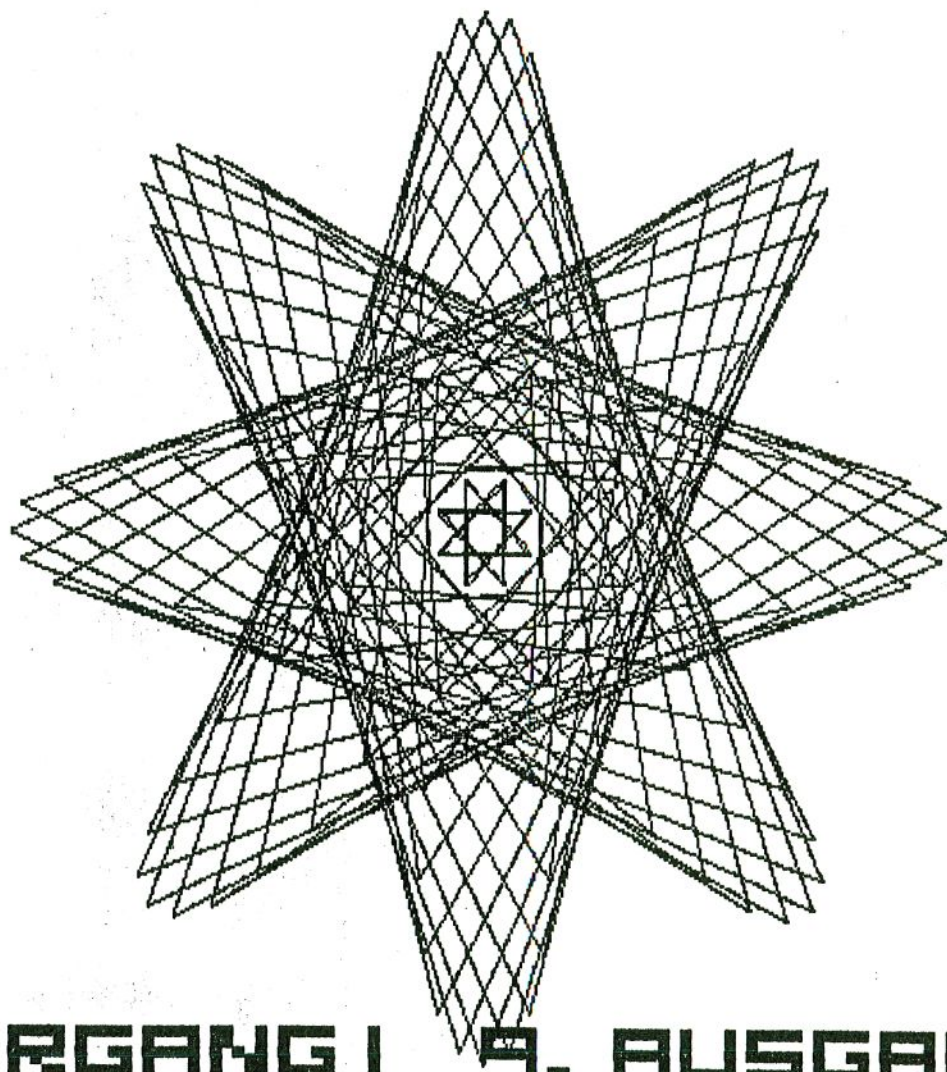


**GENIE** USER  
und ColourGenie CLUB  
**USER**  
**CLUB** **TREASON**  
**BREMERHAVEN**

CLUB-INFO  
CLUB-INFO  
CLUB-INFO  
CLUB-INFO

---



**2. JAHRGANG | 1. AUSGABE**

Red.: Peter Spieß, Trugenhofenerstr. 27, 8859 Rennertshofen 1  
\* Sortiert von: Edeltraud \*\*\* Auflage: 065 Exempl. \*\*\*\*\*

# Inhalt

## Club-Info

1	Internes
2	Adressenliste
3-5	SYS-Files und wie man sie macht v. A. Sopp
6-7	Das Problem DD SD DD oder doch SD v. O. Stark
8	Auch ein Genie kann irren v. M. Karnatz
9-10	Udo macht's möglich
11-13	Umgang mit Fremdsprachen v. A. Sopp
14	Rezept gegen Computerfieber v. P.-J. Schmitz
15-16	Endlich ein BASIC-Programm v. U. von Scheid
17-18	Die Records handhaben v. A. Sopp
19	Ein Tip zum genauen Rechnen v. P.-J. Schmitz
20-24	Ein Spooler für den EG64MBA mit HRG1b v. A. Sopp
25-26	Tune-Up des Genie mit "Speed-Up" v. A. Sopp
27	Flohmarkt / Fragen, Antworten und Tips
28 -->	Colour Genie Ecke





## INTERNES VOM BETREUER

\*\*\* Auf Grund mehrerer Anfragen möchte ich betonen, daß der Beitrag auch 1/4, 1/2 oder 1/1 jährlich bezahlt werden kann (weniger Überweisungskosten). Es muß aber im Voraus bezahlt werden ! Zur Erinnerung: Ab Januar 1985, 3,50 DM monatlich.

\*\*\* Von ein paar Mitgliedern habe ich herrliche Ansichtskarten aus ihren Urlaubsorten erhalten und möchte mich dafür herzlich bedanken.

\*\*\* Andree Opt-Hof unterbreitet zwei Vorschläge für neue Rubriken und bittet Euch um Resonanz:

1. Wie wäre es z.B. mit einer dem "Gamers Cafe" aus 80 MICRO nachempfundenen Rubrik, in der Clubmitglieder Tips, Tricks und evtl. Lösungen (Teillösungen) für Spiele und Adventures anbieten könnten ??? Es braucht aber nicht gerade die aktuelle Hiscore-Tabelle sein, sondern eben Kniffe, wie z.B. Adventures, die wohl jeder in seiner Sammlung hat, zu knacken sind. (Anm. d. Red.: Wer einen Lageplan der Schlösser von "VALKYRIE" und die Möglichkeit, die Energieeinheiten zu erhöhen, haben will, soll sich bei mir melden).

2. Da wir in der Clubzeitung ziemlich viele Assemblerprogramme haben, aber ein Teil der Mitglieder (vielleicht sogar eine größere Anzahl) nicht in der Lage ist, in Z-80 Maschinensprache zu programmieren, empfiehlt es sich, wenn sich ein oder mehrere Assemblerspezialisten dazu aufraffen könnten, einen Einführungskurs in Z-80 zu veröffentlichen (ca. 2-3 Seiten pro Heft).

\*\*\* Gleichzeitig möchte ich noch einmal alle BASIC-Spezialisten um BASIC-Programme für's Info bitten.

### GEBURTSTAG NEUE MITGL.

Werner Grajewski  
Peter Kummerow  
Holger May  
Rudolf Ring  
Horst Schmidt  
Heinz-Gerd Küster  
Hans-Dieter Schneider  
Arnulf Sopp  
Heinrich Thönnissen

Diesmal leider  
keine !

Herzlichen Glückwunsch !



ENIE/TRS80 USER CLUB MITGLIEDERLISTE \*\*\*\* LP=DRUCKER CG=COLOUR GENIE CR=KASSETTENREKORDER LW=DISKETTENLAUFWERKE \*\*\* 09.09.1984

NAME	VORNAME	M-NR.	TELEFON	STRASSE	ORT	HARDWARE
ALTHAUS	THOMAS	840441	0551/75913	WESERSTR. 35	3400 GOETTINGEN	CG,CR
BACH	SIGGI	830611	???	LEHMDENERSTR. 54	2845 DAMME 2	GENIE I, TYPENRAD
BIEWALD	MARKUS	830418	0421/471829	GESCHWISTER-SCHOLL STR 105	2800 BREMEN 41	CG, CR
BLASCHEK	MANFRED	840120	0222/6400483	INZERSDORFERSTR. 111/8/9	A-1100 WIEN	TRS 1,CR
BORNISCHLEGEL	HANS	840738	0951/73831	KOENIGSHOFSTR. 13	8605 HALLSTADT	GENIE I, 2LW
BRAKE	THILO	840413	0471/64717	KASTANIENWEG 26	2850 BREMERHAVEN	TRS80M1,2LW,LP GEMINI10X
DUERHAMMER	ULF	840446	02954/786	ECKENSTR. 8	4784 RUETHEN 13	TRS1,2LW,LPSTARDP8480
FISCHBECK	IME	840125	04421/34282	FRIEDERIKEN- 17	2940 WILHELMSHAVEN	CG,CR
FOLKERTS	RALF	840110	04223/1282	NUTZHORNERSTR. 9	2875 BOOKHOLZBERG	GENIE I, 2LW,LP NEC8023
FREY	WOLFGANG	830816	040/6958854	PILLAUERSTR. 135	2000 HAMBURG 70	GENIE I, LW
GANS	DIETMAR	840445	07633/5357	SCHWARZWALDSTR. 4	7813 STAUFEN I. BR.	TRS80III,CR,FERNSCHREIBER
GRAJEWSKI	WERNER	830507	02134/54573	ZEDERNWEG 29	4220 DINSLAKEN	GENIE I
GRUNDMANN	WALDEMAR	830815	0441/34218	BEVERBAEKSTR. 46	2900 OLDENBURG	TRS80 I,CR, LW
HILLMER	MANFRED	840443	04421/61320	RUESTERSIELERSTR 15	2940 WILHELMSHAVEN	CG,CR
HOSE	RUEDIGER	840544	0911/460012	WODANSTR. 7	8500 NUERNBERG 40	TRS80I,2LW,LPX82+TANDY M100
JEL	MARKUS	840127	05141/31133	LUENEBURGER HEERSTR. 47	3100 CELLE	GENIE I,CR,HIRE
JOURDAN	UDO	840747	06152/81704	DARMSTAEDTERSTR. 66	6080 GROSS-GERAU	GENIE I,GENIE16,4LW,MX80,RX80
KARNATZ	MICHAEL	830419	04421/53936	SCHWERINER RING 23	2940 WILHELMSHAVEN	CG,CR,LP GEMINI10X,1LW
KLEIN	GERHARD	840234	040/513159	CARL-COHN-STR. 73	2000 HAMBURG 60	TRS80 M1,1LW,LP MX80FT
KROENER	PAUL	831023	---	GRAF-ENNO-STR. 7	2970 EMDEN	GENIE I, 1LW
KRZYZANOWSKI	PROF.DR. JER	840233	---	NUR UBER BETREUER	ERREICHBAR !	GENIE I,LP ?
KUESTER	HEINZ-GERD	840748	02058/3037	SCHMACHTENBERGWEG 2	5603 WUELFRATH	GENIE I,2LW,LP PRAXIS 35
KUMMEROW	PETER	840132	0431/30647/3	STEENBECKERWEG 8/35	2300 KIEL	GENIE I,2LW(40,80TKS),LP1DS44
KUMMEROW	JENS	840336	---	HAUPTSTR. 4	5412 HUNSDORF	GENIE I,LP DP510,2LW80DSDD
KUTTER	WOLFGANG	830505	08370/1268	ILLERSTR. 18	8961 WIGGENSBACH	CG,CR,LP STAR510
LINNEWEBER	MANFRED	831224	0471/25453	AUF DER BRIGG 15	2850 BREMERHAVEN	TRS80 III,LP MX80FT
MAY	HOLGER	830508	02935/1668	MARIENSTR. 9	5768 SUNDERN 2	GENIE I,1LW NEWDOS,FERNSCHRE
MEIER	HANS-CHRISTI	840126	04421/64577	RAABESTR. 42	2940 WILHELMSHAVEN	CG,LP GEMINI10X
MUELLER	ALBRECHT	840703	0841/51962	WIRFFELSTR. 8	8070 INGOLSTADT	TRS1,2LW,LP PRAXIS35,TINTENS
OMASREITER	IRMGARD	840339	---	NUR UBER BETREUER	ERREICHBAR !!!	KOMTEK, LW,LP NEC8023
OPT-HOF	ANDREE	840851	0421/420762	HALSMUEHLENER STR. 56	2800 BREMEN 44	TSR80M1,2LW,LP SEIKO GP100A
OTEY	FRANCISCO	840337	---	W. BESSONSTR. 5	7750 KONSTANZ 16	CG,LP OLIVETTI PRAXIS
POTT	THORSTEN	840442	04223/497	UEBERN BERG 10	2875 BOOKHOLZBERG	GENIE I,1LW,MODEM
REICHELSDORFER	WOLFGANG	840129	---	HERRENBERG 25	8870 GUENZBURG/REISEN	TRS80 I,3LW(40/80SP),CR,LP I
RIT	RUDOLF	840104	0208/57280	DUISBURGERSTR. 445/304	4330 MUELHEIM/R.	CG,CR
RUETTIGERS	MARTIN	830922	---	EIFELSTR. 85 A	5190 STOLBERG-VICHT	GENIE I, LP STAR
SCHMIDT	KLAUS	830301	0471/24998	BLESSMANNSTR. 1 B	2850 BREMERHAVEN	APPLE
SCHMIDT	HORST	830302	0471/414611	KOERNERSTR. 7	2850 BREMERHAVEN	GENIE II, CR
SCHMITZ	PAUL-JUERGEN	840235	0202/401192	HAHNERBERGERSTR 111	5600 WUPPERTAL 12	GENIE I,CR,LP BROTHER CE60,2
SCHNEIDER	HANS-DIETER	830621	---	POSTFACH 1346	2943 ESENS	ABC80, CR, LP MX80FT
SOPP	ARNULF	840131	0451/791926	WAKENITZSTR. 8	2400 LUEBECK 1	GENIE I,2 LW,LP GEMINI10X
STARK	OTHMAR	840340	02236/811805	SCHILLERSTR. 112	A2340 MOEDLING	GENIE I,3LW,LPX80FT
Spieß	Peter	*30417	08434/454	Trugenhofenerstr. 27	8859 Rennertshofen 1	GENIE II,3 LW, LP NEC 8023
THALMEIER	GREGOR	840128	08091/9085	POSTFACH 1140	8011 KIRCHSEEON	TRS80 I,3LW(DD,DS),LP MX80,M
THOENNISSEN	HEINRICH	830306	0421/647762	GRAMBKERMoorER LANDSTR. 6	2800 BREMEN 77	TRS80 I, 2LW, CR, LP MX80FT
THOM	HARALD	840112	0203/337178	NECKARSTR. 9	4100 DUISBURG 1	CG,CR
TOPP	GERHARD	840749	05335/240	HEININGER WEG 1	3342 WERLABURGDORF	TRS80M1,2LW,CR,LP RX80FT
U. SCHEIDT	IME	830509	0471/85418	STROEDACKER 45 C	2850 BREMERHAVEN	TRS80I,2LW,LP RX80FT
VAN DER TOUM	WILLEM G.	840130	004117805421	TOBELRAINSTR. 2	CH-8820 WAEDENSWIL	GENIE 3,LP ITOH F10-40
VOLLMER	TORSTEN	830614	---	RHEINSTR. 42	2850 BREMERHAVEN	CG, CR
WITTMANN	REINHARD	840750	09002/2381	KLAUSENBUNNENWEG 32	8852 RAIN/LECH	GENIE I,CR
WOLF	KLAUS	840852	069/5482314	FELDSCHIEDEN STR. 44	6000 FRANKFURT 50	TRS80M1,CR,LP



## SYS-Files, und wie man sie macht

In der folgenden Tabelle sind alle SYS-Files von G-DOS 2.1b verzeichnet mit ihren Aufgaben und ihrer Länge. Die Länge deshalb, weil man sie beim Benutzen der freien SYS-Dateien kennen sollte. In den beiden letzten Spalten steht in binär und sedezimal, mit welchem Wert der Akku geladen sein muß, wenn man sie mit RST 28h aufrufen will. Manche Files erwarten außer dem Code in A weitere Parameter; Vorsicht ist also geboten!

File	Aufgabe	Sekt.	Akku bin.	hex. <sup>1)</sup>
GDOS	DOS booten	5	xxx00000	g0
INHALT	Inhaltsverzeichnis	30	xxx00001	g1
SYS0	DOS-Kern, bis 4CFFh resident	15	xxx00010	g2
SYS1	DOS-Befehle interpretieren	5	xxx00011	g3
SYS2	File-Handling	5	xxx00100	g4
SYS3	dto., JKL, DEBUG usw.	5	xxx00101	g5
SYS4	DOS-Fehlermeldungen	5	xxx00110	g6
SYS5	DEBUG	5	xxx00111	g7
SYS6	NDF, COPY, APPEND, PD, S	35	xxx01000	g8
SYS7	UHR, DATUM, AUTO, ATTRIB usw.	5	xxx01001	g9
SYS8	I (DIR), FREE	5	xxx01010	gA
SYS9	B2, BOOT, Chaining-Kommandos	5	xxx01011	gB
SYS10	BASIC-Befehle GET und PUT	5	xxx01100	gC
SYS11	BASIC-Befehl RENUM (Teil)	5	xxx01101	gD
SYS12	BASIC-Befehl REF	5	xxx01110	gE
SYS13	BASIC-Fehlermeldungen, RENUM	5	xxx01111	gF
SYS14	CLEAR, CREATE, E, LIST, @, DR	5	xxx10000	u0
SYS15	FORM, V24	4	xxx10001	u1
SYS16	Hauptteil von PD	5	xxx10010	u2
SYS17	Hauptteil von S, AIK	5	xxx10011	u3
SYS18	direkte BASIC-Kommandos	5	xxx10100	u4
SYS19	versch. BASIC-Befehle	5	xxx10101	u5
SYS20	dto.	5	xxx10110	u6
SYS21	BASIC-Befehl CMD"O"	5	xxx10111	u7
SYS22	frei	5	xxx11000	u8
SYS23	LWT <sup>2)</sup>	5	xxx11001	u9
SYS24	frei	5	xxx11010	uA
SYS25	frei	10	xxx11011	uB
SYS26	frei	5	xxx11100	uC
SYS27	frei	5	xxx11101	uD
SYS28	FORM (Druckercodes)	5	xxx11110	uE
SYS29	INFO	5	xxx11111	uF

<sup>1)</sup> x = 1 oder 0, g = gerade Hex-Ziffer, u = ungerade Hex-Ziffer

<sup>2)</sup> Diese Tabelle gilt für G-DOS 2.1b. Dort ist der Befehl LWT nicht mehr implementiert. Der Laufwerkstest funktioniert aber mit mindestens 3 Blanks und NEW LINE.

