

GENIE / TRS 80 - USER - CLUB

B R E M E R H A V E N

C L U B - I N F O

OKTOBER '83



It's hard to get by!

Liebe Clubkameraden!

Nur wenig selbstgeschriebenes findet Ihr diesmal in unserem "Clubinfo", - Dafür aber insgesamt um so mehr Informationen.

Die Abstimmung zur Umstrukturierung der Bibliothek ist positiv verlaufen; -d.h. wir werden ab sofort jegliche Programme wild untereinander tauschen 1. ohne über die Clubleitung zu gehen 2. ohne sich um etwaige Copyrights zu kümmern, denn die Weitergabe eines kommerziellen Programms ohne Entgelt an einen Bekannten zwecks dessen "Weiterbildung"(siehe auch Statuten!) ist nicht strafbar noch verboten!

Erst die weitere kommerzielle Verbreitung oder Nutzung des Programms zu kommerziellen Zwecken! steht unter Strafandrohung!

Ich erwarte also jede Menge Softwarelisten von Euch! Aber bitte!

-übt Selbstdisziplin, was Schriftgröße, Format und Umfang anbetrifft, damit die neu entstehende Bibliothek von ersten Tag an qualitativ und quantitativ besser wird als ihr Vorgänger.

Am besten, Ihr nehmt DIN A4 eng mit Maschine beschrieben und kopiert dann auf DIN A5 runter.

Zum Inhalt: Die noch fehlenden 5 Seiten des geknackten TRS 80 mischen sich bunt mit einer Umbauanleitung von Peter Spieß für den Grafikbetrieb bzw dessen Verbesserung sowie einer 2-seitigen Befehlsliste für den Z80 Maschinencode und einer Artikelsammlung für das immer wieder auftretende Problem des sog. Screenprints oder auch Hardcopy. Weiterhin habe ich meinen Artikel über die Maschinensprache fortgesetzt indem ich einen Artikel aus der GENIE-DATA über die Hardwareseite des Z80 kopierte.

Letztlich bekam ich noch von Heinrich Thönninger, Bremen, einen Fragebogen test für Einsteiger.

Da ich keine Seite ungenutzt aus dem Haus gehen lassen will kopiere ich auch schon die ersten Softwarelisten auf einige der Rückseiten.

Der Roeckrathartikel über den TRS80 ist, entgegen der Meinung mehrerer Zuschriften übrigens im vollen Umfange auch für GENIE I/II gültig und in den wesentlichen Bereichen auch für das COLOUR-GENIE.

Der Club ist natürlich im Besitz der vollständigen Romlistings und wird nach und nach diese bei entsprechenden Artikeln mit veröffentlichen.

Wie Ihr seht ist also alles in Butter, nur mir fehlt die Zeit um noch besser zu werden - auch wenn ich's möchte. Die Fremdbeteiligung durch Mitglieder ist diesmal angenehm hoch was mir eine Menge Arbeit spart. Also nur weiter so!

Ich erwähnte es schon einmal in einer früheren Ausgabe: Selbst die absurdeste Frage oder Beitrag kann den hochgestochenen und über allen Wolken schwebenden Computerfreak helfen um auf neue Bahnen zu kommen. Andererseits ist (sollte) jeder in der Lage sein sich allgemeinverständlich auch für den nur halb Sachverständigen auszudrücken, während er selber im Computerhimmel sitzt („zu sitzen glaubt!").

Klaus Schmidt

VIDEO SNOW SHOVEL

Diese Anleitung basiert auf einem Beitrag in der Zeitschrift
'80 MICROCOMPUTING' Heft 3/82.
Dortiger Titel : Video Snow Shovel

Mit der nachstehend beschriebenen Schaltung werden die dünnen schwarzen Striche, die sich vor allem bei graphikintensiven Programmen störend auf dem Bildschirm bemerkbar machen, 'ausgeschaltet'. Die Ursache für dieses störende Flimmern liegt darin, daß auf die Video - RAMs zum einen von der Z80 - CPU und zum anderen vom Videoteil zugegriffen wird. Bei einem gleichzeitigen Zugriffsversuch hat hierbei die CPU die höhere Priorität. Die folgende Schaltung vertauscht die Prioritäten. Ein Bildschirm - Bild setzt sich aus $192 * 384$ Dots zusammen. Jedes Zeichen nimmt einen Platz von $12 * 6$ Dots ein. Hieraus ergibt sich die Bildschirmgröße von 16 Zeilen zu je 64 Zeichen. Nachdem jeweils eine volle Dot - Reihe von links nach rechts geschrieben wurde, bewegt sich der Elektronenstrahl der Bildröhre ohne zu schreiben zum linken Rand zurück und beginnt die nächste Reihe. Ist der Bildschirm voll, fängt das Spiel in der linken oberen Ecke wieder von vorne an. Die CPU darf also nur auf die Video - RAMs zugreifen, wenn der Elektronenstrahl gerade nichts schreibt. Dies läßt sich sehr einfach über den WAIT - Eingang der CPU realisieren. Es werden folgende Signale vom Computer benötigt:

Signal	TRS80	GENIE
VID	Pin 8 Z36	Pin 3 Z35 CPU - Board
DLY BLANK	Pin 7 Z27	Pin 15 Z3 Interface - Board

Bauteile : 1 Stück 74LS02
1 Schalter 1 x um

Bei Spielen mit intensiver Graphik leidet die Tonausgabe etwas, da die WAIT - Zyklen die Warteschleifen verändern. Daher ist der Schalter vorgesehen. Allerdings wurde bisher bei allen bekannten Spielen nichts dergleichen bemerkt, auf den Schalter kann man genausogut verzichten.

Einbau der Schaltung:

Abbildung 1 zeigt den Zustand im Computer vor, Abbildung 2 nach dem Einbau. Es muß lediglich eine Leiterbahn durchtrennt werden. Das IC wird gemäß Abbildung 3 verdrahtet und 'piggyback' auf ein passendes IC im Computer gelötet (VCC = Pin 14 und GND = Pin 7).

Wird auf den Schalter verzichtet, so ist Pin 8 ständig mit DLY BLANK zu verbinden.

