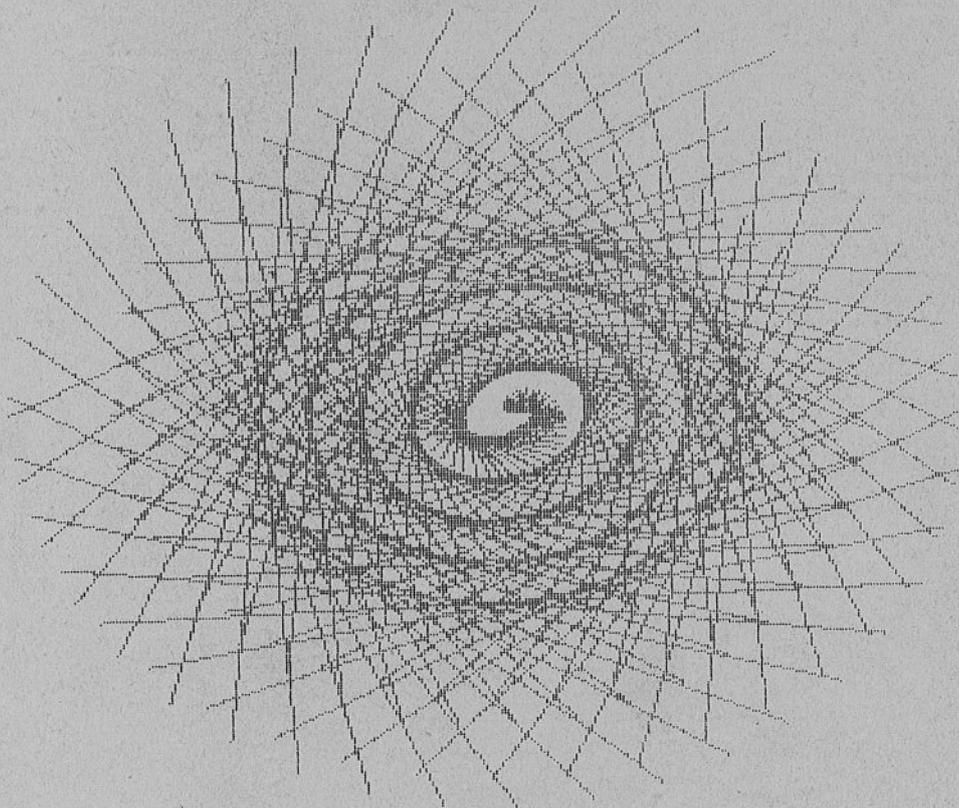


# CLUBZEITUNG



20. AUSGABE

**AUS DEM INHALT:**

-----

Unterbringung von Maschinenroutinen in BASIC-Programmen	7
Hochauflösende Softgraphic von Jürgen Degenhardt	11
PRINT -> LPRINT Umwandlung von Bernd Niedermeier	15
3.5 Mhz Modifikation für Genie II	17 *
Schnelle Datenspeicherung auf Cassette aus MICRO-EXTRA	20
VIDEO SNOW SHOVEL	22 *
Mitglieder-Adressliste	24

Mit "\*" gekennzeichnete Beiträge entstammen der AMMS-Zeitung

**Termine für Clubtreffen**

-----

Mittwoch 31.08.83 19.00 Uhr  
 Gaststätte Mathäserstuben  
 Gollnerstr. 15 8000 München 2

**Achtung!! Im September findet kein Clubtreffen statt.**

Mittwoch 26.10.83 19.00 Uhr  
 Gaststätte Mathäserstuben  
 Gollnerstr.15 8000 München 2

**Achtung! Neue Adresse: Postfach 1140 8011 Kirchseeon**

Bitte verwenden Sie künftig nur noch diese neue Anschrift.

**Clubkonto**

-----

Postscheckamt München  
 BLZ: 700 100 80  
 Kontonr.: 3452 35-800  
 Inhaber: G.Thalmeier

**DISKETTEN**

Z.Z. kann ich anbieten

MULTILIFE - Disketten  
mit Verstärkungsringen

ca. DM 5.60 Stk.  
+ Versandkosten

Verstärkungsringe einzeln  
stabilere Ausführung als oben

DM -.50 "

Ein Werkzeug zum nachträglichen Anbringen der Ringe ist im Club  
ausleihbar.

Bestellungen, die telefonisch bei mir eingehen, kann ich ggf.  
gleich bestätigen.

Gregor

**GELOCHTE DISKETTEN**

Im Club ist ein Werkzeug zum nachträglichen Lochen von  
Disketten (zwecks beidseitiger Benutzung), ausleihbar.

**ZU VERKAUFEN**

TRS-80 Mod.1 Level 2 mit 10er-Tastatur und Kleinschreibung  
Expansion-Interface mit 48K und 1 Disklaufwerk.  
Die Anlage ist neuwertig. VB DM 3500.--

Alfred Brühbach, Haydnstr. 5 3501 Fuldabrück Tel. 0561/41929

**Suche**

Bei wem kann ich die Bücher

"BASIC faster & better & other Mysteries" und  
"Machine Language Disk I/O & other Mysteries"

ausleihen?

Alfred Brühbach, Haydnstr. 5 3501 Fuldabrück Tel. 0561/41929

Bernd Niedermeier            Hirschbergweg 9            8011 Heimstetten  
Tel.: (089) 903 57 31

Hallo Computerfreaks,  
hiermit moechte auch ich mich mal den Clubmitgliedern vorstellen. Ich besitze einen TRS-80 mit 48K, Expansion Interface und zwei 5 1/4 Zoll BASF Laufwerken, den ich ein wenig aufgeböhrt habe. So ist bis zum heutigen Zeitpunkt eine Speedmodifikation (2.6 Mhz), Inverse Video und die 'Snow Shovel' eingebaut, die die von der CPU erzeugten laestigen schwarzen Striche vom Bildschirm verbannt. Der Unterschied ist einmalig.

Ich habe im Mai dieses Jahres das Abi an einem Wirtschaftsgymnasium im Muenchen erfolgreich bestanden und werde auf der FH Muenchen Elektrotechnik mit Fachrichtung Datentechnik studieren.

An meiner (ehemaligen) Schule erhielt ich die besondere Gelegenheit, meine Facharbeit in Informatik zu machen, obwohl dieses Fach kein Leistungskurs ist. Fuer die, die es interessiert: Ich schrieb eine Art Compiler in BASIC, der Programme einer PASCAL aehnlichen Programmiersprache (sie ist hauptsaechlich eingedeutschtes PASCAL und wird zu Unterrichtszwecken im Informatikunterricht verwendet) in lauffaehige BASIC-Programme uebersetzt.