Die Aufgaben der SYS-Files sind teilweise der deutschen Anleitung zu NEWDOS-80 entnommen, soweit sie dort verzeichnet sind. Den Rest habe ich selber herausgetüftelt. Da G-DOS und NEWDOS nicht völlig identisch sind, und da ich entgegen anderslautenden Gerüchten nicht unfehlbar bin, sind Irrtümer in der Spalte "Aufgabe" möglich.



Wie SYS-Files aufgerufen werden, ist im Prinzip in dem Artikel "Neuer Dreitastenbefehl <.,./>" erklärt. Die obige Tabelle erleichtert hoffentlich die Berechnung des Parameters im Akku. Der geneigte Leser muß sich nun nicht mehr mit einzelnen Bits herumschlagen, es genügt, zu wissen, worin sich eine gerade von einer ungeraden einstelligen Zahl unterscheidet. Darüber hinaus möchte ich diesmal erklären, wie ein eigenes SYS-File geschrieben werden kann. Dazu sind ein paar Vorkenntnisse nötig, deshalb wieder als hors d'oeuvre ein wenig Theorie:

Was später im Hauptspeicher stehen soll, liegt Byte an Byte auf der Diskette wie die einzelnen Töne einer Melodie auf einer Schallplatte. Es ist jedoch in einzelne Portionen, sog. Records unterteilt. Am Anfang eines jeden Records steht ein Byte, das die Art der folgenden Codes kennzeichnet. Bei einem Maschinenprogramm hat es den Wert 01. Das zweite Byte ist die Anzahl der Bytes, die dieser Record enthält. Dabei steht 00 für 256 dez.. Die relativen Bytes 3 und 4 des Records enthalten die Ladeadresse des ersten Codes (Byte 5), und zwar in der gewohnten Reihenfolge LSB-MSB. Die beiden Bytes der Adresse werden im Byte 2 übrigens mitgezählt. Die Abb. 1 zeigt ein beliebiges Beispiel. Die Bytes der Record-Organisation sind unterstrichen.

In einem nicht belegten SYS-File (22, 24, 25, 26 und 27) enthalten alle Records 256 Bytes, also ist der Zähler (rel. Byte 2) immer 00. Bei eigenen SYS-Files (wofür man diese freien Dateien ausnutzen muß) sollte man dieses Schema beibehalten.

Am Ende einer Datei schließlich steht die Einsprungsadresse des Programms. Sie wird mit dem Kenncode 02 eingeleitet. Es folgt auch hier die Anzahl der folgenden Bytes. Da es sich nur um diese Adresse handelt, lautet der Wert dieses Bytes ebenfalls 02 (2 Bytes für LSB und MSB der Adresse, s. Abb. 2a).

Die freien SYS-Files belegen 5 bzw. 10 Sektoren. Eine eigene SYS-Routine mag zwar kürzer sein, aber im Hinblick auf spätere Erweiterungen wäre es verschwenderisch, auf den Rest zu verzichten. Und das macht die Geschichte leider etwas kompliziert, wie wir später sehen werden. Grundsätzlich ist es aber ohne weiteres möglich, unter dem Namen z. B. SYS26/SYS mit EDTASM ein Maschinenprogramm zu assemblieren, das sich (außer der Länge) in nichts von einer Apparat-Systemdatei unterscheidet. Man kann sogar mit POKE-Befehlen ein Maschinenprogramm von BASIC aus zusammenschustern und mit CMD"DUMP,..." sein SYS-File kreieren.

Besser als diese Partisanen-Heimarbeit ist die Methode der sektorweisen Kopie. Dabei bleiben die Lage und Länge sowie alle anderen Eigenschaften der freien SYS-Datei erhalten. EDTASM erzeugt beispielsweise das File SYS26/CMD. Mit SUPERZAP und seinem Befehl DFS kriegt man raus, welche Sektoren dieses Programm belegt. Es sind später für den Befehl CDS die Quellsektoren. Auf die gleiche Weise findet man die Zielsektoren von SYS26/SYS.

Wenn SYS26/CMD auf einer ziemlich vollen Diskette nicht mehr zusammenhängend aufgezeichnet wurde (vorher mit SUPERZAP feststellen!), kann man leider bei der Frage SECTOR COUNT nicht mehr alles auf einen Rutsch erledigen. Wer Sorge hat, bei der Kopie auf Raten Fehler zu machen, kann für SYS26/CMD eine frische Diskette nehmen.

Zuletzt ist von Fall zu Fall noch etwas Kosmetik notwendig. Die Record-Grenzen liegen bei SYS26/SYS nicht notwendigerweise an denselben Stellen wie bei SYS26/CMD. Sie lassen sich einfach umzapfen, wie in Abb. 2a (SYS26/CMD) und 2b (SYS26/SYS) gezeigt. Die zu modifizierenden Codes sind unterstrichen. Außerdem enthalten die CMD-Sektoren vielleicht noch Müll des Assemblers (Abb. 2a), den man aus optischen Gründen auf 00 zapfen kann (Abb. 2b). Die Einsprungsadresse wird ebenfalls mit Nullen überschrieben, denn am Ende einer jeden SYS-Datei steht sie bereits: 4D00h. Das ist das Bytemuster 02 02 00 4D im letzten Sektor (der hier



nicht abgebildet ist, um nicht eine neue Infoseite zu beginnen). In Abb. 2a stehen diese Codes mitten im Sektor.

Auch bei diesem Beitrag habe ich das Gefühl, Euch eher abgeschreckt als ermutigt zu haben. Wenn Ihr Euer DOS-Original im Panzerschrank laßt und nur mit Kopien arbeitet, darf aber gerne alles schiefgehen. Und es wird einiges schiefgehen, das walte Murphy. Nach zwei Stunden Training, zwei Litern Schweiß und ebensoviel Bier habt Ihr es aber im Griff. Merke: Die teure Disco geht nur mit Hardware von der Art eines Vorschlaghammers kaputt.

		↓	1	2	3	4	5	6	7	8	9	A	B	
DRV	00	010B	0C40	C3C2	4BC3	0946	C3F2	4501	0B2D	...	\$..	K..	F..	E..-
O	10	40C3	0044	3E43	EFC3	DB4A	0121	3E40	0000	\$..	D>	C...	J..!	>\$..
OH	20	0032	1E08	530A	0500	52FF	FF00	0000	0000	.2..	S...	R.....		
	30	0000	00F5	07DC	E847	F1C9	0000	0001	1F63	.....	G.....			
DRS	40	407A	CD68	407B	F50F	0F0F	0FCD	7140	F1E6	\$...	\$.....	\$..		
5	50	0FC6	9027	CE40	2777	23C9	3B3B	1700	0112	...	'..\$'	#..;		
5H	60	0B43	0000	1123	0323	0A02	0000	C3B0	4501	.C...	#..#.....	E.		
	70	0000	0143	6B43	A540	0000	0000	0000	5A11	...	C..C..\$.....	Z.		
	80	2303	230A	0200	0011	02FF	0100	0100	0000	#..#.....				
	90	00FF	00FF	0100	0100	0000	00FF	00FF	0100	.....				
	A0	0100	0000	00FF	0071	4300	0000	0004	0100	.....	C.....			
	B0	0100	0100	000D	0D01	30B2	4300	0000	FFFF	.....	O..C.....			
FRS	C0	FC4C	0000	00FF	FF00	0000	0000	FFFF	0000	.L.....				
O	D0	0000	00FF	FF00	0080	2800	0042	0000	FF00	.....	(..B.....			
OH	E0	0000	0040	0000	00FF	FF01	0000	443E	23EF	...	\$.....	D>#.		
	F0	0000	3E63	EFC8	F53E	26EF	C309	463E	65EF	..>.....	>&...F>..			

Abb. 1

DRV	00	0162	FC4D	3063	2C30	642C	3065	2C30	662C	...	M0..	O.,	O.,	O.,
O	10	3067	2C30	6820	2D20	456E	6465	206D	6974	O.,	O..-	E.....		
OH	20	203C	4E45	5720	4C49	4E45	3E03	527D	636B	.<NEW.LINE>.	R...			
	30	6B65	6872	6164	7265	7373	653F	2028	4865	.....	?.	(H.		
DRS	40	7B3B	2066	7D72	2052	4554	206E	7572	203C	;	.....	RET.....	<	
161	50	4E45	5720	4C49	4E45	3E20	6569	6E67	6562	NEW.LINE>.....				
A1H	60	656E	2903	0202	004D	462C	3061	2C30	622C	..)	....	MF,	O.,	O.,
	70	3063	2C30	642C	3065	2C30	662C	3067	2C30	O.,	O.,	O.,	O.,	O.,
	80	6820	2D20	456E	6465	206D	6974	203C	4E45	..-	E.....	<NE		
	90	5720	4C49	4E45	3E27	0DB0	B1B2	B3B0	2009	W.LINE>'	.....			
	A0	4445	4642	0930	3348	0DB0	B1B2	B4B0	2054	DEFB.03H.....	T			
	B0	4558	5433	0944	4546	4D09	2752	7D63	6B6B	EXT3.DEFM.'	R....			
FRS	C0	6568	7261	6472	6573	7365	3F20	2848	6578	.....	?.	(H..		
1	D0	3B20	667D	7220	5245	5420	6E75	7220	3C4E	;	.....	RET.....	<N	
1H	E0	4557	204C	494E	453E	2065	696E	6765	6265	EW.LINE>.....				
	F0	6E29	270D	B0B1	B2B5	B020	0944	4546	4209	.)'	.....	DEFB.		

Abb. 2a

DRV	00	3063	0100	FE4D	2C30	642C	3065	2C30	662C	O....	M,	O.,	O.,	O.,
O	10	3067	2C30	6820	2D20	456E	6465	206D	6974	O.,	O..-	E.....		
OH	20	203C	4E45	5720	4C49	4E45	3E03	527D	636B	.<NEW.LINE>.	R...			
	30	6B65	6872	6164	7265	7373	653F	2028	4865	.....	?.	(H.		
DRS	40	7B3B	2066	7D72	2052	4554	206E	7572	203C	;	.....	RET.....	<	
135150		4E45	5720	4C49	4E45	3E20	6569	6E67	6562	NEW.LINE>.....				
547H60		656E	2903	0000	0000	0000	0000	0000	0000	..)	.....			
	70	0000	0000	0000	0000	0000	0000	0000	0000	.....				
	80	0000	0000	0000	0000	0000	0000	0000	0000	.....				
	90	0000	0000	0000	0000	0000	0000	0000	0000	.....				
	A0	0000	0000	0000	0000	0000	0000	0000	0000	.....				
	B0	0000	0000	0000	0000	0000	0000	0000	0000	.....				
FRS	C0	0000	0000	0000	0000	0000	0000	0000	0000	.....				
1	D0	0000	0000	0000	0000	0000	0000	0000	0000	.....				
1H	E0	0000	0000	0000	0000	0000	0000	0000	0000	.....				
	F0	0000	0000	0000	0000	0000	0000	0000	0000	.....				

Abb. 2b



# DAS PROBLEM, 'DD SD DD ODER DOCH SD'

=====

Othmar STARK MÖDLING AUSTRIA

Mit dem Befehl 'DISK,A bis P'habe ich es gelöst.

z.Beiispiel:

```
DISK,1=A      'DRIVE 1 IST SD EIB=2 AEIV=2
DISK,2=B      'DRIVE 2 IST DD EIB=2 AEIV=2
DISK,1=C      'DRIVE 1 IST DD EIB=3 AEIV=3
```

Durch diese Befehle braucht man nicht mehr die PD auf d.SYSTEM-DISK ändern.

## ÄNDERUNG

=====

- 1.) SUPERZAP O.A.LADEN
- 2.) DD NEW-LINE
- 3.) F,14,28,07,28 NEW-LINE
- 4.) MOD00 NEW-LINE
- 5.) 7 BYTE Ändern und auf SYSTEM-DISK abspeichern.
- 6.) RESET (Neustart).

DRV	00	1428	0728	0A02	0000	1402	1830	5328	1202	..(.(.....0S(...
0	10	0003	1802	1830	5328	1203	0003	1802	1830	.....0S(.....0
0H	20	5328	2406	0043	1802	1428	0728	0A02	0004	S(\$..C...(. (.....
	30	1402	1428	0728	1404	0044	1402	1830	5328	...(. (....D....0S(
DRS	40	1203	0007	1802	1830	5328	2406	0045	1802	.....0S(\$..E..
168	50	2850	0750	0A02	0000	2802	2850	0750	1404	(P.P....(. (P.P..
ABH	60	0040	2804	3060	5350	1203	0003	3003	3060	.S(.0.SP....0.0.
	70	5350	2406	0043	3006	1148	1328	1202	0005	SP\$..C0..H. (....
TRK	80	1102	1190	5350	1202	0003	1102	1128	1328	....SP.....(.(
9	90	0A02	0004	1102	114A	5052	1204	0003	1106	.....JPR.....
9H	A0	5120	2020	2020	2020	2051	5220	2020	2020	0.....QR.....
	B0	2020	2052	5320	2020	2020	2020	2053	5420	...RS.....ST.
TRS	C0	2020	2020	2020	2054	5520	2020	2020	2020	.....TU.....
6	D0	2055	5620	2020	2020	2020	2056	1150	0650	.UV.....V.P.P
6H	E0	0A02	0000	1102	5820	2020	2020	2020	2058	.....X.....X
	F0	1143	534F	2408	0043	1102	CACB	CCA0	BBB2	.CSD\$..C.....

Die Daten v.DISK-Befehl sind wie oben ersichtlich auf Spur 9.



ROM 1 ENTSPRICHT '2532' !! NICHT 2732 !! (ZU LANGSAM UND NICHT PINKOMP.)  
 Anstatt der vielen '0D' im ROM eine kleine SUB-Routine einbauen.  
 Die Adresse 058DH auf 0112H im ROM ändern, dadurch wird bei der Initialisierung  
 automatisch die Adresse im Arbeitsspeicher verbogen.

Die alte Adresse (058DH) kann man trotzdem verwenden mit 'POKE 4026H/4027H'.

ROM-SPEICHER

```

0100 A72B C319 1A52 4541 4459 2000 0000 0000 .(...READY.....
0110 0D0D 79F5 CDD1 0520 FBF1 32E8 37C9 000D .....2.7...
0120 0D0D 0D0D 0D0D 0D0D 0D0D 0D0D 001E 2CC3 .....
0130 A219 D7AF 013E B001 3E01 F5CF 28CD 1C2B .....>...>...(<...+
0140 FE80 D24A 1EF5 CF2C CD1C 2BFE 30D2 4A1E ...J...>...+0.J.
0150 16FF 14D6 0330 FBC6 034F F187 5F06 027A .....0...0..._...
0160 1F57 7B1F 5F10 F879 8F3C 47AF 378F 10FD .W...<B.7...
0170 4F7A F63C 571A B7FA 7C01 3E80 47F1 B778 D...<W...>B...
0180 2810 12FA 8F01 792F 4F1A A112 CF29 C9B1 (...../0.....)..
0190 18F9 A1C6 FF9F E5CD 8D09 E118 EFD7 E53A .....:
01A0 9940 B720 06CD 5803 B728 11F5 AF32 9940 .S....X..(....2.S
01B0 3CCD 5728 F12A D440 77C3 842B 212B 1922 <.W(*.S...(!(.
01C0 2141 3E03 32AF 40E1 C93E 1CCD 3A03 3E1F !A>.2.S...>...>.
01D0 C33A 03ED 5F32 AB40 C921 01FC CD21 0206 ..._2.S...!...!..
01E0 0B10 FE21 02FC CD21 0206 0B10 FE21 00FC ...!...!...!...!..
01F0 CD21 0206 5C10 FEC9 E521 00FB 181B 7ED6 !!..B...!.....