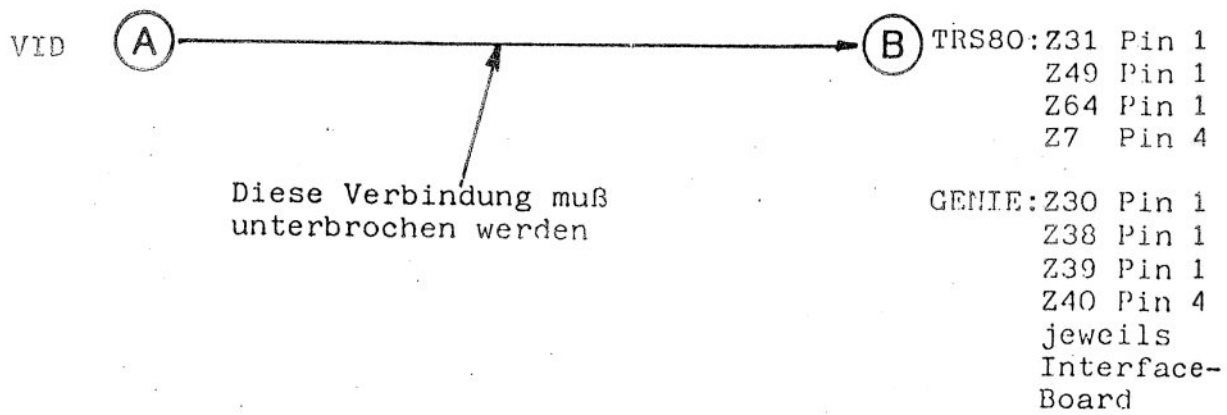


Abbildung 1

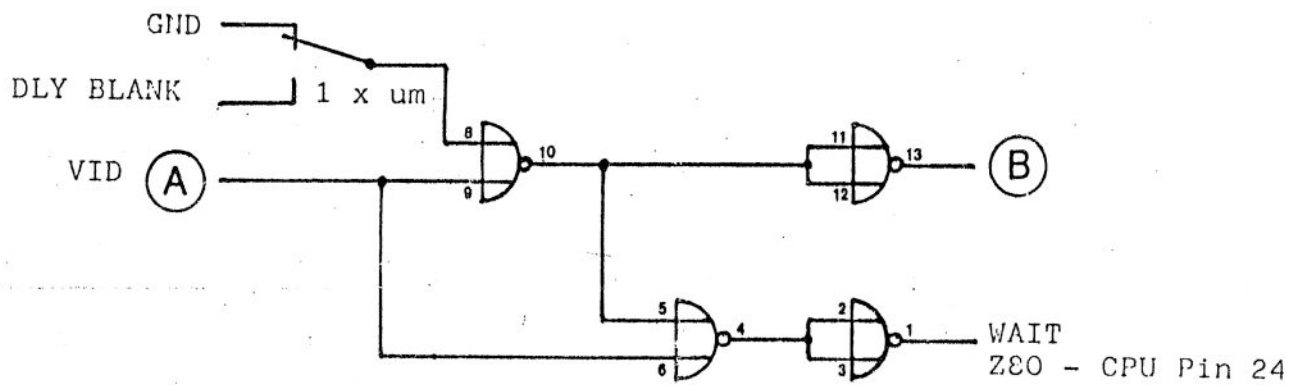


Abbildung 2

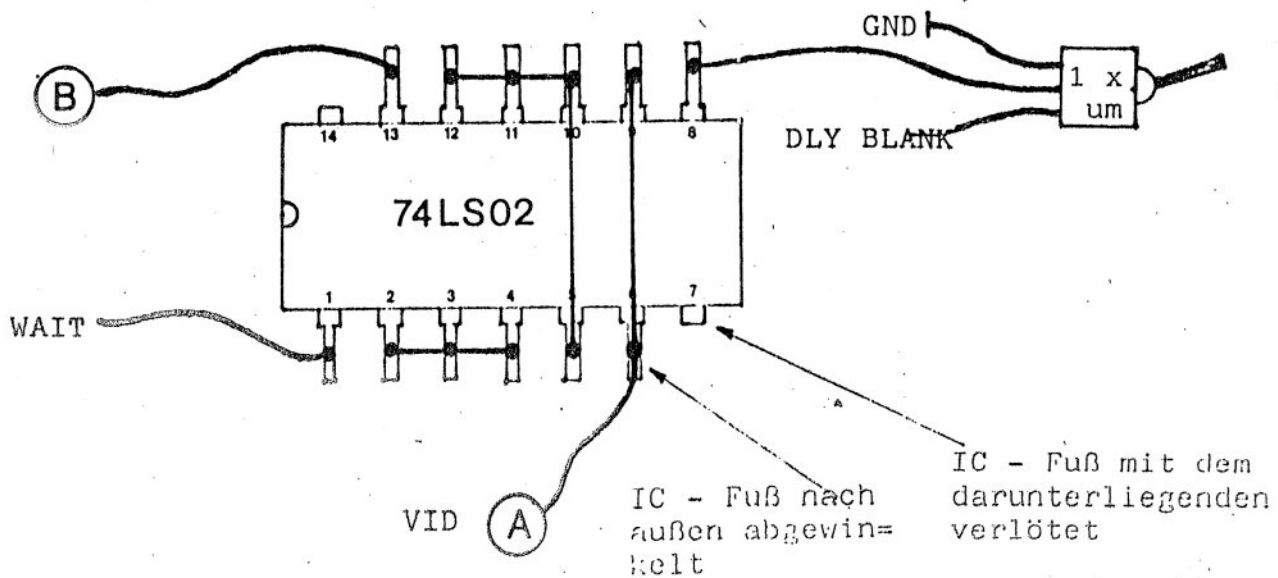


Abbildung 3

				AF BC DE HL			
1EB1	1EDD	07857	07901	GOSUB-Befehl			
1EC2	1EDD	07874	07901	GOTO-Befehl			
1ED9	1EDD	07897	07901	UL-Error			
1EDE	1F20	07902	07968	RETURN-Befehl			
1F05	1F20	07941	07968	DATA-Befehl			
1F07	1F20	07943	07968	REM-Befehl			
1F21	1F6B	07969	08043	LET-Befehl			
1F6C	1FAE	08044	08110	ON-Befehle			
1F70	1F94	08048	08084	ON ERROR-Befehl			
1F95	1FAE	08085	08110	ON GOTO (GOSUB)-Befehl			
1FAF	1FF3	08111	08179	RESUME-Befehl			
1FF4	2007	08180	08199	ERROR-Befehl			
2008	2038	08200	08248	AUTO-Befehl (Initialisierung und Sprung zur Hauptschleife)			
2039	2066	08249	08294	IF-THEN-ELSE-Befehl			
2067	2177	08295	08567	PRINT-Befehl			
2067	2067	08295	08295	Ansprung LPRINT			
206F	206F	08303	08303	Ansprung PRINT			
2072	207C	08306	08316	Wenn PRINT# Kassettenrecorder vorbereiten			
2084	20A4	08324	08356	α auswerten			
20F9	2107	08441	08455	wenn Cursor nicht in Position 0, CR ausgeben		0044	
20FE	2107	08446	08455	CR – ausgeben		0044	
2137	2168	08503	08552	TAB auswerten			
2178	217E	08568	08574	Text „?REDO(CR)“			
217F	22B5	08575	08885	INPUT, INPUT# und READ			
219A	219A	08602	08602	Ansprung INPUT und INPUT#			
21EF	21EF	08687	08687	Ansprung READ			
2286	2295	08838	08853	Text „?Extra ignored(CR)“			
22B6	2334	08886	09012	NEXT-Befehl			
2335	25D8	09013	09688	Auswertung von Ausdrücken. Der Ausdruck steht als Text (in Zwischencodes) im RAM, HL dient als Zeiger und wird entsprechend incrementiert.			
				Nach der Auswertung liegt in X das Ergebnis vor			
2335	2335	09013	09013	Ansprung für Ausdruck, der mit einer Klammer beginnt	xxxx	xxxx	xxxx P
2337	2337	09015	09015	normaler Ansprung	xxxx	xxxx	xxxx P
2406	2437	09224	09271	Ansprung der Grundrechenarten über Sprungtabelle			
2490	249E	09360	09374	HL / DE ↗ X	xxxx	xxxx	xxxx xxxx
249F	2531	09375	09521	Auswertung eines Teilausdruckes bestehend aus Variable, Konstante, NOT Ausdruck, Funktion oder einem in Klammern eingeschlossenen Ausdruck			
24CF	24D8	09423	09432	ERR-Funktion			
24DD	24E6	09437	09446	ERL-Funktion			
24EB	24FE	09451	09470	VARPTR-Funktion			
252C	2531	09516	09521	Auswertung eines Ausdrucks in Klammern (z. B.: Argument für Funktionen)	xxxx	xxxx	xxxx P
2532	253F	09522	09535	Vorzeichen (–) auswerten			
2540	254D	09536	09549	Wert einer Variablen in Ausdruck einbringen			
254E	258B	09550	09611	Argumente analysieren und Funktionen anspringen (über Tabelle)			
258C	25B7	09612	09695	Stringvergleich			
25C4	25D8	09668	09688	NOT ausführen			
25D9	25E8	09689	09704	RST 20H UP	xxxx		
				Prüft Typ des Inhaltes von X			
				INTEGER S,C P			
				SINGLE C			
				DOUBLE P			
				STRING Z,C P			
				Je nach Typ werden die Flags beeinflusst			
25E9	2602	09705	09730	AND und OR ausführen			
2603	27C8	09731	10184	Verwaltung der Variablen-tabelle			
2608	2608	09736	09736	Ansprung DIM-Befehl			
260D	260D	09741	09741	Ansprung, Variable suchen und wenn nicht vorhanden, neu einrichten. Beim Rücksprung ist die Adresse in der Variablen-tabelle in DE	xxxx	xxxx	xxxx P
260D	2651	09741	09809	Variablen-namen prüfen und Typ feststellen			
2652	268D	09810	09869	Variable in Tabelle suchen			
26A0	26CE	09888	09934	Neue Variable einrichten			
26E9	27C8	09961	10184	Verwaltung der Feldvariablen. Suchen, neu Einrichten, Dimensionieren			
273D	2741	10045	10049	BS-Error			
27C9	27D3	10185	10195	MEM-Funktion			
27D4	27F4	10196	10228	FRE-Funktion			

					AF	BC	DG	HL
27F5	27FD	10229	10237	POS-Funktion				
27F8	27FD	10232	10237	Akku als Integerzahl nach HL und X	xxxx			xxxx
27FE	2818	10238	10264	USR-Funktion				
2819	2827	10265	10279	Wert in X wird in den Typ, dessen Code im Akku steht, umgewandelt	xxxx	xxxx	xxxx	xxxx
2828	2835	10280	10293	Wenn Rechner im Direkt-Mode, ID-Error	xxxx			
2831	2835	10289	10293	ID-Error				
2836	2AEE	10294	10990	Stringspaceverwaltung und Stringfunktionen				
2836	2856	10294	10326	STR\$-Funktion				
2857	2864	10327	10340	Platz für String mit der Länge A in Stringspace schaffen und Adresse (in DE) in Zwischenspeicher eintragen	xx	xxxx	xxxx	40D3
2865	28A5	10341	10405	Stringkonstante in Zwischenspeicher und X übernehmen. Wenn Zwischenspeicher voll, ST-Error	xxxx	xxxx	40D6	P
28A1	28A5	10401	10405	ST-Error				
28A6	28BE	10406	10430	String ab HL+1 bis Anführungsstrichen oder 0 ausdrucken	xxxx	xxxx	00xx	xxxx
28BF	298E	10431	10638	In der Stringspace Platz für String mit der Länge A machen. Adresse in DE. Wenn nötig, Stringspace umsortieren und überflüssige Strings entfernen. Ist es auch so nicht möglich, genügend Platz zu schaffen, OS-Error	xx	xxxx	xxxx	xxxx
298F	29C5	10639	10693	Stringverknüpfung				
29C6	29D6	10694	10710	String in Stringspace übernehmen				
29CE	29D6	10702	10710	String mit der Länge L in Stringspace übernehmen	xxxx	xxxx	xxxx	00
				Zeiger: BC (String) und DE (Adresse in Stringspace)				
29D7	29F4	10711	10740	Letzten String aus Stringspace und Zwischenspeicher entfernen				
29F5	2A02	10741	10754	Letzten String aus Zwischenspeicher entfernen				
2A03	2A0E	10755	10766	LEN(X) ♦ X ♦ HL	xxxx	27F8	xxxx	xxxx
2A0F	2A1E	10767	10782	ASC(X) ♦ X ♦ HL	xxxx	27F8	xxxx	xxxx
2A1F	2A2E	10783	10798	CHR\$-Funktion				
2A2F	2A60	10799	10848	STRING\$-Funktion				
2A61	2A90	10849	10896	LEFT\$-Funktion				
2A91	2A99	10897	10905	RIGHT\$-Funktion				
2A9A	2AC4	10906	10948	MID\$-Funktion				
2AC5	2ADE	10949	10974	VAL-Funktion				
2ADF	2AE6	10975	10982	UP für Stringverarbeitung				
2AE7	2AEE	10983	10990	MID\$ auf linker Seite der Zuweisung				
				Nur in Disk-Basic vorhanden, daher Sprung nach 41D9.				
2AEF	2AFA	10991	11002	INP-Funktion				
2AFB	2B00	11003	11008	OUT-Befehl				
2B01	2B0D	11009	11021	Ganzzahliger Wert des Ausdrucks nach DE	xxxx	xxxx	xxxx	P
2B0E	2B1A	11022	11034	2 Argumente (< 256) für OUT analysieren				
2B1B	2B28	11035	11048	Ausdruck auswerten und ganzzahligen Wert (< 256) nach A.	xxxx	xxxx	xxxx	P
				Wird dieser Wert überschritten, FC-Error				
2B29	2B7D	11049	11133	LIST-Befehl				
2B29	2B29	11049	11049	LLIST-Ansprung				
2B2E	2B2E	11054	11054	LIST-Ansprung				
2B75	2B7D	11125	11133	UP für LIST, druckt String (Zeiger HL) bis 0.	xxxx			xxxx
2B7E	2BC5	11134	11205	UP für LIST, Zwischencode im Programmtext (Zeiger HL) in Programmtext mit Keywords zurückverwandeln, welcher nachher im I/O-Buffer vorliegt	xxxx	xxxx	xxxx	P
2BC6	2BF4	11206	11252	DELETE-Befehl				
2BF5	2C1E	11253	11294	CSAVE-Befehl				
2C1F	2CA4	11295	11428	CLOAD und CLOAD?				
2C3D	2CA4	11325	11428	Programm von Kassette laden (D=0) oder mit residentem Programm vergleichen (D=FF). Filename in E				
2CA5	2CA9	11429	11433	Text „BAD(CR)“				
2CAA	2CB0	11434	11440	PEEK-Funktion				
2CB1	2CBC	11441	11452	POKE-Befehl				
2CBD	2E52	11453	11858	USING (formatiertes Ausdrucken von Zahlen und Strings)				
2E53	2FFA	11859	12282	EDIT-Befehl (Zeileneditor)				
2E53	2E53	11859	11859	Ansprung nach SN-Fehlern				
2E60	2E60	11872	11872	normaler Ansprung				
2E60	2E6F	11872	11887	Analyse des Argumentes				
2E70	2E9A	11888	11930	Rückübersetzung des Programmtextes				
2E9B	2E9D	11931	11933	Eingabe				
2E9E	2EAF	11934	11951	Verarbeitung von Zahleneingaben				
2EB0	2F09	11952	12041	Ansprung der Unterbefehle				
2F0A	2F15	12042	12053	Space				
2F16	2F3F	12054	12095	K (Kill)				
2F1C	2F3F	12060	12095	S (Search)				
2F40	2F49	12096	12105	L (List)				
2F4A	2F64	12106	12132	D (Delete)				

2F65	2F74	12133	12148	C (Change)
2F75	2FD1	12149	12241	H (Hack and insert)
2F78	2FD1	12152	12241	X (Extend)
2F7D	2FD1	12157	12241	I (Insert)
2FD2	2FDF	12242	12255	♦ (Backspace)
2FE0	2FF5	12256	12277	Enter (Exit)
2FE3	2FF5	12259	12277	E (Save Changes and Exit)
2FF6	2FFA	12278	12282	Q (Cancel and Exit)
2FFF	2FFF	12287	12287	Ende des ROMs

I/O-Adressen

I/O-Ports:

FF	FF	00255	00255
----	----	-------	-------

Universeller Ein/Ausgabeport für Systemanwendungen:

Bit 0,1: Steuern die Spannung am AUX-Stecker: 00 und 11 = 0,45 V, 01 = 0,9 V, 10 = 0 V.

Bit 2: Wenn dieses Bit gesetzt ist, ist das Reedrelais angesteuert, also der Kassettenrecorder angeschaltet.

Bit 3: Umschaltung 64/32 cpl (0/1)

Bit 7: Eingabeport. Wenn am EAR-Stecker eine Spannung auftritt, wird dieses Bit gesetzt.

Es behält diesen Zustand, bis es durch Schreiben zurückgesetzt wird

Memory mapped I/O:

37DE	37DE	14302	14302	Communication Status Address
37DF	37DF	14303	14303	Communication Data Address
37E0	37E0	14304	14304	Interrupt Latch Address
37E1	37E1	14305	14305	Disk Drive Latch
37E4	37E4	14308	14308	Cassette Select Latch
37E8	37E8	14312	14312	Line Printer Address
37EC	37EC	14316	14316	Floppy Disc Controller Address
3800	3BFF	14336	15359	Tastatur-Adressen:

nur mit Expansion

3801	3801	14337	14337	Bit 7 G O W
3802	3802	14338	14338	Bit 6 F N V
3804	3804	14340	14340	Bit 5 E M U
3808	3808	14344	14344	Bit 4 D L T
3810	3810	14352	14352	Bit 3 C K S
3820	3820	14368	14368	Bit 2 B J R
3840	3840	14400	14400	Bit 1 A I Q
				Bit 0 @ H P
				7 6 5 4 3 2 1 0
				/ • - , ; : 9 8
				Space Br Cl CR

(Br = Break, Cl = Clear, CR = Enter) SHIFT

Bei diesen Adressen wird genau eine Zeile mit 8 Tasten abgefragt.

Wenn eine Taste gedrückt ist, ist das entsprechende Bit gesetzt

Abfrage aller Tasten außer Shift. Wenn dieser

Speicherplatz 0 ist, ist keine Taste gedrückt

Dergleichen mit Shift

Bildschirm-RAM (enthält kein Bit 6!)

RAM-Adressen

4000	4014	16384	16404	RST-Vektoren, wird einer der RSTs 8-38H ausgeführt, so wird an diese Adressen gesprungen:
4000	4000	16384	16384	RST 8H-Vektor (C3 96 1C = JP 1C96)
4003	4005	16387	16387	RST 10H-Vektor (C3 78 1D = JP 1D78)
4006	4008	16390	16390	RST 18H-Vektor (C3 90 1C = JP 1C90)
4009	4009	16393	16393	RST 20H-Vektor (C3 D9 25 = JP 25D9)
400C	400C	16396	16396	RST 28H-Vektor (C9 00 00 = RET), wird von INCH bei Drücken der Break-Taste angesprungen.
400F	400F	16399	16399	RST 30H-Vektor (C9 00 00 = RET)
4012	4012	16402	16402	RST 38H-Vektor (FB C9 00 = EI,RET) Die RSTs 30H und 38H werden im Level-II-Basic nicht verwendet, wohl aber bei Anschluß der Expansion oder eines Floppies
4015	402C	16405	16428	Device Control Blocks (DCB)
4015	401C	16405	16412	Keyboard DCB
4015	4015	16405	16405	DCB-Typ (01)
4016	4017	16406	16407	Treiberadresse (03E3)
4018	401C	16408	16412	(00 00 00 4B 49)
401D	4024	16413	16420	Display DCB
401D	401D	16413	16413	DCB-Typ (06)
401E	401E	16414	16415	Treiberadresse (0458)
4020	4021	16416	16417	Cursoradresse (3C00)