Auf die Comuterei kam ich an meiner Schule. Dort uebte ich mich zuerst auf einer alten WANG 2000, deren BASIC ich immer noch fuer eines der besten halte. Basld wurden neue Computer fuer den Informatikunterricht angeschafft. Hier sind wir eine der ganz wenigen Schulen, wenn nicht die einzige, die keine COMMODORES hat (was fuer ein Glueck). So besitzt denn zwischenzeitlich meine Schule folgendes Equipment: 5 TRS-80 Level II, davon einer mit Expansioninterface und zwei Shugartlaufwerken, einen EPSON MX80 F/T mit Graftrax 80, 2 EPSON RX80, einen AXIOM IMP Mini Printer und einen Watanabe Plotter. Demnaechst werden noch drei TRS 80 folgen.

Nach ausgiebigen Erfahrungen auf dem Schul-TRS-80 entschloss ich mich, mir selbst eine Anlage anzuschaffen, die ich nun seit Februar 82 mein eigen nenne.

Ach ja... kann sich einer von euch mal den MX80 unserer Schule anschauen? Zwei Nadeln sind kaputt und die Spannung am Ausgang scheint auch nicht ganz korrekt zu sein. Wer helfen kann und will soll sich doch mal bei mir melden.

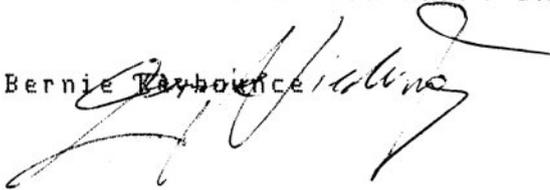
Doch nun zu meinen Hauptinteressen auf dem Computergebiet. Vor allem interessiere ich mich fuer die Steuerung eines Synthesizers mit dem TRS plus entsprechender Software. Weiterhin faende ich die Anwendung als Lichteffektsteuergeraet (auch NF-gesteuert) interessant. Wer fuer eine oder beide der genannten Hauptinteressen die selbe Antenne hat, soll sich doch mal mit mir in Verbindung setzen. Vielleicht kommt was g'scheits dabei raus.

Dieser Text wurde uebrigens auf dem AXIOM Printer ausgedruckt. Ich habe ihn freundlicherweise von meiner Schule zu leihen bekommen, da er nicht an dem 40-poligen Ausgang des Keyboards angeschlossen werden kann. Ob ein entsprechendes Interface (wie fuer EPSON) existiert...???

Vielleicht werde ich demnaechst einmal einen Testbericht ueber diesen Drucker veroeffentlichen. Soviel sei gesagt: Mit ihm ist auch Einzelpunktsteuerung moeglich.

Hiermit wuensche ich HAPPY DATAS...

Bernie Raybence



-5-

Wolfgang Wirtz  
Postfach 1372  
8013 HAAR  
Tel. 089/4304324

Hallo Computerfreaks,

wem ist es nicht schon passiert:

Da will ein Bekannter ein Programm haben, natürlich mit Handbuch. Mitgeben zum Kopieren will man es nicht, man hat ja mal in der Vergangenheit schlechte Erfahrungen gemacht. Also muß es am nächsten Tag irgendwo kopiert werden.

Mit aus diesem Grund habe ich mir einen neuen DIN A 4 - Normalpapier Kopierer angeschafft, für den ich nun für alle Clubmitglieder folgenden Service anbiete.

- \* Kopieren von Handbüchern, Anleitungen und Einzelblättern
- \* Auf Wunsch Binden mit Kunststoff - Spiralbindung oder im Heißbindeverfahren in Heftform

Eventuell kann ich auch aus meinen eigenen Beständen von ca. 100 Handbüchern und Anleitungen aushelfen, sollte Ihnen das eine oder andere fehlen. Anruf genügt hier ( ab 18 Uhr ).

Beiliegend erhalten Sie eine Preisliste sowie einige Muster.

Mit freundlichem Gruß



Anlagen  
-----



BER/NIE Software            Bernd Niedermeier        8011 Heimstetten  
Hirschbergweg 9  
Tel.: (089) 903 57 31

Unterbringung von Maschinenroutinen in BASIC-Programmen  
=====

In diesem Artikel sollen einige Moeglichkeiten zur Unterbringung von Maschinenprogrammen in BASIC-Programmen erlaeutert werden. Die meisten sind auch in BASIC FASTER & BETTER nachzulesen. Als Beispielsprogramm wird eine Unteroutine herangezogen, die den Bildschirm weiss macht.

Die gebraeuchlichste Methode besteht darin, das Maschinenprogramm aus Datazeilen zu lesen und die Werte nacheinander in die Speicherplaetze zu poken.

z.B.:

```
010 MA=&HFF00 'Maschinenroutinenanfang
020 FOR I=0 TO 13:READ A:POKE MA+I-65536,A:NEXT
030 STOP
100 DATA 33,0,60,17,1,60,54,191,1,255,3,237,176,201
```

Diese Methode hat den entscheidenden Nachteil, dass man jedesmal die Dezimalwerte ausrechnen muss, was bei laengeren Routinen sehr viel Arbeit ist. Ausserdem muss man den Speicherbereich, in dem die Routine liegt, im allgemeinen schuetzen, damit BASIC 'die Finger davon laesst'. Schreibt man die Routine so, dass sie verschiebbar ist, also keine absoluten Spruenge enthaelt, so kann man durch Aendern der Variablen MA das Programm in (fast) jeden Bereich des Speichers legen.

Eine weitere haeufig praktizierte Methode ist, das Maschinenprogramm in Strings zu legen. Die Routinen duerfen allerdings nicht laenger als 255 Bytes sein und duerfen keine absoluten Spruenge enthalten. Hier gibt es verschiedene Moeglichkeiten:

Die einfachste ist, die Routine durch Aneinanderhaengen von Charactercodes zu erzeugen. Bei unserem Beispiel wuerde das folgendermassen aussehen:

```
010 CLEAR 300
020 UP$=CHR$(33)+CHR$(0)+CHR$(60).....
```

Man koennte natuerlich die Werte auch aus DATA Statements lesen und mittels UP\$=UP\$+CHR\$(I) die Routine erzeugen. Man sieht schon, auch hier kommt man nicht umhin, die Werte ins dezimale System umzurechnen. Doch es ergibt sich ein erheblicher Vorteil gegenueber der vorher beschriebenen Methode: Da das Programm nun in einem String steht, muss man keinen Speicherplatz mehr reservieren. Nur muss man aufpassen, denn durch Stringorganisationen wird die Adresse des Strings staendig veraendert und man landet sonst in den unergruendlichen Tiefen des TRS-80. Will man die Routine aufrufen, so muss man jedesmal neu die Adresse des Strings

vor dem Aufruf ermitteln:

```

...
1000 DEFUSR=PEEK(VARPTR(UP$)+1)+256*PEEK(VARPTR(UP$)+2)
1010 ? USR(0)
...