```

```

0600 FE09 2842 FE19 2839 FE0A C0D1 7778 B728 ..(B...(9.....(
0610 CF7E 23CD 3300 0518 C7CD C901 41E1 E5C3 ..#.3.....A...
0620 E005 CD30 062B 7E23 FE0A C878 B920 F3C9 ...0...+.#.....
0630 78B9 C82B 7EFE 0A23 C82B 3E08 CD33 0004 ...+...#...+>...3..
0640 C93E 17C3 3300 CD48 03E6 072F 3CC6 085F .>...3..H.../<..._
0650 78B7 C83E 2077 23D5 CD33 00D1 051D C81B ...>...#..3.....
0660 EF37 F53E 0D77 CD33 003E 0FCD 3300 7990 .7.>...3.>...3...
0670 47F1 E1C9 D3FF 21D2 0611 0040 0136 00ED B.....!.....S.6..
0680 B03D 3D20 F106 2712 1310 FC3A 4038 E604 .==...?.....:58..
0690 C275 0031 7D40 3AEC 373C FE02 DA75 003E ...1.S;.7<.....>
06A0 0132 E137 21EC 3711 EF37 3603 0100 00CD .2.7!.7..76.....
06B0 6000 CB46 20FC AF32 EE37 0100 423E 8C77 ...F...2.7..B>..
06C0 CB4E 28FC 1A02 0C20 F7C3 0042 0118 1AC3 .N(.....B.....
06D0 AE19 C396 1CC3 781D C390 1CC3 D925 C900 .....%...
06E0 00C9 0000 FBC9 0001 E303 0000 004B 4907 .....KI..
06F0 5804 003C 0044 4F06 1201 4300 0050 52C3 X..<.DD...C..PR.

```

Nun dieses geänderte Programm (4KByte) kann man in den EPROM 2532 schießen  
 und anstelle des ROM 1 einsetzen.

Falls ein Interessent keinen EPROMMER hat, bin ich gerne bereit diesen kostenlos  
 zu schießen, bei Zusendung eines Eprom '2532'.

SUBROUTINE

```

0112 79 LD A,C y
0113 F5 PUSH AF .
0114 CDD105 CALL 05D1H ...
0117 20FB JR NZ,0114H .
0119 F1 POP AF .
011A 32E837 LD (37E8H),A 2.7
011D C9 RET .

```



## Auch ein GENIE kann irren

Im Heft 6 unseres Club-Infos stellte unser Clubkamarad Paul-Jürgen Schmitz fest, daß das Genie hinter dem Komma nicht mehr richtig rechnen kann.

Recht hat er!

Aber mein Genie bekommt wesentlich bessere Ergebnisse hin als seins.

In seiner Schleife mit einfacher (Un)genauigkeit stehen ab der dritten Stelle alles Neunen. Diese bekommt man durch runden weg. Dafür kennt das Genie den PRINT USING - Befehl. In dem von mir durchgerechneten Beispiel kommt die erste Ungenauigkeit in der dritten Stelle nach dem Komma erst nach dem 2593sten Durchlauf auf!

Der erste Fehler in der 2.Nachkommestelle trat erst nach 10722 Durchläufen auf!! Und das reicht doch wohl für die meisten Anwendungen.

Die ob Ergebnisse lassen sich auch beim Rechnen mit doppelter Genauigkeit nicht verbessern.

Beim Rechnen mit doppelter Genauigkeit reicht es nicht zu sagen (oder zu programmieren), daß das Ergebnis doppelt genau sein soll. Auch die Operanden müssen als doppelt genau angegeben werden. (Oft reicht es, nur einen Operanden als doppelt genau anzugeben - aber sicher ist sicher.)

Variable kann man entweder per DEFDBL oder durch das #-Zeichen kennzeichnen. Eine Zahl kann man durch Anhängen des #-Zeichens als doppelt genau kennzeichnen.

Hierzu einige Beispiele:

```
D=1/3      ergibt : .333333
D#=1/3     ergibt : .33333333432674408
D#=1#/3#   ergibt : .33333333333333333
```

```
A =1+0.9999 ergibt : 1.9999
A#=1+0.9999 ergibt : 1.999900102615356
A#=1+0.9999# ergibt : 1.9999
```

```
10 DEFDBL
20 FORI=1TO12
30   D=D+.01
40   PRINTD
50 NEXTI
READY
RUN
 9.999999776482582D-03
.01999999955296516
.02999999932944775
.03999999910593033
.04999999888241291
.05999999865889549
.06999999843537808
.07999999821186066
.08999999798834324
.09999999776482582
.1099999975413084
.119999997317791
```

```
10 DEFDBL
20 FORI=1TO12
30   D=D+.01
40   PRINTUSING"#.###":D
50 NEXTI
READY
RUN
 0.01
 0.02
 0.03
 0.04
 0.05
 0.06
 0.07
 0.08
 0.09
 0.10
 0.11
 0.12
```

```
10 DEFDBL
20 FORI=1TO12
30   D=D+.01#
40   PRINTD
50 NEXTI
READY
RUN
.01
.02
.03
.04
.05
.06
.07
.08
.09
.1
.11
.12
```



# Computervertrieb Udo Jourdan

Geniale Systeme – Zubehör von A-Z

Herrn  
Peter Spieß  
Trugenhofenerstr. 27

D-8859 Rennertshofen 1

Datum: 02.08.84

Betr.: Clubpreisliste Teil 1 von 2

Sehr geehrte Clubkamerad Peter ,

untenstehend erhältst du den ersten Teil der Clubpreisliste . Die Preise verstehen sich inkl. Mwst. zzgl. Porto . Ich hoffe du hast verständnis dafür, daß ich nicht alle von mir vertriebenen Artikel mit Rabatt weitergeben kann.

---- STAR - Drucker ----	Verkauf	Clubpreis
Powertype Typenrad	1598.-	1448.-
Gemini 10 x oder i	1048.-	938.-
Gemini 15 x oder i	1398.-	1248.-
Delta 10	1498.-	1348.-
Delta 15	1998.-	1828.-
Radix 10	2198.-	1998.-
Radix 15	2628.-	2428.-
Genie 16 B	5648.-	5248.-
Genie III	6398.-	5898.-
Farbmonitor ct900sr14	928.-	858.-
ct900mr14	1868.-	1748.-
ct900hr14	2198.-	1998.-
Floppygehäuse mit Netzteil	278.-	248.-
Controller steckfertig SD	398.-	358.-
DD Controller zum Aufstecken	218.-	178.-
Buskabel 50 POL mit Stecker	85.-	55.-
Floppykabel für 2 Lw	85.-	55.-

SEITE 1 von 2



# Computervertrieb Udo Jourdan

Geniale Systeme – Zubehör von A-Z

Herrn  
Peter Spieß  
Trugenhofenerstr. 27

D-8859 Rennertshofen 1

Datum: 02.08.84

Betr.: Clubpreisliste Teil 2 von 2

Sehr geehrte Clubkamerad Peter ,

untenstehend erhältst du den zweiten Teil der Clubpreisliste . Die Preise verstehen sich inkl. Mwst. zzgl. Porto . Ich hoffe du hast verständnis dafür, daß ich nicht alle von mir vertriebenen Artikel mit Rabatt weitergeben kann.

	Verkauf	Clubpreis
Doppelfloppy TCS 400/2 FC	1868.-	1728.-
TCS 800/2 FC	2598.-	2398.-
Einzelfloppy TCS 400/1 FC	1348.-	1248.-
TCS 800/1 FC	1748.-	1648.-
Colorfloppy 1 Laufwerk	1168.-	1068.-
2 Laufwerke	1678.-	1528.-
---- Einzellaufwerke Slim Line ----		
TEC 40/1 DD	648.-	598.-
40/2 DD	868.-	798.-
80/1 DD	848.-	788.-
80/2 DD	948.-	888.-
BASF 6128 DS/DD 40	848.-	798.-
6138 DS/DD 80	1028.-	958.-

----- Lieferzeiten möglich -----

SEITE 2 von 2



## Vom Umgang mit "Fremdsprachen"

Holger Mays Ausführungen in dem Artikel "Das Sechsserspiel" (Info 8/84), aber auch schon mein Briefwechsel mit ihm und Peter Spieß gaben mir sehr zu denken. Haben die Beiträge der Assembler-Freaks im Club eigentlich ein Publikum? Nun, sie haben, wie ich inzwischen sicher weiß. Drückt mir aber jemand eine hebräische "Jerusalem Post" in die Hand, komme ich mir befangen und schrecklich ungebildet vor. Wenn mir mein Freund, der mir seinerzeit Assembler einpeitschte, schreibt, ich soll mit dem RAS den MUX auf den CAS löten (aber schön fest), dann kriege ich im Kopf einen ?ST-Error, denn der String ist mir zu komplex.

Holger findet selber einen begrüßenswerten Kompromiß, indem er die BASIC-Freunde ermuntert, ohne andere einschränken zu wollen. Da ist noch ein möglicher Kompromiß, und hier folge ich einer Anregung, die mir Paul-Jürgen Schmitz gab: Da all' diese Maschinenprogramme mehr oder weniger brauchbar sind und teilweise sogar die Arbeit in BASIC erleichtern, sollten auch diejenigen sie benutzen können, für die Assembler eine Fremdsprache ist. Nämlich nach Vorlage als Setzer einer Druckerei eine "Jerusalem Post" eintippen ist gar nicht so schwierig.

Deshalb möchte ich in diesem Beitrag etwas über den Umgang mit EDTASM sagen. Dieses Dienstprogramm befähigt den User, Assembler-Listings einzugeben, so wie es der Interpreter für BASIC tut. Als Beispiel stelle ich eine unschuldige kleine Routine vor, die auf eine bewußt einfache (und damit umständliche) Weise den Bildschirm löscht (s. Listings). Sie wird von BASIC aus mitUSR aufgerufen.

Da (sofern mit EDTASM erstellt) drei verschiedene Formen eines Listings vorliegen können, möchte ich zunächst die Unterschiede erklären. Im Listing 1 ist außer dem Programmcode, einer symbolischen Adresse (LOOP) und den Angaben über den Beginn des Ladebereichs (ORG) und das Ende des Programms (END) nichts enthalten, weil weitere Informationen eben nicht von Interesse sind. Dabei stehen in der linken Spalte die Symbole (hier gibt es nur eins), daneben die sog. Operation Codes (oder einfach Opcodes). Sie kennzeichnen die Art des Befehls. Rechts daneben folgen die Operanden, sofern welche gebraucht werden (RET braucht keinen). Die Spalte ganz rechts ist Kommentaren vorbehalten, die mit REM in BASIC eine Parallele haben. Sie müssen immer durch ein Semikolon vom Programmtext getrennt sein. Es darf kein Programmtext mehr in dieser Zeile folgen (genau wie bei REM).

Im Listing 2 erscheinen zusätzlich Zeilennummern. Das ist sinnvoll, wenn im begleitenden Text auf bestimmte Programmstellen Bezug genommen wird, die man dann mit dieser Nummer benennt. Das Listing 3 zeigt ausserdem ganz links die Ladeadresse eines jeden Befehls und den Wert, der nach dem Laden des fertigen Programms in den jeweiligen Bytes steht (beides in Hex). Die Adressen und die Hexcodes werden nicht eingegeben; EDTASM ist nur so freundlich, dem interessierten Programmierer diese Angaben bei einer Art des Listens mitzuliefern.

Das Listing 4 schließlich (das EDTASM in dieser Form nicht mehr erzeugen kann; es ist eine Bildschirm-Hardcopy) enthält zusätzlich hinter jedem Befehl die Angabe des nächsten Tastendrucks, der erforderlich ist. Wie man sieht, werden die Spalten jeder Zeile mit dem Rechtspfeil voneinander getrennt. Die Zeile wird mit NEW LINE beendet wie in BASIC.

Nun muß man aber EDTASM erst einmal zu Laufen bringen. Das Laden und Starten geschieht in der für Maschinenspracheprogramme üblichen Weise: Von BASIC aus mit dem SYSTEM-Befehl oder unter DOS durch Eingabe des Filenamens. EDTASM meldet sich wie im Listing 4 zu sehen und erwartet eine Eingabe. Zum Eintippen eines Programms lautet sie I (insert). Ohne weitere Parameter nimmt EDTASM nun an, daß die erste Zeilennummer 100 lauten und das Inkrement 10 sein soll. Die erste Zeilennummer (wie auch alle weiteren) wird automatisch ausgegeben (wie nach AUTO in BASIC), und der Cursor erscheint dahinter mit einem Blank Zwischenraum.

Jetzt wird der Programmtext eingegeben wie im Listing 4 beschrieben. Wann wir damit fertig sind, erkennt EDTASM nicht von selbst, deshalb wird auch die Zeile 200 erwartet und ihre Nummer angezeigt. Man verläßt jetzt einfach mit BREAK den Insert-Modus. Zeile 200 wird damit einfach wieder "vergessen". Man kann mit BREAK auch mitten in der Arbeit



unterbrechen, um etwa einen Fehler sofort zu korrigieren (s. u.). Mit I (und evtl. einer Zeilennummer dahinter) geht es dann weiter.

Zur Kontrolle kann man mit dem P-Kommando den Text listen. Die Syntax ist ganz ähnlich wie bei LIST in BASIC. P120 listet die Zeile 120, P. die gerade aktuelle Zeile und P130:150 diesen Bereich. P allein zeigt eine Bildschirmseite von der aktuellen Zeile ab an. H statt P tut im Prinzip dasselbe auf dem Drucker. Bei P und H gibt es außerdem eine Abkürzung für die erste und die letzte Zeile: "Liste Anfang bis Ende" heißt P#:\*.

Das Programm wäre jetzt fertig - wenn man sich nicht vertippt hätte. Die Edition ist dem Verfahren in BASIC ebenfalls sehr ähnlich. Mit E und nachfolgender Zeilennummer wird die Zeile zur Edition bereitgestellt. Die Editor-Unterbefehle (L, I, K, A usw.) entsprechen genau denen von BASIC. Die notwendigen Korrekturen können jetzt durchgeführt werden.

Fehler werden aber häufig nicht gleich bemerkt. Wie BASIC erst beim Programmlauf Fehler detektiert, so findet sie EDTASM erst beim Assemblieren. Das ist der Vorgang, bei dem aus diesen Abkürzungen englischer Vokabeln (LD für load usw.) und den Zahlencodes (hier z. B. 3COOH) oder Registerbezeichnungen (z. B. HL) ein für den Z80 verständliches Programm gebastelt wird.

Das Assemblieren wird mit A veranlaßt. Ohne weitere Parameter rollt dann auf dem Bildschirm der Programtext wie in Listing 3 ab. Zusätzlich können gegf. Fehlermeldungen erscheinen. Am Ende des Textes wird die Fehlerzahl fünfstellig angegeben (die Autoren von EDTASM müssen ganz schön pessimistisch gewesen sein). Je nach Version des Programms kann auch eine Angabe des freigebliebenen Speicherplatzes folgen. Schließlich wird noch ein Verzeichnis der Symbole ausgegeben, in dem die Hexzahl erscheint, für die ein Label (= Symbol) steht (hier steht LOOP für die RAM-Adresse 6006h), rechts daneben die Zeile, in der es definiert ist und wiederum daneben die Zeilen, in denen es vorkommt. Je nach Version hat die Symboltabelle vielleicht auch ein anderes Gesicht.

Der Programmname, unter dem das File auf Cassette gespeichert wird (was EDTASM beim Assemblieren erledigt), muß in der Tape-Version mit dem Befehl A NAME eingegeben werden. Das Blank zwischen dem A-Kommando und dem Programmnamen NAME ist obligatorisch. Der Name darf bis zu sechs Zeichen enthalten. Wird nur A eingegeben, kriegt das File den Namen NO-NAME. Die Disk-Version führt den User im Dialog, so daß ich hierzu nichts erklären muß.

Das A-Kommando läßt ein paar Parameter zu, die sehr hilfreich sind. Sie werden in der Tape-Version mit Querstrich an den Filenamen angehängt. Für den BASIC-Spezialisten, der ein Programm aus dem Info nur Abtippen möchte, sei hier der Wichtigste genannt: Mit A/WE (oder A NAME/WE; für wait on error) wird veranlaßt, daß die Assemblage beim Auftreten eines Fehlers stoppt. Man kann jetzt mit BREAK unterbrechen und die betreffende Zeile edieren und berichtigen. Jede andere Taste führt zur Fortsetzung. Wie in BASIC werden auch hier nur Schreibfehler u. dergl. erkannt, keine logischen Fehler. EDTASM kann nicht ahnen, wie groß die Winkelsumme im Dreieck ist.

Es gibt eine ganze Anzahl weiterer Befehle zur direkten oder programmierten Eingabe. Eine Aufzählung würde aber zu weit führen. Mit den hier Aufgezählten kann der Leser jedes veröffentlichte Programm eingeben und zum Laufen bringen. Bleibt nur die Frage offen, wo man EDTASM hernehmen soll, wenn man es noch nicht hat. Ein billiger Weg ist im Info 4/84, S. 11 nachzulesen.

