den Variablen X oder Y oder beiden die Werte 4,-4 oder 0 zugewiesen. Die X-Variable steht für die Spaltenrichtung und die Y-Variable für die Zeilenbewegung. Da sich die Zeilen nach oben subtrahieren und nach unten

addieren, müssen wir das Vorzeichen von Y ändern. \*4 ergibt die Geschwindigkeit (0,-16,16).

### Programm\_2

Wegen der besseren Übersicht stehen die wichtigsten Befehle in einzelnen Zeilen.

```
100 CALL CLEAR :: CALL SCREEN(7) ::
CALL MAGNIFY(2)::
CALL SPRITE(1,40,1 6,100,100)
110 CALL JOYST(1,S,Z)
120 X=(X+S)*-(ABS(X)<124)
130 Y=(Y-Z)*-(ABS(Y)<124)
140 CALL MOTION(1,Y,X):: DISPLAY AT
(24,1):"Y=";Y,"X=";X :: GOTO 110
```

Zu 120 u. 130: Wörtlich bedeuten diese Statements folgendes: Wenn der absolute Wert von X (Y) kleiner als 124 dann X=X+C andernfalls X=0.

Bevor der Computer der Variablen X (Y) einen neuen Wert zuweist, berechnet er alle Ausdrücke und Funktionen. Das heißt, daß das Ergebnis von (ABS(X)<124) -1 ist, wenn es wahr ist (absolute Zahl kleiner 124). Wenn die Zahl größer oder gleich 124 ist (falsch), dann ist das Ergebnis diese Ausdruckes 0. Das Ergebnis wird negiert und mit dem linken Ausdruck multipliziert, sodaß, entweder eine positive Zahl (kleiner 124) oder 0 in der Zielvariablen steht.

### Programm\_3

```
100 CALL CLEAR :: CALL COLOR(2,7,7)
:: CALL SCREEN(12)
110 CALL HCHAR(24,1,40,64)
120 CALL VCHAR(1,31,40,96)
130 CALL SPRITE(1,42,2,17,17):: Z
SP=3
140 CALL JOYST(1,X,Y):: X=SGN(X)::
Y=-SGN(Y)
150 CALL GCHAR(Z+Y,X+SP,A):: IF A=40
THEN CALL SOUND(-50,120,7):: GOTO
140 ELSE SP=SP+X :: Z=Z+Y :: CALL
LOCATE (1,1,Z+8-7,SP+8-7):: GOTO 140
```

Zu 110: Da nur 32 Spalten am Schirm Platz haben, werden die restlichen Blöcke in der ersten Zeile platziert. Zu 120: Beginnend in der Spalte 31 werden 96 Zeilen mit Blöcken versehen. 48 Blöcke in den Spalten 31 und 32, die restlichen in den Spalten 1 und 2.

Zu 140: X erhält je nach Vorzeichen -1,0 oder 1. Y erhält das negierte Vorzeichen.

Zu 150: CALL GCHAR überprüft das folgende Zeichen. :: Wenn es ein Block ist ertönt ein Ton, wenn nicht, wird die neue Zeile und/oder Spalte errechnet und mit Hilfe von CALL LOCATE wird das Sprite neu platziert.

ANMERKUNG: Falls Sie die Bewegungen der Diagonale nach ausschalten wollen, geben Sie folgendes ein:

```
140 CALL JOYST(1,X,Y):: IF X*Y THEN
140 ELSE ....(Rest) oder IF X AND Y
THEN 140 ELSE ....(Rest)
Wörtlich: Wenn X<>0 und Y<>0 (kann nur bei Diagonalbewegungen möglich sein.), dann frage wieder ab.
```

Christian Roch



# Tips für Peripherie-Besitzer

Viele von Ihnen werden sich schon darüber geärgert haben, daß sie mit dem Vollausbau Speicherplatz verloren haben.

Das hört sich für Uneingeweihte nun paradox an, aber es ist eine Tatsache! Speicherplatz muß etwa für die Diskettendateien reserviert werden etc.

Schon beim Erwerb des XBasic-Moduls muß der Erstaunte feststellen, daß die neue Programmiersprache ihn nicht nur finanzielle Einbußen beschert. Zugleich verabschiedet sich im XBasic-Modus etwa 1/11 des Speicherplatzes. Nun, dieser kleine Verlust wird bei weitem ausgeglichen, ebenso wie der Speicherplatz-Schwund durch die Disk-Drives bei weitem durch die Vorteile dieses schnellen Festspeichers ausgeglichen werden. Außerdem haben Sie ja jederzeit die Möglichkeit durch "CALL FILES(1)" und "NEW" die Zahl der vorbereiteten Dateien auf 1 zu beschränken und somit über 1000 Bytes zu gewinnen!

Aus alledem geht hervor, daß der größtmögliche Speicherplatz bei angeschlossenen Disk-Drives in Basic nach der Exekution des obigen Befehles erreicht werden kann.

Leider haben gerade gute Basicprogramme oft die unangenehme Eigenschaft sich durch vollständiges Nutzen des Speicherplatzes auszuzeichnen. Das heißt, das Programm läßt sich mit obiger Prozedur gerade noch laden (50-54 Sektoren auf der Disk), sobald der Computer jedoch, beginnt, zu dimensionieren, erhalten Sie ein "MEMORY FULL IN XXXXX".

Das haben wohl Disk-Besitzer schon mit Frust zur Kenntnis nehmen müssen.

Nun ist es möglich, mit einem CALL LOAD(-31888,63,255) ein sinngemäßes CALL FILES(0) durchzuführen, wenn anschließend das Programm in einer Zeile editiert wird.

Um Mißverständnisse zu vermeiden: CALL LOAD geht nur mit Speichererweiterung oder mit dem Minimem-Modul!

Sie haben jetzt also die Möglichkeit, mit dem Minimem-Modul extrem lange Basicprogramme zuerst von Disk zu laden, dann die DiskDrives abzuschalten und das Programm zu starten, nachdem Sie eine Zeile verändern und wieder diese Veränderung rückgängig machen (Editieren). Jedoch hat dieser Befehl seine Tücken und es war für uns immer ein Glückspiel, die richtige Prozedur zu finden. So gab es schon verschiedene "Glaubensrichtungen" innerhalb des Clubs, die jede eine bevorzugte Prozedur vertraten.

Jetzt erreichte uns über den Ozean eine fast perfekte Lösung des Problems von Craig Miller. Dazu nun das Kochrezept!

Man nehme: Einen voll ausgebauten Ti-99/4A(Memexp,Disk,Minimem-M)

Jetzt wählt man mit angestecktem Minimem Ti-Basic und tippt "CALL

INIT". Vorsicht! Damit wird der Inhalt des MMM gelöscht. Gleichzeitig "erkennt" der MMM die Speichererweiterung.

Jetzt wird das Basicprogramm geladen: "OLD DSK1.PROGRAMM".

Der nun folgende Befehl ist der Schlüssel: "SAVE EXPMEM2".

Damit wird das Programm in die 24 Kb der Speichererweiterung verlegt, die das MMM als Expmem2 adressiert.

CALL LOAD(-31888,63,255) veranlaßt den Ti-99, keine Dateien freizuhalten.

Jetzt geben Sie "NEW" und "ENTER" ein. Jetzt erst wurde Speicherplatz im VDP-RAM verfügbar gemacht.

Da Basic-Befehle die Speichererweiterung nicht ansprechen, ist nun Ihr Basicprogramm immer noch sicher in Expmem2 aufgehoben (A000 bis FFFF).

Der nächste Befehl lautet daher: "OLD EXPMEM2" und das Programm wird aus der Memexpansion gelesen. Sie haben jetzt das Programm zweimal! Einmal in Expmem2 und einmal im Arbeitsspeicher Ihres Computers.

Nun sollte auch das größte Basicprogramm problemlos exekutiert werden.

Viel Vergnügen beim Experimentieren und Editieren und Programmieren wünscht allen Ti-Fans:

E.Reitinger

## UNTERPROGRAMME AUS DEM MODUL DATENVERWALTUNG UND ANALYSE

Wie schon im TI 99 Journal 4/83 erwähnt, bietet das Modul Datenverwaltung und Analyse einige Unterprogramme, die aus dem TI-Basic aufgerufen werden können. Natürlich muß dazu das Modul in der Konsole eingesteckt sein. Hier nun ein paar weitere Unterprogramme, wobei zur Vollständigkeit die schon bekannten nochmal aufgeführt sind.

Interessant dabei ist wohl auch noch, daß die den Extended-Basic Befehlen DISPLAY und ACCEPT AT entsprechenden CALL A und D Unterprogramme auch für numerische Variablen benutzbar sind. Die folgende Aufstellung erhebt selbstverständlich nicht den Anspruch auf Vollständigkeit, auch sind noch nicht alle Eigenschaften der Unterprogramme völlig geklärt. Dem eigenen Experimentieren sind also keine Grenzen gesetzt.

CALL D(Zeile,Spalte,Laenge,String-oder numerische Variable): Funktioniert wie DISPLAY AT im Extended BASIC. Mehrfachanweisungen hintereinander sind möglich.

CALL A(Zeile,Spalte,Länge,Funk.-Var.,Variable): Funktioniert wie ACCEPT AT, stellt aber die Zeile nicht frei. Die CLEAR-Taste arbeitet wie die ERASE-Taste im Edit-Modus. Bei Eingabe einer numerischen Variablen können nach der Variablen noch die Grenzen

bestimmt werden, z.B. (...A,1,20) läßt die Eingabe für A von 1 bis 20 zu. Bei Betätigung einer Funktionstaste wird der Variablen der letzte eingegebene Wert zugeordnet, also nicht der Wert, der auf dem Bildschirm steht. Die Werte für die Funktionstasten:

BEGINN = 6

PROCEED = 5

AID = 3

REDO = 4

BACK = 7

ENTER = 1 (bei Leereingabe 2, dann wird die Variable aber nicht gelöscht sondern behält ihren bisherigen Wert)

CALL P(10000); Reserviert Speicherplatz für das Modul. Ist aber vor dem Laden eines Programms auszuführen!

CALL H(0,6,0,X); Übergibt Anzahl der Datensätze in den Speicherplatz des Moduls.

CALL H(1,6,0,X); Übernimmt.

CALL L("File",Y):Laden einer mit dem Modul erstellten File in den vorher reservierten Speicherplatz. Funktion von Y unbekannt.

CALL S("File",Y):Saven einer File. Y siehe oben.

CALL S und CALL L arbeiten im Programmformat!

CALL G(0,Datensatz,Datenfeld,Variable):Zuordnen des Wertes der Variablen in die Datenorganisation.

CALL G(1,Datensatz,Datenfeld, Funktion,Variable): Der Variablen wird der gespeicherte Wert zugewiesen. Ist Funktion<>0 dann ist nichts gespeichert.

CALL G(2,Datensatz,Datenfeld, Variable): Löscht wahrscheinlich den gespeicherten Wert.

Diese Unterprogramme sind nun kein "Ersatz" für weitere Befehle des Extended-Basic. Allerdings läßt sich damit eine interessante Lösung für die Datenspeicherung auf Kassettenrekorder finden. Speichert das Modul doch seine Daten im Programmformat ab, das sich wesentlich schneller laden und speichern läßt als die sequentiellen Dateien.

Zur Demonstration der beiden ersten Unterprogramme hier ein kleines Programm:

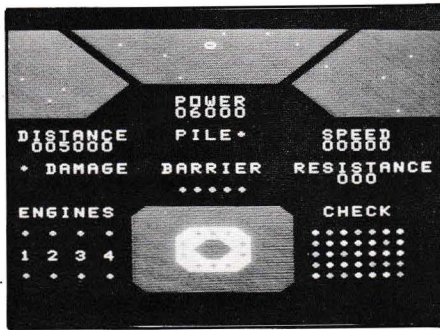
```
100 CALL CLEAR
110 CALL D(0,2,20,"1. VARIABLE:")
120 CALL D(10,2,20,"2. VARIABLE:")
130 CALL A(8,15,4,S.A,-2000,2000)
140 IF S=7 THEN 190
150 CALL A(10,15,4,S.B,-2000,2000)
160 C=A+B
170 CALL D(13,2,7,"SUMME=",13,14,5,C)
180 GOTO 110
190 END
```

Dieses Programm errechnet die Summe zweier Zahlen, die jeweils zwischen -2000 und +2000 liegen dürfen. Mit Betätigen der Taste BACK bei der Eingabe der ersten Variablen wird das Programm beendet.

Heiner Martin

Nachdruck nur mit Genehmigung des Verfassers.

# Space Killer (Basic)



Superspiel mit Grafik und Köpfchen für normales Basic.

Sie sitzen in einem Raumjäger, oder haben Sie die Schlachtschiff Variante gewählt? Nun jedenfalls ist da diese Raumstation vor Ihnen. Was heißt Station, es muß schon ein Raum-Fort sein, eine Nachschubbasis des Feindes mit schier unerschöpflichen Energiequellen. Ihre Mission bringt Sie nun, allein in Ihrem Schiff auf Angriffskurs Marke Himmelfahrt. Die Energiebank ist voll (wenigstens etwas), doch halt! Ein falscher Druck auf 'P' für 'Pile' zu deutsch Atommeiler überlädt Ihre Speicher (POWER) und läßt Ihr Schiff als Atomwolke weiterziehen. Die Raumstation wird sich um Sie erst kümmern, wenn Sie näher als 3000 Raummeilen heran sind. Dann aber heißt es, Schutzschirme voll einschalten. Die Taste 'B' errichtet eine raffinierte Raumfalle um Ihr Schiff, was natürlich ein wenig von Ihrer Energie kostet. Also Meiler recht schnell wieder einschalten, bevor der erste Schuß der Basis Sie trifft. Bei eingeschaltetem Schutzfeld leuchten die fünf Lampen unter der Schrift 'BARRIER'. Jede Steuertaste schaltet auch wieder dieselbe Funktion aus. Ausnahme sind die Bordtriebwerke. ('ENGINES'). Sie werden stufenweise eingeschaltet wobei 'E' Sie von der Raumbasis wegführt, während 'X' in Richtung Feind beschleunigt. Logischerweise werden Sie also bei Gegenbeschleunigung zuerst mit 'X' die mit 'E' eingeschalteten Triebwerke ausschalten und umgekehrt. Die Linear-Phaser, die als einzige Ihr Schutzschild von innen durchdringen können, werden mit der '-'-Taste gefeuert. Vorsicht, ein solcher Schuß kann Sie mehr Energie kosten, als Sie zur Verfügung haben. Am besten schreiben Sie sich anfangs die Tabelle mit dem Energieverbrauch der einzelnen Bordgeräte auf. Sie erscheint, wenn Sie sich für einen Schiffstyp entschieden haben. Auf dem Instrumentenbrett haben Sie noch eine Anzeige 'SPEED'. Wenn sie negativ wird, verringert sich die Distanz zur Basis.