```

Aber es gibt noch eine bessere Moeglichkeit, Maschinenroutinen in Strings zu packen mit dem Vorteil, dass man nicht mit der Umrechnung ins dezimale System belastet wird und dass man sich nicht dauernd darum kuemern muss, wo denn der String gerade im Speicher 'rumschwirrt'. Dabei schreibt man zunaechst das Maschinenprogramm wie immer, laesst es sich assemblieren und notiert die Hexcodes (wenn kein Drucker vorhanden ist). Man weiss nun, wieviel Bytes das UP hat (in unserem Fall 14) und schreibt in das BASIC-Programm zunaechst einmal einen String mit ebensovielen beliebigen Zeichen. Etwa so:

```
010 UP$='.....'
```

Nun geht man in SUPERZAP und sucht in dem BASIC-Programm die Stelle, wo dieser 'Dummy-String' steht. Man schreibt jetzt mittels des MOD-Kommandos anstelle der Punkte nacheinander die Hexcodes des Programmes hinein und hat so das UP von Anfang an in einem String stehen. Daraus ergeben sich folgende Vorteile:

1. Man muss nicht muhsam in Dezimalzahlen umrechnen
2. Solange mit der Stringvariablen (hier UP\$) keine weiteren Operationen vorgenommen werden, muss die Adresse des Strings nur einmal am Anfang mittels `PEEK(VARPTR(UP$)+1)+256*PEEK(VARPTR(UP$)+2)` ermittelt werden, da beim ersten mal die Pointer auf die Stelle, wo der String im Programm steht, zeigen. Vorsicht ist dagegen geboten, wenn man weitere Programmteile von Disk nachlaedt z.B. verschiedene Unterprogramme. Aber wenn man den UP-String an den Anfang des Programms stellt, so duerfte nichts schiefgehen.
3. Man spart Variablen. Da der Pointer auf die Adresse im Programm zeigt (s.o.), kann man die Variable oefters benutzen.

Beispiel:

```

10 UP$='oiuoiuiiiuuu' die Buchstaben zwischen den An-
fuehrungszeichen sollen das Maschinenprogramm sein
20 DEFUSR0=PEEK(VARPTR(UP$)+1)+256*PEEK(VARPTR(UP$)+2)
30 UP$='oiuouioiuoioiui'
40 DEFUSR1=PEEK(..... 'siehe Zeile 20
50 UP$='khkjhhkjhkhkjhhkjhkhjhkhjhkhkhjhkhjhkh'
60 DEFUSR2=PEEK(..... 'siehe Zeile 20
...
...

```

Dies ist moeglich, da der UP-Anfang ja unmittelbar nach der Zuweisung schon definiert wurde, das BASIC-Programm aber nicht verschoben wird. Das bedeutet, dass Strings, wenn sie definiert werden wie z.B.

```
10 A$='abcdefgh'
```

zweimal im Speicher stehen; im Programm selbst einmal und in der Variablentabelle hinter dem BASIC-Programm. Dies duerfte wohl die beste Moeglichkeit sein, Maschinenprogramme in BASIC-Programmen unterzubringen.

programme in BASIC-Programmen unterzubringen.

Eine weitere weniger gebrauchliche aber doch sehr interessante Moeglichkeit aus BASIC FASTER & BETTER moechte ich abschliessend noch erlaeuern: Maschinenprogramme, die in Integerarrays untergebracht werden. Der groesste Nachteil gleich vorweg: Sie brauchen mehr Speicherplatz als die vorher beschriebenen Methoden.

Um solche Programme zu schreiben geht man folgendermassen vor: Man assembliert das Maschinenprogramm und notiert die Hexcodes. An dieser Stelle soll unser Programm einmal im Quellcode abgedruckt werden, damit der Vorgang besser verstaendlich ist:

Hexcode	Quellcode
21003C	LD HL,15360
36BF	LD (HL),191
11013C	LD DE,15361
01FF03	LD BC,1023
EDB0	LDIR
C9	RET

Wir haben nun also eine Reihe von Hexzahlen, die in Zweierkombinationen geschrieben so aussieht:

2100 3C36 BF11 013C 01FF 03ED B0C9

Um diese Kombinationen als sieben Integervariablen hintereinander zu bekommen, muessen wir die beiden Bytes einer Kombination vertauschen. Der Grund ist der, dass die Integerzahlen im Format LSB MSB gespeichert werden. Schreiben wir also A=&H2100, so haben wir im Speicher stehen: 0021. Aus diesem Grunde lautet die Kette dann:

0021 363C 11BF 3C01 FF01 ED03 C9B0

Im Speicher steht sie dann aber richtig.

Fuer unser Beispiel wuerde das Initialisierungsprogramm dann folgendermassen aussehen:

```

010 DEFINT A-Z:J=0
020   UP(0)=&H0021:UP(2)=&H363C:UP(3)=&H11BF:UP(4)=&H3C01:
      UP(5)=&HFF01:UP(6)=&HED03:UP(7)=&HC9B0
030 DEFUSR=VARPTR(UP(0))
040 J=USR(0)
050 GOTO 50

```

Immer zwei Bytes werden in eine Integervariable gepackt. Auch hier ist zu beachten, dass man jedesmal, bevor man die Routine aufruft, DEFUSR=... ausfuehren muss.

Eine schoene Variante ergibt sich, wenn man das Programm mit NOP's 'begradigt'. Das heisst: Man schaut, dass die Bildschirmadressen jeweils in einer Integervariablen sind und nicht das LSB in der einen, das MSB dann in der naechsten Integervariablen. So kann man dann dem Maschinenprogramm sehr einfach Werte uebergeben. Nehmen wir an, wir wollen nicht grundsuetzlich den ganzen Bildschirm voll machen, sondern erst ab einer bestimmten Adresse, die vom Programmierer einzugeben ist. Weiterhin wollen wir auch den Code des zu ladenden Zeichens eingeben und wieviel mal kopiert werden soll.

Das Quellprogramm sieht dann folgendermassen aus:

Hexcode	Quellcode
---------	-----------

Hexcode	Quellcode	
00	NOP	
21003C	LD	HL,15360
00	NOP	
36BF	LD	(HL),191
00	NOP	
00	NOP	
11013C	LD	DE,15361
00	NOP	
01FF03	LD	BC,1023
EDB0	LDIR	
C9	RET	
00	NOP	
	END	

Wir erhalten die Kette

0021 003C 0036 BF00 0011 013C 0001 FF03 EDB0 C900

Mit vertauschten Bytes:

2100 3C00 3600 00BF 1100 3C01 0100 03FF B0ED 00C9

Unser Initialisierungsprogramm wuerde dann leicht abgeaendert folgendermassen aussehen:

```

010 DEFINT A-Z:J=0
020 UP(0)=&H2100:UP(1)=&H3C00'<==erste zu ladende Bild-
schirmadresse
030 UP(2)=&H3600:UP(3)=&H00BF'<==zu ladender Character
(Vorbelegung = BFH)
040 UP(4)=&H1100:UP(5)=&H3C01'<==Destinationadr.
050 UP(6)=&H0100:UP(7)=&H03FF'<==Anzahl der zu kopierenden
Zeichen
060 UP(8)=&HB0ED:UP(8)=&HC900
...
...