4022	4022	16418	16418	Cursorzeichen (00), 0 wenn Cursor ausgeschaltet, sonst das Zeichen, welches auf der Cursorposition stand (44 4F)
4023	4024	16419	16420	Printer DCB
4025	402C	16421	16428	DCB-Typ (07)
4025	4025	16421	16421	Treiberadresse (058D)
4026	4027	16422	16423	Anzahl der Zeilen pro Seite (43)
4028	4028	16424	16424	Nummer der Zeile in laufender Seite (00)
4029	4029	16425	16425	(00 50 52)
402A	402D	16426	16428	(0D 00 50 C7 00 00)
402D	4032	16429	16434	Wird angesprungen, wenn DCB-Typen nicht übereinstimmen (3E 00 C9 = LD A,0,RET)
4033	4035	16435	16437	Letzter Tastaturstatus, dient dazu festzustellen, welche Taste neu gedrückt worden ist = OUT (FF)
4036	403C	16438	16444	Wird nur vom DOS belegt, kann also im LEVEL-II-Basic z. B. für kleine USR-Routinen verwendet werden
403D	403D	16445	16445	FDC-Interrupt-Vector
403E	407F	16446	16511	Communications Interrupt Vector
4050	4051	16464	16465	25 ms „Heartbeat“-Interrupt
4052	4053	16466	16467	UP für Division
405E	405F	16478	16479	Startadresse von USR-Funktion (1E4A=FC-Error)
4080	408D	16512	16525	Zwischenspeicher für RND (40 E6 4D)
408E	408F	16526	16527	UP für INP (DB 00 C9 = IN A,(00),RET)
4090	4092	16528	16530	UP für OUT (D3 00 C9 = OUT (00),A,RET)
4093	4095	16531	16533	Speicher für INKEY\$-Funktion, enthält letzten Tastendruck während des Programmes bzw. 0(00)
4096	4098	16534	16536	Letzter Errorcode für ERR (00)
4099	4099	16537	16537	Position des Cursors in der Zeile (00)
409A	409A	16538	16538	Flag für Ausgabe (00 = Display, 80 = Kassette, 01 = Printer) (00)
409B	409B	16539	16539	Anzahl der Zeichen pro Zeile auf dem Bildschirm (wird für Zahlenausgabe benötigt) (40)
409C	409C	16540	16540	Letzte Tabulatorposition (30)
409D	409D	16541	16541	n. v.
409E	409E	16542	16542	Anfang der Stringspace (434C)
409F	409F	16543	16543	Aktuelle Zeilennummer (FFFE)
40A0	40A1	16544	16545	Anfang des Programmtextes (42E9)
40A2	40A3	16546	16547	Position des Cursors in der Zeile auf dem Bildschirm (20)
40A4	40A5	16548	16549	Anfang des I/O-Puffers (41E8)
40A6	40A6	16550	16550	Flag für Input (00 = Keyboard, sonst Kassette)
40A7	40A8	16551	16552	Zwischenspeicher für RND
40A9	40A9	16553	16553	n. v.
40AA	40AC	16554	16556	DIM-Flag (00 = kein DIM)
40AD	40AD	16557	16557	Typ-Code des Inhaltes des X-Registers
40AE	40AE	16558	16558	Flag für Zwischencodeerzeugung (nach DATA = 4E, sonst 0)
40AF	40AF	16559	16559	Letzter Speicherplatz, der für Basic zur Verfügung steht
40B0	40B0	16560	16560	Zwischenspeicher für momentan verarbeitete Strings
40B1	40B2	16561	16562	Zeiger auf nächsten freien Zwischenspeicherplatz
40B3	40D5	16563	16597	11 Zwischenspeicher für Strings (Länge, Adresse)
40B3	40B4	16563	16564	vorläufiger Zwischenspeicher, vor Eintragung in den Zwischenspeicher
40B5	40D5	16565	16597	Letztes freie Byte im Stringspace
40D3	40D5	16595	16597	Flag für Zahlenausgabe
40D6	40D7	16598	16599	Flag für Feldverwaltung
40D8	40D8	16600	16600	Zeilennummer, der DATA-Zeile, die gerade gelesen wird
40D8	40D9	16600	16601	Flag zur Sperrung von Feldvariablen (0 = Felder freigegeben)
40DA	40DB	16602	16603	Flag
40DC	40DC	16604	16604	Flag für INPUT (00 = Keyboard, sonst READ)
40DD	40DD	16605	16605	Startadresse von Objektfiles (mit SYSTEM geladen), wird auch anderweitig verwendet
40DE	40DE	16606	16606	AUTO-Flag (00 = kein AUTO)
40DF	40E0	16607	16608	AUTO-Anfangsadresse
40E1	40E1	16609	16609	AUTO-Increment
40E2	40E3	16610	16611	Adresse des Befehls, der gerade verarbeitet wird
40E4	40E5	16612	16613	Wert mit dem Stackpointer initialisiert wird, wird durch FOR, GOSUB usw. erniedrigt.
40E6	40E7	16614	16615	Zeile in der der letzte Fehler auftrat (für ERL)
40E8	40E9	16616	16617	Aktuelle Zeile für „“-Option
40EA	40EB	16618	16619	Adresse des Befehl, bei dessen Ausführung der letzte Fehler auftrat
40EC	40ED	16620	16621	Anfangsadresse der Errortraproutine.
40EE	40EF	16622	16623	Wenn kein Errortrap benutzt wird, beinhalten diese Speicherplätze 0
40F0	40F1	16624	16625	Fehlerflag: Wird beim Auftreten eines Errortraps gesetzt (FF) und durch RESUME wieder zurückgesetzt (00)
40F2	40F2	16626	16626	Zwischenspeicher für die Auswertung von Ausdrücken
40F3	40F4	16627	16628	Zeilennummer der letzten Programmunterbrechung (durch Fehler oder Break)
40F5	40F6	16629	16630	Adresse der letzten Programmunterbrechnung, 0 wenn keine aufgetreten, oder Fortführung der Programmausführung nicht möglich
40F7	40F8	16631	16632	

40F9	40FA	16633	16634	Anfang der Variablentabelle
40FB	40FC	16635	16636	Anfang der Arraytabelle
40FD	40FE	16637	16638	Ende der Arraytabelle
40FF	4100	16639	16640	Datazeiger, zeigt auf das Trennzeichen nach den zuletzt gelesenen Daten
4101	411A	16641	16666	Tabelle, enthält in alphabetischer Reihenfolge für jeden Buchstaben einen Typcode. Beinhaltet ein Variablenname keine Typbezeichnung, wird der Typ der Tabelle entnommen. Durch NEW wird 04 in die ganze Tabelle geschrieben. Durch DEFSTR, DEFINT, DEFSNG und DEFDBL wird dieser Wert geändert
411B	411B	16667	16667	TRACE-Flag (00 = TROFF, AF = TRON)
411C	411C	16668	16668	Carry für Schiebeoperationen
411D	4124	16669	16676	X-Register, wichtigstes Register für Werte aller 4 Typen. Enthält z. B. das Ergebnis eines Ausdruckes oder Argument und Ergebnis bei Funktionsaufrufen
411D	4124	16669	16676	Zahl doppelter Genauigkeit: 411D enthält das niederwertigste Byte, 4123 das höchstwertigste und 4124 den Exponenten
4121	4124	16673	16676	Zahl einfacher Genauigkeit: Abspeicherung ähnlich wie bei Zahlen doppelter Genauigkeit
4121	4122	16673	16674	Integerzahl: LSB/MSB (4121/4122)
4121	4122	16673	16674	Strings: Adresse, die auf Zwischenspeicher oder Variablentabelle zeigt, wo String abgespeichert ist
4125	4126	16675	16678	Zwischenspeicher für Arithmetik
4127	412E	16679	16686	Y-Register, wichtiges Register besonders für 16stellige Arithmetik (enthält den zweiten Operanden). Abspeicherung wie im X-Register
412F	412F	16687	16687	Carry für Schiebeoperationen
4130	4149	16688	16713	Output-Buffer, enthält nach der Rückkonvertierung von Zahlen den entsprechenden String
414A	4151	16714	16721	Zwischenspeicher für Arithmetik
4152	41E4	16722	16868	Zeiger für Erweiterung des Befehlssatzes durch das Disk-Basic:
4152	41A5	16722	16805	Zeiger für neue Disk-Basic-Befehle und Funktionen. Bei der Initialisierung des Basic wird auf alle Zeiger C3 2D 01 = JP 012D = L3-Error geschrieben
4152	4154	16722	16724	CVI-Funktion
4155	4157	16725	16727	FN-Funktion
4158	415A	16728	16730	CVS-Funktion
415B	415D	16731	16733	DEF-Befehl
415E	4160	16734	16736	CVD-Funktion
4161	4163	16737	16739	EOF-Funktion
4164	4166	16740	16742	LOC-Funktion
4167	4169	16743	16745	LOF-Funktion
416A	416C	16746	16748	MKIS-Funktion
416D	416F	16749	16751	MKSS-Funktion
4170	4172	16752	16754	MKDS-Funktion
4173	4175	16755	16757	CMD-Befehl
4176	4178	16758	16760	TIMES-Funktion
4179	417B	16761	16763	OPEN-Befehl
417C	417E	16764	16766	FIELD-Befehl
417F	4181	16767	16769	GET-Funktion
4182	4184	16770	16772	PUT-Befehl
4185	4187	16773	16775	CLOSE-Befehl
4188	418A	16776	16778	LOAD-Befehl
418B	418D	16779	16781	MERGE-Befehl
418E	4190	16782	16784	NAME-Befehl
4191	4193	16785	16787	KILL-Befehl
4194	4196	16788	16790	&-Funktion
4197	4199	16791	16793	LSET-Befehl
419A	419C	16794	16796	RSET-Befehl
419D	419F	16797	16799	INSTR-Funktion
41A0	41A2	16900	16802	SAVE-Befehl
41A3	41A5	16803	16805	LINE-Befehl
41A6	41E4	16806	16868	Zeiger zur Erweiterung vorhandener Befehle durch das Disk-Basic (z. B.: PRINT). Diese Adressen werden vom Basicinterpreter aus verschiedenen Programmsegmenten als Unterprogramme angesprochen. Bei der Initialisierung werden die Zeiger durch einen RET ersetzt
41E5	41E7	16869	16870	(3A 00 2C) Alle folgenden Adressen sind nicht die Adressen selbst, sondern die der Zeiger
40A7		16551		Input/Outputbuffer
40A4		16548		Programmtext: Zeiger auf nächste Zeile (LSB/MSB)/Zeilennummer (LSB/MSB)/Text/00... (weitere gleichartige Zeilen) ...0000
40F9		16633		Einfache Variablen: Typ/2. Buchstabe/1. Buchstabe/Wert/...
40FB		16635		Feldvariablen: Typ/2. Buchstabe/1. Buchstabe/Anzahl aller folgenden Bytes (LSB/MSB)/Anzahl der Dimensionen/Tiefe jeder Dimension (LSB/MSB)/Werte/...
40FD		16637		Freier Speicher
40E8		16616		Anfangswert des Stacks für Rücksprungadressen
40A0		16544		Anfangswert des Stacks für Programmdateien (z. B. FOR-Schleifen)
40A0+1		16544+1		Anfang des Stringspace
40D6		16598		Letztes freie Byte des Stringspace. Alles, was darüber liegt, ist mit Zeichenketten belegt
40B1		16561		Letztes RAM-Speicherbyte oder der um 2 erniedrigte Eingabewert bei MEM SIZE

8-bit Arithmetische und Logische Befehle

	B	C	D	E	H	L (HL)	A	n	(IX+d)	(IY+d)	S	Z	H	P/V	N	C
ADD	80	81	82	83	84	85	86	87	CEXX	DD8EXX	FD8EXX	*	*	*	*	0
ADC	88	89	8A	8B	8C	8D	8E	8F	CEXX	DD8EXX	FD8EXX	*	*	*	*	0
SUB	90	91	92	93	94	95	96	97	DEXX	DD9EXX	FD9EXX	*	*	*	*	1
SBC	98	99	9A	9B	9C	9D	9E	9F	DEXX	DD9EXX	FD9EXX	*	*	*	*	1
AND	A0	A1	A2	A3	A4	A5	A6	A7	EEXX	DDAEXX	FDAEXX	*	*	*	*	0
XOR	A8	A9	AA	AB	AC	AD	AE	AF	EEXX	DDAEXX	FDAEXX	*	*	*	*	0
OR	B0	B1	B2	B3	B4	B5	B6	B7	FEXX	DDBEXX	FDBEXX	*	*	*	*	0
CP	B8	B9	BA	BB	BC	BD	BE	BF	FEXX	DDBEXX	FDBEXX	*	*	*	*	1
INC	04	0C	14	1C	24	2C	34	3C		DD34XX	FD34XX	*	*	*	*	0
DEC	05	0D	15	1D	25	2D	35	3D		DD35XX	FD35XX	*	*	*	*	1

		S	Z	H	P/V	N	C
DAA	27	*	*	*	*	-	*
CPL	2F	-	-	1	-	1	-
NEG	ED44	*	*	*	*	1	*

BCD-Korrektur im Akku
Komplementiere Akku (1er-Komplement)
Komplementiere Akku (2er-Komplement)

16-bit Arithmetische und Logische Befehle

	BC	DE	HL	SP	IX	IY	S	Z	H	P/V	N	C
INC	03	13	23	33	DD23	FD23	-	-	-	-	-	-
DEC	0B	1B	2B	3B	DD2B	FD2B	-	-	-	-	-	-
ADD HL	09	19	29	39			-	-	*	-	0	*
ADC HL	ED4A	ED5A	ED6A	ED7A			*	*	*	*	0	*
SBC HL	ED42	ED52	ED62	ED72			*	*	*	*	1	*
ADD IX	DD09	DD19		DD39	DD29		-	-	*	-	0	*
ADD IY	FD09	FD19		FD39	FD29		-	-	*	-	0	*

Rotations- und Schiebepfeile

	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)
RR	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E	CB1F	DDCBXX1E	FDCBXX1E
RL	CB18	CB11	CB12	CB13	CB14	CB15	CB16	CB17	DDCBXX1E	FDCBXX1E
RRC	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBXX0E	FDCBXX0E
RLC	CB08	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBXX0E	FDCBXX0E
SRA	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E	CB2F	DDCBXX2E	FDCBXX2E
SLA	CB28	CB21	CB22	CB23	CB24	CB25	CB26	CB27	DDCBXX2E	FDCBXX2E
SRL	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E	CB3F	DDCBXX3E	FDCBXX3E