'RESISTANCE' zu deutsch Widerstand gibt Ihnen Auskunft darüber, welche Wirkung Ihre Treffer bei dem Gegner erzielt haben. Bei einem Wert von 50' wird ein Teil der Basis zerstört und die Anzeige zurück gesetzt auf '0'. 'CHECK' ist eine direkte digitale Ausgabe der umgebenden Felder und nur von digital arbeitenden Gehirnen verstehbar. Für menschliche Piloten also nicht interessant! Und wenn die Basis schießt, blinkt zur Warnung die 'DAMAGE' Lampe und ein Warnton signalisiert das Ereignis. Der Bordrechner arbeitet erst weiter, wenn Sie diesen Alarm mit 'ENTER' quittieren. Ein Probelauf: Sie starten das Programm: Jetzt wählen Sie mit '3' den Typ eines Einsatzkreuzers. Ihr Bildschirm lautet jetzt: Einsatzkreuzer Frisco  
Bank 10.000  
Gun 5.000  
Meiler 5.000  
Schirmfeld 3.000  
Engines 1 3 9 27

Das bedeutet nun für Sie, daß Ihre Geschütze 5.000 Energie-Einheiten pro Durchlauf verbrauchen, das Schutzfeld 3.000 und die Triebwerke 1,3,9 oder 27 Schubeinheiten, je nachdem, wieviele Triebwerke arbeiten. Mit dem Schwierigkeitsgrad wächst die Wahrscheinlichkeit eines Schusses der Basis. Bei '1' haben Sie pro Durchlauf mit einem Schuß zu rechnen!

Ja nun ein Wort zum Durchlauf! Wie Sie an der Beschriftung des Cockpits gesehen haben, ist Ihr Schiff englischer Herkunft. Genauer muß ich die Herkunft wohl nicht beschreiben, wenn ich erwähne daß nur ein einziger Bordcomputer existiert, der sämtliche Vorgänge steuert. Es handelt sich um ein etwas älteres, aber sehr leistungsfähiges Modell, das die Erbauer bei einer Ausverkaufaktion günstig erwerben konnten. Es ist - erraten - der legendäre TI-99/4A, der für Ihre Klimaanlage etc. verantwortlich ist. So kommt es, daß sämtliche Maschinen nacheinander gechecked werden und ein vollständiger Check heißt: 'Durchgang'. Nun viel Vergnügen und 'Raum-0'

```
90 REM SPACE-KILLER
   by TAEUBLER TI und XB
   VERS 1.84
100 CALL CLEAR
110 PRINT "BITTE WAELHEN SIE
   :": "1 JAEGER": "2 ZER
   STOERER": "3 EINSATZKREUZ
   ER": "4 SCHLACHTSCHIFF":
   :": "5 :
120 INPUT "IHRE WAHL? ":I
130 ON I GOSUB 200,220,240,2
60
```

```
140 READ WID,E1,E2,E3,E4,BAR
   GUN,POW,MEIL,SCHIFF#,B
150 PRINT : "IHRE DATEN":
   :SCHIFF#:"
   :": "GUN",GUN: "ME
   ILER",MEIL: "SCHIRMFELD",BA
   R: "SPEICHERBANK",POW: "EN
   GINES: ";E1;E2;E3;E4: :
160 INPUT "SCHWIERIGKEITSGRA
   D-1 (SCHWER) BIS 9 (LEICHT)":SC
   HW
170 GOTO 300
180 DATA 50,1,5,25,125,2000,
   2000,6000,3000," JAGDBOMBER
   JABO-ZXZ1",300000
190 DATA 80,1,4,16,64,2500,3
   000,8000,3500," ZERSTOERER
   ZST-48-16",375000
200 RESTORE 180
210 RETURN
220 RESTORE 190
230 RETURN
240 RESTORE 280
250 RETURN
260 RESTORE 290
270 RETURN
280 DATA 120,1,3,9,27,3000,5
   000,10000,5000," EINSATZKREU
   ZER FRISCO",420000
290 DATA 200,1,2,4,8,5000,80
   00,12000,8000,"SCHLACHTSCHIF
   F AMPHITREON",1040000
300 CALL CLEAR
310 CALL CHAR(114,"7F0703010
   1010000")
320 CALL CHAR(33,"FF7F3F1F0F
   070301")
330 CALL CHAR(112,"008080C0E
   0C0F0FE")
340 CALL CHAR(34,"FFFEFCFBF0
   E0C080")
350 CALL CHAR(113,"FEE0C0C0B
   0808080")
360 CALL CHAR(35,"80C0E0F0FB
   FCFEFF")
370 CALL CHAR(115,"000001030
   3070F3F")
380 CALL CHAR(36,"0103070F1F
   3F7FFF")
390 CALL CHAR(116,"000000012
   1F3FFFF")
400 CALL CHAR(88,"0000183C3C
   18")
410 CALL CHAR(117,"0F0307030
   303070F")
420 CALL CHAR(96,"0000183C3C
   18")
430 CALL CHAR(118,"FFFFD5C0B
   0000000")
440 CALL CHAR(104,"00003C424
   23C0000")
450 CALL CHAR(119,"FBF0C0E0F
   0D0F0E0")
460 CALL CHAR(105,"80C0E0F0F
   8FCFEFF")
470 CALL CHAR(106,"FFFEFCFBF
   0E0C080")
480 CALL CHAR(128,"181818181
   8181818")
490 CALL CHAR(107,"FF7F3F1F0
   F070301")
500 CALL CHAR(108,"0103070F1
   F3F7FFF")
510 CALL CHAR(37,"FFFFFFFFF
   FFFFF")
520 CALL CHAR(109,"FFFE7C3C
   3E7FFFF")
530 CALL CHAR(110,"00000010"
   )
540 CALL CHAR(111,"000000000
   000009B")
550 CALL CHAR(120,"010204081
   0204080")
```

```

560 CALL CHAR(121,"804020100
8040201")
570 CALL COLOR(1,5,1)
580 CALL COLOR(8,9,1)
590 CALL COLOR(11,16,5)
600 CALL COLOR(13,5,5)
610 CALL COLOR(9,4,1)
620 FOR I=2 TO 7
630 CALL COLOR(I,16,1)
640 NEXT I
650 CALL COLOR(10,16,5)
660 CALL COLOR(12,5,5)
670 CALL SCREEN(2)
680 FOR I=1 TO 5
690 CALL HCHAR(I,1,37,32)
700 NEXT I
710 CALL HCHAR(6,1,37,9)
720 CALL HCHAR(6,23,37,10)
730 CALL HCHAR(7,1,37,8)
740 CALL HCHAR(7,24,37,9)
750 CALL HCHAR(8,1,37,7)
760 CALL HCHAR(8,25,37,8)
770 FOR I=1 TO 3
780 CALL HCHAR(5+I,10-I+1,34
)
790 CALL HCHAR(5+I,22+I-1,33
)
800 NEXT I
810 FOR I=1 TO 5
820 CALL HCHAR(I,5+I,35)
830 CALL HCHAR(I,6+I,33)
840 CALL HCHAR(I,27-I,36)
850 CALL HCHAR(I,26-I,34)
860 NEXT I
870 FOR I=1 TO 3
880 CALL HCHAR(2+I,16-I,120)
890 CALL HCHAR(2+I,16+I,121)
900 NEXT I
910 CALL HCHAR(2,16,104)
920 CALL HCHAR(1,20,110)
930 CALL HCHAR(2,22,110)
940 CALL HCHAR(1,17,110)
950 CALL HCHAR(2,10,110)
960 CALL HCHAR(5,14,110)
970 CALL HCHAR(3,13,110)
980 CALL HCHAR(5,3,110)
990 CALL HCHAR(7,5,110)
1000 CALL HCHAR(6,6,110)
1010 CALL HCHAR(2,2,110)
1020 CALL HCHAR(3,30,110)
1030 CALL HCHAR(4,24,110)
1040 CALL HCHAR(8,29,110)
1050 CALL HCHAR(5,28,110)
1060 CALL VCHAR(3,16,128,3)
1070 FOR I=17 TO 24
1080 CALL HCHAR(I,11,37,11)
1090 NEXT I
1100 CALL HCHAR(17,11,36)
1110 CALL HCHAR(17,21,35)
1120 CALL HCHAR(24,11,33)
1130 CALL HCHAR(24,21,34)
1140 CALL VCHAR(7,14,80)
1150 CALL HCHAR(7,15,79)
1160 CALL HCHAR(7,16,87)
1170 CALL HCHAR(7,17,69)
1180 CALL HCHAR(7,18,82)
1190 CALL HCHAR(8,14,48,5)
1200 CALL VCHAR(10,3,68)
1210 CALL HCHAR(10,4,73)
1220 CALL HCHAR(10,5,83)
1230 CALL HCHAR(10,6,84)
1240 CALL HCHAR(10,7,65)
1250 CALL HCHAR(10,8,78)
1260 CALL HCHAR(10,9,67)
1270 CALL HCHAR(10,10,69)
1280 CALL HCHAR(11,4,48,6)
1290 CALL VCHAR(10,14,80)
1300 CALL HCHAR(10,15,73)
1310 CALL HCHAR(10,16,76)
1320 CALL HCHAR(10,17,69)
1330 CALL HCHAR(10,18,88)
1340 CALL VCHAR(10,24,83)
1350 CALL HCHAR(10,25,80)

```

```

1360 CALL HCHAR(10,26,69,2)
1370 CALL HCHAR(10,28,68)
1380 CALL HCHAR(11,24,48,5)
1390 CALL VCHAR(13,3,88)
1400 CALL HCHAR(13,5,68)
1410 CALL HCHAR(13,6,65)
1420 CALL HCHAR(13,7,77)
1430 CALL HCHAR(13,8,65)
1440 CALL HCHAR(13,9,71)
1450 CALL HCHAR(13,10,69)
1460 CALL VCHAR(13,13,66)
1470 CALL HCHAR(13,14,65)
1480 CALL HCHAR(13,15,82,2)
1490 CALL HCHAR(13,17,73)
1500 CALL HCHAR(13,18,69)
1510 CALL HCHAR(13,19,82)
1520 FOR I=14 TO 18
1530 CALL HCHAR(15,I,88)
1540 NEXT I
1550 CALL VCHAR(13,22,82)
1560 CALL HCHAR(13,23,69)
1570 CALL HCHAR(13,24,83)
1580 CALL HCHAR(13,25,73)
1590 CALL HCHAR(13,26,83)
1600 CALL HCHAR(13,27,84)
1610 CALL HCHAR(13,28,65)
1620 CALL HCHAR(13,29,78)
1630 CALL HCHAR(13,30,67)
1640 CALL HCHAR(13,31,69)
1650 CALL HCHAR(14,25,48,3)
1660 CALL VCHAR(17,3,69)
1670 CALL HCHAR(17,4,78)
1680 CALL HCHAR(17,5,71)
1690 CALL HCHAR(17,6,73)
1700 CALL HCHAR(17,7,78)
1710 CALL HCHAR(17,8,69)
1720 CALL HCHAR(17,9,83)
1730 CALL VCHAR(21,3,49)
1740 CALL VCHAR(21,5,50)
1750 CALL VCHAR(21,7,51)
1760 CALL VCHAR(21,9,52)
1770 FOR I=3 TO 9 STEP 2
1780 CALL HCHAR(19,I,88)
1790 CALL HCHAR(23,I,88)
1800 NEXT I
1810 CALL VCHAR(17,24,67)
1820 CALL HCHAR(17,25,72)
1830 CALL HCHAR(17,26,69)
1840 CALL HCHAR(17,27,67)
1850 CALL HCHAR(17,28,75)
1860 FOR I=19 TO 23
1870 CALL HCHAR(I,23,88,7)
1880 NEXT I
1890 REM BASIS
1900 CALL HCHAR(19,14,108)
1910 CALL HCHAR(19,15,109,3)
1920 CALL HCHAR(19,18,105)
1930 CALL HCHAR(20,14,109)
1940 CALL HCHAR(20,15,106)
1950 CALL HCHAR(20,17,107)
1960 CALL HCHAR(20,18,109)
1970 CALL HCHAR(21,14,109)
1980 CALL HCHAR(21,15,105)
1990 CALL HCHAR(21,17,108)
2000 CALL HCHAR(21,18,109)
2010 CALL HCHAR(22,14,107)
2020 CALL HCHAR(22,15,109,3)
2030 CALL HCHAR(22,18,106)
2040 GOTO 2620
2050 REM SCHUSS
2060 CALL COLOR(12,16,5)
2070 CALL COLOR(12,5,5)
2080 CALL SOUND(100,-5,0)
2090 RETURN
2100 REM CHECK
2110 FOR I=23 TO 29
2120 CALL HCHAR(INT(5*RND)+1
9,I,88)
2130 NEXT I
2140 FOR I=23 TO 29
2150 CALL HCHAR(INT(5*RND)+1
9,I,96)
2160 NEXT I
2170 RETURN
2180 REM BESCHUSS
2190 PENG=0
2200 PSE=SCHW

```

```

2210 ZIX=ZIX+1
2220 CALL SOUND(100,-7,0)
2230 ON ZIX GOSUB 2250,2470,
2530,2350,2370,2490,2410,243
0,2450,2270,2390,2510,2330,2
590,2550,2290,2570,2310,4370
2240 RETURN
2250 CALL HCHAR(19,14,115)
2260 RETURN
2270 CALL HCHAR(19,15,116)
2280 RETURN
2290 CALL HCHAR(19,16,116)
2300 RETURN
2310 CALL HCHAR(19,17,116)
2320 RETURN
2330 CALL HCHAR(19,18,112)
2340 RETURN
2350 CALL HCHAR(20,14,117)
2360 RETURN
2370 CALL HCHAR(20,15,113)
2380 RETURN
2390 CALL HCHAR(20,17,114)
2400 RETURN
2410 CALL HCHAR(20,18,119)
2420 RETURN
2430 CALL HCHAR(21,14,117)
2440 RETURN
2450 CALL HCHAR(21,15,112)
2460 RETURN
2470 CALL HCHAR(21,17,115)
2480 RETURN
2490 CALL HCHAR(21,18,119)
2500 RETURN
2510 CALL HCHAR(22,14,114)
2520 RETURN
2530 CALL HCHAR(22,15,118)
2540 RETURN
2550 CALL HCHAR(22,16,118)
2560 RETURN
2570 CALL HCHAR(22,17,118)
2580 RETURN
2590 CALL HCHAR(22,18,113)
2600 RETURN
2610 GOTO 2060
2620 REM PROGRAMM
2630 DIST=5000
2640 BANK=POW
2650 GOSUB 3970
2660 CALL SOUND(100,1047,10)
2670 GOSUB 2100
2680 T=T+1
2690 CALL KEY(0,KEY,ST)
2700 IF KEY=69 THEN 3060
2710 IF KEY=88 THEN 3160
2720 IF KEY=66 THEN 2940
2730 IF KEY=46 THEN 3690
2740 IF KEY=80 THEN 3610
2750 SPEED=SPEED+ENG
2760 DIST=DIST+SPEED
2770 IF DIST<3000 THEN 2840
2780 PSE=PSE-1
2790 BANK=BANK+PIL-ABS(10*EN
G)-SCREEN
2800 IF INT(BANK)<0 THEN 376
0
2810 IF INT(BANK)>POW THEN 3
760
2820 GOSUB 3970
2830 GOTO 2660
2840 IF PSE>0 THEN 2780
2850 IF 1=INT(RND*SCHW)+1 TH
EN 2880
2860 BANK=BANK+PIL-ABS(10*EN
G)-SCREEN
2870 GOTO 2800
2880 GOSUB 4480
2890 IF SCREEN=0 THEN 2930
2900 IF SCREEN>=INT(B/DIST) T
HEN 4320
2910 SCREEN=INT(B/DIST)
2920 GOTO 4320
2930 IF WID<INT(B/DIST) THEN
3760 ELSE 4320
2940 REM BARRIER
2950 IF SCREEN=0 THEN 3010
2960 SCREEN=0
2970 FOR I=14 TO 18

```

# HARDWARE:

## Rekordkabel-Selbstbau

Viele jener TI 99-Besitzer, die um die Jahreswende zu den Ausverkaufspreisen noch rasch eine Gerät erwarben, stehen seither vor dem Problem, keine Kabel für den Anschluß des Kassettenrekorders zu bekommen.