```

Wenn wir nun eine Eingaberoutine schreiben:

```

...
...
1000 CLS:INPUT"1. ZU KOPIERENDES ZEICHEN";HL
1010 INPUT"DESTINATION";DE
1020 INPUT"BYTE COUNT";BC
1030 INPUT"CHARACTER";CH
1040 UP(1)=HL:UP(3)=CH:UP(5)=DE:UP(7)=BC
1050 DEFUSR=VARPTR(UP(0))
1060 J=USR(0)
1070 GOTO 1070
...

```

S0 koennen einfach Daten dem Maschinenprogramm uebergeben werden. Diese muessen sich natuerlich nicht nur auf Bildschirmadressen beziehen. Auch ganze Strings koennen bei entsprechender Programmierung uebergeben werden.

Welche der beschriebenen Methoden der einzelne am besten findet wird nicht zuletzt von der Anwendung abhaengen.

Somit genug fuer heute...

Euer Bernd Niedermeier

Hochauflösende Softgraphic für den TRS80 Model I (Teil 1)

Ich bin ein frischgebackenes Clubmitglied und möchte meinen Einstand geben mit einem kleinen Maschinenprogramm, zu dem ich durch einen Artikel im Aprilheft der Zeitschrift mc angeregt wurde. Ich werde das Programm in mehreren Folgen besprechen und möchte Euch jetzt schon auffordern, reichlich Kritik zu üben. Das Programm ist nämlich noch gar nicht fertig, Vorschläge und Verbesserungen können noch eingebaut werden. Die Struktur des Programms, als Paket von Hilfsroutinen für das normale BASIC, fordert - wie ich finde - geradezu heraus, Erweiterungen zu erfinden. Wenn viele Köpfe daran arbeiten würden, könnte bestimmt etwas Brauchbares dabei herauskommen.

Doch genug der Vorrede.

Dreh- und Angelpunkt des Programms ist ein Speicherbereich im RAM, 8 mal so groß wie der Bildschirmspeicher des TRS80. Er soll ebenso leicht mit der Pixelgraphic gefüllt werden können wie der Bildschirm. In X-Richtung sind 256 und in Y-Richtung 192 Pixel einzeln ansteuerbar, insgesamt also 49152. Im Condensed Mode des EPSON MX80 füllt dieses Format fast eine ganze Druckseite.

Die Hardcopy-Routine braucht wohl nicht näher erläutert zu werden. Falls man den Epson mit TRS80 Zeichensatz fährt, sind die Zeilen 590 bis 610 zu löschen! Eventuell muß diese Routine noch an andere Drucker angepaßt werden in den Zeilen 530 und 690. Ich kenne leider nicht die Steuerzeichen anderer Drucker.

Die nächste kleine Routine (CLEAR) löscht den 8 kByte großen Speicherbereich, indem sie das Zeichen 80H hineinschreibt. Auch diese Routine erklärt sich wohl fast von selbst.

Diese Beiden sollen auch schon die einzigen für dieses Mal vorgestellten Routinen sein. Die Zeilen 740 bis 840 deuten schon darauf hin, was noch alles kommen wird. Wer neugierig ist, möge sich das oben erwähnte mc-Heft ansehen. Dieses Mal möchte ich viel lieber noch Eure Aufmerksamkeit auf die Zeilen 180 bis 440 lenken. Hier steckt der Verteiler für die einzelnen Routinen. Ich bin ziemlich stolz auf dieses Programmstück und bin sehr gespannt darauf, ob jemand einen Verteiler kennt, der kürzer ist und dennoch das Gleiche leistet. Bitte versteht mich jetzt richtig: Ich möchte nicht mit meinen Programmierkenntnissen prahlen. Vielmehr möchte ich Euch zur Kritik herausfordern, da ich der Meinung bin, daß es eine wunderbare Aufgabe für einen Userclub sein kann, optimale Programme (-Teile) gemeinsam zu entwickeln. So ist Fortschritt möglich, den es auf dem Gebiet der Software genauso gibt (geben muß), wie im Hardware-Bereich.

Der Verteiler ist so eingerichtet, daß Erweiterungen sehr leicht möglich sind. Es brauchen lediglich Änderungen in der Zeile 160 und die Eintragung des Einsprungpunktes der neuen Routine in die Sprungadressen-Tabelle (320 bis 440) vorgenommen werden. Auf diese Weise können bis zu 128 Rou-

tinem angesprungen werden, indem von BASIC aus mit  
USR(code) aufgerufen wird. Diese Methode ist also auch für  
Nicht-DiskBASIC-Benutzer mit nur einem USR-Befehl geeignet.  
Der Verteiler selbst ist voll relocatierbar.

Ich habe die jetzigen Programmteile absichtlich noch nicht  
mit festem ORG-Statement assembliert, da es dem Benutzer  
später möglichst freigestellt bleiben soll, wo er das Pro-  
gramm ablegt.

Im nächsten Teil werde ich die Routine VIEW besprechen,  
also den Teil des Programmpakets, der den 8 K Speicher auf  
dem Bildschirm sichtbar macht. Vielleicht bleibt auch noch  
Platz für SET, RESET und POINT.