Arnulf Sopp, Tel. 0451-791926



```

1:
    ORG      6000H          ;H für hex
    LD       DE,4000H      ;Ende Bildschirmspeicher +1
    LD       HL,3C00H      ;Anfang Bildschirmspeicher
LOOP  LD      (HL), ' '    ;Blank dorthin laden
    INC      HL            ;nächste Bildschirmstelle
    RST      18H          ;vergleicht HL und DE
    JR       NZ,LOOP      ;weiter, falls Ende noch
                        ;nicht erreicht
    RET                      ;zurück ins BASIC
    END

```

```

2:
00100      ORG      6000H          ;H für hex
00110      LD       DE,4000H      ;Ende Bildschirmspeicher +1
00120      LD       HL,3C00H      ;Anfang Bildschirmspeicher
00130 LOOP  LD      (HL), ' '    ;Blank dorthin laden
00140      INC      HL            ;nächste Bildschirmstelle
00150      RST      18H          ;vergleicht HL und DE
00160      JR       NZ,LOOP      ;weiter, falls Ende noch
00170                        ;nicht erreicht
00180      RET                      ;zurück ins BASIC
00190      END

```

```

3:
6000      00100      ORG      6000H          ;H für hex
6000 110040 00110      LD       DE,4000H      ;Ende Bildschirmspeicher +1
6003 21003C 00120      LD       HL,3C00H      ;Anfang Bildschirmspeicher
6006 3620   00130 LOOP  LD      (HL), ' '    ;Blank dorthin laden
6008 23     00140 INC      HL            ;nächste Bildschirmstelle
6009 DF     00150      RST      18H          ;vergleicht HL und DE
600A 20FA   00160      JR       NZ,LOOP      ;weiter, falls Ende noch
        00170                        ;nicht erreicht
600C C9     00180      RET                      ;zurück ins BASIC
0000      00190      END

```

00000 mal gepennt  
34747 Zeichen verfügbar

LOOP 6006 00130 00160

4:

Genie-DOS Editor-Assembler

\*I nl

```

00100 rp      ORGrp  6000Hrp rp      ;rp = Rechtspfeil nl
00110 rp      LDrp   DE,4000Hrp      ;nl = NEW LINE nl
00120 rp      LDrp   HL,3C00Hrp      ;brk = BREAK nl
00130 LOOPrp  LDrp   (HL), ' 'rp      ; ... nl
00140 rp      INCrp  HLrp   rp      ; ... nl
00150 rp      RSTrp  18Hrp   rp      ; ... nl
00160 rp      JRrp   NZ,LOOPrp      ; ... nl
00170 rp      rp      rp      ;reine Kommentarzeile nl
00180 rp      RETrp  rp      rp      ; ... nl
00190 rp      ENDnl
00200 brk
*_

```

Erklärung zu

;rp = Rechtspfeil nl  
 ;nl = NEW LINE nl  
 ;brk = BREAK nl  
 ; ... nl  
 ; ... nl  
 ; ... nl  
 ; ... nl  
 ;reine Kommentarzeile nl  
 ; ... nl

\* = EDTASM-Prompt



## Rezept gegen Computerfiber

Wie vor kurzem einschlägige Forschungsarbeiten nachgewiesen haben, breitet sich die tückische und gefährliche Krankheit epedemieartig weiter aus: Besonders Jugendliche und junge Erwachsene (bis ca.35 Jahre), vereinzelt aber auch ältere Menschen und Pensionäre werden davon befallen: vom Computerfiber (Fibra dualis).

Symptome: Heiße Stirn und zitterige Hände von der Eingabe an der Tastatur; Augenbrennen und -flimmern (vom Monitor); akute Kontaktstörungen und Nervosität; Schwindelgefühle; Schlaflosigkeit und Essensunlust; Neigung zur Depression.

### 1. Therapie:

Der von der Krankheit Befallene nehme einen kleinen langen Schraubendreher und einen isolierten Seitenschneider. Die ca.8 Schrauben am Boden des Video Genie entfernen und vorsichtig den Deckel abziehen. Vorher den Netzstecker ziehen!

Jetzt sieht man rechts hinten das Netzteil im schwarzen Kasten.

Die vier Schrauben oben lösen und Gitterverkleidung abnehmen. Vergewissern, daß der Netzstecker auch wirklich aus der Steckdose gezogen ist!

Jetzt mit dem Seitenschneider ca.3 cm der blauen Leitung des Netzkabels von der Trafoseite her abtrennen und an sicherem Ort (vor Kindern schützen) deponieren. Von der Netzleitung her das Ende der Leitung isolieren.

Zur Sicherheit die schwarze Leitung einfach durchtrennen.

GESCHAFFT! Jetzt zurücklehnen und entspannen; danach mindestens eine Stunde spazieren gehen.

Der Computer ist jetzt entschärft.

### 2. Therapie: (Alternativ zur 1.Therapie und in schweren Fällen anzuwenden).

Computer wie oben öffnen und dann die länglichen Platten mit den kleinen schwarzen Klümpchen abschrauben. Die Tastatur abmachen und den Kindern zum Spielen geben: Die Tasten lassen sich z.B. bei Mensch-ärgere-Dich-nicht als Figur verwenden.

Eventuell hartnäckige Leitungsverbindungen abschneiden.

Jetzt: Gehäusedeckel aufschrauben und in die verbleibende Aussparung Blumen einpflanzen; diese Methode ist nicht nur (nachweislich!) nervenschonend, sondern auch überaus dekorativ!

ACHTUNG: Für eventuell wiedererwartend/auftretende Schäden kann der Autor keine Haftung übernehmen.

Good Byte!

Paul-Jürgen Schmitz

22.8.1984



Uwe von Scheidt

2850 Bremerhaven, xx.xx.1984  
Ströddacker 45c  
☎ 0471/85418

TRS80 Mod.1 48K, 2 BASF LW (SSSD), EPSON RX80 F/T

Hallo Clubfreunde,  
da in den letzten Club - Infos immer wieder die Basicprogrammierer angesprochen wurden, möchte ich ein Programm bringen welches sich an die User der Spiele wendet und ihnen ermöglicht ihre HIGHSCORES zu speichern. Das Programm habe ich in sehr einfacher Form in Englisch bekommen, übersetzt und erweitert.  
Ich hoffe einigen unter euch damit einen Wunsch erfüllt zu haben.

PS. Ich habe ein Angebot über 10 \* 2000 Blatt Endlospapier erhalten. Der Preis für 2000 Blatt würde 25 DM + Porto betragen.  
Sollte jemand Interesse daran haben, bitte bei mir melden.

```
100 REM *****
110 REM Highscoreprogramm
120 REM *****
130 CLEAR5000:DEFINTC-Z:DEFSTRA
140 DIMA(100),B(100):EN=0
150 CLS
160 PRINT$13,"H i g h s c o r e p r o g r a m m"
170 FORI=24TO93:SET(I,3):NEXT:I=320
180 PRINT$200,"Bitte waehle eine der folgenden Funktionen"
190 PRINT$332,"<1> Ein neues Programm eingeben"
200 PRINTTAB(12)"<2> Einen neuen Highscore eintragen"
210 PRINTTAB(12)"<3> Liste sortieren"
220 PRINTTAB(12)"<4> Liste von Disk lesen"
230 PRINTTAB(12)"<5> Liste auf Disk speichern"
240 PRINTTAB(12)"<6> Liste auf Bildschirm"
250 PRINTTAB(12)"<7> Liste auf Drucker"
260 PRINTTAB(12)"<8> Einzelnes Programm suchen"
270 A=INKEY$:IFA=""THEN270ELSEIFASC(A)<49ORASC(A)>56THEN270
280 ONVAL(A)GOSUB310,360,420,530,630,690,930,590
290 GOTO150
300 REM NEUES PROGRAMM EINGEBEN
310 CLS:PRINT"Willst du wirklich ein Programm eingeben (J/N)";
320 GOSUB750:IFFL=0THENRETURN
330 EN=EN+1:GOSUB780:A(EN)=HI$
340 CLS:PRINT"Moechtest du noch ein Programm eingeben (J/N) ?";:GOTO320
350 REM HIGHSCORE EINGEBEN
360 CLS:PRINT"Willst du wirklich einen neuen Highscore eingeben (J/N)":GOSUB750
370 IFFL=0THENRETURN
380 CLS:PRINT"Bitte gebe den Namen des Programms ein"
390 GOSUB790:FORI=1TOEN:IFA(I)=HI$THEN400ELSENEXT:CLS:PRINT"Programm nicht in de
r Liste":GOTO900
400 CLS:PRINT"Bitte gebe den neuen Highscore fuer "HI$" ein";:INPUTB(I):RETURN
410 REM LISTE SORTIEREN
420 CLS:PRINT$24,"S o r t i n g":FORI=48TO74:SET(I,3):NEXT
430 I=0:J=0:M=0
440 FORI=1TOEN-1
450 M=I
```



```

460 FORJ=I+1TOEN
470 IFA(J)<A(M) THENM=J
480 NEXTJ
490 A(O)=A(I):A(I)=A(M):A(M)=A(O):B(O)=B(I):B(I)=B(M):B(M)=B(O)
500 PRINT$158,EN-I;:NEXTI
510 RETURN
520 REM LISTE VON DISK LESEN
530 CLS:PRINT"Moechoest du wirklich die Liste laden (J/N) ?";:GOSUB750:IFFL=OTHE
N RETURNELSECLS
540 PRINT"Daten werden geladen"
550 OPEN"I",1,"DATEN/IND:O"
560 INPUT#1,EN:FORI=1TOEN:INPUT#1,A(I),B(I):NEXTI
570 CLOSE1:RETURN
580 REM PROGRAMM SUCHEN
590 CLS:PRINT"Bitte gebe den Namen des gesuchten Programms ein";:GOSUB790
600 FORI=1TOEN:IFA(I)<>HI$THENNEXTI:CLS:PRINT"Programm nicht gefunden":GOSUB900:
RETURN
610 CLS:PRINT"Der Highscore fuer "HI$" ist "B(I)" Punkte":GOSUB900:RETURN
620 REM LISTE AUF DISK LADEN
630 CLS:PRINT"Moechoest du die Liste wirklich abspeichern (J/N) ?";:GOSUB750:IFF
L=OTHE N RETURN
640 CLS:PRINT"Liste wird auf Disk geschrieben"
650 OPEN "O",1,"DATEN/IND:O"
660 PRINT#1,EN:FORI=1TOEN:PRINT#1,CHR$(34);A(I);CHR$(34);B(I):NEXTI:CLOSE1
670 RETURN
680 REM LISTE AUF BILDSCHIRM
690 CLS:PRINT"Name";:PRINT$28,"Highscore";:FORI=0TO7:SET(I,3):NEXT:FORI=56TO73:8
ET(I,3):NEXT
700 Z=0
710 FORI=1TOEN
720 Z=Z+1:PRINT$Z*64+64,A(I);:PRINT$Z*64+92,USING"#####";B(I);:IFZ<13THENNEXTI:
GOTO740ELSEZ=0
730 IFINKEY$<>" "THENFORX=128TO1023STEP64:PRINT$X,STRING$(40," ");:NEXTX:NEXTIELS
E730
740 IFINKEY$<>" "THENRETURNELSE740
750 AA=INKEY$:IFAA="J"ORAA="j"THENFL=1ELSEIFAA="n"ORAA="N"THENFL=0ELSE750
760 RETURN
770 REM eingabe
780 CLS:PRINT"Bitte gebe den Namen ein, dann druecke <ENTER>"
790 PRINT$128,"";:FORI=1TO25:PRINTCHR$(95);:NEXT:HI$=""
800 PRINT$128,"";:I=128
810 IFI=153THEN890ELSEA=INKEY$:IFA=""THEN810ELSEB=ASC(A)
82 IFB=8ANDI>128THENI=I-1:PRINT$I,CHR$(95);:GOTO810
830 IFB=32THENPRINT$I,CHR$(32);:I=I+1:GOTO810
840 IFB>64ANDB<91THENPRINT$I,CHR$(B);:I=I+1:GOTO810
850 IFB>47ANDB<59THENPRINT$I,CHR$(B);:I=I+1:GOTO810
860 IFB>95ANDB<128THENPRINT$I,CHR$(B);:I=I+1:GOTO810
870 IFB=13THEN890
880 GOTO810
890 FORB=15488TOB+I-129:HI$=HI$+CHR$(PEEK(B)):NEXT:RETURN
900 PRINT$128,"Bitte <Enter> druecken"
910 A=INKEY$:IFA=""THEN910ELSEIFASC(A$)=13THENRETURNELSE910
920 REM LISTE AUF DRUCKER
930 LPRINTTAB(5)"Name";:LPRINTTAB(20)"Highscore";:LPRINTTAB(45)"Name";:LPRINTTAB
(60)"Highscore"
940 LPRINTTAB(5)"====";:LPRINTTAB(20)"=====";:LPRINTTAB(45)"=====";:LPRINTTAB
(60)"====="
950 LPRINT
960 EL=EN:EM=0
970 FORI=1TOEN
980 EM=EM+1:EL=EL-1
990 IFEM=2THEN1020
1000 LPRINTTAB(5)A(I);:LPRINTTAB(20)" ";:LPRINTUSING"#####";B(I);:NEXTI
1010 IFEL=0THENLPRINT:GOTO1040
1020 LPRINTTAB(45)A(I);:LPRINTTAB(60)" ";:LPRINTUSING"#####";B(I):EM=EM-2
1030 NEXTI
1040 RETURN

```



## Die Records handhaben

In den beiden Beiträgen "Neuer Dreitastenbefehl ..." und "SYS-Files ..." war bereits von der Record-Organisation die Rede. Mich läßt das dumpfe Gefühl nicht los, daß ich mich dort etwas zu global, nur für den Experten verständlich ausdrückte. Da meine literarischen Absonderungen letztenendes für die Praxis auch des weniger geübten Infolesers etwas bringen sollen, möchte ich einen weiteren Beitrag zu diesem Thema nachschieben. Wer bei seiner Zapperei möglichst wenige Fehler machen will, muß mit Records einfach umgehen können.

Es mag auf Anhieb paradox klingen, daß ich als Beispiel eine Datei der Systemdiskette nehme, die als einzige eben nicht in Records gegliedert ist: Habt Ihr schon einmal versucht, GDOS/SYS (BOOT/SYS) zu disassemblieren? Disassembler erwarten als allererstes Byte des Files den Code 01. Er signalisiert ein Maschinenprogramm bzw. ein Datenfeld, das ähnlich einem Programm einem ganz bestimmten Speicherbereich zugeordnet ist. In GDOS/SYS findet sich aber nur der Maschinencode dieses Urladers, nichts von Adressen, kein Kenncode 01. Das liegt daran, daß es DOS erst laden muß und selbst vom Microsoft-ROM geladen wird.

Für unser Problem ist zunächst nur der erste Sektor von GDOS, der eigentliche Urlader interessant und hiervon auch nur die ersten 238 Bytes. Der Rest des Sektors enthält einen Copyright-Vermerk und der Rest des Files Daten zur späteren Verwendung.

Zur Wiederholung: Das erste Byte eines Maschinenprogramms lautet immer 01. Das zweite hält die Anzahl der zu diesem Record, dieser Portion des Programms gehörigen Bytes. Dabei steht 00 für 100h (256d). Das dritte und vierte Byte (die bei der Anzahl bereits mitgezählt werden) stellen in der Folge LSB-MSB die Ladeadresse des ersten zum Maschinencode gehörenden Bytes (des fünften) dar. Erst jetzt folgt das eigentliche Programm. Am Ende des Files schließlich finden wir mit dem Kenncode 02 die Einsprungsadresse des Programms. Nach 02 folgt wieder ein Bytezähler. Er lautet ebenfalls 02, weil nur noch die Adresse (2 Bytes) folgt.

Um z. B. für DSMBLR oder DISASSEM aus GDOS ein lesbares File zu machen, müssen wir für Record-Codes sorgen. Es wäre schade, die ersten vier Bytes dafür zu überschreiben. Davor ist dem File aber kein Sektor mehr zugeordnet, den man dafür verwenden könnte. Die Lösung ist simpel: Wir "borgen" uns ein unbenutztes File, das mindestens zwei Sektoren Platz bietet. Die bereits früher erwähnten freien SYS-Dateien eignen sich hervorragend. Hier kommt willkürlich SYS22/SYS zur Anwendung.

In dessen zweiten Sektor kopieren wir den ersten Sektor von GDOS. Das geht gut mit DEBUG oder SUPERZAP. Die letzten vier Bytes des ersten SYS22-Sektors halten jetzt für die Record-Organisation her. Es beginnt mit 01. Weshalb es ausgerechnet mit FE weitergeht, erkläre ich später. Danach folgt die Ladeadresse 4200h, mit dem LSB beginnend.

Die 252 Nullen davor würde der Disassembler aber ebenfalls nicht verzeihen, deshalb werden sie mit entsprechenden Record-Codes zu dem erklärt, was in BASIC REM heißt: Der Code 05 bezeichnet Bereiche auf der Diskette, die Kommentare enthalten und ansonsten ignoriert werden sollen. Es muß auch hier ein Bytezähler folgen, damit DOS "weiß", ab wo es wieder interessant wird. In unserem Beispiel lautet er FAh (250d Bytes bis zum nächsten Record-Header).

Erinnern wir uns: Die letzte Dump-Zeile von GDOS, das ist jetzt der zweite Sektor von SYS22, enthält nur noch den Copyright-Vermerk. Seine



letzten vier Buchstaben (s. Abb.) sind ohnehin hochgestapelt, also können sie für die Einsprungsadresse dienen. Sie wird, wie gesagt, mit der Bytefolge 02 02 1b mitargestellt, wobei in diesem Fall konkret 00 42 für 4200h einzusetzen ist. Zwischen dem Kenncode 02 und dem letzten Zählbyte im ersten Sektor liegen 254 Bytes, und so erklärt sich der Zähler FE dort.

Alles klar? Alles klar.