Diesen Mangel kann aber leicht selbst abgeholfen werden, als mit ganz wenig Aufwand dieser für den Betrieb doch unerläßliche Bauteil mit dem Lötkolben als einziges Arbeitsgerät selbst gemacht werden kann, wobei als positiver Nebeneffekt die Preisgünstigkeit zusätzlich motivierend sein dürfte.

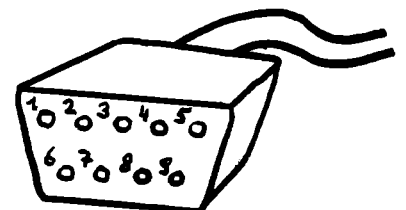
Was benötigen Sie? Zunächst für den Anschluß am TI einen neunpoligen Subminiaturstecker mit Gehäuse, einen runden Meter sechspoliges Kabel (abgeschirmt erwünscht), aber keine unbedingte Anforderung), sowie die an Ihrem Rekorder notwendigen DIN- oder Klinkenstecker.

Sollte Ihr Rekorder keine REMOTE-Buchse besitzen und wenn Sie ein bißchen Elektronikkenntnisse besitzen, können Sie diesen Mangel gleich in einem beheben - suchen Sie den Rekordernetzteil - irgendwo muß ein deutlich als Trafo erkennbarer Bauteil stecken - verfolgen Sie die Sekundärleitungen bis über den Gleichrichter hinaus und trennen Sie die positive Spannungsversorgung auf. Von der Auftrennungsstelle weg führen Sie zwei Anschlüsse zu einer neu zu legenden Buchse (eventuell an unbenutzte Pins, falls Sie als Mikrofoneingang eine fünfpolige DIN-Buchse haben).

Der TI schaltet die positive Versorgungsspannung mittels eines Schalttransistors, der von einem Optokoppler getrieben wird, durch (klappt es nicht auf Anhieb, vertauschen Sie die zum TI führenden Anschlüsse!), die REMOTE-Leitung ist polar, als je ein Anschluß zu Kollektor und Emitter des Schalttransistors führt.

Nachfolgende Skizze gibt die Pinnnummern am neunpoligen Stecker wieder (seitenverkehrt zu den TI-Pinnummern!).

An Pin 1 liegt mit der zugehörigen Masseleitung auf Pin 3 der Mikrofon-Eingang, Pin 7 (Masse Pin 6) gehört zum externen Lautsprecher oder Kopfhörerausgang und Pin 5 stellt den Kollektor, Pin 4 den Emitter des Schalttransistors dar. (Alle Zählungen auf den Stecker bezogen!).



```

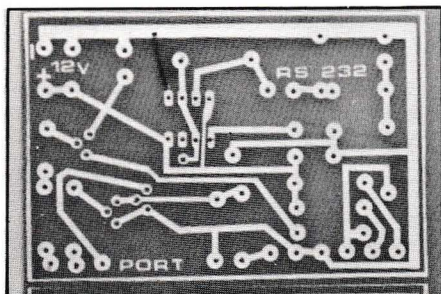
2980 CALL HCHAR(15,1,88)
2990 NEXT I
3000 GOTO 2750
3010 SCREEN=BAR
3020 FOR I=14 TO 18
3030 CALL HCHAR(15,1,96)
3040 NEXT I
3050 GOTO 2750
3060 REM ENG+
3070 IF ENG=-E4 THEN 3540
3080 IF ENG=-E3 THEN 3500
3090 IF ENG=-E2 THEN 3460
3100 IF ENG=-E1 THEN 3420
3110 IF ENG=0 THEN 3380
3120 IF ENG=E1 THEN 3340
3130 IF ENG=E2 THEN 3300
3140 IF ENG=E3 THEN 3270
3150 IF ENG=E4 THEN 3270
3160 REM ENG-
3170 IF ENG=E4 THEN 3300
3180 IF ENG=E3 THEN 3340
3190 IF ENG=E2 THEN 3380
3200 IF ENG=E1 THEN 3420
3210 IF ENG=0 THEN 3460
3220 IF ENG=-E1 THEN 3500
3230 IF ENG=-E2 THEN 3540
3240 IF ENG=-E3 THEN 3580
3250 IF ENG=-E4 THEN 3580
3260 REM TRIEBWERKE
3270 ENG=E4
3280 CALL HCHAR(19,9,96)
3290 GOTO 2750
3300 ENG=E3
3310 CALL HCHAR(19,9,88)
3320 CALL HCHAR(19,7,96)
3330 GOTO 2750
3340 ENG=E2
3350 CALL HCHAR(19,7,88)
3360 CALL HCHAR(19,5,96)
3370 GOTO 2750
3380 ENG=E1
3390 CALL HCHAR(19,5,88)
3400 CALL HCHAR(19,3,96)
3410 GOTO 2750
3420 ENG=0
3430 CALL HCHAR(19,3,88)
3440 CALL HCHAR(23,3,88)
3450 GOTO 2750
3460 ENG=-E1
3470 CALL HCHAR(23,3,96)
3480 CALL HCHAR(23,5,88)
3490 GOTO 2750
3500 ENG=-E2
3510 CALL HCHAR(23,5,96)
3520 CALL HCHAR(23,7,88)
3530 GOTO 2750
3540 ENG=-E3
3550 CALL HCHAR(23,7,96)
3560 CALL HCHAR(23,9,88)
3570 GOTO 2750
3580 ENG=-E4
3590 CALL HCHAR(23,9,96)
3600 GOTO 2750
3610 REM PILE
3620 IF PIL=0 THEN 3660
3630 PIL=0
3640 CALL HCHAR(10,18,88)
3650 GOTO 2750
3660 PIL=MEIL
3670 CALL HCHAR(10,18,96)
3680 GOTO 2750
3690 REM FEUERN
3700 GOSUB 2060
3710 PENG=PENG+INT(GUN/DIST)
3720 BANK=BANK-GUN
3730 IF PENG<50 THEN 3750
3740 GOSUB 2180
3750 GOTO 2750
3760 CALL SCREEN(10)
3770 GOSUB 3970
3780 CALL SOUND(500,-7,0)
3790 CALL COLOR(8,16,10)
3800 CALL CHAR(88,"04456A3C7
C3C5690")
3810 CALL COLOR(8,2,10)
3820 CALL CHAR(88,"0000183C3
C180000")
3830 CALL SOUND(500,-7,0)
3840 CALL COLOR(9,16,10)
3850 CALL CHAR(96,"04456A3C7
C3C5690")
3860 CALL COLOR(9,2,10)
3870 CALL CHAR(96,"0000183C3
C180000")
3880 FOR I=0 TO 2000
3890 NEXT I
3900 CALL CLEAR
3910 PRINT "          VERLORE
N"
3920 PRINT : : : : : : : :
: : :
3930 FOR I=0 TO 3000
3940 NEXT I
3950 CALL CLEAR
3960 END
3970 X=14
3980 Y=8
3990 STELLEN=5
4000 ZAHL$=STR$(INT(BANK))
4010 GOSUB 4200
4020 X=4
4030 Y=11
4040 STELLEN=6
4050 ZAHL$=STR$(INT(DIST))
4060 GOSUB 4200
4070 X=24
4080 Y=11
4090 STELLEN=5
4100 ZAHL$=STR$(INT(SPEED))
4110 GOSUB 4200
4120 X=25
4130 Y=14
4140 STELLEN=3
4150 ZAHL$=STR$(INT(PENG))
4160 GOSUB 4200
4170 IF SCREEN=0 THEN 4190
4180 SCREEN=BAR
4190 RETURN
4200 IF VAL(ZAHL$)<0 THEN 42
30
4210 CALL HCHAR(Y,X-1,32)
4220 GOTO 4250
4230 CALL HCHAR(Y,X-1,45)
4240 ZAHL$=STR$(ABS(VAL(ZAHL
$)))
4250 CALL HCHAR(Y,X,48,STELL
EN-LEN(ZAHL$))
4260 FOR I=1 TO LEN(ZAHL$)
4270 DX=X+STELLEN-LEN(ZAHL$)
+I-1
4280 SIGN=ASC(SEG$(ZAHL$,I,1
))
4290 CALL HCHAR(Y,DX,SIGN)
4300 NEXT I
4310 RETURN
4320 CALL SOUND(100,-1,0)
4330 CALL HCHAR(13,3,96)
4340 CALL HCHAR(13,3,88)
4350 CALL KEY(0,KEY,ST)
4360 IF KEY=13 THEN 2790 ELS
E 4320
4370 CALL CLEAR
4380 CALL COLOR(8,16,1)
4390 PRINT SCHIFF$:"
: : :
SIEG": : : : "ZIEL:
ROMULANISCHE RAUM": : " B
ASIS DER ULAT-KLASSE": : :
: : "ZIELABSCHUSS IN";T;"SEC
.": : :
4400 CALL SOUND(500,262,0)
4410 CALL SOUND(500,330,0)
4420 CALL SOUND(500,392,0)
4430 CALL SOUND(2000,330,0,3
92,0,524,0)
4440 FOR I=0 TO 6000
4450 NEXT I
4460 CALL CLEAR
4470 END
4480 CALL COLOR(13,10,5)
4490 CALL SOUND(300,784,3,-1
,2,880,3)
4500 CALL COLOR(13,5,5)
4510 RETURN

```



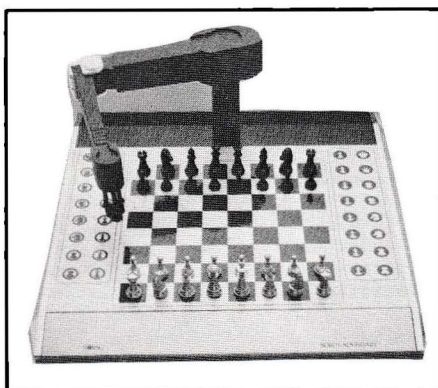
# HARDWARE: Sprites in Basic mit Minimem

V24-Schnittstelle am Joystick-Port Platine erhältlich!



Als Nachtrag zur Selbstbauanleitung der RS232/V24-Schnittstelle im TI99-Journal Nr.4, können wir Ihnen als Service professionell gefertigte **PLATINEN zur Selbstbestückung** anbieten. Damit haben wir die Anregung von zahlreichen Lesern aufgegriffen: Da zwar jeder technisch Interessierte eine so einfache Platine bestücken, aber die wenigsten Platinen, ohne großen Aufwand, herstellen können.  
**BEZUG:** TI99-JOURNAL  
**PREIS:** 6S 125,- (inkl. Versand)  
**ACHTUNG:** Es können nur **SPESEFREIE** Einzahlungen berücksichtigt werden!

## Schachroboter mit Greifarm



Mit modernster Technologie ausgestattet, bietet ein führender Schachcomputer-Hersteller ein revolutionäres Gerät an: den Schachroboter mit Greifarm. Wenn Sie Ihre Figur (ohne Druck!) von Feld zu Feld bewegen, ist dieser Zug augenblicklich registriert. Dann greift der Arm des Roboters seine Figur und setzt sie auf die neue Position. Das Gerät fasziniert durch ein enorm starkes, 32K Schachprogramm, eine Eröffnungsbibliothek von 5500 Halbzügen und eine große Anzahl von Spezialfunktionen. Als netten Gag findet man eine EMOTION-Taste: nun zeigt der Roboter auch Freude oder Ärger durch wildes Geklapper mit dem Greifarm und ebensolchen Tönen. Für nähere Auskünfte steht Ihnen Hr. Fiedler gerne zur Verfügung.

## SPRITES IN BASIC MIT MINIMEM

Auch ohne Maschinenprogramme sind Sprites am Minimem möglich, eine Eigenheit, die manchen am XBasicmodul lieb gewonnenen Befehl auch hier zur Verfügung stellt.

Sprites werden durch Poken ins VDP-RAM möglich, wobei bis zu drei ruhende Sprites zulässig sind, die aber durch ein Basicprogramm (Neudefinition an pixelbenachbarter Stelle) bewegt werden können.

Der Pokebefehl muß dabei folgenden Aufbau haben:

CALL POKEV(SADR,YP,XP,CH,FB,200)

Die Spriteadresse SADR ist dabei  
768 für Sprite#1  
772 für Sprite#2  
776 für Sprite#3

YP ist die um eins verminderte Pixelreihe, in der das linke obere Ende des Sprites liegt. 0<=YP<191 ist erlaubt.

XP ist die um eins verminderte Pixelspalte. 0<=XP<255

CH ist der um den Screenoffset 96 vermehrte ASCII-Code des abzubildenden Zeichens. 0<=CH<255 ist zulässig, nur werden Zeichen mit Code <126 selten sichtbar sein.

FB ist das um eins verminderte Farbbyte des Sprites, wobei der Farbcode den üblichen Basicwerten entspricht. 0<=FB<15.

Eine kurze Übersicht über die Steuerbefehle:

CALL MAGNIFY(n) ...  
CALL LOAD(-31788,223+n)  
CALL DELSPRITE(ALL) ...  
CALL LOAD(768,208)

CALL COINC(ALL) kann durch PEEKen von -31777 und Vergleichen, ob das Coincidenzbit gesetzt ist, realisiert werden, wie bei MAGNIFY zu beachten ist, daß ein CALL KEY folgen muß, da nur dann der Wert von >B3D4 ins VDP-Register übernommen wird.

Nachfolgend ein kleines Demoprogramm für Sprites am Minimem:

```
100 REM *****
110 REM * SPRITES IN BASIC *
120 REM * UND MINI-MEMORY *
130 REM *****
140 REM * VON REIDLINGER *
150 REM * BUCHENWEG 18 *
160 REM * 2020 HOLLABRUNN *
170 REM *****
180 REM * ERGAENZT VON *
190 REM * KARL HAGENBUCHNER*
200 REM *****
210 CALL CLEAR
220 CALL SCREEN(16)
230 CALL CLEAR
240 PRINT "MAGNIFIED(1)
    MAGNIFIED(2)
    MAGNIFIED(3)
    MAGNIFIED(4)"
250 INPUT "SIZE: ":K
260 IF (K<1)+(K>4)=-1 THEN 2
30
270 ON K GOTO 280,300,320,34
0
280 CALL LOAD(-31788,224)
290 GOTO 350
300 CALL LOAD(-31788,225)
310 GOTO 350
320 CALL LOAD(-31788,226)
330 GOTO 350
340 CALL LOAD(-31788,227)
350 PRINT " PRESS ANY KEY
    TO START "
360 CALL KEY(0,K,S)
370 IF S=0 THEN 360
380 CALL CLEAR
390 F=1
400 D=96
410 REM 96 ZU ASCII-CODE ADD
    IEREN
420 Z=0+33
430 REM SPRITE #1
440 CALL POKEV(768,0,8,Z,F,2
    08)
450 REM SPRITE #2
460 CALL POKEV(772,90,90,Z+1
    ,F,208)
470 REM SPRITE #3
500 REM SPRITE #1 POSITION A
    ENDERN
510 CALL POKEV(768,K,K)
520 F=F+1
530 IF F<16 THEN 560
540 F=1
550 REM SPRITE #2 FARBE AEND
    ERN
560 CALL POKEV(775,F)
570 Z=Z+1
580 IF Z<0+127 THEN 510
590 Z=0+33
600 REM SPRITE #3 ZEICHENCOD
    E AENDERN
610 CALL POKEV(778,Z)
620 NEXT K
630 GOTO 210
```



## Der prämierte Tip Der prämierte Tip Der prämierte Tip

### TI-JOYSTICKS (Hardware-Tip für Jedermann!)

Kennen Sie das? Sie haben bei Parsec soeben 'level 6' erreicht oder Ihrem Fußballgegner trotz starker Gegenwehr gerade das 7. Tor in die linke Ecke geschossen, als das Hochgefühl in Ihrer Brust von einem weniger angenehmen Gefühl in Ihrem Daumen übertönt wird. Wenn dann nach dem 8. Tor oder bei Level 7 der Krampf so arg wird, daß Sie ans Aufgeben denken, dann haben Sie drei Möglichkeiten:

1. Sie gehen zum Orthopäden (der verschreibt Ihnen Einlagen, das tut er immer).
2. Sie hören mit diesen infantilen Spielereien auf und wenden sich hochgeistigem Programmieren zu.
3. Sie lesen diesen Artikel zu Ende.