Bis dann

Euer Jürgen Degenhardt

```

003B      00120 PRINT EQU 3BH ;Printer Output Routine
0080      00130 ROWLEN EQU 128 ;Tableau size is 256x192
          00135 ;pixels
2000      00140 TABLEN EQU 2000H ;= 8 kByte
1997      00150 SYNERR EQU 1997H ;Syntax Error Routine
000D      00160 JPTLEN EQU 13 ;Number of Subroutines
          00170 ;
0000 CD7F0A 00180 START CALL 0A7FH ;get code from BASIC
0003 2600   00190 LD H,0
0005 7D     00200 LD A,L
0006 FE0D   00210 CP JPTLEN ;test if code allowed
0008 D29719 00220 JP NC,SYNERR ;ERROR if it is not so
000B CB25   00230 SLA L ;displacement = 2 * code
000D 111600 00240 LD DE,JMPTBL
0010 19     00250 ADD HL,DE ;(HL)= LSB of jump address
0011 5E     00260 LD E,(HL)
0012 23     00270 INC HL ;point to MSB
0013 56     00280 LD D,(HL)
0014 EB     00290 EX DE,HL ;HL = jump address
0015 E9     00300 JP (HL) ;execute subroutine
          00310 ;
0005 6900   00320 JMPTBL DEFW RESET ;table of start addresses
0018 6A00   00330 DEFW SET ;for several subroutines
001A 6B00   00340 DEFW POINT
001C 6C00   00350 DEFW VIEW
001E 3E00   00360 DEFW HDCOPY
0020 3000   00370 DEFW CLEAR
0022 6D00   00380 DEFW FUNCIN
0024 6E00   00390 DEFW EVAL
0026 6F00   00400 DEFW ROUND
0028 7000   00410 DEFW STRMOV
002A 7100   00420 DEFW FRLLOOP
002C 7200   00430 DEFW FSLOOP
002E 7300   00440 DEFW FPLLOOP
          00450 ;
0030 217400 00460 CLEAR LD HL,TABLE ;this subroutine clears
0033 3680   00470 LD (HL),80H ;the whole tableau by
0035 117500 00480 LD DE,TABLE+1 ;filling it with 8 K
0038 010020 00490 LD BC,TABLEN ;graphic blanks
003B EDB0   00500 LDIR
003C C9     00510 RET
          00520 ;
003E 3E0F   00530 HDCOPY LD A,15 ;this subroutine makes
0040 CD3B00 00540 CALL PRINT ;a hardcopy of the tableau
0043 217400 00550 LD HL,TABLE ;on the EPSON MX/80.
0046 0E40   00560 LD C,64 ;# OF ROWS
0048 0680   00570 LLOOP LD B,ROWLEN ;LENGTH OF A ROW
004A 7E     00580 FETCH LD A,(HL)
004B FE80   00590 CP 80H ;TEST IF GRAPHICS
004D 3B02   00600 JR C,CHARAC
004F C620   00610 ADD A,20H ;SHIFT TO EPSON GRAPHIC
0051 CD3B00 00620 CHARAC CALL PRINT
0054 23     00630 INC HL
0055 10F3   00640 DJNZ FETCH
0057 3E0D   00650 LD A,0DH
0059 CD3B00 00660 CALL PRINT
005C 0D     00670 DEC C
005D 20E9   00680 JR NZ,LLOOP
005F 3E12   00690 LD A,18 ;CANCEL CONDENSED MODE
0061 CD3B00 00700 CALL PRINT ;see line 530
0064 3E0D   00710 LD A,0DH
0066 C33B00 00720 JP PRINT ;PRINT & RETURN
          00730 ;

```

```
0069 C9      00740 RESET   RET
006A C9      00750 SET     RET
006B C9      00760 POINT   RET
006C C9      00770 VIEW    RET
006D C9      00780 FUNCIN  RET
006E C9      00790 EVAL    RET
006F C9      00800 ROUND   RET
0070 C9      00810 STRMOV  RET
0071 C9      00820 FRL00P  RET
0072 C9      00830 FSL00P  RET
0073 C9      00840 FPL00P  RET
              00850 ;
0074          00860 TABLE EQU    $
402D          00880      END    402DH
00000 TOTAL ERRORS
27586 TEXT AREA BYTES LEFT
```

;these are 'Dummy Sub-  
;routines' for now

;start of the tableau

Bernd Niedermeier  
BER/NIE Software

Hirschbergweg 9

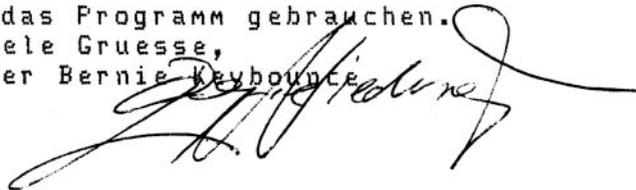
8011 Kirchheim  
Tel.:(089) 903 57 31

\*\*\*\*\*  
\* PRINT => LPRINT \*  
\*\*\*\*\*

Liebe Clubfreunde,  
der Beitrag von Bernard Haible aus der Clubzeitung Nr.12 hat mich dazu inspiriert, etwas Aehnliches zu schreiben. Ich haette durchaus Bernard Haibles Version umschreiben koennen, doch als Assemblerneuling wollte ich es ganz neu schreiben, ohne einen Blick darauf zu werfen. Was mich vor allem stoerte war, dass alle PRINT's in LPRINT's umgewandelt werden. Oft moechte man aber nur den Druckerausgabeteil des Programms konvertieren und die restlichen Bildschirmausgaben belassen. So wird bei meiner Version bei folgenden zwei Faellen nicht konvertiert:  
Format 1: PRINT "Text"  
Entscheidend ist ein Space zwischen dem PRINT und dem Anfuehrungszeichen.  
Format 2: PRINT@xx,"Text"  
Entscheidend ist das PRINT at-Statement.

Alle anderen PRINT's werden zu LPRINT's konvertiert. Das Programm liegt ab FF00H. MEMSIZE muss also mit 65280 beantwortet werden. (Diskbasic: BASIC,65280)  
Man laedt dann das Konvertierungsprogramm und das zu konvertierende Programm. Der Entrypoint des Programms ist FF02H bzw 65282 dezimal. Level II Anwender muessen folgende Pokeanweisung ausfuehren:  
POKE16526,2:POKE16527,255  
Der Aufruf des Programms erfolgt dann mit ?USR(0).

Ich hoffe, jemand kann das Programm gebrauchen.  
Viele Gruesse,  
euer Bernie Keybounce



```

00110 ;**UMWANDLUNG*PRINT=>LPRINT*UNTER*BEST:**VORAUSSETZUNGEN**
00120 ;* BER/NIE SOFTWARE Bernd Niedermeier Hirschbergweg 9 *
00130 ;* 8011 Heimstetten Tel.: 089/903 57 31 *
00140 ;* *
00150 ;* Keine Umwandlung bei 'PRINT "... ' (Space nach PRINT) *
00160 ;* Keine Umwandlung bei 'PRINT@xx, "... ' (PRINT at) *
00170 ;* Umwandlung bei 'PRINT"... " *
00180 ;*****
00190 ;
00200 ;