000000:	00FE	30F3	21EC	3736	FF36	D023	3600	2336	..0.!.76.6.#6.#6	GDOS/SYS, Sektor 0
000010:	0011	0501	D931	E041	21FF	51CD	5242	FE20	.....1.A!.Q.RB.	
000020:	4730	2957	CD52	424F	CD52	425F	1012	CD52	60)W.RB0.RB....R	
000030:	4257	0D0D	2CCC	5542	7E12	130D	20F6	18DB	BW....UBB... ..	
000040:	10F9	CD52	4257	1AFE	A513	D5C8	21E5	42C3	...RBW.....!.B.	
000050:	C342	2C7E	C0D9	060A	21E1	3736	01D5	C57B	.B.B....!.76...ä	
000060:	D612	3803	5F36	0921	EC37	CDCE	42ED	53EE	..B._6.!.7..B.S.	
000070:	3736	1BCD	CE42	3688	11EF	3701	0051	CDD7	76...B6...7..Q..	
000080:	427E	E683	E281	421A	0203	CB4E	C287	42CB	BB....B....N..B.	
000090:	4EC2	8742	CB4E	20EF	CB46	2808	CB4E	20E7	N..B.N..F(..N.	
0000A0:	CB7E	28E6	7E36	D0C1	D1E6	FC20	0C1C	7BD6	.B(.B6..... ..ä.	
0000B0:	2420	0314	1E00	D97E	C9CD	D742	360B	1098	# .....B...B6...	
0000C0:	21DD	427E	FE03	28FB	23CD	3300	18F5	CDD7	!.BB..(.#.3.....	
0000D0:	42CB	4620	FC7E	C93E	063D	20FD	C91C	1F52	B.F.B.>.= ....R	
0000E0:	4553	4554	031C	1F47	2D44	4F53	3F03	0000	ESET...G-DOS?...	
0000F0:	4027	3832	2F38	3420	5443	532F	<u>536F</u>	<u>7070</u>	\$'82/84 TCS/Sopp	

000000:	<u>05FA</u>	2020	2020	2020	2020	2020	2020	2020	..	SYS22/SYS, Sektor 0
000010:	2020	2020	2020	2020	2020	2020	2020	2020		
000020:	4465	7220	436F	6465	2030	3520	6265	2D20	Der Code 05 be-	
000030:	7A65	6963	686E	6574	2065	696E	2020	2020	zeichnet ein	
000040:	4665	6C64	2C20	6461	7320	766F	6D20	2020	Feld, das vom	
000050:	444F	5320	6967	6E6F	7269	6572	7420	2020	DOS ignoriert	
000060:	7765	7264	656E	2073	6F6C	6C2E	2046	4120	werden soll. FA	
000070:	6461	6869	6E74	6572	2069	7374	2064	6173	dahinter ist das	
000080:	5A7B	686C	6279	7465	2C20	6461	6D69	7420	Zählbyte, damit	
000090:	6465	7220	666F	6C67	656E	6465	2020	2020	der folgende	
0000A0:	4B65	6E6E	636F	6465	2030	3120	2873	2E20	Kenncode 01 (s.	
0000B0:	752E	2920	6765	6675	6E64	656E	2020	2020	u.) gefunden	
0000C0:	7769	7264	2E20	2020	2020	2020	2020	2020	wird.	
0000D0:	2020	2020	2020	2020	2020	2020	2020	2020		
0000E0:	2020	2020	2020	2020	2020	2020	2020	2020		
0000F0:	2020	2020	2020	2020	2020	2020	<u>01FE</u>	<u>0042</u>	...B	

000100:	00FE	30F3	21EC	3736	FF36	D023	3600	2336	..0.!.76.6.#6.#6	SYS22/SYS, Sektor 1
000110:	0011	0501	D931	E041	21FF	51CD	5242	FE20	.....1.A!.Q.RB.	
000120:	4730	2957	CD52	424F	CD52	425F	1012	CD52	60)W.RB0.RB....R	
000130:	4257	0D0D	2CCC	5542	7E12	130D	20F6	18DB	BW....UBB... ..	
000140:	10F9	CD52	4257	1AFE	A513	D5C8	21E5	42C3	...RBW.....!.B.	
000150:	C342	2C7E	C0D9	060A	21E1	3736	01D5	C57B	.B.B....!.76...ä	
000160:	D612	3803	5F36	0921	EC37	CDCE	42ED	53EE	..B._6.!.7..B.S.	
000170:	3736	1BCD	CE42	3688	11EF	3701	0051	CDD7	76...B6...7..Q..	
000180:	427E	E683	E281	421A	0203	CB4E	C287	42CB	BB....B....N..B.	
000190:	4EC2	8742	CB4E	20EF	CB46	2808	CB4E	20E7	N..B.N..F(..N.	
0001A0:	CB7E	28E6	7E36	D0C1	D1E6	FC20	0C1C	7BD6	.B(.B6..... ..ä.	
0001B0:	2420	0314	1E00	D97E	C9CD	D742	360B	1098	# .....B...B6...	
0001C0:	21DD	427E	FE03	28FB	23CD	3300	18F5	CDD7	!.BB..(.#.3.....	
0001D0:	42CB	4620	FC7E	C93E	063D	20FD	C91C	1F52	B.F.B.>.= ....R	
0001E0:	4553	4554	031C	1F47	2D44	4F53	3F03	0000	ESET...G-DOS?...	
0001F0:	4027	3832	2F38	3420	5443	532F	<u>0202</u>	<u>0042</u>	\$'82/84 TCS/...B	



## Ein Tip zum "genauen" Rechnen

Lieber Wolfgang Reichelsdorfer! Liebe Clubfreunde!

Nachdem im letzten Beitrag das Thema "Irren ist GENIEal und menschlich" auf die Rechengenauigkeit anspricht (was mich freut) möchte ich hier einen kleinen, aber wirksamen Trick vorstellen, wie man zumindest bei einfacher (Un-)genauigkeit auch ohne Benutzung des USING-Befehls mit exakten Ergebnissen rechnen kann.

```
50 DEFINT I           : ' Damit es schneller geht!
100 FOR I=1 TO 1000
120  A=.01:A=A*100:E=E+A:D=E/100
130  PRINT D;
199 NEXT
```

Mit der Umwandlung in Zeile 120 erhält man zumindest für einfache Berechnungen genaue Ergebnisse. Diese Routine läßt sich z.B. bei Eingaben von Daten oder Berechnungen, bei denen das Ergebnis auch nach dem Komma stimmen muß, einfügen. Natürlich ist das nur eine "Krücke" die zudem den Nachteil hat, nur bei kleinen Summen, die nicht zu einem Überlauf führen, wirksam zu sein.

Bei doppelt genauer Berechnung ist aber Vorsicht geboten, denn kleine Unterschiede werden ja hierbei mitgenommen. Der Fehler kann zwar mit obiger Berechnung und der Zuweisung doppelt genauer Werte (also wie im letzten Beitrag .01# oder auch etwa .0100000000000000) erheblich verringert werden. Vergleiche solcher Werte gegen andere Variable z.B. bei IF-THEN Abfragen sollten aber besser als "größer-gleich" bzw. "kleiner-gleich" ausgeführt werden.

Für Techniker wird insbesondere auch interessant sein, daß man auch den Ergebnissen trigonometrischer und sonstiger Berechnung des BASIC-Interpreters nur ungefähr trauen kann. Manche (nicht aber alle) Taschenrechner können hier genauere Ergebnisse liefern. Für Computerspiele reicht allemal; wer aber die Statik von Brücken berechnet; sollte den Fehler gleich mit einbeziehen.

Good Byte!

Paul-Jürgen Schmitz

17.8.1984



Spooler sind Programme, die bei der Ausgabe von Zeichen auf den Drucker Druckgut zunächst in einem reservierten Speicher auf Eis legen, bis es auf irgendein Signal hin ausgeprintet wird. Auf diese Weise kann man die Warterei umgehen, die damit bei längeren Texten oder Listings verbunden ist.

Einige von Euch haben das Modul HRG 1b für hochauflösende Graphik. Es stellt eigene RAMs von 12 kB zur Verfügung, die beliebig und nicht nur als Bildschirmspeicher genutzt werden können. Folgerichtig hat dazu 1983 ein Dieter Bolz (das geht aus einem nicht angezeigten Copyrightvermerk hervor) einen Spooler geschrieben, der das Druckmaterial in diesem externen Speicher zwischenlagert.

Dieser Spooler hat eine Reihe von Nachteilen, die mich natürlich zu einer Alternative reizten. Am meisten störte mich, daß er bei jedem Interrupt aktiv wird. Auf diese Weise verzögert sich die INT-Behandlung entsprechend, was den ganzen Computer verlangsamt. Obendrein ist der Spooler auch noch brav in Partisanenheimarbeit in die INT-Kette eingeschleift, wie es das Handbuch vorschreibt. Das kostet Bytes. In einem früheren Artikel beschrieb ich, wie man sich das verkneifen kann. Auch die Länge seiner residenten Teile (was nach der Initialisierung im Speicher bleiben muß) ist für meinen Geschmack zu großzügig.

Ein paar weitere Nachteile konnte ich mit Hilfe des EG 64 MBA vermeiden. Da der HRG-Besitzer nicht auch noch unbedingt dieses Bauteil haben muß, ist Herr Bolz dafür allerdings nicht zu zeihen. So nutzt er beispielsweise das Himem (schade drum), weil es ohne MBA nun mal nicht anders zu machen ist. Gleichwohl kann mein Spooler durch Änderung der entsprechenden Adressen ebenfalls an beliebiger Stelle (also auch ohne MBA) arbeiten.

Aufgerufen wird er mit dem Dreitastenbefehl <567> (ähnlich <JKL>, <123>, <DFG>). In einem zurückliegenden Info ist im Zusammenhang mit <.,./> für das Banking beschrieben, wie so ein Befehl zu implementieren ist. Deshalb will ich darauf jetzt nicht mehr eingehen (notfalls bitte anrufen). Dieser Befehl lädt und startet das DOS-Modul SYS26/SYS, das dem Anwender zur freien Verfügung steht. In meinem SYS26 ist noch etliches mehr untergebracht. Deshalb sind die hier nicht interessanten Teile im Assembler-Listing durch \*LIST OFF ausgespart.

Zum Verständnis des Programmablaufs will ich nur das Allernötigste sagen, um bei diesem doch recht komplexen Programm nicht ein ganzes Info zu füllen: Wurde <567> gedrückt, springt die CPU in SYS26 zum Segment SPOOL (Z1. 3680ff.). Dort wird durch einen Vergleich zwischen der Adresse im Tastatur-DCB und der Adresse des Spooler-Tastaturtreibers geprüft, ob er z. Zt. aktiv ist. Falls ja, wollte der Anwender mit erneuter Eingabe von <567> den Spooler löschen. Falls nein, wollte er ihn in Gang bringen. Es erscheint eine entsprechende Meldung, und das Nötige wird veranlaßt:

Beim Einklinken des Spoolers werden der Pufferteil (zwischenspeichern des Druckmaterials) und der Druckteil (ausdrucken) vom Speicherbereich von SYS26 in den Bereich ab 3000h (Level-4-ROM) geladen. Um diese Teile auch ansteuern zu können, müssen zwei Zeiger verbogen werden: Im Drucker-DCB die Adresse der Printer-Routine (es soll nicht mehr geprintet, sondern gepuffert werden) und im Keyboard-DCB die Adresse des Tastaturtreibers.

Wieso gerade diese? Das Ausdrucken während der Interrupts ist ungünstig (s. o.). Zeit dafür ist ohne Verzögerung immer dann, wenn gerade eine Eingabe vom Anwender erwartet wird. Der tippt nämlich auch bei traumhafter Geschwindigkeit immer noch so langsam, daß der Computer jede Menge Zeit hat, nebenher den Drucker mit Material zu beschicken. Dies



scheint mir die einzige Situation zu sein, wo das Drucken keine Zeit stiehlt oder Verzögerungen verursacht. Mit anderen Worten, immer dann, wenn der Cursor blinkt, wird gedruckt. Währenddessen kann man gerne seine Befehle eingeben, der User merkt nichts (außer akustisch) vom derzeitigen Doppeljob seiner Maschine.

In meinem SYS26 wird zusätzlich ein Teil der INT-Bearbeitung bei der Initialisierung des Spoolers mit verschoben. Dieser Teil residiert bei mir sonst ab 0072h, wo jetzt aber der Zeichenpuffer liegt. Das geht natürlich auch nur mit dem MBA. In einem früheren Artikel legte ich dieses Programmsegment in den DOS-Befehlspeicher. Nach dieser Initialisierungsphase springt das Programm an die alte Stelle zurück. Das darf gerne mitten in einem anderen Programm sein. Es wird normal weiterlaufen.

Wenn der Spooler mit <567> aber gelöscht werden sollte, werden die alten Zeiger wiederhergestellt. Das restliche Druckmaterial im Puffer wird ausgedruckt (was man mit <BREAK> unterbrechen kann). Der Bereich 0000-3FFFh des Hauptspeichers, also der Bereich, den der MBA auf RAM umschalten kann, wird anschließend im Segment INIT wieder auf die Normalbelegung geändert. Die Vollzugsmeldung erscheint, das alte Programm läuft ebenfalls ungestört weiter.

Im Pufferteil (PRDRV) wird zunächst geprüft, ob im Puffer noch Platz ist. Falls nein, wird welcher geschaffen, und zwar entweder durch erzwungenes Drucken (hier hat jeder Spooler seine Grenzen) oder durch Umschauen der Daten: Was bereits gedruckt ist, fliegt raus. Anschließend wird das neue Zeichen gespeichert. Der Zähler für noch zu druckende Zeichen (COUNT), der Zeiger auf die nächste freie Adresse (INADR) und der auf die Adresse des als nächstes zu druckenden Zeichens (OUTADR) werden jeweils aktualisiert.

Im Druckerteil (KBDRV, weil der Tastaturtreiber hier seine Arbeit beginnt) wird überprüft, ob es überhaupt etwas zu drucken gibt oder der Drucker bereit ist. Andernfalls geht es normal weiter. Liegt aber Arbeit für den Drucker an, wird das Zeichen geholt und gedruckt, der Zeichenzähler (COUNT) wird erniedrigt und der Zeiger auf das nächste Zeichen (OUTADR) erhöht. Jetzt wird die alte Keyboard-Routine angesprungen. Die beiden Adreßbytes dieses Sprungbefehls dienen gleichzeitig als Speicher für die alte Adresse im DCB, die ja beim Deaktivieren des Spoolers restauriert werden muß.

Der Spooler greift, um arbeiten zu können, in die Keyboard- und die Printer-Routine ein, indem er die Adressen in den beiden DCBs auf seine Routinen abändert. Daraus folgert eine wichtige Einschränkung: Jedes Programm, das selbst eine eigene Tastatur- oder Druckeroutine zur Verfügung stellt, stört den Spooler. In manchen Fällen sind beide einfach nicht vereinbar. Versucht man es dennoch, sind die Folgen bestenfalls harmlos, schlimmstenfalls fatal. Die hier vorliegende Version arbeitet beispielsweise noch nicht mit der JKL-Option zusammen. Verbesserungen werden sich im Lauf der Zeit ergeben, die dann jeweils im Info nachzulesen sind. Wunder darf man nicht erwarten, denn der Spooler ist ein Programm, und wie jedes Programm hat er seine Grenzen.

Mir ist vollkommen klar, daß diese sehr knappe Erklärung auch für den Experten nur ausreicht, um den Programmablauf beim Lesen des Listings nachvollziehen zu können. Um jedoch dem Assembler-Neuling die Routine plausibel zu machen, hätte nur die Form des Fortsetzungsromans eine Chance. Am Telefon werde ich gerne ausführlicher.