	S	Z	H	P/V	N	C
RR/RL	*	*	0	*	0	*
RRC/RLC	*	*	0	*	0	*
SRA/SLA	*	*	0	*	0	*
SRL	*	*	0	*	0	*

Rotiere Reg rechts/links durch Carry
Rotiere Reg rechts/links
Shift Reg rechts/links arithmetisch
Shift Reg rechts/links logisch

	S	Z	H	P/V	N	C
RRCA	0F	-	-	0	-	0
RLCA	07	-	-	0	-	0
RRA	1F	-	-	0	-	0
RLA	17	-	-	0	-	0
RLD(HL)	ED6F	*	*	0	*	0
RRD(HL)	ED67	*	*	0	*	0

Rotiere Akku rechts
Rotiere Akku links
Rotiere Akku rechts durch Carry
Rotiere Akku links durch Carry
Rotiere Ziffer links zwischen Akku und (HL)
Rotiere Ziffer rechts zwischen Akku und (HL)

Einzelbitbefehle

	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)
SIT 0	CB40	CB41	CB42	CB43	CB44	CB45	CB46	CB47	DDCBXX4E	FDCBXX4E
BIT 1	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E	CB4F	DDCBXX4E	FDCBXX4E
BIT 2	CB50	CB51	CB52	CB53	CB54	CB55	CB56	CB57	DDCBXX5E	FDCBXX5E
BIT 3	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E	CB5F	DDCBXX5E	FDCBXX5E
BIT 4	CB60	CB61	CB62	CB63	CB64	CB65	CB66	CB67	DDCBXX6E	FDCBXX6E
BIT 5	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E	CB6F	DDCBXX6E	FDCBXX6E
BIT 6	CB70	CB71	CB72	CB73	CB74	CB75	CB76	CB77	DDCBXX7E	FDCBXX7E
BIT 7	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E	CB7F	DDCBXX7E	FDCBXX7E
RES 0	CB80	CB81	CB82	CB83	CB84	CB85	CB86	CB87	DDCBXX8E	FDCBXX8E
RES 1	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E	CB8F	DDCBXX8E	FDCBXX8E
RES 2	CB90	CB91	CB92	CB93	CB94	CB95	CB96	CB97	DDCBXX9E	FDCBXX9E
RES 3	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E	CB9F	DDCBXX9E	FDCBXX9E
RES 4	CBA0	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6	CBA7	DDCBXXAE	FDCBXXAE
RES 5	CBA8	CBA9	CBAA	CBAB	CBAC	CBAD	CBAE	CBAF	DDCBXXAE	FDCBXXAE
RES 6	CBB0	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6	CBB7	DDCBXXBE	FDCBXXBE
RES 7	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE	CBBF	DDCBXXBE	FDCBXXBE
SET 0	CB00	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBXX0E	FDCBXX0E
SET 1	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBXX0E	FDCBXX0E
SET 2	CB00	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBXX0E	FDCBXX0E
SET 3	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBXX0E	FDCBXX0E
SET 4	CB00	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBXX0E	FDCBXX0E
SET 5	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBXX0E	FDCBXX0E
SET 6	CBF0	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6	CBF7	DDCBXXFE	FDCBXXFE
SET 7	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE	CBFF	DDCBXXFE	FDCBXXFE

Flagbeeinflussung:	S	Z	H	P/V	N	C
BIT	?	*	1	?	0	-
SET	-	-	-	-	-	-
RES	-	-	-	-	-	-

Flag-Register

Bit	7	6	5	4	3	2	1	0
	S	Z	X	H	X	P/V	N	C

	gesetzt	nicht gesetzt	wird gesetzt bei
C Carry-Flag	C	NC	Uebertrag von Bit 7
N Add-/Subtract-Flag			Subtraktionen
P/V Parity-/Overflow-Flag	PE	PO	gerader Paritaet
H Half-Carry-Flag			Uebertrag von Bit 3
Z Zero-Flag	Z	NZ	Ergebnis 0
S Sign-Flag	M	P	neg. Ergebnis
X nicht verwendet			

Beeinflussung:	1 gesetzt
	0 zurueckgesetzt
	* abhaengig vom Ergebnis einer Operation
	- nicht beeinflusst
	? unbestimmt

Kurze Maschinenprogramme lassen sich auch ohne Assembler schnell und einfach mit den hexadezimalen Operationscodes schreiben. Heinz-Peter Curdts hat sich die Mühe gemacht, die Z80-Maschinenbefehle übersichtlich zusammenzustellen.

8-Bit-Ladebefehle

	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(nn)	n
LD A, ..	7F	78	79	7A	7B	7C	7D	7E	0A	1A	3AXXX	3EXX
LD B, ..	47	40	41	42	43	44	45	46				05XX
LD C, ..	4F	48	49	4A	4B	4C	4D	4E				0EXX
LD D, ..	57	50	51	52	53	54	55	56				15XX
LD E, ..	5F	58	59	5A	5B	5C	5D	5E				1EXX
LD H, ..	67	60	61	62	63	64	65	66				25XX
LD L, ..	6F	68	69	6A	6B	6C	6D	6E				2EXX
LD (HL), ..	77	70	71	72	73	74	75					35XX
LD (BC), ..	02											
LD (DE), ..	12											
LD (nn), ..	32XXXX											

	A	B	C	D	E	H	L
LD .., (IX+d)	DD7EXX	DD4EXX	DD4EXX	DD5EXX	DD5EXX	DD6EXX	DD6EXX
LD .., (IY+d)	FD7EXX	FD4EXX	FD4EXX	FD5EXX	FD5EXX	FD6EXX	FD6EXX
LD (IX+d), ..	DD77XX	DD70XX	DD71XX	DD72XX	DD73XX	DD74XX	DD75XX
LD (IY+d), ..	FD77XX	FD70XX	FD71XX	FD72XX	FD73XX	FD74XX	FD75XX

LD (IX+d), n	DD36XXXX	LD (IY+d), n	FD36XXXX
--------------	----------	--------------	----------

		S	Z	H	P/V	N	C
LD A, I	ED57	*	*	0	*	0	-
LD A, R	ED5F	*	*	0	*	0	-
LD I, A	ED47	-	-	-	-	-	-
LD R, A	ED4F	-	-	-	-	-	-

16-Bit-Ladebefehle

	BC	DE	HL	SP	IX	IY
LD .., nn	D1XXXX	11XXXX	21XXXX	31XXXX	DD21XXXX	FD21XXXX
LD .., (nn)	ED4BXXXX	ED5BXXXX	2AXXXX	ED7BXXXX	DD2AXXXX	FD2AXXXX
LD (nn), ..	ED43XXXX	ED53XXXX	22XXXX	ED73XXXX	DD22XXXX	FD22XXXX
LD SP			F9		DDF9	FDF9

	BC	DE	HL	AF	IX	IY
PUSH ..	C5	D5	E5	F5	DDE5	FDE5
POP ..	C1	D1	E1	F1	DDE1	FDE1

EX (SP), HL	ED		EX DE, HL	EB
EX (SP), IX	DD		EX AF, AF'	0B
EX (SP), IY	FD		EXX	D9 BC-BC' DE-DE' HL-HL'

Blocktransfer- und Suchbefehle

		S	Z	H	P/V	N	C
LDI	ED40	-	-	0	*	0	-
LDIR	ED41	-	-	0	0	0	-
LDD	ED42	-	-	0	*	0	-
LDHR	ED43	-	-	0	0	0	-
CPI	ED44	*	*	*	*	1	-
CPHR	ED45	*	*	*	*	1	-

LD(DE), (HL); INC HL; INC DE; DEC BC
wie LDI, wiederholen bis BC=0
LD(DE), (HL); DEC HL; DEC DE; DEC BC
wie LDD, wiederholen bis BC=0
LD A, (HL); INC HL; DEC BC
wie CPI, wiederholen bis BC=0 oder Zeichen gefunden

CPD	EDA9	*	*	*	*	1	-	LD A, (HL); DEC HL; DEC BC
CPDR	EDB9	*	*	*	*	1	-	wie CPD, wiederholen bis BC=0 oder Zeichen gefunden

Springbefehle

	Z	NZ	C	NC	PE	PO	M	P
JP	CAXXXX	C2XXXX	DAXXXX	D2XXXX	EAXXXX	E2XXXX	FAXXXX	F2XXXX
CALL	CCXXXX	C4XXXX	DCXXXX	D4XXXX	ECXXXX	E4XXXX	FCXXXX	F4XXXX
RET	C9	C0	D9	D0	E9	E0	F9	F0
JR	28XX	20XX	38XX	30XX				

	unbedinst	(HL)	(IX)	(IY)
JP	C3XXXX	E9	DDE9	FDE9
CALL	CDXXXX			
RET	C9			
JR	18XX			

DJNZ	10XX	DEC B; JR NZ, d
RETI	ED4D	zurueck vom Interrupt
RETN	ED45	zurueck vom nicht maskierbaren interrupt

CPU-Steuerbefehle

		S	Z	H	P/V	N	C
NOP	00	-	-	-	-	-	-
HALT	76	-	-	-	-	-	-
CCF	3F	-	-	*	-	0	*
SCF	37	-	-	0	-	0	1
EI	FB	-	-	-	-	-	-
DI	F3	-	-	-	-	-	-
IM 0	ED46	-	-	-	-	-	-
IM 1	ED56	-	-	-	-	-	-
IM 2	ED5E	-	-	-	-	-	-

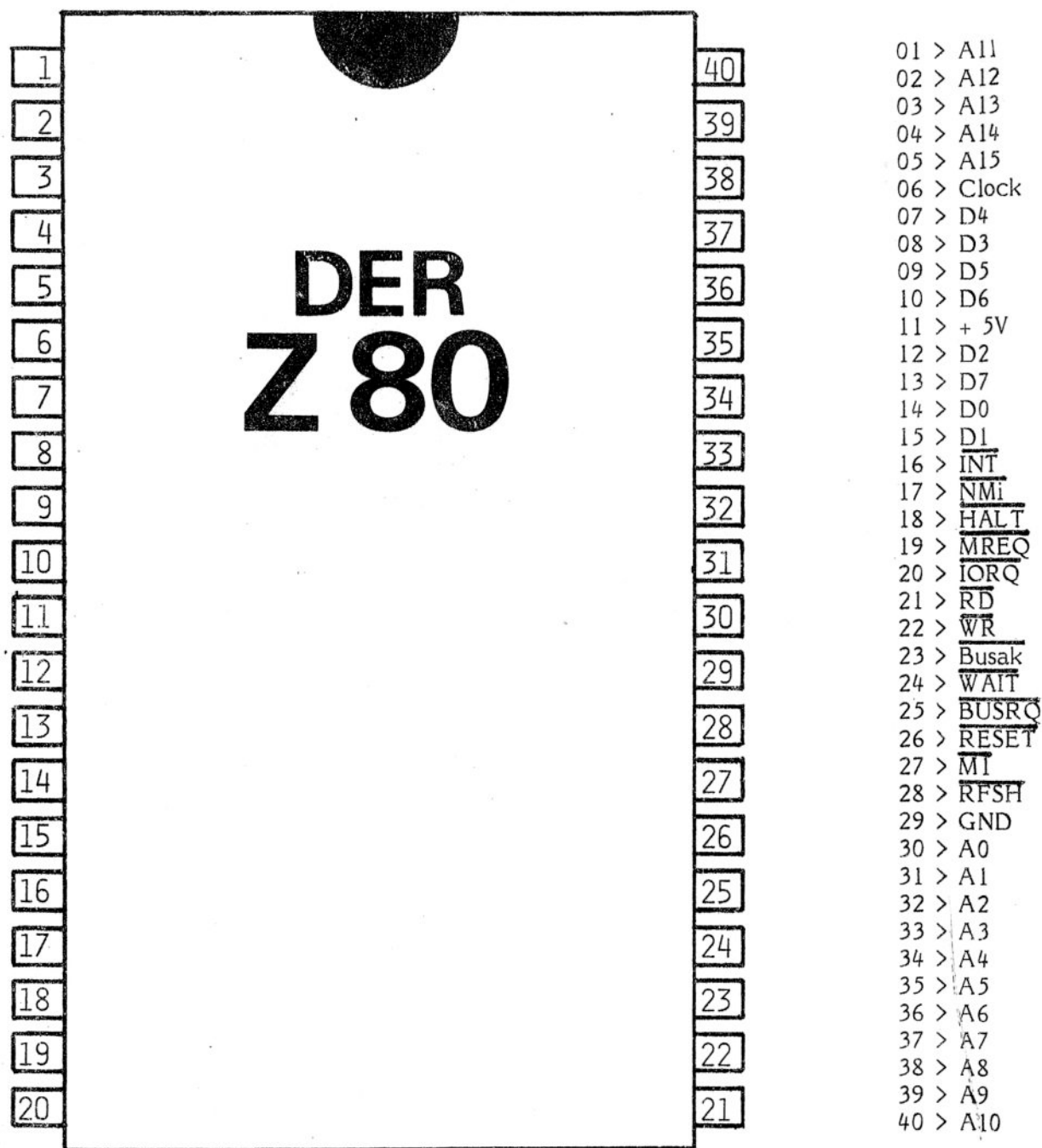
Leerbefehl
Komplementiere Carry-Flag
Setze Carry-Flag
Interrupts freigeben
Interrupts sperren
Interrupt-Modus 0
Interrupt-Modus 1
Interrupt-Modus 2

Ein-/Ausgabebefehle

		A	B	C	D	E	H	L	S	Z	H	P/V	N	C
IN .., (C)	ED78	ED40	ED48	ED50	ED58	ED60	ED68		*	*	0	*	0	-
OUT (C), ..	ED79	ED41	ED49	ED51	ED59	ED61	ED69		-	-	-	-	-	-

		S	Z	H	P/V	N	C
IN A, n	DBXX	-	-	-	-	-	-
OUT n, A	D3XX	-	-	-	-	-	-
INI	EDA2	?	*	?	?	1	-
INIR	EDB2	?	1	?	?	1	-
IND	EDAA	?	*	?	?	1	-
INDR	EDBA	?	1	?	?	1	-
OUTI	ED43	?	*	?	?	1	-
OTIR	EDB3	?	1	?	?	1	-
OUTD	EDAB	?	*	?	?	1	-
OTDR	EDBB	?	1	?	?	1	-

(HL), (C); INC HL; DEC B
wie INI, wiederholen solange B()0
IN (HL), (C); DEC HL; DEC B
wie IND, wiederholen solange B()0
OUT (C), (HL); INC HL; DEC B
wie OUTI, wiederholen solange B()0
OUT (C), (HL); DEC HL; DEC B
wie OUTD, wiederholen solange B()0



Nicht nur in Genie Computern und im TRS 80 sitzt als Mikroprozessor ein Z80. Der Z80 ist einer der am häufigsten verwendeten Mikroprozessoren auf der Welt. Was liegt also näher, als Ihnen einmal etwas über ihn zu berichten? Wie Sie sehen, hat der Z80 vierzig Beinchen, in der Fachsprache Pin's genannt. Wenn man im Wörterbuch nachschaut, findet man unter Pin das Wort Nadel und in der Tat sehen ja die Anschlußstifte fast aus wie Nadeln. Übrigens nennt man die Stifte nicht nur beim Z80 Pin, sondern bei allen IC's. (IC = Integrated Circuit = Integrierter Schaltkreis.)