```

```

F00 00210 ENDE EQU 0FF00H
F02 00220 ORG 0FF02H
F02 2AF940 00230 LD HL,(40F9H) ;Anfang der
;Variablentabelle
F05 2B 00250 DEC HL ;Ende des Basicpr
F06 2200FF 00260 LD (ENDE),HL
F09 2AA440 00270 LD HL,(40A4H) ;Anfang des Baspr
F0C 23 00280 INC HL
F0D 23 00290 INC HL
F0E 23 00300 INC HL
F0F 23 00310 LOOPA INC HL
F10 EB 00320 LOOPB EX DE,HL
F11 2A00FF 00330 LD HL,(ENDE)
F14 DF 00340 RST 18H
DB 00360 RET C ;Wenn Ende, dann
;=> BASIC
;erreicht
F16 EB 00390 EX DE,HL
F17 AF 00400 XOR A
F18 BE 00410 CP (HL) ;Zeilenende?
F19 2815 00420 JR Z,ZENDE ;ja=>5Bytes weiter
F1B 3EB2 00430 LD A,0B2H ;'PRINT'
F1D BE 00440 CP (HL) ;'PRINT'?
F1E 20EF 00450 JR NZ,LOOPA ;nein=weilersuchen
F20 23 00460 INC HL
F21 3E20 00470 LD A,20H
F23 BE 00480 CP (HL) ;' ' nach PRINT ?
F24 28EA 00490 JR Z,LOOPB ;ja=>keine Umwdlg
F26 3E40 00500 LD A,40H
F28 BE 00510 CP (HL) ;'@' nach PRINT ?
F29 28E5 00520 JR Z,LOOPB ;ja=>keine Umwdlg
00530 ;Es handelt sich um ein umzuwandelndes PRINT
2B 2B 00540 DEC HL ;PRINT Adr
36AF 00550 LD (HL),0AFH ;LPRINT Token einset
n
2E 18DF 00560 JR LOOPA
30 23 00570 ZENDE INC HL
31 23 00580 INC HL
32 23 00590 INC HL
33 23 00600 INC HL
34 18D9 00610 JR LOOPA
00 00620 END
000 TOTAL ERRORS
695 TEXT AREA BYTES LEFT

```

DE	FF00	00210	00260	00330	
OPA	EE0E	00310	00450	00560	00610
OPB	FF10	00320	00490	00520	
NDE	FF30	00570	00420		

Ihr Computer wird d o p p e l t s o s c h n e l l

**Vorbemerkung:**

Diese Modifikation kann nur von denen vorgenommen werden, die sich einigermaßen in Elektronik auskennen und löten können. Auf eine genaue Beschreibung, die jeden Handgriff vorschreibt wurde deshalb bewußt verzichtet.

Eine Garantie für die Funktion dieser Modifikation kann nicht übernommen werden, noch können Ansprüche für Schäden geltend gemacht werden, die durch den Einbau dieser Modifikation entstehen. Der Einbau erfolgt ausschließlich auf eigenes Risiko.

Es kann Fälle geben, bei denen diese Modifikation Schwierigkeiten macht oder überhaupt nicht funktioniert. Dies kann an zu langsamen ROM's, Video-RAM's etc. liegen. Es gibt auch Floppy-Disk-Controller IC's (1771), die bei 3.5 MHz nicht arbeiten.

Bevor Sie mit dem Lötkolben anfangen zu hantieren, überzeugen Sie sich, daß Ihr Computer in Ordnung ist. Nachdem Sie sich ganz sicher sind, können Sie anfangen.

1. Tauschen Sie die CPU gegen eine Z80 A aus.
2. Die dynamischen RAM's 4116 in der Tastatur dürfen eine maximale Zugriffszeit von 300 ns haben.
3. Die dynamischen RAM's 4116 im Expander EG 3014 dürfen maximal eine Zugriffszeit von 200 ns haben.  
Falls die entsprechenden Speicher nicht eingebaut sind, müssen sie gegen solche ausgetauscht werden.
4. Bauen Sie den Schalter nach Abbildung 2 in die Tastaturplatine ein und führen Sie die Änderungen so durch, wie sie beschrieben sind. Achten Sie vor allem auf kurze Verbindungen und auf saubere Ausführung dieser Arbeiten. Diese Schaltung ermöglicht ein Umschalten der CPU-Taktfrequenz zwischen 1.7 MHz und 3.5 MHz. Weiter ist eine Verringerung der Zeitabstände der beiden Signale RAS und MUX notwendig. Dies wird in Abbildung 3 beschrieben.

Nach Abschluß dieser Arbeiten kontrollieren Sie nochmals sehr sorgfältig die ausgeführten Tätigkeiten und beachten Sie vor allem, daß keine ungewollten Brücken durch Lötzinn spritzer entstanden sind. Nun bauen Sie alles wieder zusammen und prüfen die ganze Anlage sehr ausführlich, um späteren Ärger zu vermeiden. Zuerst wird bei normaler Taktfrequenz ( 1.7 MHz ) das System gebootet und ein Speichertest durchgeführt. Danach werden Programme geladen und ausgeführt, die im Bereich von 8000H bis FFFFH arbeiten. Weiterhin sollte der Arbeitsspeicher mit einem bestimmten Bitmuster geladen werden und der Inhalt nach einigen Stunden überprüft werden. Auch sollten andere Betriebssysteme wie zum Beispiel CP/M oder PASCAL ausprobiert werden. Sind diese Tests erfolgreich verlaufen, so wird die gleiche Prozedur mit der doppelten Taktfrequenz wiederholt.

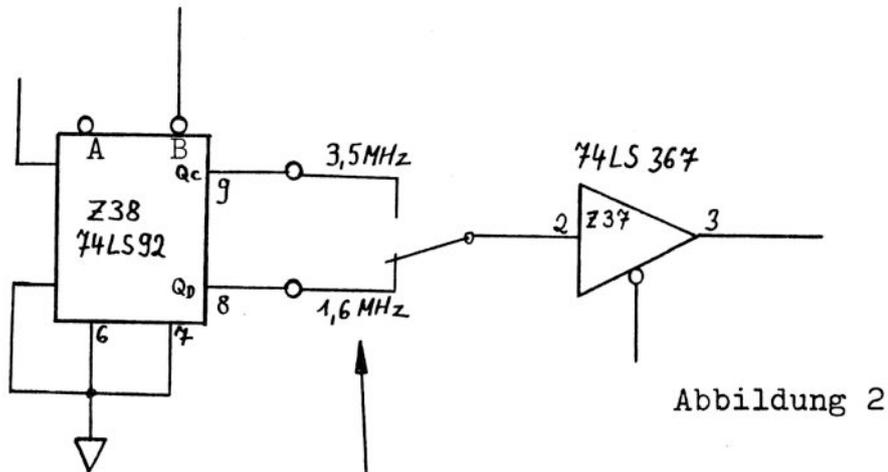
Sind auch die letzten Tests erfolgreich verlaufen, so können Sie Ihren Computer jetzt mit doppelter Taktfrequenz betreiben. Bei Verwendung von NEWDOS80 2.0 ändern Sie den Systemparameter BJ von 1 auf 2. Jetzt läuft das Betriebssystem auch bei 3.5 MHz. Die 3.5 MHz Taktfrequenz ist auch notwendig, wenn sie unter Verwendung des 5/8 Zoll Single/Double Density Controllers auf 8 Zoll-Laufwerken booten wollen oder 8 Zoll-Double-Density fahren wollen.