Da Sie sich für Punkt drei entschieden haben, werden Sie in Kürze Ihre Joysticks so umbauen können, daß deren Bedienung ein Langzeitvergnügen wird.

Benötigtes Werkzeug und Material:

- 1 Schere
- 1 Kreuzschlitzschraubendreher
- 1 Stück Moosgummi (rund, Durchmesser ca. 4 cm; erhältlich z.B. bei Franz Nuschei KG, A-1071 WIEN, Neubaug. 31. Preis ca. s 25,- pro Meter).

Zuerst schneiden Sie sich vom Moosgummi 4 gleichlange Stücke von 7 - 10 mm Länge ab. Die Schere können Sie nun bereits wieder ins Nähkästchen Ihrer Gattin oder Mutter zurücklegen; Sie brauchen nur mehr den Schraubendreher. Während der ganzen folgenden Arbeit halten Sie den Joystick verkehrt herum, das heißt, Steuerknüppel nach unten. Sie lösen nun die beiden Schrauben und haben den Deckel nach oben ab (keine Sorge, es fliegt keine Feder oder ähnliches davon). Wenn Sie sich an dem recht einfachen Innenleben des Joysticks sattgesehen haben, nehmen Sie den Steuerknüppel und dann den darunterliegenden Gummiring aus der, in Ihrer Hand ruhenden, Gehäusehälfte. Dort wo der Ring war, sehen Sie nun eine durchsichtige Folie mit silbrigen Leiterbahnen links, rechts, oben und unten. Auf diese Leiterbahnen stellen Sie die vorbereiteten Moosgummistücke (bei 12 mm Durchmesser klemmen Sie ein wenig, wodurch sie beim Zusammenbau des Gehäuses kaum verrutschen können) und setzen dann den Steuerknüppel wieder so ein, daß die Wülste auf seiner Auflageseite auf dem Moosgummi liegen. Nun setzen Sie den Deckel so vorsichtig auf, daß der Moosgummi nicht verrutscht, drücken die beiden Gehäusehälften zusammen und drehen die Schrauben wieder ein. Probieren Sie den Schraubenansatz so lange, bis Sie ins bereits vorhandene Gewinde hinein finden, denn

Kunststoffgewinde lieben es gar nicht, immer wieder neu geschnitten zu werden! Und nun viele Stunden vergnügliches Spiel!

**ACHTUNG:** Wir prämiieren auch Ihren Tip. Diesen Tip sandte uns Herr Bader aus A-1030 WIEN. Die Redaktion hat diesen Brief zum Tip des Heftes gewählt! Herr Bader erhält als Anerkennungsprämie einen Kugelschreiber mit Quarzuhr.

## Leser-Ecke

Liebe Ti-Fans!

Immer wieder wurden wir gebeten, im TI99-Journal auch eine persönliche Seite für und von dem Leser zu bringen. In dieser Seite sollten sowohl eventuell auftretende Schwierigkeiten betreffend die Besorgung von Hardware als auch die tätige Hilfe für Software-Fragen behandelt werden. Nun da ist sie: Die spezielle Leserseite für Ihre Probleme.

## BESTELLSCHEIN

Ich bestelle ein TI-99 Journal-ABO (6 Ausgaben) zum  
Preis von öS 462,— BRD, Schweiz  
sonstige Länder auf Anfrage  
öS 420,— für Österreich (inkl. Porto und Verpackung)

Lieferung ab Nummer (Ausgabe): .....

- ☐ ich habe einen Scheck beigelegt (Bitte, **nur** für Österreich)
- ☐ ich habe den Betrag am ..... auf das PSK-Konto WIEN 6.333.544 überwiesen. (ACHTUNG: Es können nur SPESENFREIE Überweisungen berücksichtigt werden!)

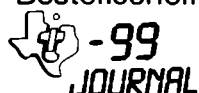
Absender: Name .....

Anschrift .....

Tel.: .....

Ich besitze folgende Geräte: .....

Bestellschein in Kuvert stecken und senden an:



Thomas M. Fiedler

**A-1150 WIEN, FELBERSTRASSE 24/26**  
**Tel. 0222/92 29 033 (10—13, 19—20 Uhr)**

Zögern Sie nicht, uns Ihre Wünsche und Anregungen zu schreiben. Bei allgemein interessanten Fragen werden wir diese, samt Antwort, gerne veröffentlichen. Durch die plötzliche Ausweitung unseres Clubs ist es uns kaum mehr möglich, die Briefflut individuell zu beantworten. Trotzdem wird selbstverständlich jeder Brief von uns gelesen und in unserer Redaktion thematisch berücksichtigt. Allgemein gültige Probleme, die nicht in einem der nächsten Hefte besprochen werden, werden wir dann in dieser speziellen Leser-Ecke näher behandeln.

Das bringt uns zu der Frage der überlangen Zeilen. Ich zitiere hier auszugsweise einen Brief von Herrn Ing. Schieb aus Köln: zustandekommen. Für einen allgemeinen Hinweis wäre ich dankbar, da ich alle anderen Möglichkeiten anhand von Ausdrucken mehrfach geprüft habe... Ihre Programme im Journal wie in der Sammlung sind hervorragend... Eine Kritik an der Politik Ihres Hauses möchte ich aber noch anmelden. Ich finde es nicht fair, daß die guten Programme Ihres Journals derart massiv mehrfach veröffentlicht werden, wie z.B. im neuen 'Chip Sonderband'. Und das ansonsten seriöse 'Chip'

versteigt sich zu dem Versprechen, bisher nie veröffentlichte Programme zu bieten. P.S. Läßt sich Ihr Cassettenservice nicht professionalisieren? 30 - 50% der gegenwärtigen Preise halte ich für angemessen. Die Nachfrage dürfte dann auch 'explodieren'. Red.: Wir danken für Ihre Vorschläge und haben einen guten Teil davon bereits verwirklicht. Zunächst zu den 'zusätzlichen' Zeilen. Sie haben sowohl in Ti-Basic, als auch in XBasic die Möglichkeit, Zeilen zu verlängern, aber nur in einem nicht vorhersagbaren Ausmaß. Um im XBasic-Modus eine 6. Zeile anzufügen, tippen Sie bitte zunächst volle 5 Bildschirmzeilen ein. Dann <Enter>. Anschließend holen Sie die Zeile mit <Fctn 8> (Redo) wieder und bringen den Cursor zum Ende der Zeile. Sie können nun mindestens eine, oft sogar noch 3 zusätzliche Zeilen eintippen. Warum? Nun, mehr Speicherausnutzung, höhere Geschwindigkeit beim Programmablauf und bequemere Codierung umständlicher Algorithmen. Bitte verzeihen Sie uns im Namen unserer Leser diese bislang fehlende Information. Wir haben angenommen, dieser Trick wäre schon Allgemeingut! In Ti-Basic werden die erlaubten 4(!) Zeilen eingetippt. Dann wieder <Enter>. Anschließend wird die Zeile zum Editieren durch Eintippen der Zeilennummer und <Fctn X> wieder

auf den Bildschirm geholt und da haben Sie Ihre zusätzliche Zeile! Wußten Sie übrigens, daß auch in Ti-Basic statt des unständlichen Befehles 'EDIT' einfach <FCTN X> verwendet werden kann? (Steht in keinem Basic-Manual) Betreffend die Doppelveröffentlichung haben wir bereits im 'Spezial' darauf hingewiesen, daß eine eventuelle Veröffentlichung einzelner Programme in der in Kürze erscheinenden Chip Programmsammlung zu erwarten ist. Die Aussage in der o.a. Zeitschrift entstammt nicht unserer Redaktion. Zum Cassettenservice haben wir für unsere Leser erfreuliche Neuigkeiten! Auf allgemeine Anregung, auch im Sinne Ihres Schreibens, konnten wir, durch bessere Organisation, die Preise für Software auf Kassette oder Disk auf das von Ihnen vorgeschlagene Niveau senken! Lesen Sie dazu bitte unter Hinweise im Inneren dieses Journals.

Herr Paaperl, Wien, schreibt uns zu unserem letzten regionalen Clubtreffen: Ein voller Erfolg! Mann ist das nächste. Vorschlag von mir wäre, sich bei jedem Treff ein Hauptthema zu stellen. Dieses kann dann jeder Interessierte dazu etwas beitragen. (Programmiertricks, Assemblerschule, Hardwarebesprechung etc.) Fixpunkte: Imbiß, Kennenlernen, Programmtauschen. Neuigkeiten vorführen. "Fragestunde" sollten beibehalten werden! Ich bin wieder dabei. Das ganze war die S 30,- wert!

Weiße man schon näheres über eine Sammelbestellung von Herrn Becherers Maschinenprogrammen? Red.: Wir freuen uns, daß unser Clubtreffen so großen Anklang gefunden hat und werden uns auch in Zukunft bemühen, regelmäßige Treffen so attraktiv wie möglich zu gestalten. Clubmitglieder ernennen nähere Einzelheiten ihren Mitteilungsblättern. Die Maschinenspiele von Herrn Becherer sind bereits über den Club erhältlich. Näheres entnehmen Sie bitte der Infoseite im Blattinneren.

Herr Dr. Apflauer, Pöchlarn, sandte uns folgenden Tip zum Programm Sprite-Film aus dem 'Chip'-Spezial: Ich halte den Erfolg für verblüffend.  
550 CALL PATTERN(0,T,A)::NEXT T::NEXT A  
555 CALL MAGNIFY(INT(RND\*4)+1)  
557 CALL SCREEN(INT(RND)+1)  
560 FLI=INT((3\*RND)+1)::IF FLI=2 THEN 270


Den zahlreichen Lesern, die besorgt auf das verzögerte Erscheinen des Journals reagierten und schriftlich bei uns angefragt haben, ob wir ihre Bestellung erhalten hätten, danken wir für die bewiesene Geduld. Wie sie an der gegenwärtigen Ausgabe erkennen können, hat sich das Warten gelohnt. Die Leserbriefe, die wir aus Platzgründen nicht veröffentlichen konnten, werden wir uns bemühen, in den folgenden Ausgaben themenzentriert zu behandeln.

MICROCOMPUTER  
**Opesoft**  
SOFTWARE

Dr. A. PESCHETZ - Neubergsiedlung 26  
A-2431 ENZERSDORF/FISCHA - ÖSTERREICH

Die PROFISOFTWARE für den  
**TI-99/4A**

Nach heute GRATISINFO anfordern!



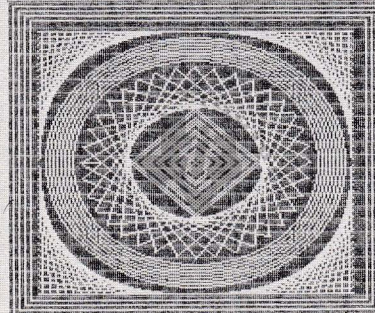
GRAPHIKPAKETE in TMS9900 ASSEMBLER:

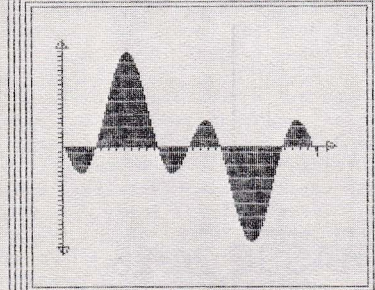
- EXPANDED GRAFIC BASIC (APE003/C, APE003/D)
- TI GRAFIC PLOTTER (APE004/D)
- GRAFIC GENERATOR/INTERPRETER (APE009/D)

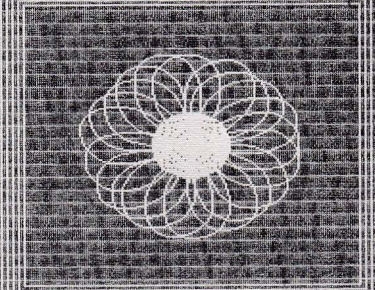
Preise: Diskette (D) 6S 1810,- DM 220,-  
(incl. MWST) Kassette (C) 6S 1610,- DM 195,-

Versand nur bei Nachnahme (+ Porto u. Verp.)!

Diese Anzeige wurde mit dem OPESOFT  
Graphikpaket TI GRAFIC PLOTTER kreiert.







# BASIC-KURS

## (LETZTER TEIL)

### LÖSUNGSVORSCHLÄGE zu BASIC-KURS NR4

Im letzten Teil des BASIC-Kurses im TI99-Journal Nr.4 haben wir Ihnen abschließend drei Aufgaben vorgeschlagen. Hier finden Sie nun deren mögliche Lösungen: (Anfragen richten Sie bitte an Hr.Berger.)