Wir haben die einzelnen Pin's durchnummeriert, um Ihnen das Auffinden zu erleichtern. Nun aber zur Belegung:

A0 - A15

Über die Pin's A0 - A15 werden die Adressen, die der Z80 ansprechen möchte, gekennzeichnet. Und zwar folgendermaßen:

Nehmen wir an, der Prozessor möchte aus dem Speicher mit der Adresse 4 einen Wert lesen. Er geht dann mit allen Pin's auf 0 Volt Spannung (low), nur Pin A3 bleibt hoch. Wenn wir das nun in duale Schreibweise umwandeln, bedeutet eine Eins an dritter Stelle von links dezimal 4. Und damit haben wir auch schon die gewünschte Adresse. Weiterhin können wir daran sehen, daß er in der Lage ist 64K Speicher direkt zu adressieren.

Anmerkung: Bei manchen I/O Operationen gelten nur die unteren 8 Bit's als Adresse, woge-

gen die anderen 8 Bit's den Inhalt des Akkumulators enthalten.

D \emptyset - D7

Das sind die Datenleitungen des Z80 Prozessors. Über diese acht Leitungen sendet oder empfängt der Prozessor Daten. (Datenbus).

Pin 16 - $\overline{\text{INT}}$ (Interrupt)

Nach dem Beenden eines laufenden Befehls, nimmt der Z80 über dieses Pin einen Interrupt (Unterbrechung) an, sofern das interne Interruptenable (Unterbrechungsfreigabe)-Register ein ist und kein $\overline{\text{BUSRQ}}$ (bus request/siehe unten) vorliegt. Bei einem Befehl, der sehr viele Maschinenzyklen lang ist, wird ein Interrupt auch während der Ausführung dieses Befehls - z. B. Blocktransfer - angenommen.

Pin 17 - $\overline{\text{NMI}}$ (Non maskable Interrupt)

Ein $\overline{\text{NMI}}$ (Nicht maskierbare Unterbrechung) ist der mächtigste unter den Interrupt's. Er wird am Ende eines Befehls immer angenommen und veranlaßt den Z80 zu einem Sprung auf die Speicherzelle 66H. Nur ein $\overline{\text{BUSRQ}}$ oder $\overline{\text{WAIT}}$ kann den $\overline{\text{NMI}}$ davon abhalten, sofort wirksam zu werden.

Pin 18 - $\overline{\text{Halt}}$

Der Z80 wird hierdurch gestoppt, das heißt, er führt während eines Halts ständig NOP's aus, so daß der Refresh für dynamische RAM's trotzdem korrekt ausgeführt wird.

Pin 19 - $\overline{\text{MREQ}}$ (Memory Request/Speicher Bedarf)

$\overline{\text{MREQ}}$ zeigt an, daß an den Adreß-Leitungen A0 - A15 eine gültige Adresse ansteht.

Pin 20 - $\overline{\text{IORQ}}$ (Input/Output Request/ Eingabe/Ausgabe Bedarf)

Hiermit zeigt der Prozessor an, daß an der unteren Hälfte des Adreßbuses eine gültige Adresse für eine I/O - Operation anliegt.

Pin 21 - $\overline{\text{RD}}$ (Read/Lesen)

Der Z80 zeigt an, daß er Daten vom Speicher oder einem Peripherie-Gerät erwartet.

Pin 22 - $\overline{\text{WR}}$ (Write/Schreiben)

Mit $\overline{\text{WR}}$ teilt der Prozessor mit, daß er Daten, die am Datenbus anliegen in den Speicher schreiben oder einem Peripherie-Gerät übergeben möchte.

Pin 23 - $\overline{\text{BUSAk}}$ (Bus Aktiv)

Hiermit zeigt der Z80 an, daß ein $\overline{\text{BUSRQ}}$ korrekt erledigt wurde.

Pin 24 - $\overline{\text{WAIT}}$ (Warte)

Dem Prozessor wird durch $\overline{\text{WAIT}}$ angegeben, daß ein Speicher oder Peripherie-Gerät Daten noch nicht verarbeitet hat oder Daten noch nicht bereitgestellt wurden. Während $\overline{\text{WAIT}}$ ak-

tiv ist erfolgt kein Refresh.

Pin 25 - $\overline{\text{BUSRQ}}$ (Bus Request/Bus Anforderung)
Durch $\overline{\text{BUSRQ}}$ wird vom Prozessor verlangt, nach Beendigung des momentanen Zyklus, seinen Daten-, Adreß- und Steuerbus in einen TRI-State Zustand zu überführen.

Pin 26 - $\overline{\text{RESET}}$ (Zurücksetzen)

Nach einem $\overline{\text{RESET}}$ wird der Programm-Counter (Programm-Zähler) auf Null gesetzt, d. h., ein Programm ab Adresse Null wird gestartet. Außerdem werden interne Register so geschaltet, daß das $\overline{\text{INT}}$ -Verhalten wie beim 8080 Prozessor ist.

Pin 27 - $\overline{\text{RFSH}}$ (Refresh/Auffrischen)

Auf A0 - A6 liegt eine 7Bit lange Refresh Adresse für dynamische RAM's. A7 ist nicht verwendet und A8 - A15 zeigen den Inhalt von I (Interruptregister).

Pin 6 - Clock (Taktfrequenz)

Takteingang (TTL-kompatibel).

Zum Schluß zeigen wir Ihnen noch den Registersatz des Z80 (Bild 1) und den Aufbau des Flag-Registers F (Bild 2).

BILD 1

A	F
B	C
D	E
H	L

I	R
IX	
IY	
SP	
PC	

BILD 2

S	Z	X	H	X	PV	N	C
---	---	---	---	---	----	---	---

Schnelle Hardcopy

HARDCO (Hardcopy Programm fuer Colour Genie) / GENIE I + II

COLOUR 1 GENIE TRS 80

```

10      ORG 7F00H
20 START LD HL, 4400H 3C00H
30      LD E, 24
40 MARKE LD D, 40 64D
50 LOOP  CALL AUSGA
60      INC HL
70      DEC D
80      JR NZ, LOOP
90      CALL FEED
100     DEC E
110     JR NZ, MARKE
120 BASIC JF 0066H
130 AUSGA LD A, (HL)
140     PUSH DE
150     CALL 003BH ✓
160     POP DE
170     RET
180 FEED  LD A, 0DH
190     PUSH DE
200     CALL 003BH ✓
210     POP DE
220     RET
230     END BASIC
    
```

MEMSIZE BEI 16K = 32511

HARDCO HEXDUMP

```

7F00 21 00 44 1E 18 16 28 CD 17 7F 23
7F0B 15 20 F9 CD 1E 7F 1D 20 F1 C3 66
7F16 00 7E D5 CD 3B 00 D1 C9 3E 0D D5
7F21 CD 3B 00 D1 C9 FF FF FF 00 FF FF
    
```

2.

TRS-80, Video Genie: Schnelle Hardcopy mit einer BASIC-Zeile

Die Idee für diesen Trick stammt aus 'BASIC FASTER AND BETTER' (Hofacker Verlag). Die Lösung dort ist zwar etwas umständlicher, aber dennoch: 'BASIC FASTER AND BETTER' ist das beste Buch über den TRS-80, das ich je gelesen habe.

Zu jeder String gehört ein 3-Byte-Vektor des Inhalts

Byte 1 : Stringlänge

Byte 2 : LSB Adresse

Byte 3 : MSB Adresse

In der Variablen-Tabelle ist nur der Vektor abgelegt. Die String selbst steht entweder im Programmtext (Stringkonstante) oder im mit CLEAR reservierten Bereich (Stringvariable).

Das Programm tut nun nichts weiter, als diesen Vektor so zu 'verbiegen', daß er nacheinander auf je eine der 16 Zeilen des Video-RAMs zeigt. Der Befehl LPRINT V\$ bewirkt dann den Ausdruck der Zeile.

Die Adresse der ursprünglichen String wird mit Hilfe der VARPTR-Funktion festgestellt. Die beiden Blindzuweisungen 'I=I' und 'V=V' stellen sicher, daß nach VARPTR keine neue Variable initialisiert wird.

Wenn Sie eine Kleinschrift-Modifikation eingebaut haben, muß der 'Driver' aktiv sein.

1000 I=I:V=V:V\$="" :V=VARPTR(V\$):

```

POKE V,64:FOR I=15360 TO 16320
STEP 64:POKE V+1,I AND 255:POKE
V+2,INT (I/256):LPRINT V$ NEXT:
RETURN
    
```

P. Wollschläger, Hildesheim ■

Für COLOUR's : -V, 40 - -1 = 17408 TO 18383 STEP 40 -

Falsche Eingabe

Eine Fehlerbehandlungsroutine von Michael Karnatz

Es ist allgemein ueblich, nach einem INPUT zu ueberpruefen, ob die Eingabe in der richtigen Groessenordnung erfolgte.

Das ist auch erforderlich, denn es ist schon recht aergerlich wenn der Computer wegen einer falschen Eingabe aussteigt.

Aber oft wird, wenn eine Zahl erwartet wird, nur mit dem Variablentyp fuer Zahlen gearbeitet. Wird nun ein Buchstabe oder Zeichen eingegeben, meldet sich der Computer mit REDO? und versaut einem das ganze Bildschirmbild.

Man sollte daher grundsaeztlich nach einer Stringvariablen abfragen, denn dieser Variablentyp nimmt Buchstaben, Zeichen und Zahlen an.

In Basic sieht das dann so aus:

```
10 INPUT "Welche Zahl (1...4)";T$
```

Mit folgender Zeile wird der Zahlenwert von T\$ ermittelt:

```
20 T=VAL(T$)
```

Wenn das erste Zeichen von T\$ keine Zahl ist, ergibt sich T=0.

In der naechsten Zeile ueberpruefen wir die richtige Groessenordnung von T.

```
30 IF T<1 OR T>4 THEN GOSUB 50000
```

Dabei habe ich vorausgesetzt, dass die Fehlerbehandlungsroutine in der Zeile 50000 steht. Es ist zweckmaessig, diese Routine in allen Programmen an die gleiche Stelle zu schreiben, damit weiss man immer wo sie steht.

Etwas schwieriger wird es, wenn die Zahl auch Null sein kann.

Denn wie wir oben gesehen haben, ergeben auch Buchstaben den Wert Null.

Also muessen wir nach dem Feststellen des Wertes abfragen, ob das erste Zeichen von T\$ eine Null ist.

```
30 IF T=0 AND LEFT$(T$,1) <>"0" OR T<0 OR T>4 THEN GOSUB 50000
```

Nun nimmt der Computer nur Zahlen zwischen 0 und 4 an.

Bei der Betaetigung aller anderen Tasten geht er in die Zeile 50000.

Allerdings mit drei Ausnahmen: bei <, > und <:> meldet er sich mit:

EXTRA IGNORED

Die BREAK Taste hat die uebliche Wirkung,

Nun zur Routine selbst. Fehlerbehandlung ist eigentlich falsch.

Denn die eigentliche Aufgabe dieser Routine ist es, den Bediener zu wecken.

Denn das scheint noetig zu sein. Schliesslich hat er schon einmal Mist eingegeben, denn sonst waere diese Routine nicht angesprungen worden.

Deshalb hat es keinen Sinn den Bediener irgendeine Taste druecken zu lassen. Er soll richtig aufwachen und ganz gezielt eine Taste druecken.

In diesem Programm ist es die RETURN Taste.

Um den Weckvorgang zu erleichtern, wird die Tonleiter gespielt und die Anzeige blinkt in der letzten Zeile.

Wenn man sich die letzte Zeile in allen Programmen fuer solche Routinen freihaelt, macht man sich auch sein Bildschirmbild nicht kaputt.