Annulf Sopp, Tel. 0451-791926



4D00	00100	ORG	4D00H	
	00110 *LIST OFF			
	00160 *LIST ON			
4D09 FEFC	00170 CP		0FCH	; <567>?
4D0B CA3850	00180 JP		Z, SPOOL	; Spooler
	00190 *LIST OFF			
	01160 *LIST ON			
4DC5 F3	01170 INIT	DI		; Störungen wären fatal
4DC6 0604	01180 LD	B, 04H		; 4 Codes
4DC8 3E0E	01190 LD	A, 0EH		; 1. Code
4DCA D3DF	01200 LOOP1 OUT	(0DFH), A		; bank selection
4DCC 3D	01210 DEC	A		; nächster Code
4DCD 10FB	01220 DJNZ	LOOP1		
4DCF 3D	01230 DEC	A		; 0Ah überspringen
4DD0 D3DF	01240 OUT	(0DFH), A		; 09h ausgeben
4DD2 010036	01250 LD	BC, 3600H		; Zähler f. ROM-Kopie
4DD5 61	01260 LD	H, C		; Quelle und Ziel = 0000h
4DD6 69	01270 LD	L, C		
4DD7 51	01280 LD	D, C		
4DD8 59	01290 LD	E, C		
4DD9 EDB0	01300 LDIR			; ROM auf RAM kopieren
4DDB 70	01310 LD	(HL), B		; (3600h) <- 0
4DDC 1C	01320 INC	E		; Ziel = 3601h
4DDD 01FF09	01330 LD	BC, 09FFH		; Zähler f. zero memory
4DE0 EDB0	01340 LDIR			; (3600h - 3FFFh) = 00h
4DE2 117200	01350 LD	DE, 0072H		; INT-Bearb. im ROM-Ber.
4DE5 011000	01360 LD	BC, 0010H		; Länge INT-Erweiterung
4DE8 CD854F	01370 CALL	MOVINT		; Code übertragen
4DEB DBDF	01380 IN	A, (0DFH)		; reset MBA
4DED 3E08	01390 LD	A, 08H		; "read RAM 0000-2FFFh"
4DEF D3DF	01400 OUT	(0DFH), A		; auf Banking-Port
4DF1 3E0F	01410 LD	A, 0FH		; RESET MBA n. m. R.-Taste
4DF3 D3DF	01420 OUT	(0DFH), A		; dto.
4DF5 FB	01430 EI			
4DF6 C9	01440 RET			
	01450 *LIST OFF			
	01540 *LIST ON			
1ED6	01550 OFFSET EQU	\$-3000H		; für die Relokation
4ED6 3A2038	01560 DEVINT LD	A, (3820H)		; Tastatur
4ED9 FED0	01570 CP	0D0H		; <,./>?
4EDB 2806	01580 JR	Z, RST28		
4EDD 3A1038	01590 LD	A, (3810H)		; Tast.
4EE0 FEE0	01600 CP	0E0H		; <567>?
4EE2 C0	01610 RET	NZ		; norm. weiter, falls nein
4EE3 F61C	01620 RST28 OR	1CH		; A für RST vorber.
4EE5 EF	01630 RST	28H		; SYS26/SYS anspringen
3010	01640 KBDIV EQU	\$-OFFSET		; Adresse des Spoolers
4EE6 E5	01650 PUSH	HL		; Register retten
4EE7 D5	01660 PUSH	DE		
4EE8 F5	01670 PUSH	AF		
4EE9 2AAD30	01680 LD	HL, (COUNT)		; wiew. Bytes im Puffer?
4EEC 7C	01690 LD	A, H		; 0 ?
4EED B5	01700 OR	L		
4EEE 2821	01710 JR	Z, LEAVIT		; falls Puffer leer
4EF0 CDD105	01720 CALL	05D1H		; Drucker bereit?
4EF3 201C	01730 JR	NZ, LEAVIT		; falls nicht bereit
4EF5 3E08	01740 LD	A, 08H		; "read RAM 0000-2FFFh"
4EF7 F3	01750 DI			; Störungen wären fatal
4EF8 D3DF	01760 OUT	(0DFH), A		; auf Banking-Port
4EFA 2AAB30	01770 LD	HL, (OUTADR)		; Adr. aktuelles Zeichen
4EFD 7E	01780 LD	A, (HL)		; Byte laden
4EFE D3ED	01790 OUT	(0DFH), A		; und drucken
4F00 AF	01800 XOR	A		; "read ROM 0000-2FFFh"
4F01 D3DF	01810 OUT	(0DFH), A		; Bank zurückschalten
4F03 2AAD30	01820 LD	HL, (COUNT)		; Bytezähler
4F06 2B	01830 DEC	HL		; -1
4F07 22AD30	01840 LD	(COUNT), HL		; aktualisieren
4F0A 2AAB30	01850 LD	HL, (OUTADR)		; Stelle im Puffer
4F0D 23	01860 INC	HL		; +1
4F0E 22AB30	01870 LD	(OUTADR), HL		; aktual.
4F11 F1	01880 LEAVIT POP	AF		; Register restaurieren
4F12 D1	01890 POP	DE		
4F13 E1	01900 POP	HL		
303E	01910 CHNGCD EQU	\$-OFFSET		; Code kann geänd. werden
4F14 FB	01920 EI			
4F15 C3	01930 DEFB	0C3H		; JP-Opcode
3040	01940 KBBUF EQU	\$-OFFSET		; zum Originaltreiber
4F16 0000	01950 DEFW	0000H		; spätere Sprungadresse
3042	01960 PRDRV EQU	\$-OFFSET		; neuer Druckertreiber
4F18 E5	01970 PUSH	HL		; Register retten
4F19 D5	01980 PUSH	DE		



4F1A F5	01990	PUSH	AF		
4F1B 2AA930	02000	LD	HL, (INADR)		;Adr. höchstes Byte + 1
4F1E 7C	02010	LD	A, H		;Puffer voll?
4F1F D630	02020	SUB	30H		; (falls 3000h Zeichen)
4F21 B5	02030	OR	L		
4F22 F3	02040	DI			
4F23 203F	02050	JR	NZ, GOBUFF		; falls nicht voll
4F25 ED5BAB30	02060	LD	DE, (OUTADR)		; wieviele Z. gedruckt?
4F29 7A	02070	LD	A, D		
4F2A B3	02080	OR	E		; falls welche gedr.:
4F2B 2016	02090	JR	NZ, MOVBUF		; garbage collection
4F2D E5	02100	PUSH	HL		; wird verändert
4F2E 2100C9	02110	LD	HL, 0C900H		; NOP, RET
4F31 223E30	02120	LD	(CHNGCD), HL		; KBDRV als UP vorbereiten
4F34 CDD105	02130	CALL	05D1H		; Drucker bereit?
4F37 20FB	02140	JR	NZ, WTPRT		; falls nein
4F39 CD1030	02150	CALL	KBDRV		; 1 Zeichen ausdrucken
4F3C 21FBC3	02160	LD	HL, 0C3FBH		; EI, JP-Opcode
4F3F 223E30	02170	LD	(CHNGCD), HL		; Programm restaur.
4F42 E1	02180	POP	HL		
4F43 3E08	02190	MOVBUF	LD	A, 08H	; "read RAM 0000-2FFFh"
4F45 D3DF	02200	OUT	(0DFH), A		; auf Banking-Port
4F47 3C	02210	INC	A		; "write ..."
4F48 D3DF	02220	OUT	(0DFH), A		
4F4A AF	02230	XOR	A		; A=0, Cy=0
4F4B ED52	02240	SBC	HL, DE		; neue höchste Adresse
4F4D 22A930	02250	LD	(INADR), HL		; speichern
4F50 22AD30	02260	LD	(COUNT), HL		; Anz. zu druckender Z.
4F53 C5	02270	PUSH	BC		; wird verändert
4F54 42	02280	LD	B, D		; Anzahl der zu verschie-
4F55 4B	02290	LD	C, E		; benden Bytes
4F56 57	02300	LD	D, A		
4F57 5F	02310	LD	E, A		; DE=0000h, Ziel f. LDIR
4F58 ED53AB30	02320	LD	(OUTADR), DE		; speichern
4F5C E5	02330	PUSH	HL		; brauchen wir noch
4F5D EDB0	02340	LDIR			; Puffer bereinigen
4F5F AF	02350	XOR	A		; "read ROM 0000-2FFFh"
4F60 D3DF	02360	OUT	(0DFH), A		; Bank zurückschalten
4F62 E1	02370	POP	HL		
4F63 C1	02380	POP	BC		
4F64 3E09	02390	GOBUFF	LD	A, 09H	; "write RAM 0000-2FFFh"
4F66 D3DF	02400	OUT	(0DFH), A		; auf Banking-Port ausg.
4F68 71	02410	LD	(HL), C		; Zeichen puffern
4F69 3E01	02420	LD	A, 01H		; "write ROM 0000-2FFFh"
4F6B D3DF	02430	OUT	(0DFH), A		; dto.
4F6D 23	02440	INC	HL		; Zeig. auf nächste Stelle
4F6E 22A930	02450	LD	(INADR), HL		; aktualisieren
4F71 2AAD30	02460	LD	HL, (COUNT)		; Zähler erhöhen
4F74 23	02470	INC	HL		
4F75 22AD30	02480	LD	(COUNT), HL		; aktualisieren
4F78 F1	02490	POP	AF		; restaur.
4F79 D1	02500	POP	DE		
4F7A E1	02510	POP	HL		
4F7B FB	02520	EI			
4F7C C9	02530	RET			
30A7	02540	FRBUF	EQU	\$-OFFSET	; Puffer f. Originaltr.
4F7D 0000	02550	DEFW	0000H		; Platz f. Treiberadresse
30A9	02560	INADR	EQU	\$-OFFSET	; P. f. Eingabeadresse
4F7F 0000	02570	DEFW	0000H		
30AB	02580	OUTADR	EQU	\$-OFFSET	; P. f. Ausgabeadresse
4F81 0000	02590	DEFW	0000H		
30AD	02600	COUNT	EQU	\$-OFFSET	; Zähler für Dr.-Ausgabe
4F83 0000	02610	DEFW	0000H		
4F85 D5	02620	MOVINT	PUSH	DE	; Adr. d. Prg.-Erweiterg.
4F86 21D64E	02630	LD	HL, DEVINT		; dto. d. Prg.-Codes
4F89 EDB0	02640	LDIR			; Code übertragen
4F8B 3ECD	02650	LD	A, 0CDH		; CALL-Opcode
4F8D 32D345	02660	LD	(45D3H), A		; 45D3h: CALL INT-Erweit.
4F90 E1	02670	POP	HL		; Adresse
4F91 22D445	02680	LD	(45D4H), HL		
4F94 C9	02690	RET			
	02700	*LIST	OFF		
	03680	*LIST	ON		
5038 2A1640	03690	SPOOL	LD	HL, (4016H)	; Adr. Tastaturtreiber
503B 111030	03700		LD	DE, KBDRV	; Treibererweit. des Sp.
503E DF	03710	RST	18H		; Vergl.: Spooler aktiv?
503F F3	03720	DI			
5040 2043	03730	JR	NZ, INISPO		; nein, initialisieren
5042 3E08	03740	LD	A, 08H		; "read RAM 0000-2FFFh"
5044 D3DF	03750	OUT	(0DFH), A		; auf Banking-Port
5046 2AA930	03760	LD	HL, (INADR)		; Obergrenze Druckmaterial



5049	ED5BAB30	03770	LD	DE, (OUTADR)	;dto. Untergrenze
504D	B7	03780	OR	A	;Cy <- 0
504E	ED52	03790	SBC	HL, DE	;Anzahl Zeichen
5050	EB	03800	EX	DE, HL	; (HL) = 1. Byte
5051	7A	03810	LD	A, B	;Zähler bereits 0?
5052	B3	03820	OR	E	
5053	2816	03830	JR	Z, BRKWIP	;falls ja
5055	3A4038	03840	LD	A, (3840H)	;Tastatur
5058	FE04	03850	CP	04H	;BREAK gedrückt?
505A	280F	03860	JR	Z, BRKWIP	;dann abbrechen
505C	DBFD	03870	IN	A, (0FDH)	;Drucker bereit?
505E	E6F0	03880	AND	0F0H	;linkes Nibble d. Status
5060	FE30	03890	CP	30H	;bereit, falls Bits 4&5=1
5062	20F1	03900	JR	NZ, WTWIP	;falls nicht bereit
5064	7E	03910	LD	A, (HL)	;Zeichen holen
5065	D3FD	03920	OUT	(0FDH), A	;und drucken
5067	23	03930	INC	HL	;Zeiger nachstellen
5068	1B	03940	DEC	DE	;dto. Zähler
5069	18E6	03950	JR	WIPE	;nächstes Zeichen
506B	DBDF	03960	IN	A, (0DFH)	;MBA rücksetzen
506D	3E0A	03970	LD	A, 0AH	; "read RAM 3000-35FFh"
506F	D3DF	03980	OUT	(0DFH), A	;auf Banking-Port
5071	2A4030	03990	LD	HL, (KBBUF)	;alte Tast.-Treiberadr.
5074	221640	04000	LD	(4016H), HL	;restaurieren
5077	2AA730	04010	LD	HL, (PRBUF)	;dto. Druckertr.
507A	222640	04020	LD	(4026H), HL	
507D	CDC64D	04030	CALL	INIT+1	;ROM wiederherstellen
5080	21C450	04040	LD	HL, TEXT5	;Meldung
5083	1827	04050	JR	DSPRET	;anzeigen und zurück
5085	3E0A	04060	LD	A, 0AH	; "read RAM 3000-35FFh"
5087	D3DF	04070	OUT	(0DFH), A	;auf Banking-Port ausg.
5089	3C	04080	INC	A	; "write RAM 3000-35FFh"
508A	D3DF	04090	OUT	(0DFH), A	;dto.
508C	22164F	04100	LD	(KBBUF+OFFSET), HL	;Treiberadr. retten
508F	2A2640	04110	LD	HL, (4026H)	;Druckertreiberadr.
5092	227D4F	04120	LD	(PRBUF+OFFSET), HL	;retten
5095	ED531640	04130	LD	(4016H), DE	;auf neuen Tr. umleiten
5099	214230	04140	LD	HL, PRDRV	;neuer Druckertreiber
509C	222640	04150	LD	(4026H), HL	;umleiten
509F	110030	04160	LD	DE, 3000H	;neue Programmadresse
50A2	01AF00	04170	LD	BC, MOVINT-DEVINT	;Länge des Programms
50A5	CD854F	04180	CALL	MOVINT	
50A8	FB	04190	EI		
50A9	21B250	04200	LD	HL, TEXT4	;Meldung
50AC	CDA64B	04210	CALL	4BA6H	;anzeigen und zurück
50AF	AF	04220	XOR	A	;A <- 0
50B0	57	04230	LD	D, A	; "567" nicht anzeigen
50B1	C9	04240	RET		;ins Betriebssystem
50B2	53	04250	DEFB	'Spooler aktiviert'	
50C3	0D	04260	DEFB	0DH	
50C4	53	04270	DEFB	'Spooler deaktiviert'	
50D7	0D	04280	DEFB	0DH	
		04290	*LIST OFF		

00000 mal gepennt  
22729 Zeichen verfügbar

COPYRIGHT 1984 BY ARNULF SOPP, TEL. 0451-791926



## Tune-Up des Genie mit "Speed-Up"

Manchmal treibt es mir die Tränen in die Augen, zu sehen, daß gewisse Mickymaus-Computer mit dem Z80A laufen und traumhaft schnell getaktet werden, während das Genie, ein ernstzunehmendes Arbeitspferd für mäßige Ansprüche, mit 1,77 MHz auskommen muß. Dabei wäre eine höhere Taktfrequenz durchaus möglich, denn der einfache Z80 soll 3 MHz klaglos vertragen. Vermutlich hatten die ersten TRS80 noch recht langsame RAMs oder was weiß ich.

Die Fa. Udo Jourdan Computervertrieb (anscheinend ein Ein-Mann-Betrieb, dessen Inhaber unser aller teures Hobby mit eigenen Hard- und Softentwicklungen finanziert) bietet eine kleine Zusatzplatine zum nachträglichen Einbau an, die das Genie benchmarkfit macht. Unter der Bezeichnung "Speed-Up" wird ein fertig aufgebautes Teil geliefert, das die CPU wahlweise mit 2,65 oder 3,54 MHz taktet. Dabei ermöglicht ein Schalter die Rückschaltung auf die gewohnten 1,77 MHz.

Die Version für 2,65 MHz umfaßt eine kleine Platine mit 5 DIL-Chips, einen Schalter und die Einbauanleitung. Die Montage ist zwar sehr einfach und kann wohl in 20 Minuten durchgeführt werden, aber sie kostet natürlich Nerven. Immerhin fummelt der Käufer dieses Turboladers mitten im Gehirn seines besten Stücks herum. Dabei müssen zwei Leiterbahnen der CPU-Platine aufgetrennt und vier Kabel der Zusatzplatine dort angelötet werden.

Im Lieferumfang der 3,54 MHz-Version sind zusätzlich noch eine Z80A-CPU und ein weiteres IC enthalten, das einem anderen der Platine huckepack aufgelötet werden muß. Ein paar weitere Modifikationen auf der Zusatzplatine sind erforderlich, die jedoch ebenfalls sehr schnell gehen.

Bei allen Arbeiten ist die sehr ausführliche Anleitung eine wertvolle Hilfe. Sie ist jedem auf Anhieb verständlich, der noch nie ein elektronisches Bauteil in der Hand hielt. Zwei oder drei Punkte, die noch Anlaß zur Kritik boten, als ich die Lieferung bekam, sind inzwischen abgestellt. Z. B. war auf meiner Anleitung nicht erklärt, in welcher Richtung man die Pins eines DIL-Chips zählt. So war die Anweisung "Pin X von Chip A an Pin Y von Chip B löten" zu jenem Zeitpunkt dem Laien noch unverständlich. Das hat sich inzwischen, wie eine mir zugesandte neue Version der Anleitung zeigt.

Der mitgelieferte Schalter ist, gemessen an seiner simplen Ein-Aus-Funktion, ein Monstrum. Er kann 2X2mal umschalten. Ich legte ihn in mein elektronisches Schatzkästlein für spätere, kompliziertere Aufgaben und benutzte stattdessen einen gewöhnlichen Ein-Schalter für'n Groschen. Herr Jourdan erklärt hierzu, daß er immer möglichst günstige Angebote wahrnimmt, so daß ein Teil schon einmal aufwendiger als benötigt ausfallen kann.

Die hohe Arbeitsgeschwindigkeit der CPU kann Probleme mit dem Floppy-Betrieb verursachen. Bei meinem Gerät liefen alle Schreiboperationen fehlerlos, allerdings erhielt ich etwa bei jedem fünften Lesen eine unsinnige Fehlermeldung (Diskette schreibgeschützt, Bauteil nicht erreichbar o. ä.). Die Anleitung weist bereits darauf hin. Deshalb sollte man vor dem Lesen auf 1,77 MHz zurückschalten. Zu diesem Zweck enthält die Platine einen zusätzlichen Pin, der mit dem Motor-On-Signal der Floppy zu verbinden ist (Pin 16 des Floppy-Controllers). Solange der Motor läuft, ist der Arbeitstakt 1,77 MHz, anschließend schaltet der Baustein wieder auf die vorher eingestellte Frequenz.