#### TATZL

```
100 REM DIESE DREI KURZPROGR
AMME GEBEN NUR MOEGELICHE LOE
SUNGSVARIANTEN. UND SIND DAH
ER REIN EXEMPLARISCH!!!
110 REM LOESUNG FUER 1.BEISP
IEL "BASIC-KURS IV"
120 CALL CLEAR
130 PRINT "EINGABE VON JAHRE
SZAHLEN ZWISCHEN 1901 UND
2099!"
140 INPUT "?":A
150 IF A<1901 THEN 210
160 IF A>2099 THEN 210
170 IF A-INT(A)<>0 THEN 210
180 ZA=ZA+1
190 PRINT ZA:".EINGABE"
200 GOTO 140
210 PRINT "FALSCH EINGABE"
220 GOTO 130
230 REM VON ALEXANDER BERGER
```

#### TATZL1

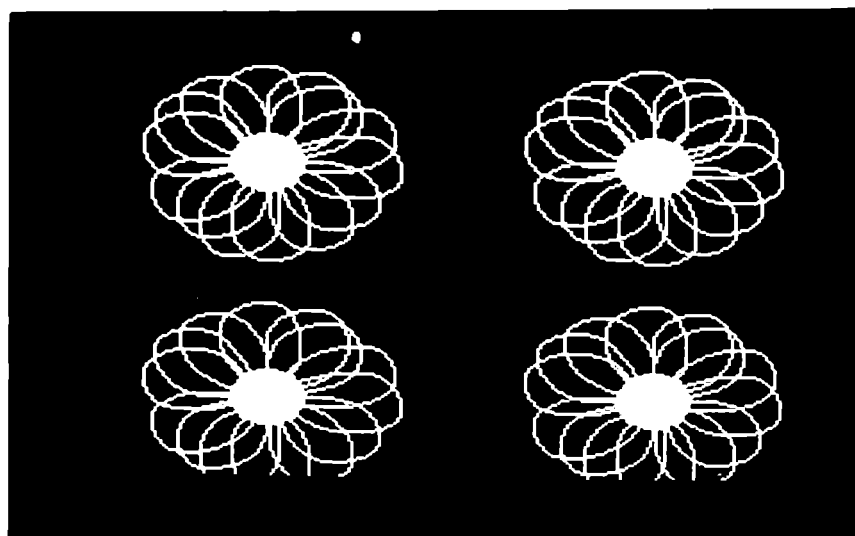
```
100 REM LOESUNG FUER 2.BEIS
PIEL "BASIC-KURS IV"
110 CALL CLEAR
120 PRINT "EINGABE VON ZAHLE
N UND UNTERSCHIEDUN
G IN GERADE, UNGERADE
UND DEZIMALZAHLEN"
130 INPUT "?":A
140 IF A-INT(A)<>0 THEN 160
150 IF A/2=INT(A/2) THEN 180
ELSE 200
160 PRINT "DEZIMALZAHL"
170 GOTO 130
180 PRINT "GERADE ZAHL"
190 GOTO 130
200 PRINT "UNGERADE ZAHL"
210 GOTO 130
220 REM VON ALEXANDER BERGER
```

#### TATZL2

```
100 REM LOESUNG FUER 3.BEI
SPIEL "BASIC-KURS IV"
110 CALL CLEAR
120 PRINT "EINGABE VON POSIT
```

```
IVEN ZAHLENUND ANGABE DER AN
ZAHLE DER STELLEN"
130 INPUT "?":A
140 IF A-INT(A)<>0 THEN 160
150 IF ABS(A)<>A THEN 160 EL
SE 200
160 PRINT "KEINE DEZIMALZAH
EN"
```

```
170 GOTO 130
180 PRINT "KEINE NEGATIVEN Z
AHLEN"
190 GOTO 130
200 A$=STR$(A)
210 L=LEN(A$)
220 PRINT "DIE EINGEBENE ZAH
L IST ":L:"-STELLIG."
230 GOTO 130
240 REM VON ALEXANDER BERGER
```



### Apesoft-Programme

DIESE LEITUNGSFAHIGEN PROGRAMME ERWEITERN IHREN TI99/4A UM BIS ZU 38 AUSGEFEILTE GRAFIKBEFEHLE. DIE GRAFIKGENERIERUNG ARBEITET SUPERSCHNELL: DER VDP-CHIP WIRD DIREKT ÜBER DEN 16BIT-PROZESSOR ANGESPROCHEN.

#### EXPANDED GRAFIK-BASIC:

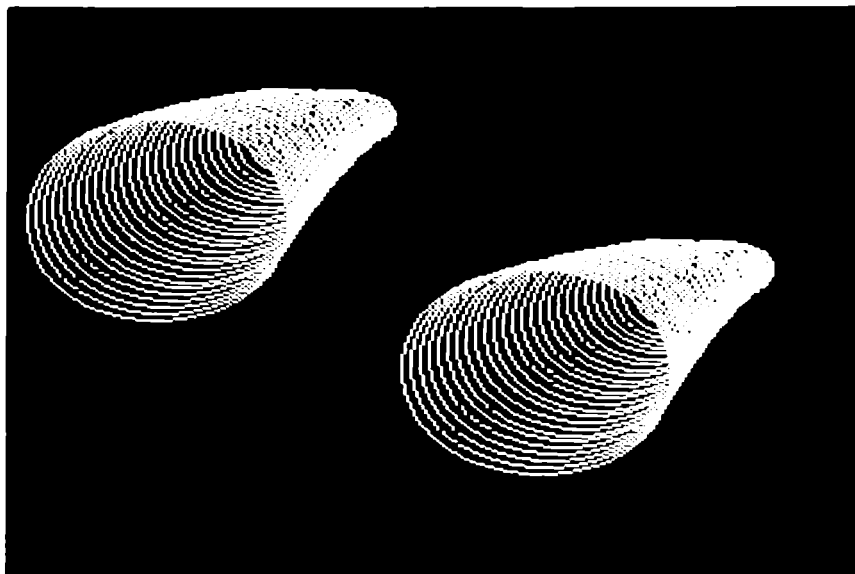
Bis zu 38 Grafikbefehle, wie MOVETO, TURN, ELLIPS, AXIS, CIRCLE, stehen Ihnen für Ihre Programme in BASIC

bzw. XBASIC zur Verfügung. ZUR ANWENDUNG: DIE BEFEHLE WERDEN IN DAS BASIC/XBASIC-PRGR. EINFACH WIE BASIC-BEFEHLE VERWENDET.

Kassettenvers.: MINIMEM  
Diskvers.: XBASIC (od.ASS od. MINIMEM). 32K-Erweiterung  
**BEZUG: TI99-JOURNAL-KLUB**  
Felberstraße 24  
A-1150 WIEN

**PREISE:** 6S 1810.- (Diskvers.)  
6S 1610.- (Kass.vers.)  
(+ 6S 25.- Versandkosten)

**ACHTUNG:** Es können nur SPESENFREIE Überweisungen berücksichtigt werden.





# HINWEISE

## Kassettenroutine

Sie haben auch nach dem Erscheinen der Kass.anweisungen auf dem Bildschirm noch die Möglichkeit aus der Rekorderroutine auszusteigen: Drücken Sie einfach **Taste E!** Das ermöglicht Ihnen nach versehentlich eingegebenen OLD oder SAVE einen problemlosen Ausstieg. Erst wenn **RECORDING** bzw. **READING** erscheint, bleibt Ihnen nur mehr die Möglichkeit auf die Fehlermeldung zu warten.

## Abgestürzte Disketten

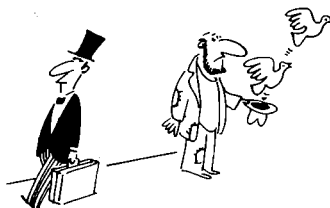
Ist es Ihnen vielleicht schon einmal passiert, daß Sie beim Einleseversuch einer bespielten Diskette die Fehlermeldung "DISKETTE NICHT INITIALISIERT" erhalten haben? Oder stimmte die Anzahl der belegten Sektoren im Disk-katalog nicht mit der tatsächlichen Anzahl überein? Nun, da ist irgendetwas auf Ihrer Disk durcheinander geraten. Wenn Sie diese oder ähnliche Probleme mit einer Disk haben, verzweifeln Sie nicht, sondern wenden Sie sich einfach an uns. **Wir retten (fast) jede verwirrte Diskette.** Um jedoch sich und uns unnötige Arbeit zu ersparen, rufen Sie zuerst einmal an (Hr. Fiedler) und beschreiben den aufgetretenen Fehler. Vielleicht findet sich bereits auf diesem Weg eine Lösung.

## Regionale Clubs

**ACHTUNG!** Wir unterstützen gerne regionale TI99-Klubs und veröffentlichen Ihre Termine für Veranstaltungen und Treffs. Zu den Wiener Klubtreffen sind auch Klubmitglieder aus den Bundesländern und S-Deutschland angereist. Dies ist nicht unbedingt notwendig, da in fast allen grösseren Städten in Deutschland und Österreich genügend TI99-User und TI99-Journal-Leser vorhanden sind, um eigene, regionale Klubabende zu veranstalten. Falls Sie Interesse haben, so einen regionalen Treffpunkt zu gründen, setzen Sie sich mit uns in Verbindung. Wir stehen Ihnen gerne mit Rat und Tat zur Seite, bzw. kommen auch gerne, um den ein oder anderen Klubabend bei Ihnen zu gestalten. Ihre Anregungen, usw. richten Sie bitte an Hr. Reitingen.

## Neuerscheinung:

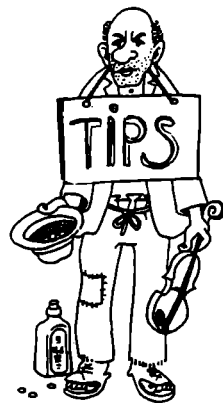
Der zweite Band des bei TI erschienen Programmierhandbuches des bekannten Autorenteam **APESOFT** ist bereits erhältlich. Auf knapp **500 Seiten** werden ausgehend von einfachen, spielerischen Programmen die grafischen, wissenschaftlichen und musikalischen Möglichkeiten des TI99 eindrucksvoll verdeutlicht. Dabei wird an Hand einer Vielzahl von Programmen in APESOFT-Qualität, durch fortgeschrittene Techniken der Programmierung in BASIC, XBASIC, ASSEMBLER und PASCAL geführt. Dieses **99 Special II** ist eine Erweiterung Ihres Systems und wird vom TI99-Journal uneingeschränkt empfohlen. **BEZUG:** Fachbuchhandel



# SUPERTIP

## Disk-Schutz

Bisher hatten Sie nur die Möglichkeit Ihre Programme durch die PROTECTED Funktion vor unerlaubtem Zugriff zu schützen. Die Vervielfältigung der Programme oder Datenfiles mit dem Diskmanager Modul konnte aber nicht verhindert werden. Ab sofort können Sie aber auch den "Raubkopierern" Einhalt gebieten. Die dazu notwendige Funktion ist im Diskmanager-Modul mit enthalten, wurde aber von TI nicht weiter dokumentiert. Was ist also zu tun? Ganz einfach: Sie nehmen eine leere Diskette (auf die Sie nach erfolgtem Schutz die Programme oder Datenfelder abspeichern möchten) und wählen die Funktion "Diskette initialisieren". Nun drücken Sie bei niedergehaltener **FCFN-Taste zehn Mal die X-Taste.** Sie hören einen tiefen Signalton und damit ist die Diskette kopiergeschützt. Nun können Sie wie gewohnt weitermachen. **ACHTUNG:** Diese Funktion bleibt bis zur neuen Wahl des Diskmanagers eingeschaltet!



## Software-Service für

## Klubmitglieder

Wie bereits in der "Leserecke" angekündigt, haben wir das Software Service (für Programme auf Kassette oder Diskette) durch gezielte Rationalisierung um **50% gesenkt.** Ab sofort beträgt der Preis für "jedes weitere KB" nur mehr **öS 10,-.** Ebenso ist der Grundpreis reduziert. Die neuen Preise gelten selbstverständlich auch für alle bisher erschienen Softwarelisten. Näheres entnehmen Sie, bitte, den. Ihrem Journal beiliegenden Klubmittlungsblättern!

## Kleinanzeigen

**SUPER GRAFIK-PAKET in ASSEMBLER**  
30 BASIC (XBASIC)-Befehle stehen zur Verfügung. Bereits mit dem MINIMEM-Modul einsetzbar!!!  
**BEZUG:** TI99-JOURNAL-KLUB  
**PREIS:** Kassette öS 1.610,-  
Diskette öS 1.810,-

**ASSEMBLER-HANDBUCH in DEUTSCH**  
Wir bieten Ihnen das umfangreichste deutschsprachige Handbuch für den TMS9900-Assembler des MINI MEMORY MODULS.  
(Siehe auch unter BUCHTIPS!!!)  
**BEZUG:** TI99-JOURNAL-KLUB  
**PREIS:** s. BUCHTIP in d. JOURNAL

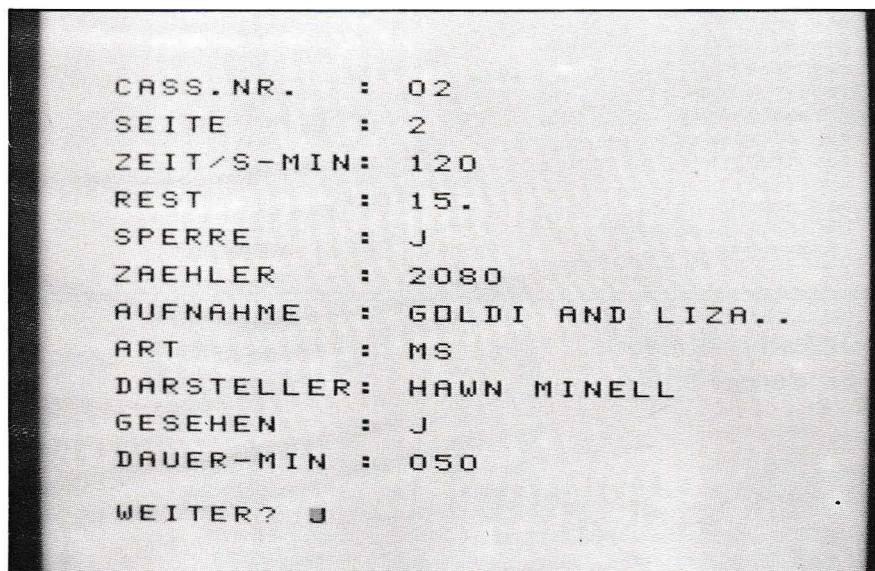
### VERKAUFE:

**Verkaufe** TI99/44+alle Erweiterungen+Software kompl. od.einzeln Angebote an:  
Dr. Gotfryd, Frankenweg 52,  
D-6236 ESCHBORN 2

**Box** mit Floppy,32k,P-Code,Pascal Miniem, Assembler, Ext.Schnittstelle, etc. **06/02624/3817**

**ASSEMBLERPROGRAMM: PIXELGRAFIK** für MINIMEMORY. Zeichnet Gerade durch 1 Basicbefehl uä öS 300,-. BASIC-ASSEMBLER/DISASSEMBLER für MINIMEMORY. 50 Label. Verschieben von Programmen usw. öS 140,- Info und weitere Programme bei Herwig Reidlinger, Buchenweg 16 A-2020 HOLLABRUNN (+ Versandk.)

# Videothek



Dieses Programm ermöglicht Ihnen das Verwalten Ihrer Videobänder auf Diskette. Das Grundmenü zeigt Ihnen die 5 Möglichkeiten, die zur Verfügung stehen:

- 1 NEU - Satz- und Seitennummer wird Ihnen vorgegeben.
- 2 ANDERN - eines Datensatzes
- 3 AUSWAHLEN - Dieser Punkt ist das eigentliche Herzstück des Programmes. Hier können sämtliche Daten, die in der Datei gespeichert sind abgefragt werden. D. h., Sie suchen ein Band mit einer Restlaufzeit von mind. 25 Minuten. Dann werden Ihnen alle Kassettenseiten, die 25 Minuten und mehr Restlaufzeit (wird aufgrund der vorhandenen Daten vom Programm errechnet) haben, der Reihe nach angezeigt.
- 4 ANSEHEN - wenn Sie sich eine Seite gezielt ansehen wollen.

5 ENDE

Bei NEU können Sie folgende Daten eingeben (die Länge der Eingabe ist Ihnen durch Punkte vorgegeben):

CASS.NR. - wird vorgegeben  
SEITE - wird vorgegeben  
ZEIT/S-MIN - Zeit pro Seite in Min.  
REST - wird errechnet  
SPERRE - J oder N, je nach dem, ob Kassette gesperrt für weitere Aufnahmen oder nicht.  
ZAEHLER - Zählerstand = Beginn der Aufnahme  
AUFNAHME - Titel  
ART - Hier steht es Ihnen frei, zwei Buchstaben Ihrer Wahl - zur näheren Bezeichnung Ihres Filmes (L-lustig, K-Krimi, usw.).

DARSTELLER  
GESEHEN - J/N  
DAUER-MIN

Von ZAEHLER bis DAUER-MIN können Sie pro Seite bis zu vier Sendungen speichern.

Da das Programm für VIDEO 2000 geschrieben worden ist, müssen Sie für BETA- oder VHS-System folgendes ändern:  
140 DEF SNR=VAL(T\$(1))-1  
360 T\$(2)="1" : IF VAL(T\$)<10....  
420 Sie müssen hier die Spielzeit pro Seite eingeben, die Ihrem System entsprechen.