Das entscheidende tut sich aber, wenn die RETURN Taste gedrueckt wurde und der Computer an die Stelle zurueckkehrt, die dem Befehl GOSUB 50000 folgt.

Zur Zeit steht dort noch nichts. Das wollen wir jetzt aendern. Als erstes muss der Cursor wieder in die richtige Zeile gelangen und der vorher eingegebene Mist muss geloescht werden. Um die richtige Zeile wiederfinden zu koennen, muessen wir erst die Zeile 10 modifizieren:

```
10 ?$400.'Welche Zahl (0...4)';:INPUT T$
```

Der eingegebene Wert steht also ab \$420. Um den vorher eingegebenen Mist zu loeschen erzeugen wir uns einen Leerstring mit 10 Leertaster zu Beginn des Programms

```
5 L$=STRING$(10,32)
```

und legen diesen Leerstring ueber die falsche Eingabe:

```
?$420 L$
```

Nachdem wir die falsche Eingabe geloescht haben, muss nun erneut abgefragt werden. Es muss also ein GOTO 10 folgen.

Das komplette Programm sieht also wie folgt aus:

```
5 L$=STRING$(10," ")
```

```
9 CLS :COLOUR 1
```

```
10 PRINT$400,"Welche Zahl (0...4)";:INPUT T$
```

```
20 T=VAL(T$)
```

```
30 IF T=0 AND LEFT$(T$,1) <>"0" OR T<0 OR T>4 THEN GOSUB 50000 : PRINT$420,L$ : GOTO 10
```

```
40 PRINT"Eingabe richtig!"
```

```
50 STOP
```

```
50000 'F a l s c h e   E i n g a b e   *****
```

```
50010 FOR OK =2 TO 6 : 'Oktaven
```

```
50020 FOR NO =17 TO 23 : 'Noten (neue ROMs)
```

```
50030 COLOUR RND(15)
```

```
50040 PRINT$960,"Falsche Eingabe! Bitte RETURN druecken";
```

```
50050 PLAY(1,OK,NO,11)
```

```
50060 FOR I=1 TO 100 : 'Dauer Schrift und Ton
```

```
50070 NEXT I
```

```
50080 PRINT$960,CHR$(30); : 'Bis Zeilenende loeschen
```

```
50090 PLAY(1,1,1,0)
```

```
50100 FOR I=1 TO 100 : 'Dauer keine Schr. und Ton
```

```
50110 NEXT I
```

```
50120 T$=INKEY$:IF T$<>CHR$(13) THEN 50130 : ELSE 50180
```

```
50130 NEXT NO
```

```
50140 NEXT OK
```

```
50150 GOTO 50000
```

```
: 'Wenn alle Oktaven durch
```

```
50180 COLOUR 1
```

```
50190 RETURN
```

Einbau von Umlauttasten in den TRS 80

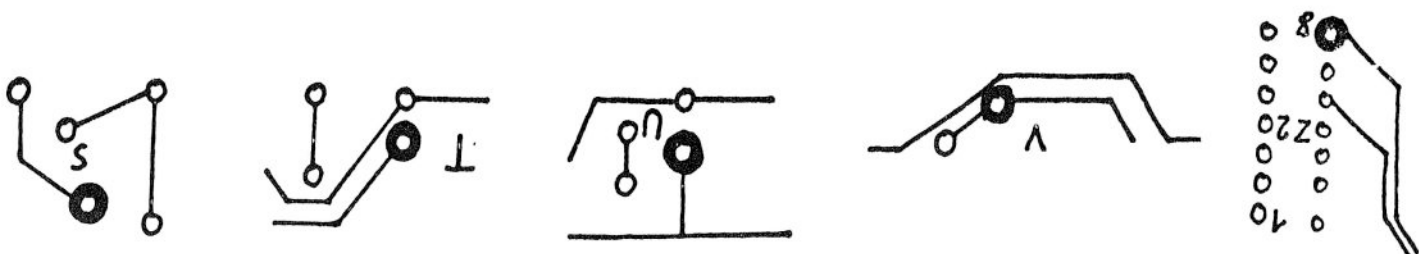
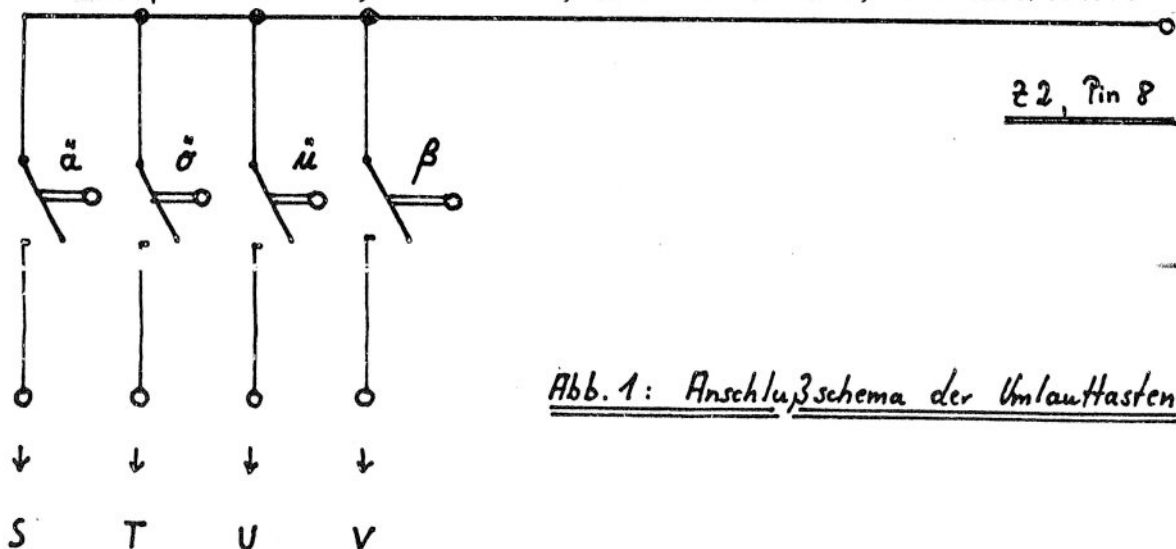
Von Hause aus ist das TRS 80 Modell I leider weder mit der Kleinschreibung noch mit den Umlauten ä, ö, ü, und ß ausgerüstet. Inzwischen weiß aber wohl jeder TRS 80 User, wie man diesem Übel abhelfen kann, entsprechende Bauteile sind ohne Probleme zu erhalten, man schaue ggfs. mal in den Kleinanzeigenteil einschlägiger Fachzeitschriften.

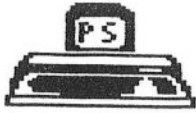
Es bleibt die unangenehme Tatsache, daß die Umlaute nach Austausch des Zeichengenerators nur über eine Tastenkombination erreichbar sind, so zum Beispiel mit Shift-Null-a für ä usw. Mit einer einfachen Bastelararbeit kann man dies ändern :

Benötigt werden 4 Tastaturtaster, die im Electronic-Fachhandel zu bekommen sind. Öffnet man nun den TRS 80 und klappt die Tastaturplatte nach vor, so befindet sich oben links der Baustein Z2, zu sehen ist in dieser Lage die Leiterbahnseite. Man suche Pin 8 von Z2 und verbinde damit alle Tasten, den freibleibenden Kontakt pro Taste verbindet man mit dem Lötpoint der Tasten für S (ergibt ä), T (ö), U (ü) und V (ß). Die entsprechenden Lötunkte sind auf der Platine gekennzeichnet. Es geht am schnellsten, wenn man bei eingeschaltetem Rechner mit einer Prüflleitung Z2 Pin 8 mit den Lötunkten kurz verbindet, hat man den richtigen Lötpunkt erwischt, erscheint am Bildschirm der zugehörige Umlaut. (Unter Newdos erscheint allerdings ÄÖÜ Rechtspfeil, denn :)

Die Abfrage der neuen Umlauttasten ist allerdings softwareabhängig, so erzeugt Newdos die o.a. Zeichen, die Kleinbuchstaben und ß erreicht man über Shift. Ein kleiner Zap kann dies ändern. Für Scripsit stimmt die Abfrage aber, hier liegt wohl auch das Hauptanwendungsgebiet der neuen Tasten. Viel Spaß beim Umbau !

Christoph Wachendorf, Almastr. 50, 4200 Oberhausen 1, Tel. 0208/854354





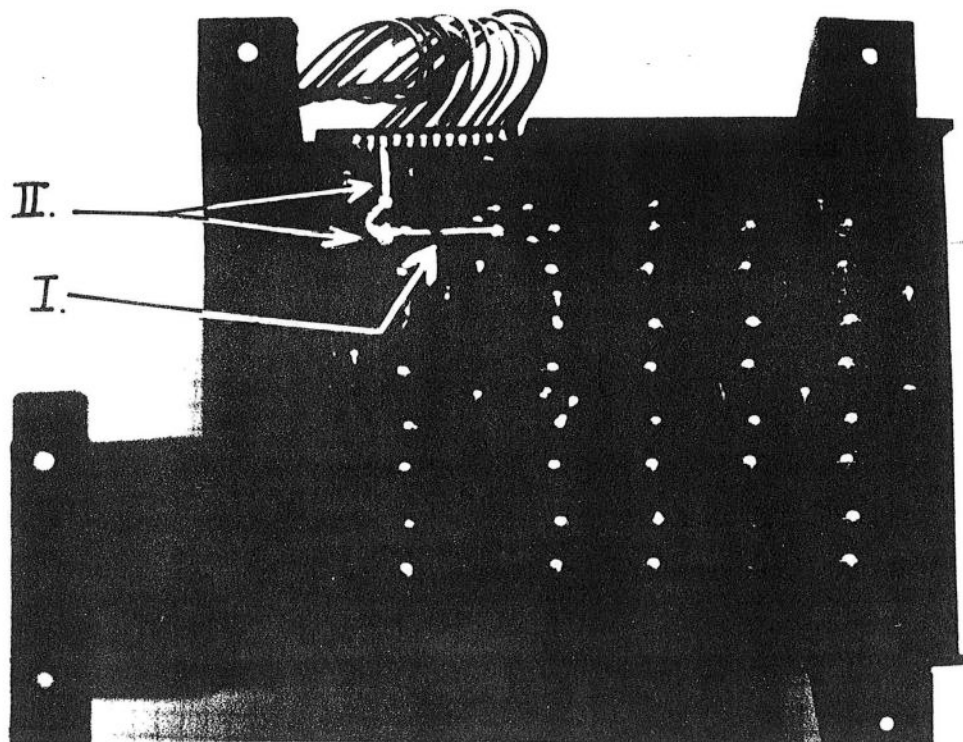
Peter Spieß, Trugenhofenerstr. 27
8859 Rennertshofen 1
Tel.: 08434/454

Umbauanleitung zur Darstellung des "ä" und "Ä" auf der F1 Video Genie II

1. Gehäuseoberteil des Computers entfernen.
2. Tastaturplatine des Zehnerblocks herausnehmen (5 Schrauben, eine Steckverbindung).
3. Platine wie auf der Zeichnung ablegen.
4. Änderung I: Leiterbahn mit spitzem Gegenstand unterbrechen.
5. Änderung II: Drahtbrücke wie gekennzeichnet einlöten.
6. Tastatur und Gehäusedeckel wieder festschrauben.

Die Grafikzeichen der Taste F1 sind jetzt durch die Umlaute "Ä" und "ä" ersetzt.

Viel Spass !



Home-Computer maßgeschneidert

Die CHIP-Computer-Aktion berät Sie individuell bei der Auswahl des für Sie passenden Home-Computers.

Komplizierte Technik und ein unübersichtlicher Markt machen es dem Einsteiger schwer, sich im Reich der Home-Computer zurechtzufinden. Wer weiß schon, bevor er beginnt, was die technischen Details, die in Anzeigen und Prospekten ausgiebig geschildert werden, wirklich bedeuten? Wie soll ein Neuling erraten, was der präsentierte Home-Computer nun wirklich kann und was er nicht kann. Denn gerade über den letzten Punkt schweigen die Kaufleute nur zu oft.

Die CHIP-Computer-Aktion will allen Einsteigern helfen, für sich persönlich den richtigen, den am besten geeigneten Home-Computer zu finden. In Zusammenarbeit mit den Horten-Computer-Centren in 28 Filialen des Kaufhauskonzerns wird der nebenstehende Fragebogen ausgewertet, um individuell jedem Einsender den auf seinen Bedarf zugeschnittenen Rechner zu empfehlen. Die Teilnahme ist kostenlos und verpflichtet zu nichts.

Bevor man sich einen Home-Computer anschafft, sollte man nachdenken. Was will ich mit meinem Rechner tun? Sollen vor allem intelligente Spiele über den Bildschirm zu Hause mitemmachen? Will ich die Buchhaltung meines Haushalts oder meines Kleinbetriebs nun endlich in Computergeschwindigkeit in den Griff bekommen? Geht es mir darum, eine Sammlung zu verwalten, ein Archiv zu führen? Will ich lediglich die Geburtstage der Verwandtschaft zur automatischen Erinnerung in das Elektronengehirn füttern? Soll der Zugbetrieb meiner Modelleisenbahn in Zukunft elektronisch gesteuert laufen? Oder weiß ich vielleicht noch gar nicht so genau, was ich mit einem Computer machen möchte?