Sie werden von dem Geschwindigkeitszuwachs begeistert sein!



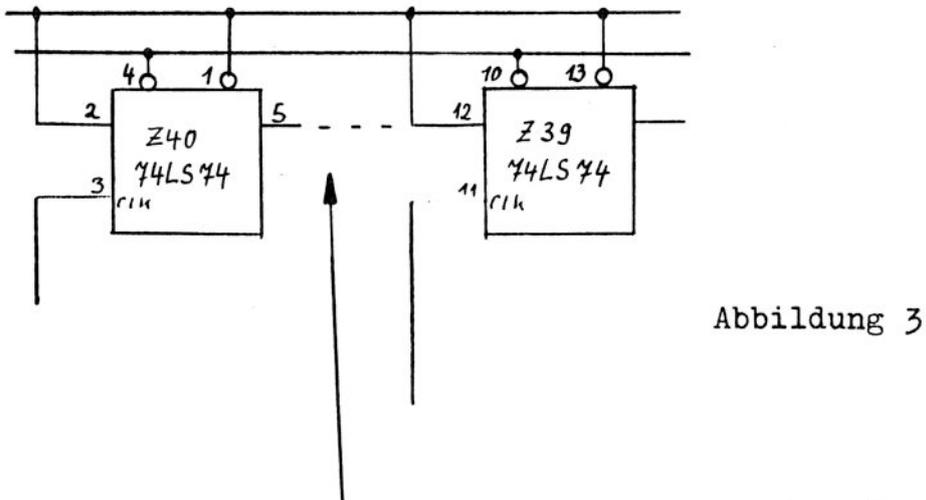
# GENIE 3,5MHz-Modifikation

## Änderung I :



Verbindung zwischen Pin 8 von Z38 und Pin 2 von Z37 auftrennen. Hierzu muß die CPU - Platine ausgebaut werden, da die entsprechende Leiterbahn auf der Unterseite verläuft. Umschalter 1 x um wie in obiger Abbildung einbauen.

## Änderung II :



Verbindung zwischen Pin 5 von Z40 und Pin 12 von Z39 auftrennen. dies geschieht am besten auf der Platinen= oberseite dicht bei Pin 5 von Z40. Neue Verbindung (Drahtbrücke) zwischen Pin 12 und Pin 13 von Z39 herstellen.

# TRS 80 – Schnelle Datenspeicherung auf Cassette

von Ulrich Heidenreich

Nachdruck mit freundlicher Genehmigung der MICRO EXTRA-Redaktion

Der TRS 80 – und auch andere Modelle, die das gleiche Microsoft-Basic verwenden – verfügt nicht über die Befehle OPEN und CLOSE bei Datenspeicherung auf Cassette. Dies hat zur Folge, daß bei Übergabe jedes Einzeldatums Datenvor- und Nachspann geschrieben werden, was zu einer nicht geringen Verlangsamung bei der Verarbeitung vieler kurzer Einzeldaten führt. Dieser Beitrag zeigt nun, wie durch Zusammenfassen zu Datenblöcken die Speicher- und Ladezeit verkürzt werden kann.

## Zum Prinzip

Die zu bearbeitenden Daten werden zu Datenblöcken à 249 Zeichen zusammengefaßt und auf Cassette gespeichert bzw. von diesen geladen. Laut „Basic manual“ sollen zwar 255 Zeichen (= Länge des I/O-Puffers) auf und von Cassette gelesen werden können; jedoch zeigten praktische Versuche, daß nur die ersten 249 Zeichen verarbeitet werden!

In Anlehnung zur Datenspeicherung auf Diskette möchte ich diese Datenblöcke im Folgenden als Sektoren bezeichnen.

## Zusammenfassen der Daten zu Sektoren

Jedes einzelne Datum stehe als Zeichenkette unter LIN\$ zur Verfügung. Numerische Daten können leicht mit dem Basic-Statement STR\$ in die nötige Form gebracht werden.

Aufgabe des „UP zur Komprimierung“ ist es nun, diese Datenzeilen zu den oben erwähnten Sektoren zusammenzusetzen. Außerdem wird jeder Datenzeile ihre Länge vorausgestellt, um beim Dekomprimieren der Daten keine Zeit mit der Suche nach Trennzeichen zu verschenken.

Das erste bis dritte Byte jeder Datenzeile enthält die erwähnte Länge; ein- und zweiziffrige Zahlen werden mit Blanks auf drei Stellen ergänzt. Ab dem vierten Byte folgen die eigentlichen Daten.

Daten, die die Grenze des Sektors überschreiten, werden zum Teil in den einen, zum Teil in den anderen Sektor geschrieben. Um ein sauberes Zählen bei der Dekomprimierung der Daten zu gewährleisten, erfolgt diese Trennung der Daten so, daß sie nicht innerhalb der ersten 3 Byte erfolgt, da sonst die Datenlänge nicht korrekt gelesen werden kann; außerdem beginnt jeder Datensektor mit der Anzahl der zum jeweiligen Datum gehörenden Zeichen, d. h. auch, wenn es sich um den Rest der Daten aus dem vorhergehenden Sektor handelt.

Bei der Zusammenfügung der Daten zum Sektor SEK\$ ist nun noch eine Eigenart des Basic zu beachten: Bei der Umwandlung der Datenlänge (= numerische Größe) in eine Zeichenkette wird aus z. B. L = 9 der String L\$ = "9". Dieses – fürs Vorzeichen freigehaltene – führende Blank wird beim Lesen der Stringvariable ignoriert, bei der Bestimmung der Länge jedoch mitgerechnet. Um ein „Verzählen“ beim Dekomprimieren der Daten zu vermeiden, wird dieses Blank vor der Zusammenfügung des Sektors mit Hilfe von RIGHT\$ entfernt.

## Speichern der Daten

Die so zu Sektoren zusammengeführten Einzeldaten können nun als SEK\$ (SEK) wie üblich auf Cassette gespeichert werden. Den Sektoren sollte vorher Ihre Anzahl mit auf Cassette geschrieben werden, dann muß beim Laden der Daten nicht nach einem EOF (End of File) – Zeichen gesucht werden, sondern nur die spezifizierte Anzahl Sektoren geladen werden.

## Laden der Daten

Die gespeicherten Daten werden als SEK\$ (SEK) in den Rechner geladen und dem UP zur Dekomprimierung übergeben.