Christian Roch

## VIDEOTHEK

```

100 REM VIDEOTHEK VIDEO 2000
      VON CHRISTIAN ROCH
110 OPTION BASE 1
120 DIM T$(29),TG$(5),TC$(11),TL(29)
130 ON ERROR 940
140 DEF SNR=VAL(T$(1)&T$(2))-
      (VAL(T$(1))*8+3):ERRECHN. S
      ATZNR.
150 OPEN #1:"DSK1.VIDEODATEN",
      RELATIVE, FIXED(158)
160 DATA "1 NEU","2 AENDERN",
      "3 AUSWAHLEN","4 ANSEHEN",
      "5 ENDE","CASS.NR. : ..","S
      EITE : ..","ZEIT/S-MIN: .
  
```

```

..","REST : "
170 GOTO 190 : A,A$,A1$,B,B
$,C,C$,I,I1,I2,J,J1,L,M,M1,T
,Z$ : CALL ERR : CALL CLEA
R : CALL HCHAR
180 'SP-
190 DATA "SPERRE : N","ZA
EHLE : 0000","AUFNAHME :
.....","ART
: ..","DARSTELLER: .....
...."
200 DATA "GESEHEN : N","DA
UER-MIN : 000",2,1,3,3,1,4,1
6,2,11,1,3,4,16,2,11,1,3,4,1
6,2,11,1,3,4,16,2,11,1,3
210 FOR I=1 TO 5 : READ TG$(
I):: NEXT I : FOR I=1 TO 1
1 : READ TC$(I):: NEXT I :
FOR I=1 TO 29 : READ TL(I)
: NEXT I
220 M=3 : M1=8
230 CALL CLEAR
240 J=7 : FOR I=1 TO 5 : D
ISPLAY AT(J,9):TG$(I):: J=J+
2 : NEXT I
250 DISPLAY AT(24,1):"MENUE?"
:M
260 ACCEPT AT(24,9)SIZE(-1)V
ALIDATE(DIGIT)BEEP:M
270 IF M<1 OR M>5 THEN 260
280 RESTORE #1
290 CALL CLEAR
300 ON M GOTO 310,570,790,57
0,1090
310 REM *** NEU ***
320 T$(1)=" " : T$(4),T$(11
),T$(17),T$(23),T$(29)="000"
330 IF EOF(1)=0 THEN LINPUT
#1:Z$ : GOTO 330
340 IF REC(1)=0 THEN T$(1).T
$(2)="0" : GOTO 360
350 T$(1)=SEG$(Z$,1,2):: T$(
2)=SEG$(Z$,3,1)
360 IF T$(2)="1" THEN T$(2)=
"2" : T$(3)=SEG$(Z$,4,3)ELS
E IF VAL(T$(1))<10 THEN T$(1
)="0"&STR$(VAL(T$(1))+1):: T
$(2)="1" ELSE T$(1)=STR$(VAL
(T$(1))+1)
370 J=1 : FOR I=1 TO 11
380 DISPLAY AT(J,1):TC$(I)
390 IF I=1 OR I=2 OR(I=3 AND
T$(2)="2")THEN DISPLAY AT(J
,13):T$(I)
400 J=J+2 : NEXT I
410 IF T$(2)="1" THEN ACCEPT
AT(5,13)VALIDATE(DIGIT)SIZE
(-3)BEEP:T$(3)
420 IF T$(3)<>"060" AND T$(3
)<>"120" AND T$(3)<>"180" AN
D T$(3)<>"240" THEN 410
430 ACCEPT AT(9,13)VALIDATE(
"JN")SIZE(-1)BEEP:T$(5)
440 FOR I=1 TO 4 : GOSUB 92
0
450 DISPLAY AT(24,1):"WEITER
? J" : ACCEPT AT(24,9)VALID
ATE("JN")SIZE(-1)BEEP:A$
460 IF A$="N" THEN 480 ELSE
CALL HCHAR(24,3,32,28)
470 NEXT I
480 DISPLAY ERASE ALL AT(22,
1):"SPEICHERN/LISTEN/ENDE"
490 DISPLAY AT(24,1):"S/L/E?"
L : ACCEPT AT(24,8)SIZE(-
1)VALIDATE("SLE")BEEP:A1$
500 IF A1$<>"S" THEN 510
510 FOR I=1 TO 29 : IF LEN(
T$(I))<TL(I)THEN T$(I)=T$(I)
&RPT$(,"",TL(I)-LEN(T$(I)))
  
```



```

)
520 NEXT I
530 PRINT #1,REC SNR:T$(1);
540 FOR I=2 TO 29 :: PRINT #
1:T$(I):: NEXT I
550 IF A1$<>"L" THEN T$(1),T
$(8)=" " :: T$(2),T$(5),T$(
10)=" " :: T$(3),T$(4),T$(11
)=" " :: T$(6)=" " :: T
$(7),T$(9)=RPT$(" ",16):: GO
TO 230
560 GOSUB 660 :: GOTO 480
570 DISPLAY AT(1,1):TC$(1)
580 DISPLAY AT(3,1):TC$(2)
590 ACCEPT AT(1,13)VALIDATE(
DIGIT)SIZE(-2)BEEP:T$(1)
600 ACCEPT AT(3,13)VALIDATE(
"12")SIZE(-1)BEEP:T$(2)
610 LINPUT #1,REC SNR:Z$
620 GOSUB 640 :: IF M=4 THEN
230 ELSE 480
630 !ZUORDNUNG
640 L=1 :: FOR I=1 TO 29 ::
T$(I)=SEG$(Z$,L,TL(I)):: L=L
+TL(I):: NEXT I
650 !ANZEIGE
660 J=1 :: FOR I=1 TO 11 ::
DISPLAY AT(J,1):TC$(I):: DIS
PLAY AT(J,13):T$(I):: J=J+2
670 NEXT I
680 IF M=2 AND A1$<>"L" THEN
I=1 :: GOSUB 950
690 FOR I=2 TO 4
700 DISPLAY AT(24,1):"WEITER
? J" :: ACCEPT AT(24,9)SIZE(
-1)VALIDATE("JN")BEEP:A$
710 IF A$="N" THEN RETURN
720 J=11 :: CALL HCHAR(24,3,
32,9):: FOR I1=6*I TO 6*I+5
730 DISPLAY AT(J,13):T$(I1):
J=J+2 :: NEXT I1
740 IF M=2 AND A1$<>"L" THEN
GOSUB 950
750 NEXT I
760 DISPLAY AT(24,1):"WEITER
? J" :: ACCEPT AT(24,9)SIZE(
-1)VALIDATE("JN")BEEP:A$
770 IF A$<>" " THEN RETURN
780 RETURN
790 !AUSWAEHLEN
800 J=1 :: FOR I=3 TO 11 ::
DISPLAY AT(J,1):I:" " :SEG$(T
C$(I),1,10):: J=J+2 :: NEXT
I
810 DISPLAY AT(24,1)BEEP:"IH
RE WAHL?":M1 :: ACCEPT AT(24
,12)SIZE(-2)VALIDATE(DIGIT):
M1
820 RESTORE #1
830 IF M1<3 OR M1>11 THEN 81
0
840 DISPLAY ERASE ALL AT(24,
1):TC$(M1):: ACCEPT AT(24,13
)SIZE(TL(M1)):B$
850 IF M1>5 THEN J1=4 ELSE J
1=1
860 A=0 :: FOR I=1 TO M1-1 :
A=A+TL(I):: NEXT I
870 IF EOF(1)<>0 THEN 230 EL
SE LINPUT #1:Z$
880 B=A :: FOR I2=1 TO J1
890 C$=SEG$(Z$,B+1,TL(M1))
900 IF M1=4 THEN IF SEG$(C$,
2,1)="." THEN C$=SEG$(C$,1,1
)ELSE IF SEG$(C$,3,1)="." TH
EN C$=SEG$(C$,1,2)
905 IF M1=4 THEN IF VAL(C$)>
=VAL(B$)THEN GOSUB 640 :: GO
TO 920
910 IF (M1=8 AND POS(C$,SEG$
(B$,1,1),1))OR POS(C$,B$,1)T
HEN GOSUB 640 :: GOTO 920 EL

```

```

SE B=B+37 :: GOTO 930
920 DISPLAY AT(24,1):"NAECHS
TE SEITE/CASSETTE? J" :: ACC
EPT AT(24,26)SIZE(-1)VALIDAT
E("JN"):A$ :: IF A$="N" THEN
230 ELSE 870
930 NEXT I2 :: GOTO 870
940 CALL ERR(C,T):: IF C=130
THEN PRINT "SATZ NICHT VORH
ANDEN": "BITTE WARTEN" :: RUN
ELSE RETURN
950 ACCEPT AT(5,13)VALIDATE(
DIGIT)SIZE(-3)BEEP:T$(3)
960 IF T$(3)<>"060" AND T$(3
)<>"120" AND T$(3)<>"180" AN
D T$(3)<>"240" THEN 950
970 ACCEPT AT(9,13)VALIDATE(
"JN")SIZE(-1)BEEP:T$(5)
980 I1=6*I
990 ACCEPT AT(11,13)SIZE(-4)
VALIDATE(DIGIT)BEEP:T$(11)
1000 I1=I1+1 :: ACCEPT AT(13

```

```

,13)SIZE(-16)VALIDATE(DIGIT,
UALPHA)BEEP:T$(I1)
1010 I1=I1+1 :: ACCEPT AT(15
,13)VALIDATE(UALPHA)SIZE(-2)
BEEP:T$(I1)
1020 I1=I1+1 :: ACCEPT AT(17
,13)SIZE(-11)BEEP:T$(I1)
1030 I1=I1+1 :: ACCEPT AT(19
,13)SIZE(-1)VALIDATE("JN")BE
EP:T$(I1):: I1=I1+1
1040 ACCEPT AT(21,13)SIZE(-3
)VALIDATE(DIGIT)BEEP:T$(I1)
1050 T$(4)=STR$(VAL(T$(3))-V
AL(T$(11))-VAL(T$(17))-VAL(T
$(23))-VAL(T$(29)))
1060 IF VAL(T$(4))<0 THEN DI
SPLAY AT(24,1)BEEP:"FALSCH
E DAUER" :: GOTO 1040
1070 DISPLAY AT(7,13):T$(4)
1080 RETURN
1090 CLOSE #1
1100 END

```

## Diskettenlaufwerk für den TI 99/4A

Zum Anschluß an den TI-Diskcontroller sind nicht nur die Original TI Laufwerke geeignet. Besonders jetzt, da wohl keine Laufwerke von TI mehr zu bekommen sind, ist dies besonders interessant. Zum Beispiel ein BASF 6106-Laufwerk läßt sich ganz einfach anschließen: Es ist nur das Kabel zu verbinden und man muß darauf achten, daß Pin 1 des Steckers am Laufwerk auch mit Pin 1 am TI-Controller verbunden wird.

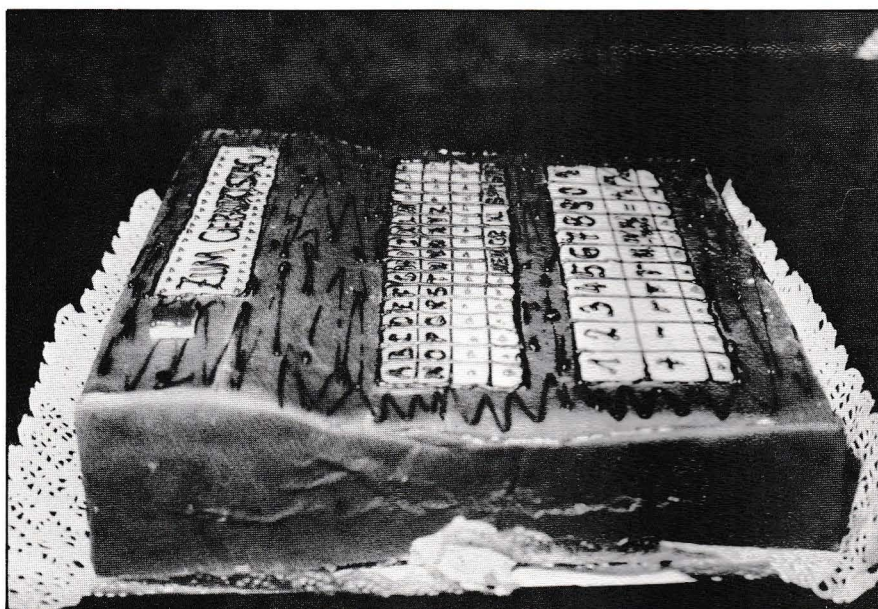
In der Bundesrepublik Deutschland kursiert das Gerücht, daß man mit dem TI-Controller auch in doppelter Dichte (double density) arbeiten könnte, entsprechende Laufwerke vorausgesetzt. Das stimmt nicht! Im deutschen Teil des Disk-Manager 2 liegt offensichtlich ein Fehler vor. Es wird beim Initialisieren einer Diskette abgefragt "Einfache Dichte J/N". Gibt man hier Nein ein, so versucht der Diskettencontroller die Diskette doppelseitig zu formatieren. Das geht natürlich nur mit doppelseitigen Laufwerken. Im deutschen Teil des Disk-Manager 2 sind also die Abfragen "Einfache Dichte" und "Einseitig" vertauscht.

Heiner Martin

## Geburtstagsstorte

Diese eßbare Geburtstagskonsole hat uns Fr.Briditte KASENBACHER aus Landenzersdorf "gebaut". Ich möchte ihr auf diesem Wege nochmals recht herzlich für diese liebe Idee danken.

(über Herausgeber)



## Achtung Achtung Achtung

# Neuerscheinungen



NACHDEM UNS SEHR VIELE TI99-FANS UND LESER DES TI99-JOURNALS ANGEREGT HABEN, ZUSÄTZLICH ZUM JOURNAL PROGRAMMSAMMLUNGEN HERAUSZUGEBEN, FREUT ES UNS, IHNEN DIE ERSTEN BEIDEN, IN VORAUSSICHTLICH ZWEI- BIS DREIMONATLICH, NEBEN DEM TI99-JOURNAL ERSCHEINENDEN, SONDERHEFTE VORSTELLEN ZU KÖNNEN.

### TI 99-Spezial-Basic 1

PROGRAMMSAMMLUNG MIT PROGRAMMEN IN BASIC UND EXTENDED BASIC.  
HIER EIN KLEINER ÜBERBLICK ÜBER DIE PROGRAMME:

46WINNT...BALLONJAGD...BIORHYTHMUS...BOMBER  
BLACK-JACK...BACKGAMMON...HISTOGRAMM...ECONOMY  
MUNCH-MAN...REVERSER.....

### TI 99-Spezial-Basic 2

HP-RECHNER-SIMULATION...EINZELER...SCHNAPS  
(SUPERLANGES PROGRAMM)...KONTO-VERWALTUNGSPRG  
Rem-Remover...Zahl>Text...Uboot...Lauftext...  
Konverter...Roboter.....

### TI 99-Spezial-Assembler 1

PROGRAMMSAMMLUNG MIT PROGRAMMEN IN  
TMS9900-ASSEMBLER-Maschinensprache.  
HIER EIN KLEINER ÜBERBLICK ÜBER DIE PROGRAMME:

Bildschirmseiten abspeichern...SCROLL UP/DOWN  
BIT-MAP-GRAFIK.....DISPLAY AT/ACCEPT AT.....  
TEXTMODUS (Textsystem mit 40 Zeichen/Zeile und  
Utilities)....GOTO n, GOSUB n, RESTORE n....