Die Möglichkeiten der Beschäftigung mit dem Home-Computer sind zahllos. Man kann sich auch in das Computerhobby hineintreiben lassen. Besser und billiger ist es aber auf alle Fälle, schon klare Vorstellungen zu haben, ehe man beginnt. Hinterher wird der „Appetit“ mit dem „Essen“

ohnehin immer größer. Deshalb auch sollte der Home-Computer am Anfang auf keinen Fall zu knapp geplant sein. Wer schon unbedingt jetzt sparen will, sollte lieber klein anfangen mit einem Modell, das sich später ausbauen läßt – auch wenn dann die Computerleistung oft teurer kommt als in einem Stück gekauft.

Wenn Sie den nebenstehenden Fragebogen ausfüllen und an die CHIP-Redaktion unter dem Stichwort „Computer-Aktion“ einsenden, vermitteln Sie uns ein ungefähres Bild von Ihren Wünschen und wie sie sich für die Wahl des für Sie passenden Home-Computers auswirken. Die Wünsche, die Ansprüche, die Vorstellungen addieren sich auf, so daß am Ende die Größenklasse und die Leistung eines Computers herauskommt, der das meiste von dem kann, was Sie von Ihrem Home-Computer erwarten.

Teilnahmebedingungen

Die CHIP-Computer-Aktion „Welcher Home-Computer paßt zu mir?“ will Einsteigern einen Hinweis auf für sie geeignete Home-Computer geben. Zur Teilnahme ist jeder mit einem Fragebogen berechtigt.

In Zusammenarbeit mit den Horten-Computer-Centern in 28 Filialen des Kaufhauskonzerns wird aufgrund der Antworten für jeden Teilnehmer individuell mit Hilfe eines Computerprogramms ein geeigneter Home-Computer ausgewählt. Teilnehmer, die ihren Fragebogen an die CHIP-Redaktion einsenden, erhalten einen Brief mit dem entsprechenden Vorschlag.

Die ausgefüllten Teilnahmebögen sind in einem ausreichend frankierten Briefumschlag zu senden an: CHIP-Redaktion, Computer-Aktion, Bavariaring 8, 8000 München 2. Letzter Einsendetermin ist der 1. November 1983 (Poststempel). Die Teilnahme an der CHIP-Computer-Aktion ist für die Einsender kostenlos und unverbindlich.

Verlag und Redaktion übernehmen für den vom Computer ausgewählten Vorschlag keine Gewähr. Der Rechtsweg bleibt ausgeschlossen. Unvollständiges Ausfüllen des Fragebogens kann zu abweichenden Ergebnissen führen.

Nun sind die Augen oft größer als der Geldbeutel. Es mag sein, daß Ihr Wunschcomputer nicht ganz mit Ihren Preisvorstellungen übereinstimmt, daß es den Computer mit der von Ihnen gewünschten Leistungsfähigkeit in der gewünschten Preisklasse nicht gibt. Wir haben bewußt den Preis nicht zum allein ausschlaggebenden Kriterium erhoben. Erstens fallen die Preise noch immer, so daß es durchaus sein könnte, daß Ihr Wunschcomputer in absehbarer Zeit für Sie erschwinglich sein wird. Andererseits ist die Nutzung eines Home-Computers schließlich nicht nur eine Frage von Kosten und Nutzen. Computerei soll ein Hobby sein. Also ist die Frage nach dem maßgeschneiderten Home-Computer vor allem auch von der Freude bestimmt, die er bringen kann, und von der Leistung, die er erzielt.

Ihr Teilnahmebogen, den Sie einsenden, wird von einem Computer ausgewertet. Übrigens – um Ihnen zu demonstrieren, was ein Home-Computer alles vermag –, von einem Home-Computer der oberen Leistungsklasse. Die CHIP-Redaktion hat in Zusammenarbeit mit der Firma Triumph-Adler ein Programm entwickelt, das unabhängig von einzelnen Herstellerinteressen die Antworten bewertet und die Palette der heute angebotenen Home-Computer daraufhin durchforstet, welches für Sie der Richtige ist.

Natürlich kann Ihnen die Computerempfehlung nicht die Kaufentscheidung abnehmen. Sie sollten sich auf jeden Fall beim Kauf noch einmal intensiv im Geschäft beraten lassen. Nur so können Sie sicher sein, daß Sie mit Ihren Wünschen richtig verstanden werden, daß keine für Sie wichtige Frage offenbleibt. Der Computer-Rat, den wir Ihnen geben, soll eine Hilfe bei dieser Entscheidung sein.

Gerade der Anfang ist beim Computerhobby erfahrungsgemäß am schwersten. Und oft werden in dieser Phase auch die meisten Fehler gemacht, die sich hinterher nur schwer wieder reparieren lassen. Denn dies würde ja bedeuten, einen Computer für einige hundert Mark umsonst gekauft zu haben. Die Teilnahme an der CHIP-Computer-Aktion kann Ihnen helfen, daß Ihnen das nicht passiert.

Liste meiner Software:

Abkürzungen: G=Game/B=Basic/M=Maschinensprache/N=Mathematisch-
Naturwissenschaftlich/G=Geschäftsprogramm/C=Casette/
D=Diskette/E=Eigenentwicklung

Spiele:

Praesident	G/B/C/D	Burg	G/B/C/D
Umwelt	G/B/C/D	Glücksrad	G/B/C/D
Bandit	G/B/D/C	Labyrinth	G/B/C/D
Dame	G/B/C/D	Jäger	G/B/C/D
Superhirn	G/B/C/D	Snoopy	G/B/C/D
Muenze	G/B/C/D	Backgammon	G/B/C/D
Mondlandung	G/B/C/D	Kalah	G/B/C/D
ROULETTE	G/B/C/D	Startrek III	G/B/C/D
Bowlen	G/B/C/D	Erode	G/B/C/D
Bluffen	G/B/C/D	Eliza	G/B/C/D
Biorhythmus I	G/B/C/D	Blackjack	G/B/C/D
Biorhythmus II	G/B/C/D	Springer	G/B/C/D
Dr Marcus	G/B/C/D	Pferderennen	G/B/C/D
Aimfire	G/B/C/D	Türme v. Hanoi	G/B/C/D
Kopf/Zahl	G/B/C/D	Malen	G/B/C/D
U-Boot	G/B/C/D	Zufall	G/B/C/D
Nova	G/M/D	Reaktion	G/B/C/D
Galaxy	G/M/D	Bilder	G/B/C/D
Cosmic	G/M/D	Suche	G/B/C/D
Fight	G/M/D		

Mathematisch-Naturwissenschaftliche Programme:

Darlehn	B/C/D	Matrix	B/C/D/E
Abgezinste Geldm.	B/C/D	Molgeschwindigkeit	B/C/D/E
Effektivzins	B/C/D	Rotation	B/C/D/E
Kreditkonto	B/C/D	Maxwell	B/C/D/E
Rückzahlung	B/C/D/E	Bernoulli	B/C/D/E
Balkengraphik	B/C/D	Fourier	B/C/D/E
Lissajo Figur	B/C/D/E	Geometr. Reihe	B/C/D/E
Korrelation	B/C/D	Pi	B/C/D/E
Lineare Funktion	B/C/D	Plancksches Quant	B/C/D/E
Wiss. Notation	B/C/D	Diffusionspotent.	B/C/D/E
Lineare Optimierung	B/C/D	Waermefluß	B/C/D/E
Kationen Trenn.gang	B/C/D	Mittlere fr. Weglänge	B/C/D/E
Lineare Regression	B/C/D/E	Kinetik	B/C/D/E

COLOUR GENIE

Klaus Schmidt, 2850 Bremerhaven, Bleßmannstr.1b, Tel.: 0471/ 24998

Versendung der Programme nur bei Zusendung von Rückporto, entsprechender Versandtasche (die meisten kann man doppelt verwenden!) und einer Schutzgebühr von 3,-DM pro programm. Pro Copie (bei Hardbuts z.B.) bitte noch-, 10DM beilegen. Die Seitenzahl steht in der Liste. ~~Ein~~weiges Listing der Programme mit anfordern wenn kein Drucker vorhanden!

Bei Tausch alles kostenlos! "Suchliste" wird ergänzt !

HABE

Z 80 Betriebssystem

5 Seiten Handout

12KByte

Assembler, Editor, Disassembler u. Maschinenspr.-monitor.

ca.9KB Basic u. 3KB Masch. ca.2KB frei (16kVersion) speziell zur Entwicklung von USR-Routinen!

Karteikasten

7KByte

Art und Umfang der Karten frei definierbar!

z.B. 60Karten a 4Angaben und 16 Bytes Ja/Nein-Infos
Kommandos: Def/Cas/Eing/Sort/Find/Loesch/Anz/Druck
Find-Kommando kennt Schnitt-, Differenz und Teilmenge sucht nach Angaben und Infocodes.

Haushaltsfinanzen

5KByte Basic

Ermittelt aus den in Datazeilen stehenden Angaben die monatlichen Fixkosten eines Haushalts (Miete/Strom etc.)
Unregelmäßige Zahlungen werden berücksichtigt.
Druck Jahresbilanzen und Durchschnittsmonat.

Funktinenplotter

6KByte Basic

HGR!-Graphik beliebiger math. Funktionen mit Drucker ausgabe, Maßstabswahl und Bereichswahl.

Zeicheneditor

s. Handbuch!

Colour-Compiler

setzt Basic in Maschinensprache 32K erforderlich 20Seiten

Copyclone

Copiert beliebiges Maschinenprogramm bis 14KByte!

liegt auf ~~54k~~ F400H! (CLOAD/SAVE/VERIFY)

TRS80BASICLOADER

Lädt TRS80/GENIE-Basicprogramme in das COLOUR System fremde Befehle müssen umgeschrieben werden.

PRINT# - 1, ...

mit nur 20 Nullbytes (ohne Gebühr!)

PRINTHGR,

in Arbeit!

SUCHE

Jegliche Software aus allen Bereichen

Speziell: COLOUR SCHACH, Graphik Programme

Lohnsteuer- Einkommensteuerberechnung

Allgemein: Gute Maschinenspr.programme!

Schon eingetauscht: COLASM/COLMON/SHAPE

TRS80/GENIE I/II

softwareliste, holger may / marienstr.9 / 5768 sundern 2

+++++++ alle programme in basic und auf cassette ++++++++
sp=speicherplatzbed. / sg=schutzgebuehr
zur sg kommt je sendung noch dm 3.00 fuer porto und verpackung

name	beschreibung	sp	sg
mau-mau	das bekannte kartensp.geg.den computer	5229	3.00
mondphasen uhr	konstellation v.erde,mond,sonne	4901	3.00
zahlen aussprechen	zahlen in worte umwandeln	3977	3.00
waagerechte wurf	mit grafischer darstellung	3055	3.00
das periodensystem	alle daten der elemente abfragb.	5984	3.00
autorennen 2	gutes spiel mit 2 strecken	8800	4.40
briefe schreiben	ein briefeditor (drucker erf.)	6598	3.30
morsen	uebers. text in zeichen und toene	2954	3.00
sechser spiel	iteres. wuerfelspiel comp.spielt mit	3417	3.00
stoneair 1	fliegen sie durch die gesteinsbrocken	2771	3.00

Programm - Listing :
=====

- | | |
|-------------------|---|
| 1. KORRBRIEF/BAS | : BRIEFPROGRAMM MIT KORREKTUR |
| 2. ZINSBERE/BAS | : PROGRAMM ZUR ZINSBERECHNUNG |
| 3. ADRESSEN/BAS | : ADRESSEN-SPEICHER-PROG. FÜR DISK |
| 4. EFEKTIVZ/BAS | : ZUR BERECHNUNG DES EFFETIVZINS |
| 5. HEIRATEN/SPI | : SPIEL FÜR DIE GANZE FAMILIE |
| 6. FORMELN/DAT | : MATHEM. FORMELN (ABGELEITETE FUNKTIONEN) |
| 7. DARLEHNR/BAS | : DARLEHNSRECHN. MIT ZINS UND TILGUNG |
| 8. GEBRIEF/BAS | : GESCHAEFTS - UND PRIVATBRIEF MIT ANREDEN |
| 9. WARENKAL/BAS | : WARENKALKULATIONS - PROGRAMM |
| 10. PROVISION/BAS | : PROVISIONS - BERECHNUNGS - PROGRAMM |
| 11. AUFKLEBE/BAS | : ADRESS - AUFKLEBER - PROG. FÜR FORM 510 |
| 12. STROMREC/BAS | : PROG. ZUM NACHRECHNEN DER STROMRECHNUNG |
| 13. QUERPVB/BAS | : TRÄGHEITSBERECHNUNG (STATIK) |
| 14. QUERTVB/BAS | : QUERSCHNITTSBERECHNUNG AUS TEILFLÄCHEN |
| 15. FUND1/BAS | : QUERSCHNITTSBERECHNUNG M. VERSAGENDER ZUGZONE |
| 16. GAUSS/BAS | : BERECHNUNG V. VARIABLEN U. UNBEKANNTEN |
| 17. WKURSBER/BAS | : WECHSELKURSBERECHNUNG EWG U. ANDERE STAATEN |
| 18. OHMSCHEG/BAS | : BERECHNUNG VON OHM, AMPERE, WIDERSTAND, VOLT |
| 19. VARIABLE/TXT | : PROGRAMMIERHILFE - VARIABLENLISTE |

Softwareliste von Michael Karnatz

Schweriner Ring 23 2940 Wilhelmshaven

Verwendete Hardware: -Colour Genie
-Cassettenrecorder
-Drucker STAR DP 510

1 Großbuchstaben

Die Überschrift wurde mit diesem Programm erstellt.
Das Programm erstellt zeilenweise Grossbuchstaben. Nach Eingabe des Wortes wird die Wortlänge berechnet und angezeigt sowie die Druckkopfposition berechnet die fuer mittigen Druck erforderlich ist.
Umlaute in Gross- und Kleinschrift sind im Zeichensatz enthalten.