Cassettenoperationen darf man auf jeden Fall nur mit 1,77 MHz durchführen, weil das Timing der Baudrate mit Warteschleifen bewerkstelligt wird, die sonst natürlich ebenfalls schneller durchlaufen würden. Die Fehlerquote dürfte bei Formel-I-Cassetten erheblich höher als



normal sein. Und "normal" ist bei unseren Computern bereits eine Katastrophe.

An einen weiteren zusätzlichen Pin kann eine LED angeschlossen werden, die bei erhöhtem Takt leuchtet. So wird der User erinnert, daß er jetzt mit CSAVE usw. Ärger bekommen wird, und daß er mit Reaktionsspielen wohl keinen Blumentopf mehr gewinnen kann.

Eine Besonderheit fiel mir auf, die jedoch harmlos ist: Im Adreßbereich 0000-3FFFh werden nach dem Einschalten im parallelen RAM neben dem Microsoft-ROM, dem TCS-EEPROM und dem memory-mapped-I/O-Bereich seltsame Zufallscodes abgelegt. Wer keinen Banking-Adapter hat, wird davon nichts merken. Bei meinem System wird durch eine Modifikation in SYS0/SYS beim Booten zunächst etwa eine Millisekunde lang das RAM wieder eingenordet, und alles ist eitel Wonne. Ob diese Merkwürdigkeit auch mit dem normalen Z80 auftritt, weiß ich nicht, denn ich habe die Version für 3,54 MHz.

Sonst läuft bisher alles vollkommen normal - wenn man die veränderthalbfachte bzw. verdoppelte Arbeitsgeschwindigkeit des Computers als normal bezeichnen will. So schrieb beispielsweise Peter Spieß in einem früheren Info, daß die Titelbildgraphik ein paar Stunden beansprucht habe. Mit Speed-Up dauert es ein paar halbe Stunden. Und damit ist auch schon gesagt, was das alles soll.

Die Version für 2,65 MHz kostet DM 75,-, die für 3,54 MHz DM 95,- (frei Haustür mit Nachnahmegebühren DM 101,50). Wer überwiegend in Maschinensprache arbeitet, mag den Umbau mit Recht als teuren Luxus ansehen. In BASIC ist das Geld aber zur Schonung der Nerven gut angelegt.

Herr Jourdan empfiehlt vorsichtigerweise die schnellere Version nur für die Genies neuerer Bauart. Ob die RAMs und die sonstige Elektronik beim alten Video-Genie schnell genug sind, um der CPU auch noch bei 3,54 MHz zu folgen, ist nicht sicher. Andererseits bestehen mit 2,65 MHz Chancen, daß sogar der Cassettenbetrieb beschleunigt durchgeführt werden kann. Eine Version für alle drei Taktfrequenzen wird leider nicht angeboten.

Mein zuvor abgedrucktes Hardcopy-Programm brauchte zum Durchlauf 85 Sekunden. Mit 3,54 MHz benötigte es - 85 Sekunden. Nanu?! Beim Stoppen der Zeit mag ich einen kleinen Fehler gemacht haben, so daß wir von einer Differenz von vielleicht einer halben Sekunde ausgehen können. Das zeigt, daß die eigentliche Rechenzeit (bei 1,77 MHz) weniger als eine Sekunde beträgt. Der Drucker ist auch mit 120 Zeichen/Sekunde eben doch extrem lendenlahm gegenüber dem Computer, so daß die Arbeit des Speed-Up in diesem Falle überhaupt nicht wahrgenommen werden kann - dies nur ein Apperçu am Rande.

Mit Sicherheit liegt es nicht am Speed-Up, das einwandfrei und deutlich sichtbar arbeitet; mit der BASIC-Befehlsfolge

```
10 CLS: FOR I=0TO63: PRINT CHR$(191);: NEXT: GOTO 10
```

läßt sich der Unterschied klar beweisen. Am oberen Rand des Displays entsteht je nach Schalterstellung in gewohnter Geschwindigkeit oder doppelt so schnell ein Graphikbalken.

Diese Umschaltung kann man übrigens mitten in der Arbeit vornehmen. Die Anleitung warnt zwar davor, daß das System dabei abstürzen könnte, aber das ist bei mir bisher kein einziges Mal (von vielleicht hundert Malen) passiert.

Also kaufen? Wer so computerkrank ist wie ich, wird über diese Frage nicht lange nachdenken. Ob die Anschaffung letztendendes den Verzicht auf 30 große Bier oder 500 Zigaretten wert ist, mag jeder für sich erwägen.

Arnulf Sopp, Tel. 0451-791926

Nachbemerkung: Udo Jourdan ist inzwischen Mitglied unseres Clubs geworden. Hoffentlich wird er es nicht bereuen, wenn wir feste bei ihm bestellen und "unter Freunden" alles billiger haben wollen!



### Bei der Betreuungsadresse Können ab sofort EPROM's programmiert werden. Es stehen die Typen 2716, 2732, 2532 (2764, 27128) zur Auswahl. Wer diesen Service nutzen möchte, schickt bitte die entsprechende Anzahl EPROM's ("volle" Können gelöscht werden) und das Programm als HEX-Dump auf Diskette. Weiterhin besteht die Möglichkeit, den Zeichensatz des Genie I+II (TRS-80 M.1+3 in Vorbereitung) individuell zu ändern. Benötigt wird ein 2716 und eine Liste über die gewünschten Änderungen.

### Gesucht wird: Ein Englisch - Deutsches Wörterbuch-Programm mit etwa 10000 - 20000 Stichwörtern. Gibt es überhaupt so etwas ? Außerdem: muMATH/muSIMP 2.0, Electric Webster, sowie Geschäftsprogramme aller Art (Fibu, Steuer, Fakt., usw.... (Anm. d. Red.: Diese Anzeige war hinten auf einem Briefumschlag draufgeschrieben; leider ohne Absender. Da ich in letzter Zeit sehr viel Post bearbeitet habe, weiß ich nun nicht mehr, wer der Absender ist. Ich bitte dies zu entschuldigen und mich zu benachrichtigen.)

### Bei der Betreuungsadresse gibt es die Möglichkeit, alte Farbbandkassetten mit neuem Farbband auffüllen zu lassen. Es wird garantiert neues Material verwendet (Kein Wiederauffrischen der alten Farbbänder). Die Standardfarben sind schwarz, blau und braun (Lieferzeit ca. 14 Tg.). Auf Wunsch gibt es noch die Farben grün und rot (Lieferzeit ca. 3-4 Wochen).

Preise:

	ITOH/NEC	EPSON	EPSON
	8510,8023	MX 80	MX 100
Schwarz	: 8,50	14,--	21,--
Braun,blau	: 12,--	20,--	30,--
Sonderfarben	: 18,--	30,--	45,--

Alle Preise incl. MwSt + Porto und Verpackung. Bei Bestellung bitte alte Kassette mitschicken. Andere Druckertypen auf Anfrage.

### Diskettenangebot: BASF-Disketten aber ohne Firmendruck double sided, double density 96 TPI, 10er Pack 55,-- DM.

Noch eine Bitte:  
 Die Originale für Infobeiträge bitte mit  
 einem guten schwarzen Farbband schreiben,  
 da sonst der Druck sehr schwach wird.

\*\*\*\*\*

## Fragen, Antworten und Tips

Andree Opt-Hof hat folgende Frage:

Das Erscheinen x-beliebiger Zeichen auf dem Bildschirm und plötzliches, unerwartetes Booten der Floppy liegt ja meistens am Verbindungskabel Exp.-Int. ---> Keyboard. Um soetwas zu beheben, habe ich bisher immer die Kontaktstellen mit Benzin vorsichtig gereinigt, aber das hilft auch nicht immer. Nun meine Frage: Wer weiß Rat, was man noch tun könnte und wer hat schon Erfahrungen mit den GOLD-PLUG-Kontakten, die nicht oxydieren können, gemacht ???



# Las-Vegas- Spielautomat

Auf dem Bildschirm rotieren drei Walzen. Durch Stoppen und Neustarten soll erreicht werden, daß alle drei den gleichen Wert auf dem Bildschirm sichtbar werden lassen.

Durch bestimmte Joker-Konstellationen ergeben sich Serien- und Super-Serienspiele.

Gute Grafik und guter Sound zeichnen dieses Programm aus.



```

10 REM .....
20 REM *
30 REM *
40 REM *
50 REM *
60 REM *
70 REM .....
80 CLS
90 CLEAR1000
100 DEFINIT A,CIDEFINT E-Z
110 P$=""+"CHR$(230)+""+CHR$(230)+""+CHR$(230)"+""+CHR$(230)
120 P$=P$+P$+P$+P$+P$+P$+P$+P$+""+CHR$(230)+""+
130 P$=""+"CHR$(233)
140 P$(1)="+"P$:P$(2)="+"P$:P$(3)="+"P$:P$(4)="+"P$:P$(5)+"P$:P$(6)"
150 COL$=P$+P$+P$+P$+P$+P$(5)=P$+P$+P$+P$+P$+P$(6)=P$
160 PRINTSTRING$(7,197); " LAS VEGAS SPIEL-AUTOMAT "; STRING$(8,197)
170 PRINT
180 COLOUR7
190 PRINT" Leertaste = START/STOP-Taste"
200 COLOUR4
210 PRINT:"PRINT" Gewinnausloesung, wenn;"
220 COLOUR1
230 PRINT"1. Ausssen zwei gleiche Gewinne und in "
240 PRINT" der Mitte ###."
250 PRINT"2. Ausssen der gleiche Gewinn wie in der"
260 PRINT" Mitte."
270 PRINT"3. Nur ### in der Mitte (0.30 DM)"
280 COLOUR4
290 PRINT:"PRINT" Serienausloesung, wenn
300 COLOUR1
310 PRINT"1. ### ## Superserie (10 Spiele)"
320 PRINT" ####"
330 PRINT
340 PRINT:"PRINT"2. ##### Serie (5 Spiele)"
350 PRINT" XXXX"
360 PRINT:"PRINT"Bei Serienspielen jeder Gewinn = 1 DM"
370 PRINT
380 COLOUR 6:"PRINT"START --> RETURN"
390 GOSUB 2320:GOSUB 2230:GOSUB 2280
400 O=1:N=1:M=3:I(9)=1
410 I=-1
420 PRINTB49,P$S
430 PRINTO980:P$(0):O=0+1:IFO=7:IO=1
440 GOSUB 2420
450 GS=INKEY$:IFGS="" :PRINTB41,P$S:GOTO 420
460 PLAY(1,"I",0)
470 FORA=5270-STEP-1:POKE17149,A:LGR:NEXT
480 DIM B$(14)
490 DIM B(14)
500 COLOUR6
510 REM ***** GRAFIK *****
520 CLS
530 S5=STRING$(29,217)
540 IF L<35 THEN PRINTB728,N$
550 H$=CHR$(202)*STRING$(7,211)+CHR$(202)
560 JS=CHR$(202)*STRING$(7,218)+CHR$(202)
570 KS=CHR$(207)!LS=CHR$(219)
580 PRINTB126,G$
590 PRINTB206,G$
600 PRINTB166,K$!PRINTB195,KS
610 PRINTB287,H$!PRINTB367,J$!PRINTB327,K$!PRINTB335,L$
620 PRINTB416,H$!PRINTB498,J$!PRINTB456,K$!PRINTB464,L$
630 PRINTB385,H$!PRINTB395,J$!PRINTB345,K$!PRINTB353,L$
640 PRINTB686,G$
650 PRINTB766,G$
660 PRINTB726,K$!PRINTB754,L$
670 PRINTB610,STRING$(21,217)
680 PRINTB450,K$!PRINTB670,L$
690 COLOUR3
700 PRINTB251,CHR$(255):PRINTB669,CHR$(253)
710 PRINTB4,STRING$(33,233)
720 PRINTB884,STRINGS(33,202)
730 FOR A=44 TO 844 STEP 40:PRINTBA,CHR$(233):NEXT

```



```

1080 B5(11)="####":B(11)=.3
1090 B5(12)="2,--":B(12)=2
1100 B5(13)="3,--":B(13)=3
1110 B5(14)="####":B(14)=.3
1120 N5=" "
1130 Q5=" " START "
1140 W5=" " STOP "
1150 E5=" " STOP "
1160 R5=" " STOP "
1170 V5=" " VERLOREN "
1180 G5=" " GEWONNEN "
1190 COLOUR5
1200 PRINT@728,Q5
1210 COLOUR1
1220 K5=INKEY$:IF K5="" THEN 1220
1230 L7=3:GOSUB 2290
1240 D=D-.3
1250 PRINT@172,USING"###.##";D
1260 PRINT@728,N5
1270 GOTO 1320
1280 X=RND(11)
1290 L=50+RND(50)
1300 RETURN
1310 REM **** LAUF DER WALZEN ****
1320 GOSUB 1280
1330 FOR I=1 TO L
1340 GOSUB 2310
1350 PRINT@329,B5(X)
1360 B1=B(X)
1370 X=X+1:IF X>14 THEN X=1
1380 IF I=20:COLOUR5:PRINT@728,W5:COLOUR1
1390 IF I>20 AND I<35:Q15=INKEY$:IF Q15="" :L7=3:GOSUB 2290:GOTO 1420
1400 IF I=35 THEN PRINT@728,N5
1410 NEXT
1420 PRINT@728,N5
1430 GOSUB 1280
1440 FOR I=1 TO L
1450 PRINT@347,B5(X)
1460 B2=B(X)
1470 X=X+1:IF X>14 THEN X=1
1480 GOSUB 2310
1490 IF I=20:COLOUR5:PRINT@728,R5:COLOUR1
1500 IF I>20 AND I<35:Q25=INKEY$:IF Q25="" :L7=4:GOSUB 2290:GOTO 1530
1510 IF I=35 THEN PRINT@728,N5
1520 NEXT
1530 PRINT@728,N5
1540 GOSUB 1280
1550 FOR I=1 TO L
1560 PRINT@458,B5(X)
1570 B3=B(X)
1580 X=X+1:IF X>14 THEN X=1
1590 GOSUB 2310
1600 IF I=20:COLOUR5:PRINT@728,E5:COLOUR1
1610 IF I>20 AND I<35:Q35=INKEY$:IF Q35="" :L7=5:GOSUB 2290:GOTO1640
1620 IF I=35 THENPRINT@728,N5
1630 NEXT
1640 COLOUR3
1650 REM ***** AUSWERTUNG *****
1660 IF B1=B(11) AND B2=B(11) THEN 1780
1670 IF B1=B(11) AND B3=B(11) THEN 1690
1680 GOTO 1720
1690 S=S-1:IF S>0 THEN 2180:GOSUB 2190
1700 GOSUB 2230
1710 D=D+B2:PRINT@728,G5:FOR I=0 TO 500:NEXT:GOTO 930
1720 IF B2=B(11) AND B3=B(11) THEN 1740
1730 GOTO 1770
1740 S=S-1:IF S>0 THEN 2180
1750 GOSUB 2230
1760 GOSUB 2190:D=D+B1:PRINT@728,G5:FOR I=0 TO 500:NEXT:GOTO 930
1770 IF ((B1=B2) AND (B1=B3)) OR((B1=B2) AND (B3=B(11))) OR ((B1=B3) AND(B2=B(11))) OR ((B2=B3) AND (B1=B(11))) THEN 1890
1780 IF B3=B(11) AND B1=B(11) AND B2=B(11) THEN 2100
1790 IF B1=B(11) AND B2=B(11) THEN 2010

```

```

1800 IF B3=B(11) THEN 1960
1810 GOTO 1830
1820 PRINT@728,G5:D=D+B3:FOR I=0 TO 500:NEXT:GOTO 930
1830 PRINT@728,V5
1840 S=S-1:IF S>0 THEN 2210
1850 GOSUB 2280
1860 IF S=0 THENGOSUB 2190
1870 IF D<0.3 :H=1
1880 FOR I=0 TO 500:NEXT:GOTO 930
1890 PRINT@728,G5
1900 GOSUB 2230
1910 S=S-1:IF S>0 THEN 2180
1920 IF S=0 THEN GOSUB 2190
1930 IF B1=B(11) THEN D=D+B2 ELSE D=D+B1
1940 FOR I=0 TO 500:NEXT:GOTO 930
1950 GOTO 930
1960 S=S-1:IF S>0 THEN 2180
1970 GOSUB 2230
1980 IF S=0 THEN GOSUB 2190
1990 PRINT@728,G5:D=D+0.30
2000 FOR I=0 TO 500:NEXT:GOTO 930
2010 COLOUR 4
2020 S=S-1:IF S>0 :S=S+5:GOTO 2180N1920
2030 GOSUB 2190
2040 GOSUB 2230
2050 PRINT@182,"SERIE"
2060 FOR J=1TO 3:GOSUB 2230:NEXT
2070 S=5
2080 PRINT@652,S,"Serienspiele"
2090 D=D+1:PRINT@728,G5:FOR I=0 TO 500:NEXT:GOTO 930
2100 COLOUR 4
2110 S=S-1:IF S>0:S=S+10:GOTO 2180
2120 GOSUB 2190
2130 PRINT@182,"SUPERSERIE"
2140 FOR J=1 TO 4:GOSUB 2230:NEXT
2150 S=10:D=D+1
2160 PRINT@652,S,"Serienspiele "
2170 PRINT@728,G5:FOR I=0 TO 500:NEXT:GOTO 930
2180 D=D+1:PRINT@728,G5:COLOUR4:PRINT@652,S,"Serienspiele ":GOSUB 2230:FOR I=0 T
O 500 :NEXT:GOTO 930
2190 PRINT@652," " :PRINT@182," "
2200 RETURN
2210 PRINT@728,V5:COLOUR4:PRINT@652,S,"Serienspiele ":GOSUB 2280:FOR I=0 TO 500:
NEXT:GOTO 930
2220 REM ***** MELODIE *****
2230 PLAY(1,4,1,15):FOR I=0 TO 40:NEXT
2240 PLAY(1,4,3,15):FOR I=0 TO 40:NEXT
2250 PLAY(1,4,5,15):FOR I=0 TO 50:NEXT
2260 PLAY(1,1,1,0):PLAY(2,1,1,0):PLAY(3,1,1,0)
2270 RETURN
2280 PLAY(1,3,1,15):FOR I=0 TO 90:NEXT:PLAY(1,1,1,0):PLAY(2,1,1,0):PLAY(3,1,1,0)
:RETURN
2290 PLAY(1,L7,1,15):PLAY(2,L7,3,15):PLAY(3,L7,5,15):FOR I=0 TO 50:NEXT
2300 PLAY(1,1,1,0):PLAY(2,1,1,0):PLAY(3,1,1,0):RETURN
2310 PLAY(1,1,5,15):PLAY(1,1,1,0):RETURN
2320 FOR H1=1 TO 8
2330 FOR H2=1 TO 12
2340 PLAY(1,H1,H2,15)
2350 NEXT:NEXT
2360 FOR H1=8 TO 1 STEP -1
2370 FOR H2=12 TO 1 STEP -1
2380 PLAY(1,H1,H2,15)
2390 NEXT:NEXT
2400 PLAY(1,1,1,0)
2410 RETURN
2420 IFA=0:PLAY(1,M,1,12):M=M+1:IFM=8:A=1
2430 IFA:PLAY(1,M,1,12):M=M-1:IFM=1:A=0
2440 RETURN

```



dto. bei Colour Genie

## Autostart bei Video-Genie

Beim Video-Genie werden Maschinenprogramme mit Hilfe eines »System-Befehls« geladen; danach kann man sie automatisch starten lassen.