### TI 99-Spezial-Assembler 2

Joystick-RS232-Software....Interruptgetriebene  
Uhr...Memory-Image-Saved XBASIC-File-Loader...  
Life-Spiel...Right\$/Left\$...Revers\$/Mirror\$...  
Textmodus für XBASIC.....



### BEZUG:

===== Diese Programmsammlungen, mit  
vielen noch unveröffentlichten Tips, erhalten  
Sie bei folgenden Vertriebsstellen:

**GENERALVERTRIEB**  
für ÖSTERREICH:  
**ERB-VERLAG**

Amerlingstraße 1  
A-1061 WIEN  
Tel.: 0222/566209

**Klubbestellungen**  
und AUSLAND:  
**TI99-Journal-Klub**  
Felberstraße 24  
A-1150 WIEN

**PREIS:** öS 180,- (+ Versandkosten)

**VERSAND:** A) Vorkassa: öS 18,- (Inland)  
öS 25,- (Ausland)

B) Nachnahme: öS 35,- (NUR INLAND)

**ACHTUNG:** Es können nur **SPESEFREIE**  
Überweisungen akzeptiert werden!

**HÄNDLERANFRAGEN ERWÜNSCHT!!!**



# Neuerscheinung

DIESES HANDBUCH ERMÖGLICHT DEM ANFANGER EINEN UNKOMPLIZIERTEN EINSTIEG IN DAS MINIMEMORY-MODUL UND BEREITET IN EINFACHER WEISE KOMPLEXE PROBLEME DER MASCHINENSPRACHE TMS9900-ASSEMBLER AUF. ABER EBENSOWIE WIRD DER FORTGESCHRITTENE NÜTZLICHE TIPS UND HINWEISE FINDEN. ZAHLREICHE PROGRAMMBEISPIELE VERANSCHAULICHEN THEORETISCH ERKLÄRTES.

ES MUSS FÜR JEDEN MINIMEMORY-BENÜTZER.

**BEZUG:** TI99-JOURNAL-KLUB  
T.FIEDLER-VERLAG  
Felberstraße 24  
A-1150 WIEN

**PREIS:** öS 350,-

**VERSAND:** A) Vorkassa: öS 18,- (Inland)  
öS 25,- (Ausland)  
B) Nachnahme: öS 35,- (NUR INLAND)

**ACHTUNG:** Es können nur SPESEFREIE Überweisungen berücksichtigt werden.



## Neue Assembler-Software Exklusive bei TI 99 Journal

DIESE ASSEMBLER-SOFTWARE ZEIGT IHNEN DIE ENORMEN "KRAFTRESERVEN" IHRES TI99/4A. OB EINES DER SPIELE ODER DER GRAFIKPROGRAMME. JEDES PROGRAMMPAKET FASZINIERT DURCH SEINEN BETRIEBUNGSKOMFORT (MENÜLEISTE). DIE IN VERSTÄNDLICHEM DEUTSCH GEHALTENE, AUSFÜHRLICHE BETRIEBUNGSANLEITUNG UND DEN HOHEN QUALITÄTS-STANDARD.

## Eat-Man

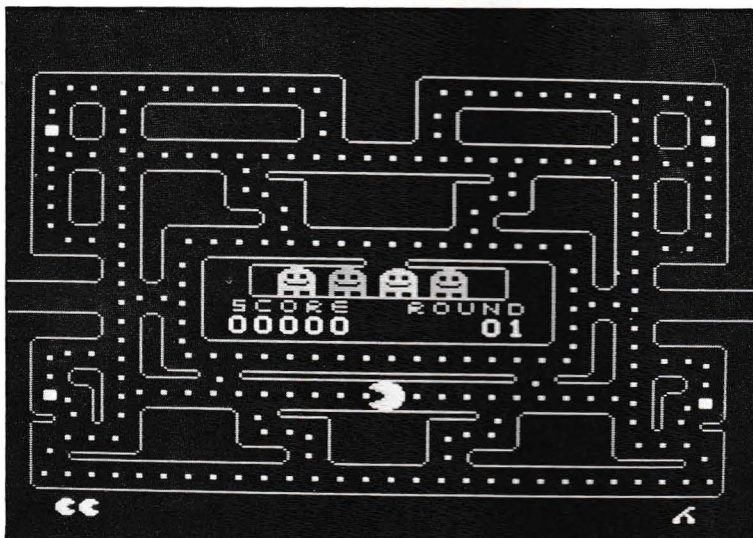
=====

VERSUCHEN SIE MIT IHREM EAT-MAN MÖGLICHSST VIELE KEKSE (PUNKTE) ZU FRESSEN. JEDOCH MÜSSEN SIE SICH VOR DEN VIER WÄCHTERN IN ACHT NEHMEN: EINE BERÜHRUNG WÄRE TÖTLICH. VIER SUPER-KEKSE GEBEN IHNEN ABER SO VIEL KRAFT. DAB SELBST DIE WÄCHTER ERBLASSEN UND DIESMAL VOR IHREM EAT-MAN FLÜCHTEN. GESCHICKLICHKEIT, REAKTION UND TEMPO SIND DIE MERKMALE DIESSES SPIELES IN SUPERQUALITÄT.

## Worm-Attack

=====

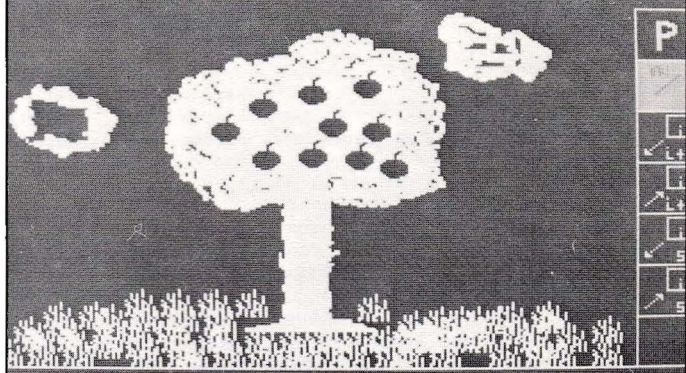
VERSUCHEN SIE MIT IHRER LASER-KANONE DEN ANGREIFENDEN WURM ZU STOPPEN. ES ERWARTEN SIE EINIGE ÜBERRASCHUNGEN (FLIEGEN, SPINNEN....) WENN SIE EIN WIRMSEGMENT TREFFEN SO VERWandelt ES SICH IN EINEN TOTENKOPF UND DER WURM TEILT SICH. DIESSES ASSEMBLER-PROGRAMM ZÄHLT SICHER ZU DEN RASCHESTEN SPIELEN UND VERLANGT EINIGES AN REAKTIONSVERMÖGEN.





## Grafic-Master

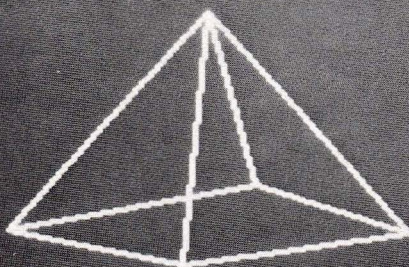
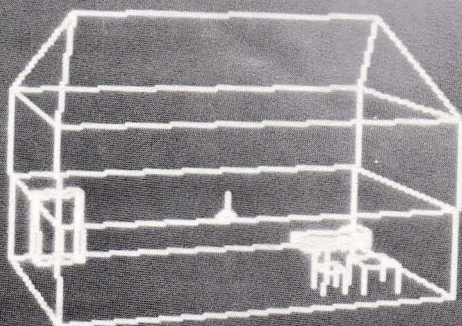
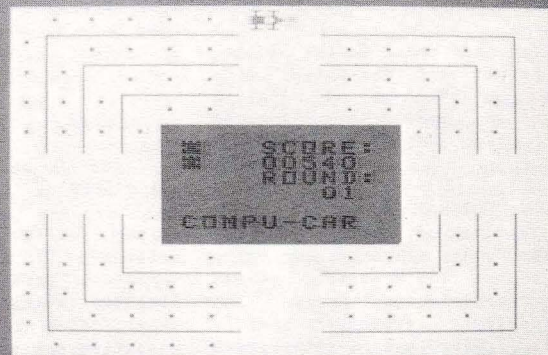
MIT DIESEM UMFANGREICHEN PROGRAMM-PAKET HABEN SIE DIE MÖGLICHKEIT MIT SELBSTDEFINIERTEN SYMBOLEN ODER DIREKTEM ZEICHNEN FARBBGRAFIKEN ZU ERSTELLEN. DAS ZEICHENFELD HAT 232 x 168 PUNKTE, DIE JEWEILS EINZELN, UNEINGESCHRÄNKT ANSPRECHBAR SIND, EBENSO STEHEN IHNEN ALLE 16 FARBEN ZUR VERFÜGUNG. AUF EINER 16 x 16 MATRIX KÖNNEN SIE BIS ZU 100 EIGENE SYMBOLE DEFINIEREN. EBENSO KÖNNEN SIE DIREKT AUF DEM ZEICHENFELD EIN 56 x 56 PUNKTE GROSSES SYMBOL FESTLEGEN UND DIESES DANN WIE DIE KLEINEN SYMBOLE WEITERVERWENDEN. SELBSTVERSTÄNDLICH LASSEN SICH ALLE BILDER AUCH ABSPEICHERN. EIN GROSSTEIL DER PROGRAMMBEDIENUNG KANN AUCH ÜBER JOYSTICK VORGEGENOMMEN WERDEN. DIESES PAKET GEHÖRT ZU DEN FASZINIERENDSTEN PROGRAMMEN DIE DEM TI99 ZUR VERFÜGUNG STEHEN!



CHOOSE ANY MENU-FUNCTION.

## 3D-Welt

ERSTELLEN SIE ÄUßERST KOMFORTABEL DREIDIMENSIONALE KÖRPER. NUN HABEN SIE DIE MÖGLICHKEITEN EINES PERSONALCOMPUTERS: DREHEN UM JEDE ACHSE, VERKLEINERN UND VERGRÖßERN IN JEDER RICHTUNG IN WÄHLBARER GESCHWINDIGKEIT ODER IN ALLE RICHTUNGEN GLEICHZEITIG, VERÄNDERN EINZELNER TEILE (Z.B.: UMLAZIERUNG DES KASTENS IN IHREM HAUS.), AUS- UND EINSCHALTEN EINZELNER STRUKTUREN, UND DAS ALLES MIT ENORMER GESCHWINDIGKEIT...



## Compu-Car

LENKEN SIE IHREN RENNWAGEN DURCH DAS LABYRINTH UND SAMMELN SIE DABEI IHRE PUNKTE. DOCH SO LEICHT WIRD ES IHNEN NICHT GEMACHT: GEISTERAUTOS VERSUCHEN SIE ZU ÜBERROLLEN. PLANEN SIE DEN FAHRSPURWECHSEL, REAGIEREN SIE RASCH UND VORALLEM RICHTIG, AUCH BEI DIESEM SPIEL FASZINIERT DIE AUSGEREIFTE PROGRAMMIERTECHNIK.

### BEZUG:

===== Diese ASSEMBLER-Programme benötigen ein Diskettenlaufwerk und die 32K-Karte. Der Versand erfolgt am Tage des Zahlungs-, bzw. Bestellungseinganges.

### GENERALVERTRIEB

für ÖSTERREICH	für DEUTSCHLAND
TI99-JOURNAL-KLUB	HEGA-SOFT
Felberstraße 24	Weitlstraße 63
A-1150 WIEN	D-8000 MÜNCHEN 45

PREISE: 1 Spiel 6S 420,-  
 2 Spiele " 780,-  
 3 Spiele " 1140,-  
 1 Grafik " 595,-  
 2 Grafik " 1120,-

Die Preise inkl. Versand u. Anleitung.  
 Nachnahme: + 6S 25,- (NUR für ÖSTERREICH)

ACHTUNG: Es können nur SPESENFREIE Überweisungen berücksichtigt werden!



# FORTH AM TI99/4A

Nachdem aus dem amerikanischen 99'er-Userjournal schon seit einigen Monaten zu entnehmen war, daß dort ein von TI selbst vertriebenes Modul mit der Programmiersprache FORTH zu kaufen ist und davon unabhängige Anbieter ebenfalls bereits von Modulen unabhängige Versionen angeboten haben, ein Versuch, über TI hier in Österreich zu einem FORTH-Interpreter zu kommen, gescheitert ist, konnte nun durch einen Direktimport diese neuerdings stark aufgewertete Programmsprache angeschafft werden.

Damit ist nun neben Basic, XBasic, Logo, Pascal und Assembler eine weitere Sprache am TI möglich geworden und jede neue Sprache erhöht die Attraktivität des Gerätes.

Was bringt FORTH? Da FORTH in der Mitte zwischen Assembler und den bekannten Hochsprachen anzusiedeln ist, bringt FORTH primär einmal einen unglaublichen Gewinn an Geschwindigkeit gegenüber BASIC, speziell TI-Basic. Ein nachfolgend aufgelisteter Benchmark-Test soll dies mehr als deutlich unterstreichen.

Diesem Geschwindigkeitsgewinn stehen allerdings gravierende Fakten gegenüber: FORTH kann speziell am TI nur bei bereits guten Kenntnissen von Assembler programmiert werden, da fortgesetzt auf Systemadressen zugegriffen wird. Jede geplante Operation ist in allen Konsequenzen durchzudenken, ein einfaches Drauflosprogrammieren wie in Basic führt in keinem Fall zum Ziel.

FORTH-Programme sind ähnlich schwer zu lesen wie Assemblerprogramme, die Dokumentation ist der wichtigste Programtteil. FORTH verlangt auch ein neues TI-Gefühl, denn die Sprache verwendet UPN-Eingabe, wie sie sonst nur von HP-Geräten bekannt ist. FORTH verwendet kaum Variable, sonder legt alles in Stacks ab, mit denen gearbeitet wird. (Anklänge an die TI-Taschenrechnerreihe SR56, TI58 u.ä.)

Dies klingt sehr kompliziert und man muß zugeben: es ist relativ schwer, sein erstes laufendes FORTH-Programm zu schreiben. Was macht nun FORTH gerade am TI aber so interessant?

Neben dem völlig überraschenden Tempo einige erstaunliche Eigenschaften: man bekommt mit Einwortbefehlen die Möglichkeit, zwischen Text-, Graphic- und Bit-map-Mode umzuschalten! Grafik wird blitzartig erstellt. FORTH unterstützt selbstverständlich alle File-Operationen an Diskettenlaufwerk und Druckerausgaben. Sprites und Ton können auch angesprochen werden.

Sicherlich aber am wichtigsten ist die Erstellung eines bildschirmorientierten Editors, in dem der Cursor mittels der Pfeiltasten auf jeden Bildschirmpunkt gesetzt werden kann und die Eingabe dort erfolgt (wie am TI-Writer).

Die wohl am bemerkenswertesten Eigenart von FORTH ist, daß jedes einmal eingegebene Programm ab der Eingabe ein fixer Bestandteil des Betriebssystems wird, somit wird das Betriebssystem nach jedem Einsatz größer. Es können allerdings mit dem Schlüsselwort FORGET Befehle außerhalb des geschützten Katalogs wieder gelöscht werden.

Wohl am einfachsten zu programmieren sind Sprachübersetzer: So meldet sich FORTH nach Eingabe von

```
: BAUM ." TREE" ;
```

bei jeder Eingabe von "BAUM" sofort mit "TREE" ok.

Welche Konfiguration ist für FORTH notwendig? Für die uns zur Verfügung stehende Version ist Diskettenlaufwerk, RAM Expansion und XBasic-Modul notwendig, Kassettenversionen, die auch mit MINIMEM oder EDITOR/ASSEMBLER zusammenarbeiten, (RAM-Expansion wird immer benötigt), könnten besorgt werden.