2 Übungsprogramm

Dieses Programm habe ich als Rechen- und Gedächtnistrainer fuer meine Kinder geschrieben.
Die Groesse der Zahlen sowie das Vorzeichen werden zum Beginn abgefragt. Bei falschen Loesungen wird eine Hilfe gegeben (Balkendarstellung bei Addition und Subtraktion / das 1*n bei Multiplikation und Division) und das Ergebnis erneut abgefragt.
Bei 10 richtigen Loesungen darf das Kind (ich spiele auch damit) drei Freispiele machen, bei 9 nur 1 Freispiel.
Ein Unterprogramm schult das Gedächtnis. Eine Zahl wird kurz gezeigt und muss wiedergegeben werden. Die Länge der Zahl waechst staendig. Die maximale Länge ist zu Beginn einstellbar.
Auch hier gibt es zur Beleohnung Freispiele.

3 Verbrauchsstatistik

In dieses Programm habe ich meine Gas-, Strom- und Wasserverbraeuche der letzten Jahre eingegeben. Das Programm kann fuer die Verbrauchsarten getrennt Tabellen und Balkendiagramme erstellen.
Durch Aendern der Massstabsfaktoren laesst sich das Programm auch fuer andere Zwecke verwenden.

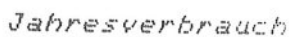
4 Lottozahlengenerator

Der Name sagt (fast) alles.
Das Programm erzeugt die Zahlen auf dem Bildschirm. Vom Bildschirm laesst sich eine Hardcopie ziehen

Fuer die Programme 1-3 sind 32K erforderlich

Trommeschläger Programme

Assembler
Monitor
Shaper



	0	50	100	150	200	250	300	350	
76	[Bar chart showing consumption for day 76]								116
77	[Bar chart showing consumption for day 77]								114
78	[Bar chart showing consumption for day 78]								115
79	[Bar chart showing consumption for day 79]								137
80	[Bar chart showing consumption for day 80]								161
81	[Bar chart showing consumption for day 81]								167
82	[Bar chart showing consumption for day 82]								163
83	[Bar chart showing consumption for day 83]								

155

2 7 1 6 8 01

Bildschirmbild nach falscher Loesung

Christoph Wachendorf Almastr. 50 4200 Oberhausen 1
Liste 5/83

System : TRS 80 Mod. I, Shugart-Doppelfloppy SS/3D, Typenraddrucker

101	Finpro	Finanzprognoseprogramm - Liquiditätsstatus aus monatl. prognostizierten Zahlungsreihen	
102	Rendite	Berechnung des Kurswertes festverzinslicher Wertpapiere, sog. optimales Portefeuille	
103	AfA	Berechnung der Investitionsvorteilhaftigkeit bei alternativen Sätzen der degr. AfA	
104	AfATab	Erstellung einer Abschreibungstabelle	
105	Optibest	Berechnung der optimalen Bestellmenge	
106	Endwert	Berechnung der End- und Barwerte verschiedener Zahlungsreihen bei verschiedenen Zinszahlungsweisen	
106	Tilgung	Tilgungsplan eines Kredites	
107	Kurs	Rendite festverzinslicher Wertpapiere	
108	Proko	Produktionsprogrammplanung im einstufigen Betrieb mit Kapazitätsengpaß	
109	Preis	Preissteigerungsrücklage nach § 74 EstDV	
110	Rechnung	Fakturierungsprogramm	
111	EST	Berechnung der Einkommen- bzw. Lohnsteuer-erstattung bzw. -nachzahlung	
201	Elim	Lösung kleiner linearer Gleichungssysteme durch vollständige Elimination	
202	Simplex	Lineare Optimierung nach der Simplex-Methode	
203	Phasen	Lineare Optimierung mittels 2-Phasen-Methode	
204	Trans	Minimierung von Transportwegen	
205	Plotter	graphische Darstellung von Funktionen	
206	Sort	Zahlensortierer	
301	TRSDOS	Diskettenbetriebssystem von Tandy	(A= 70)
302	Newdos 1	Newdos Vers. 1.0 von Apparat	
303	Newdos 2	Newdos Vers. 2.0 von Apparat	(A=270)
304	Newdos 2+	Newdos Vers. 2.052 mit erweiterter Graphik	
305	CP/M 1.4	Betriebssystem von Digital Research	
306	CP/M 1.5	Betriebssystem von Digital Research	(A= 38)
307	VTOS 3.0.5		
308	Cobol	Cobol Compiler (mit CP/M 1.4)	(A= 76)
309	Fortran80	Fortran 80	(A=130)
310	Pascal80	Pascal 80	(A= 42)
311	Pascal	UCSD - Pascal (4 Disketten)	
312	ZBasic	Basic-Compiler von Simutek Vers. 2.2	
313	Lcdriv	Kleinschreibungstreiber für TRSDOS	
314	TRSMon	Monitor für den TRS 80 von E. Pese	
315	APL80	APL 80	(A= 25)
316	G-Dos	Diskettenbetriebssystem für Video Genie	

Anm.: "A" gibt die Seitenzahl des zugehörigen Manuals an.

401	VisiCalc	Bildschirmorientiertes Kalkulationsprogramm	(A=175)
402	Fibu 80	Finanzbuchhaltung, 1000 Buchungen / Monat	(A= 30)
403	Fibu II	Fibu von Playtron, 1500 Buchungen / Monat	(A= 38)
404	Scriptit	Textverarbeitungssystem m. Umlauten	(A= 70)
405	Superscriptit	stark erweitertes Scriptit v. Tandy	(A=152)
406	TRSText	Textverarbeitung für TRS und VGS	(A= 40)
407	Microfiles	Dateiverwaltungssystem von Tandy	(A= 40)
408	Profile	dito	(A= 24)
409	Versafile	dito	(A= 25)
410	Unidat 80	z.Zt. effektivste Dateiverwaltung, Vers.5.24	(A= 45)
411	Superut22	Superutility 22 Plus	(A= 25)
412	Trakcess	Diskettenutility	
413	Editor 80	Macroeditor, Assembler und Linker	(A=220)
414	Packer 48	"packt" Basic-Programme	
415	Superdir	Superdirectory für alle Disk	(A= 5)
416	Edtasm	Editor u. Assembler von Tandy m. Erweiter.	(A= 4)
417	Step 80	Debugger, Disassembler und Stepper	(A= 5)
418	Clone	dupliziert Tapes und Disk	(A= 2)
419	Lager	Inventory Control System (Lagerverwaltung)	(A= 28)
420	Tasmon	Tasmon-Monitorprogramm Vers. 2.12	
421	Discat	Disketenverwaltung für SD- und DD-Disk	
422	Music	viel Musik, über Recorderausgang des TRS 80	
423	Superbak	Diskettenbackup - Utility	
424	Wahl	Diskettenverwaltungsprogramm	
425	Dirmap	Directory Mapper	
426	RSM	RSM-Monitor	
427	Powerdraw	Graphikprogramm mit vielen Möglichkeiten	(A= 20)
428	Protext 80	Textverarbeitung	
429	XASMZ865	6502-Maschinenprogramme für den TRS 80	(A= 5)
430	TDCS	Tandy Drive Controller System	
431	DDT	Disk Drive Timer	
432	Dis	Cursor orientierte Directory	
433	GAP	Geschäftsadressenprogramm v. Tandy	
434	Fakt	Fakturierungsprogramm	
435	Kleber	erstellt Adressenaufkleber	
436	Lohn	Lohnberechnung für max. 30 Arbeitnehmer	
437	Bascom	Basic - Compiler	
438	GrBas	Graphik-Basic, für grafikintensive Programme	
439	Floppydoc	umfangreiches Floppy-Testprogramm	
440	Graph3D	erstellt dreidimensionale Graphik	
441	Taberst	zur Erstellung von Tabellen in Basic	
442	Electric Pen	Textverarbeitungsprogramm Electric Pencil	(A= 58)
443	Perspect	Perspektivisches Zeichnen in Basic	
444	Formatter	Bildschirmwurfprogramm	
445	Diskexec	ähnlich dem LMOFFSET des Newdos	
446	Vardoc	erzeugt Referenzliste	
447	Random	erzeugt Zufallszahlen nach ver. Kriterien	
448	Ramtest	RAM - Tester	
449			
450			
451			
452			

Bei Programmtausch erbitte ich die Erstattung (ggfs. anteiliger) Kopierkosten (DM 0.10 / Blatt).

Suche auch Software für TRS 80 Modell II !

Teilnahmebogen

Bitte füllen Sie den folgenden Fragebogen vollständig und gut lesbar aus, damit unser Computer-Programm für Sie den passenden Home-Computer auswählen kann.

Senden Sie den ausgefüllten Fragebogen bitte bis spätestens 1. November 1983 (Poststempel) an: CHIP-Redaktion, Computer-Aktion, Bavaria-ring 8, 8000 München 2

1. Was wollen Sie mit ihrem Home-Computer machen?

- | | |
|---|--------------------------|
| Programmieren lernen | <input type="checkbox"/> |
| Computer-Spiele | <input type="checkbox"/> |
| Vokabeln (oder anderes) lernen | <input type="checkbox"/> |
| den Haushalt verwalten | <input type="checkbox"/> |
| Briefe schreiben und speichern | <input type="checkbox"/> |
| Sammlungen archivieren
(Schallplatten, Briefmarken etc.) | <input type="checkbox"/> |
| Elektrische Geräte steuern | <input type="checkbox"/> |
| Adressen/Telefonnummern speichern | <input type="checkbox"/> |
| Berufliche Anwendungen
(Kalkulation, Buchhaltung usw.) | <input type="checkbox"/> |
| Weiß ich noch nicht | <input type="checkbox"/> |

2. Werden dabei viele Daten anfallen?

- | | |
|--|--------------------------|
| mehr als 1 Schreibmaschinen-Seite
(vollgeschrieben) | <input type="checkbox"/> |
| mehr als 5 Seiten | <input type="checkbox"/> |
| mehr als 20 Seiten | <input type="checkbox"/> |
| mehr als 100 Seiten | <input type="checkbox"/> |

3. Wollen Sie auch Programme aus Büchern und Zeitschriften in Ihren Computer eintippen?

ja/nein

4. Wollen Sie andere Geräte an Ihren Computer anschließen?

ja/nein

welche

- | | |
|--------------------------------------|--------------------------|
| Fernsehgerät | <input type="checkbox"/> |
| Computer-Monitor | <input type="checkbox"/> |
| Elektronische Schreibmaschine | <input type="checkbox"/> |
| Drucker | <input type="checkbox"/> |
| Steuerknüppel für Spiele, Drehregler | <input type="checkbox"/> |
| Lichtgriffel | <input type="checkbox"/> |
| Elektronische Schaltungen | <input type="checkbox"/> |
| Meßgeräte | <input type="checkbox"/> |
| andere, welche? | <input type="checkbox"/> |

5. Legen Sie Wert auf eine schnell und leicht zu bedienende Tastatur?

ja/nein

Welche wäre Ihnen am liebsten?

- | | |
|-------------------------------|--------------------------|
| eine Schreibmaschinentastatur | <input type="checkbox"/> |
| eine Hartgummi-Tastatur | <input type="checkbox"/> |
| eine Folientastatur | <input type="checkbox"/> |

6. Wollen Sie vor allem auch Ihre eigenen Programme entwickeln? ja/nein

7. Wollen Sie häufiger Programme fertig kaufen? ja/nein

welche Art? Spiele ☐
Lernprogramme ☐
Verwaltungsprogramme
(für Haus und Beruf) ☐
Programmierhilfen ☐

8. Was soll Ihr Home-Computer kosten?

unter 300 Mark ☐
300 bis 500 Mark ☐
500 bis 1000 Mark ☐
1000 bis 1500 Mark ☐
über 1500 Mark ☐

9. Besitzen Sie schon einen Home-Computer? ja/nein

10. Können Sie schon programmieren? ja/nein
Falls ja, halten Sie sich für einen

Anfänger ☐
Fortgeschrittenen ☐
„Profi“ ☐

In welcher Sprache können Sie programmieren?

in BASIC ☐
in COBOL ☐
in Pascal ☐
in Assembler ☐

11. Wollen Sie sich persönlich mit dem Home-Computer beschäftigen, ist er für die ganze Familie gedacht oder vor allem für die Kinder?

für mich selbst ☐
Familie ☐
Kinder ☐

12. Haben Sie Interesse an einführender Literatur zu Ihrem neuen Home-Computer? ja/nein

13. Wollen Sie eventuell einen Kurs für Computer-Einsteiger oder Programmierkurs selbst besuchen? ja/nein

14. Wie wollen Sie sich über die weitere Computer-Entwicklung auf dem laufenden halten?

durch das Angebot im Fachgeschäft ☐
durch Zeitschriften ☐
durch Computer-Bücher ☐
durch persönliche Kontakte ☐

15. Sind Sie mehr an neuen Geräten oder an Berichten über neue Programme interessiert?

Geräte ☐
Programme ☐

16. Würden Sie sich einem Computer-Club anschließen? ja/nein

Was erwarten Sie sich davon?

Kurse ☐
Hilfen zum Einstieg ☐
Erfahrungsaustausch ☐
Geselligkeit ☐
Vertiefung der Computerkenntnisse ☐

Und nun noch Ihre Adresse (bitte deutlich schreiben, damit unsere Antwort Sie erreicht):

Name: _____

Alter: _____ männlich/weiblich: _____

Beruf: _____

Straße: _____

PLZ, Ort: _____