Die Ausführung des System-Befehls steht im ROM ab 2B2H und hat am Anfang ein CALL 41E2H. Normalerweise steht in 41E2H ein C9H (RET), das heißt der Computer springt sofort wieder zurück und führt dann den System-Befehl aus. Nun befindet er sich im »System-Modus«, in dem ein Maschinenprogramm geladen werden kann.



Nach dem Laden des Programmes startet der Computer den System-Befehl von neuem. Er bleibt im »System-Modus«, um das Programm durch »/« in Gang zu setzen. Bei diesem System-Befehl Restart wird zwangsläufig auch das CALL 41E2H noch einmal durchlaufen. Wenn in 41E2H nach dem

Laden ein Befehl stehen würde, der das Programm startet, wäre der Autostart perfekt. Um dieses Ziel zu erreichen, wollen wir uns einmal ansehen, wie beim Video-Genie ein Maschinenprogramm auf Kassette aufgezeichnet wird:

Vorspann: 255 Nullbytes; Synchronisation: A5H; Filename-Block: Filename-Headercode (55H), 6 Bytes Filename; n Datenblöcke: Data-Headercode (3CH), Blocklänge (1 — 256; 00 entspricht 256), Blockanfangsadresse (LSB/MSB), Datenbytes 8-Bit Prüfsumme über Blockanfangsadresse und alle Datenbytes; Entrypoint-Block: Entrypoint-Headercode (78H), Entrypoint (LSB/MSB)

Der Entrypoint des Programmes wird vom Computer in das Registerpaar HL gelesen. Anschließend wird der System-Befehl zum zweiten Male gestartet.

Man muß also die Aufnahme des Programmes derart verändern, daß der Computer ein »JP(HL)« einliest und in 41E2H abspeichert. Dadurch würde dann das Programm sofort nach dem Laden gestartet.

Folgende Bytes müssen in die Aufnahme eingefügt (alle hexadezimal) werden:

BEGIN	LD	HL,7000H	;Pointer (beliebige Adresse)
	CALL	215H	;Kassettenrecorder an
	CALL	296H	;Synchronbyte lesen
LESEN	CALL	235H	;Byte in Akku lesen
	LD	(HL),A	;abspeichern
Bild 1	INC	HL	;inkrementiere Pointer
	JP	LESEN	;weiter
BEGIN	LD	HL,7000H	;Pointer (beliebige Adresse)
	CALL	215H	;Kassettenrecorder an
	CALL	287H	;255 Nullbytes und Synch.-Bytes schreiben
PRINT	LD	A,(HL)	;Inhalt von HL in Akku
	CALL	264H	;und abspeichern
	LD	A,L	;LSB von Pointer in Akku
	CP	XXH	;Vergleich mit LSB der Endadresse des Programms im Speicher
	JP	NZ,PRINT	;Nein, weiter
	LD	A,H	;MSB des Pointers in Akku
	CP	YYH	;Vergleich mit MSB der Endadresse
	JP	NZ,PRINT	;Nein, weiter
Bild 2	CALL	1F8H	;Kassettenrecorder aus

3C	Data-Headercode,
01	Anzahl der Datenbytes:
	1,
E2	LSB der Blockanfangs-
	adresse,
41	MSB der Blockanfangs-
	adresse,
E9	entspricht »JP (HL)«,
0C	Prüfsumme.

Man muß das Programm Byte für Byte (das heißt mit Data-Headercode) einlesen. Anschließend fügt man die Bytes für den Autostart ein und speichert sie wieder Byte für Byte ab. Zum Einlesen kann das Programm laut Bild 1 dienen. Da man nicht genau weiß, wann das Programm, das man lesen will, auf der Kassette endet, sollte man obiges Leseprogramm starten und mit RESET unterbrechen, sobald an der Volumenanzeige kein Ausschlag mehr erkennbar ist.

Den Autostartzusatz fügt man der Übersicht und Einfachheit halber hinter den letzten Datenblock ein, dort, wo normalerweise der Entrypoint-Block steht.

Beispiel:	3C	Dataheader
	.	.
	.	Daten
	.	.
	nn	Prüfsumme
Einfü-		
gung -	78	Entrypoint-
		Header
	nn	
	nn	

Die drei Bytes des Entrypoint-Blocks werden dann hinter die Einfügung geschrieben (78:LSB:MSB).

Zum Abspeichern kann das Programm laut Bild 2 dienen.

Das Programm, das man mit Autostart versehen will, muß vom Filename-Header (55H) bis zum Entrypoint-Block wieder auf Kassette abgespeichert werden. Bei

Maschinenprogrammen, die nach der Ausführung den Computer wieder ins Basic springen lassen, ist besondere Vorsicht geboten: In 41E2H steht dann immer noch JP (HL). Man muß eine Routine in das Programm einfügen, die bei 41E2H wieder das normale C9H abspeichert.



Das Laden von Maschinensprachprogrammen von Cassette war beim Colour Genie schon immer ein Problem:

Da hat man eine Cassette mit mehreren Programmen und möchte ein ganz bestimmtes laden. Den Anfang hat man schnell gefunden, bloß ist es das gesuchte Programm ? Nein. Aber welches ist es und wo bin ich gerade auf der Cassette ?

Dem schafft mein Programm abhilfe, denn schreibt den Programmnamen des geraden ladenden Programmes auf den Bildschirm.

Gleichzeitig wird das Programm eingeladen.

Ich habe dieses Programm bei mir in den obersten Speicherbereich (bfa0) abgelegt, so daß es auch nach einem eventuellen Basic-Warmstart existent ist.

Nun wünsche ich noch viel Spaß mit diesem kleinen Programm.

bfa0	cdc901	CALL 01c9 H		
bfa3	115a44	LD DE,445a H		
bfa6	21c8bf	<del>XXXXXXXXXXXX</del> LD HL,0bc8 H		
bfa9	011000	LD BC,0010 H		
bfac	edb0	LDIR		
bfae	cd4c02	CALL 024c H		
bfb1	cded01	CALL 01ed H		
bfb4	fe55	CP 55 H		
bfb6	20f9	JR NZ,\$-05 H		
bfb8	216444	LD HL,4464 H		
bfbB	0606	LD B,06 H		
bfbD	cded01	CALL 01ed H		
bfc0	77	<del>XXXXXXXX</del> LD (HL),A		
bfc1	23	INC HL		
bfc2	10f9	DJNZ \$-05 H		
bfc4	cdea02	CALLO2ea H		
bfc7	c9	RET		
bfc8	49	LD C,C		
bfc9	63	LD H,E		
bfca	68	LD L,B		
bfcbb	206c	JR NZ,\$+6e H		
bfcD	61	LD H,C		
bfce	64	LD H,H		
bfcf	65	LD H, <del>X</del> L	bfd5	68 LD L,B
bfd0	3a206e	LD A,(6e20 H)	bfd6	74 LD (HL),H
bfd3	69	LD L,C	bfd7	73 LD (HL),E
bfd4	68	LD H,E		



# Die Colour Genie

EC  
KE



Auszug aus dem Info des Colour-Genie User-Club

## NEUE SPRACHEN FÜR DAS CG:

Neben den beiden PASCAL-Compilern und dem Fortran-Compiler, die wir im letzten TEAMWORK ausführlich beschrieben haben, stehen für das CG noch zwei weitere Sprachen zur Verfügung, auf die wir nun genauer eingehen wollen.

### **FORTH:**

Diesen Compiler, der im Angebot von fast allen Software-Anbietern erhältlich ist, gibt es in zwei verschiedenen Versionen. Die erste Version, die für 95,- DM zu bekommen ist, ist nur für den Kassettenbetrieb gedacht, während die zweite um 10,- DM teurere Version auch für den Diskettenbetrieb zu geeignet ist. Für beide Versionen benötigt man 32 KByte RAM, sowie die neuen ROMs, die seit April '83 in allen CGs eingebaut sind. Bei diesem FORTH-Compiler handelt es sich um eine angepaßte Version des FIG-FORTH, die allerdings um viele, für das CG nützliche Befehle erweitert wurde (z. B. Befehle für Graphik u. Ton, usw.). Der wesentliche Vorteil, des schwerer zu erlernenden FORTH gegenüber BASIC, ist die extrem schnelle Ausführungszeit der Programme, die in dieser FORTH-Version noch durch Benutzung des eingebauten Assemblers gesteigert werden kann. Zu beiden Versionen wird ein ausführliches deutsches Handbuch mitgeliefert.

### **LISP:**

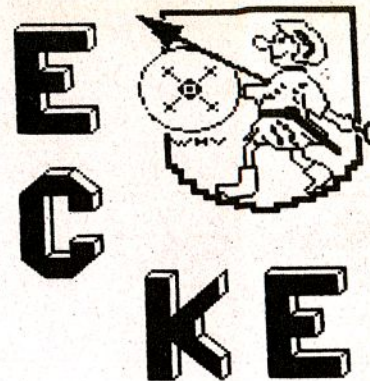
Bei dieser LISP-Version handelt es sich um einen in BASIC geschrieben und daher auch nicht besonders schnellen und leistungsfähigen LISP-Interpreter. Folgende Möglichkeiten und Besonderheiten bietet dieses Programm:

Die Verwendung von Symbolen mit mehr als 9 Zeichen ist möglich, jedoch wird bei Ablage mit (SAVE) jedes Objekt auf die ersten 9 Zeichen beschnitten. Pascal-ähnliche Iterationen wie DOWHILE-Schleifen oder DOUNTIL stehen zur Verfügung. Mit Hilfe der Funktionen READ, PATOM und PRINT kann ein Programm bzw. Funktion Informationen vom Terminal eiholen bzw. ausgeben. Die Sonderfunktion PP erlaubt die Ausgabe selbstdefinierter Funktionen. Eingegebene Programme können mit Hilfe von SAVE bzw. LOAD abgespeichert bzw. geladen werden. Der Programmspeicherung erfolgt mittels der Level II Befehle PRINT#-1 und INPUT#-1. Somit können auch relativ umfangreiche Funktionsumwelten relativ schnell geladen und gespeichert werden.

Preis und Vertrieb für dieses doch recht gelungene, obwohl in BASIC geschriebene Programm, sind uns nicht bekannt.



# Die Colour Genie



Auszug aus dem Info des Colour-Genie User-Club

**Hier noch einige POKES von Christian Klein, die recht interessante Auswirkungen mit sich bringen:**

POKE 16863,195:POKE 16864,24:POKE 16865,26  
bewirkt, daß der LIST-Befehl blockiert ist.

POKE 16396,23 sperrt die BREAK-Taste.

Die Voraussetzung für die folgenden POKES ist, daß die Zeilennummer 0 existiert.

POKE 16863,195:POKE 16864,169:POKE 16865,30  
damit reagiert der LIST-Befehl wie RUN!

POKE 16396,195:POKE 16397,169:POKE 16398,30  
Damit wird nach Drücken der BREAK-Taste das Basic-Programm neu gestartet und alle Variablen werden gelöscht.

Das gleiche, aber ohne Variablen löschen sieht dann so aus:  
POKE 16396,195:POKE 16397,172:POKE 16398,30

Durch diese POKES kann man ein Basicpgr. nicht mehr mit <BREAK> verlassen, aber immer noch mit den beiden <RST>-Tasten. Diese lassen sich nicht ausschalten. Durch einen Trick kann man aber auch zum Ziel kommen: Die Speicherstelle 16813 wird vom Interpreter immer dann angesprungen, wenn sich der Rechner im Direkt- (=Eingabe-) Modus befindet. (Für Neugierige: Hauptschleife der Programmeingabe: 1A1C CALL 41AC).

Durch Drücken der beiden <RST>-Tasten (= sog. nicht maskierte Interrupt NMI, der den PC auf 0066 setzt), führt das CG einen Warmstart aus (Stack berichtigen, usw.) und CALLED dann nach 41AC, wo normalerweise 'RET' steht. Hier springt der Rechner z. B. bei Colour-Kong zu einer Speicherlöschroutine, und das Programm ist vernichtet, nachdem man die beiden <RST>-Tasten gedrückt hat.

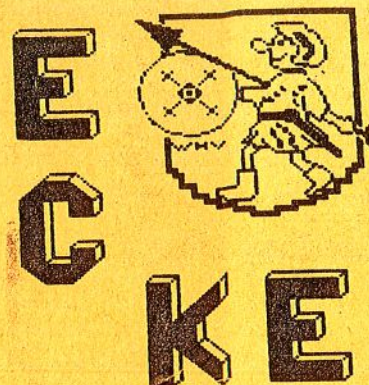
Den sog. Kaltstart = Drücken der <R>-Taste und der beiden <RST>-Tasten kann softwaremäßig nicht beeinflußt werden.

Die folgenden POKES sind nur für Basic interessant:

POKE 16813,169:POKE 16814,30:POKE 16812,195  
bewirkt, daß nach einem NMI das Basicpgr. neu gestartet wird. Das entspricht in Basicsprache 'CLEAR:GOTO 0'.  
Nun das gleiche ohne die Variablen zu löschen => GOTO 0:  
POKE 16813,172:POKE 16814,30:POKE 16812,195



# Die Colour Genie



Auszug aus dem Info des Colour-Genie User-Club

Natürlich kann man sich diesen Trick auch in Maschine zunutze machen, indem man ab 41ACH einfach 'JP NN' hineinschreibt. Nach einem NMI springt der Rechner nach NN, wo meinetwegen der Hinweis ausgedruckt wird: 'He was soll denn das', und das Programm neu startet.

Den Normalzustand erreicht man, wenn man in 16812 = 41ACH 'RET' hineinpoked: POKE 16812,&HC9

Nun noch einige 'DCB' POKEs, bei denen die Zeilennummer 0 keine Voraussetzung mehr ist.

POKE 16414,227 schaltet den Display-DCB ab, POKE 16414,228 wieder an. (DCB=Device Control Block)

Um das etwas zu verdeutlichen schrieb ich folgendes Programm:

```
10 POKE 16414,227
20 INPUT A$
30 POKE 16414,228
40 PRINT A$
50 RUN
```

Mit POKE 16406,250 erreicht man, daß der Keyboard-DBC etwas eingeeengt wird: Es ist jetzt nicht mehr möglich mit <CTRL>+<MOD SEL> in den FGR-Modus zu gelangen.

POKE 16406,247 bewirkt das gleiche, nur ist jetzt kein Cursor im Direktmodus mehr sichtbar,

Mit POKE 16406,86 kann man die Tastaturabfrage abschalten. Gerade bei Basic-Spielprg. ist dieser POKE besonders interessant, da bei jedem Tastendruck der Programmablauf verzögert wird und das Programm somit langsamer wird. Mit diesem POKE kann man diese Verzögerung abschalten, aber die Tastatur immer noch mit PEEK aus den Speicherstellen F801 - F880 abfragen. Hierbei muß man aber beachten, daß der Rechner nicht in den Direktmodus gelangt, bevor der Normalzustand wieder hergestellt ist, da sonst keine Eingabe über die Tastatur mehr möglich ist.

Mit POKE 16406,227 ist der Normalzustand wieder hergestellt.

Um den Ausgangszustand der anderen POKEs wieder zu erreichen, fragt Ihr am besten mit PEEK(XX) die jeweilige Speicherstelle einfach ab, bevor Ihr irgendwas hineinPOKed und merkt Euch diesen Wert.