Interessenten an der uns zur Verfügung stehenden Version können sich mit dem Autor in Verbindung (via Club) setzen.

Nun zum angegebenen Benchmarktest. Die Basicversion benötigt für einen Durchlauf 220 Sekunden, die FORTH-Version 5.3 sec.

```
100 N=1000
110 FOR X=10 TO 99
120 FOR Y=0 TO 99
130 IF (X+Y)*(X+Y)=N THEN PRINT N
140 N=N+1
150 NEXT Y
160 NEXT X
```

```
BASE $ DECIMAL : TASK :
2 LOAD 3 LOAD ( #IN. BELL )
( HOUSE$S RUN BENCHMARK ONCE )
: HOUSE$S 1000 100 10 DO 100 0 DO
DUP J I + DUP =
IF DUP . ENDIF 1+ LOOP LOOP DROP
```

```
( TIMEIT : RUN BENCHMARK MANY TIMES )
: TIMEIT CR CR ." House Numbers
Benchmark "
```

```
CR CR ." How many times ?"
#IN DROP 1+
1 DO CR I . HOUSE$S LOOP
BELL :
```

Der Programmaufruf erfolgt dann mit TIMEIT

Das Umschalten zwischen den einzelnen Betriebsmoden des TI99 wird z.B. durch die Programme TEXTMODE und GRAPHICSMODE ermöglicht:

```
: CLS 0C EMIT ;
: TEXTMODE 0F0 DUP 83D4 C! 1
!VDPREGISTER
28 SCREEN-WIDTH ! 13 7
!VDPREGISTER CLS
84FC 8484 8484 FC84 8 0 DO IF 8 *
800 + 1 +
!VDP 2 +LOOP ;
```

```
: GRAPHICSMODE 0E0 DUP 83D4 C! 1
!VDPREGISTER
20 SCREEN-WIDTH ! CLS 3A0 380 DO
13 I C!VDP
LOOP 380 300 DO 0 I C!VDP LOOP ;
```

Um nun aus einem Modus in den anderen zu kommen, müssen lediglich die Schlüsselwörter TEXTMODE oder GRAPHICSMODE eingegeben werden. Jene unter Ihnen, die mit Assembler vertraut sind, werden in den angegebenen HEX-Zahlen mehr oder minder vertraute Werte wiedererkennen, die auch beim Assemblerprogramm zum TEXTMODUS gesetzt werden müssen.

Wie kurz doch wieder manches FORTH-Programm relativ komplexe Probleme löst, soll das abschließende Programm demonstrieren: ein HEX-Dump, der neben der Adresse auch noch angibt, ob der gespeicherte Wert zufällig auch ein druckbares ASCII-Zeichen darstellt:

```
( HEXDUMP ) DUMP
: LINE DUP DUP 8 + SWAP DO I C$ 3 .R
LOOP ;
```

```
: WERT DROP 20 ;
: TI99/4A DUP 7D = IF WERT THEN DUP
FD = IF WERT THEN
DUP 1C = IF WERT THEN DUP
7E = IF WERT THEN
DUP 1D = IF WERT THEN DUP
9C = IF WERT THEN
DUP 1E = IF WERT THEN DUP
FE = IF WERT THEN
DUP 1F = IF WERT THEN DUP
FF = IF WERT THEN
DUP 90 = IF WERT THEN :
```

```
: ASCII DUP DUP 8 + SWAP DO I C$
TI99/4A EMIT
2 SPACES LOOP ;
```

```
: HDUMP CR DUP . SPACE LINE CR 8
SPACES ASCII 0 + ;
```

```
: DUMP 0 DO HDUMP LOOP;
```

Der Aufruf erfolgt mit Startadresse Endadresse DUMP.

Das uns zur Verfügung stehende FORTH entspricht in allen Teilen dem internationalen Standard fig-FORTH, Version 1.0, abgeändert wurde lediglich der Standardbefehl KEY in ?KEYBOARD, die Befehle VLIST oder CATALOG sind durch HELP ersetzt, INPUT-LINE, CURSOR-POS, COLUMN-SKIP sind spezifisch für das TI-FORTH, auch arbeitet die DO... +LOOP-Schleife etwas vom Standard abweichend.



Alles in allem ist FORTH eine wesentliche Bereicherung für den TI99/4A, da es seine Möglichkeiten voll nutzt und nach einer Eingewöhnungsphase in Basic kaum zu realisierende Programme erlaubt, die aber dennoch noch leichter als gleich in Assembler zu erstellen sind.

## 128 K RAM-EXPANSIONSKARTE

Nachdem der Nachschub von TI an Peripheriegeräten durch den Ausstieg aus der Produktion unterbunden wurde, ist es naheliegend, nach Ersatzgeräten für jene zu suchen, die zwar noch den "Schlußverkauf" nutzen konnten, jetzt aber auf der Suche nach Peripheriegeräten sind.

Schon seit einiger Zeit bietet eine amerikanische Firma RAM Expansionskarten an, wobei eine 128-K-Karte sicherlich die größte Aufmerksamkeit verdient, da sie einerseits ungeahnte Speichermöglichkeiten ergibt, andererseits preislich nur wenig über jenem Niveau liegt, das hier bei uns die 32K-Karte ohnehin die längste Zeit hatte.

Wie funktioniert nun dieser Traumspeicher, wo der direkte Adressierungsbereich des TMS9900 doch auf 64K beschränkt ist?

Diese Karte, geliefert in genau jenem stabilen Metallgehäuse, das Sie vielleicht von den anderen Einschüben schon kennen und deren Ausführung die Qualität vergleichbarer Produkte andere Homecomputer immer schon um Längen geschlagen hat, ist in vier Banken zu je 32 K organisiert. Jede dieser Banken belegt genau den gleichen Adressbereich, der schon bei der alten 32K-Karte bekannt war - den unteren Bereich >2000 - >3FFF und einen oberen Bereich >A000 - >FFFF. Zwischen diesen Banken kann mittels SB0- und SBZ-Instruktionen umgeschaltet werden. Als Speicher selbst kommen 4164, dynamische RAMs, zum Einsatz, die von einem TMS 4500 als DRAM-Controller verwaltet werden.

Welche Möglichkeiten und Grenzen ergeben sich aus dem Einsatz dieser Karte?

Zunächst einmal ist für den Anwender sicherlich enttäuschend, auf den Befehl "SIZE" hin die gewohnte Aussage von "11840 BYTES OF STACK FREE, 24480 BYTES OF PROGRAM SPACE FREE" zu erhalten. Dies ist aus dem "bank switching" aber unmittelbar abzuleiten. Somit steht der große Speicher auch nicht den Modulen TI-Writer, Multiplan zur Verfügung.

Die Stärke der 128K-Karte liegt in der mitgelieferten Software, der DSR-Option. Sie enthält als sicherlich interessanteste Variante die Verwendung als RAM-Disk, die mittels "DSKX" angesprochen, und mit OPEN, PRINT und INPUT als File verwendbar ist. Es ist damit auch

möglich, mit nur einem Laufwerk einfacher Disketten zu kopieren, als der Speicherbereich groß genug ist, einen kompletten Disketteninhalt aufzunehmen - dann erst die Disk zu wechseln! - und auf eine neue Diskette zu übertragen. Jeder, der mit einem Laufwerk eine Diskette voller kurzer Programme oder Files kopiert hat, wird die Bedeutung dieser Routine voll zu würdigen wissen.

Damit nicht genug, steht in Basic ein Datenspeicher ähnlich dem File "MINIMEM" von hier aber ganzen 96K zur Verfügung, das mittels OPEN#1:"MEM96" zu öffnen ist. Dabei steht dieses File bereits in normalem TI-Basic zur Verfügung!

Als weitere Option steht unter Xbasic die Möglichkeit offen, gleichzeitig mehrere Programme auf verschiedenen Banken geladen zu haben, die aus einem Programm mittels "RUN" ein weiteres Programm somit ein Superprogramm bis 128K aufzubauen erlauben!

Nun, einige der letztgenannten Optionen sind nur bei gleichzeitigem Erwerb des von der gleichen Firma angebotenen

Programms Dis-Emulator möglich, hingewiesen aber muß auch noch darauf, daß bei Einsatz des Minims letzlich sogar 132 K-RAM für Maschinenprogramme zur Verfügung stünden...

Was nun aber wohl am meisten interessiert, dürfte der Preis sein - inklusive Fracht nach Österreich sind der Bestellung ein auf eine amerikanische Bank bezogener Scheck über \$ 242.55 beizulegen - die Lieferfirma hat folgende Anschrift:

FOUNDATION  
74 Claire Way  
Tiburon, CA 94920

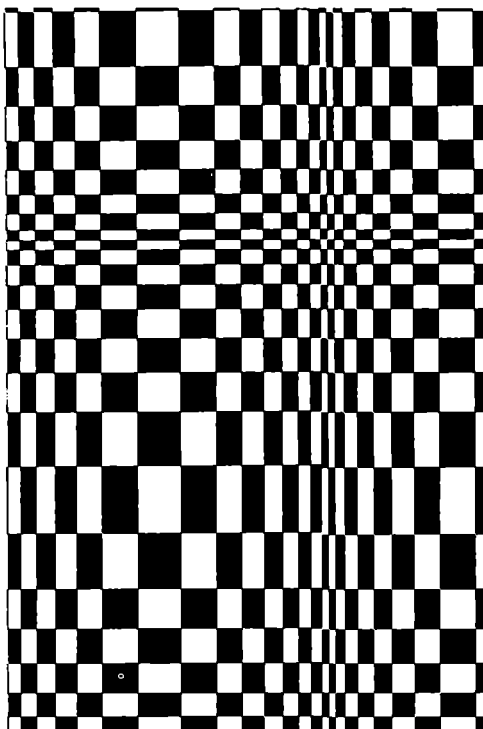
Wahrscheinlich klar, aber vielleicht doch erwähnenswert für Ihre persönliche Kalkulation, wenn Sie bereits eine herkömmliche 32K-Karte besitzen: da diese den gleichen Adressbereich belegt, kann sie nicht weiter verwendet werden. Sie werden sie daher verkaufen müssen, womit die Investition in die 128K-Karte sicherlich bereits wesentlich entschärft werden wird!

## Vorschau TI 99 Journal Nr. 6

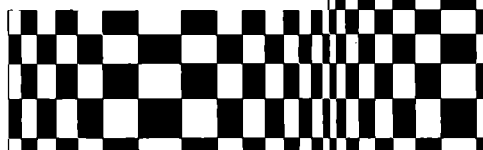
DAS NÄCHSTE TI99-JOURNAL BRINGT UNTER ANDEREM FOLGENDE, INTERESSANTE ARTIKEL UND PROGRAMME:

128K-RAM-KARTE....BASIC-COMPILER.... TMS9900 im Vergleich...FORTH am TI99 HARDWARE: Eigenbau-Speechsynthesizer schreibt Programme selbst....XBASIC-Tips...Disketten zweiseitig gemacht. Poker (Prgr.).....

**Ich freue mich, daß  
ich in Ihrer Zeitung  
erscheinen darf.**



**Die Anzeige.**



HOME COMPUTER

TEXAS INSTRUMENTS



# TI-99 ITALIAN USER CLUB

[WWW.TI99IUC.IT](http://WWW.TI99IUC.IT)

[INFO@TI99IUC.IT](mailto:INFO@TI99IUC.IT)

- Thanks to 99'er User: *Alfredo Cevolini*, for the Scan of this document.

- Reworked by TI99 Italian User Club ([info@ti99iuc.it](mailto:info@ti99iuc.it)) - 2014

*Downloaded from [www.ti99iuc.it](http://www.ti99iuc.it)*



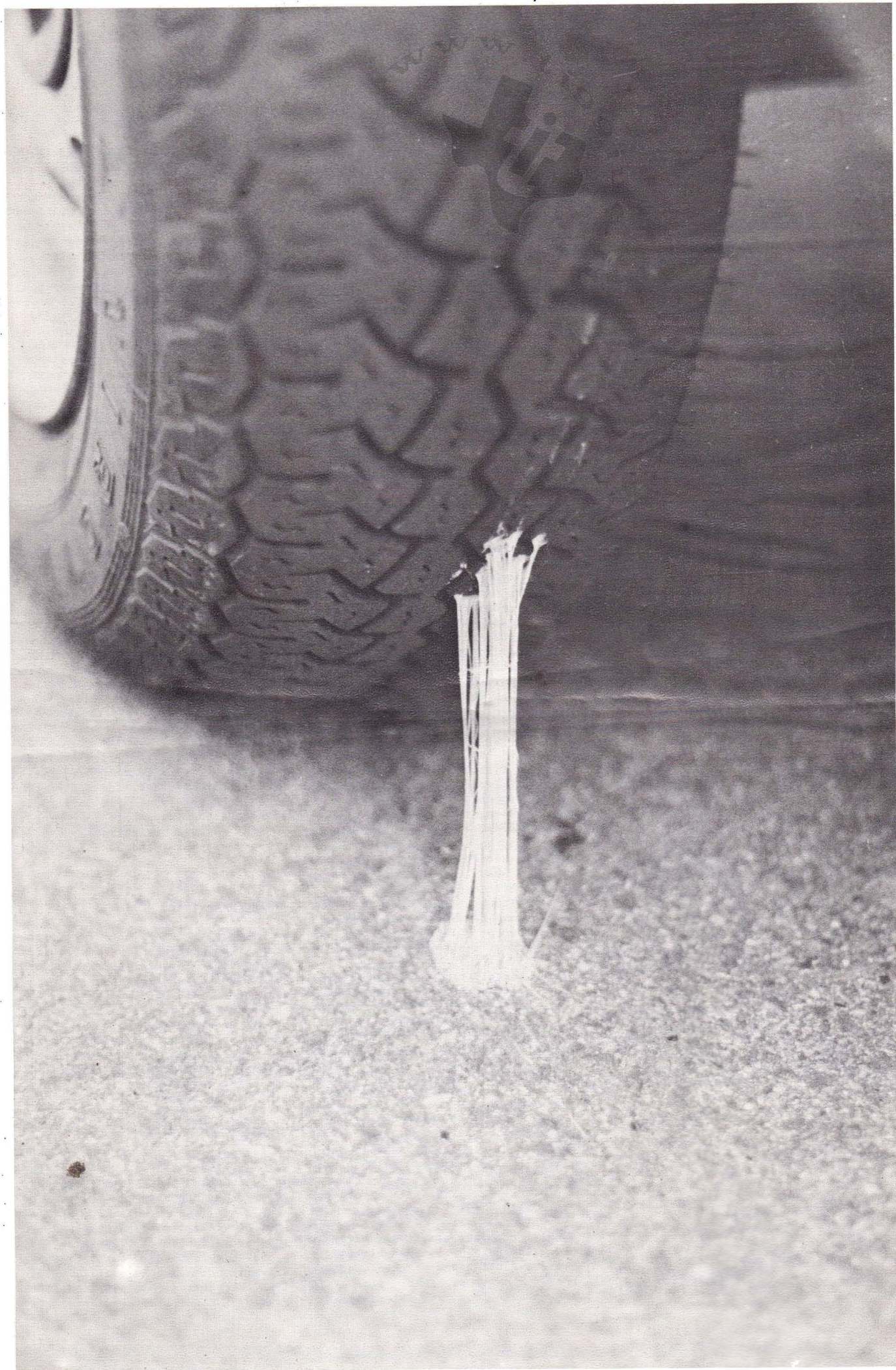


Foto: T. Fiedler