## Herauslösen der Daten aus dem Sektor

Wie oben erwähnt, erfolgt das Herauslösen der ursprünglichen Daten aus den Sektoren nicht durch Suche nach Trennzeichen, sondern einfach durch Ausgabe der vor jedem Einzeldatum angegebenen Anzahl Zeichen. Wird das Sektorende erreicht, so wird überprüft, ob der nächste Sektor mit einem numerischen Zeichen beginnt. Ist dies der Fall, gibt diese Zahl die Anzahl der noch auszugebenden Zeichen an. Dies ist entweder der Rest des Datums aus dem vorhergehenden Sektor – dieser wird dem Datum angefügt – oder ein

```

7900 REM
7910 REM      UP ZUR KOMPRIMIERUNG
7920 REM
7930 REM  INPUT PTR      : ZEIGER AUF FREIRAUM IM SEKTOR
7940 REM      SEK      : FREIER SEKTOR
7950 REM      LIN$     : DATENZEILE
7952 REM      LTD      : MAXIMALE SEKTORLAENGE+1  (*)
7955 REM
7960 REM  OUTPUT SEK$(SEK) : KOMPRIMIERTE DATEN
7970 REM      SEK      : NAECHSTER FREIER SEKTOR
7980 REM      PTR      : ZEIGER AUF FREIRAUM IM SEKTOR
7990 REM
8000 L$=STR$(LEN(LIN$)):L$=RIGHT$(L$,LEN(L$)-1)
8010 LIN$=L$+STRING$(3-LEN(L$),32)+LIN$
8020 IFPTR>LTD-4,PTR=1:SEK=SEK+1:LIN=-1
8030 IFPTR+LEN(LIN$)<LTD,SEK$(SEK)=SEK$(SEK)+LIN$:PTR=PTR+LEN(LIN$):RETURN
8040 SEK$(SEK)=SEK$(SEK)+LEFT$(LIN$,LTD-PTR):SEK=SEK+1:LIN=-1
8050 SEK$(SEK)=RIGHT$(LIN$,LEN(LIN$)+PTR-LTD)
8060 L$=STR$(LEN(SEK$(SEK))):L$=RIGHT$(L$,LEN(L$)-1)
8070 SEK$(SEK)=L$+STRING$(3-LEN(L$),32)+SEK$(SEK)
8080 PTR=LEN(SEK$(SEK))+1:RETURN
8500 REM
8510 REM      UP ZUR DEKOMPRIMIERUNG
8520 REM
8530 REM  INPUT PTR      : ZEIGER AUF ZEICHEN IM SEKTOR
8540 REM      SEK      : ZU BEARBEITENDER SEKTOR
8550 REM      SEK$(SEK) : KOMPRIMIERTE DATEN
8560 REM
8570 REM  OUTPUT SEK    : BEARBEITETER SEKTOR
8580 REM      PTR      : ZEIGER AUF NAECHSTE DATEN ODER DATEIENDE
8590 REM      LIN$     : DATENZEILE
8600 REM      EFI     : DATEIENDE-FLAG
8610 REM
9000 MAX=LEN(SEK$(SEK))+1:L=VAL(MID$(SEK$(SEK),PTR,3)):IFL<>0,9020
9010 IFVAL(LEFT$(SEK$(SEK+1),3)<>0,SEK=SEK+1:PTR=1:LIN=-1:GOTO9000ELSEEFI=-1:RE
TURN
9020 PTR=PTR+3:IFPTR+L<=MAX,LIN$=MID$(SEK$(SEK),PTR,L):PTR=PTR+L:LIN=LIN+1:RETUR
N
9030 LIN$=RIGHT$(SEK$(SEK),MAX-PTR):SEK=SEK+1:PTR=1:LIN=-1
9040 L=VAL(LEFT$(SEK$(SEK),3))
9050 LIN$=LIN$+MID$(SEK$(SEK),4,L):PTR=PTR+3+L:LIN=LIN+1:RETURN
9060 REM
9070 REM  (*) : FUER EG3003 LTD=250, DA NUR 249 ZEICHEN IM SEKTOR MOEGlich !
9080 REM  BITTE INDIVIDUELL KONTROLLIEREN, OB LTD=256 MOEGlich !
9090 REM

```

neues Datum. Beginnt der neue Sektor nicht mit einem numerischen Zeichen, so ist das Dateiende erreicht; EFI = 1. Dies muß zwingend so sein, da jeder beschriebene Sektor mit der Anzahl der Zeichen beginnt; steht am Sektorenanfang keine Anzahl, so ist der Sektor leer!

### Hinweise zum Programmlisting

Im Programmkopf der beiden UP's ist jeweils unter INPUT angegeben, welche Parameter und Daten das Pro-

grammsegment benötigt; sowie unter Output, welche Daten es liefert bzw. welche Parameter nachgestellt werden. Beide UP's stellen außerdem die Variable LIN nach, die die Anzahl der bearbeiteten Datenzeilen repräsentiert. LTD bezeichnet die Sektorlänge + 1; hier ist 250 einzusetzen (vgl. oben). ■

V I D E O   S N O W   S H O V E L

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.

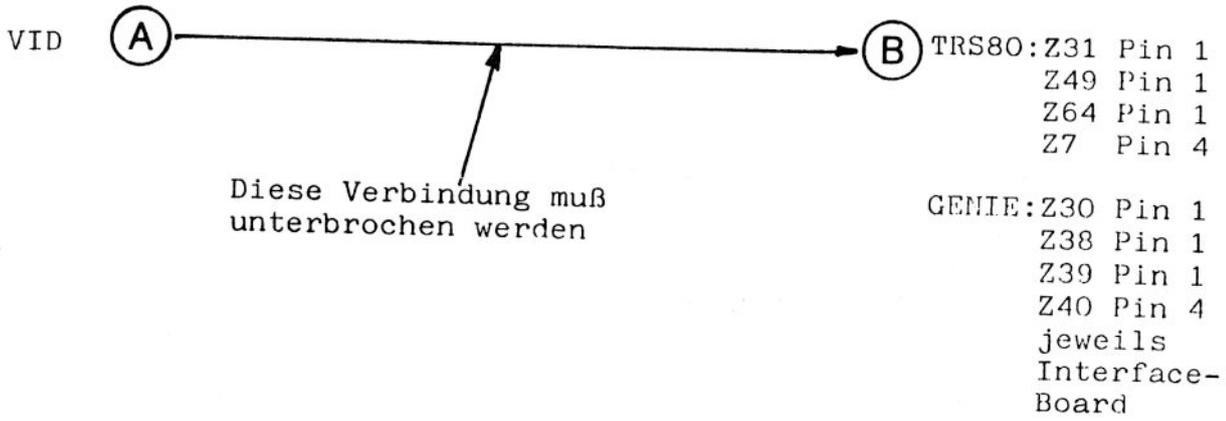


Abbildung 1

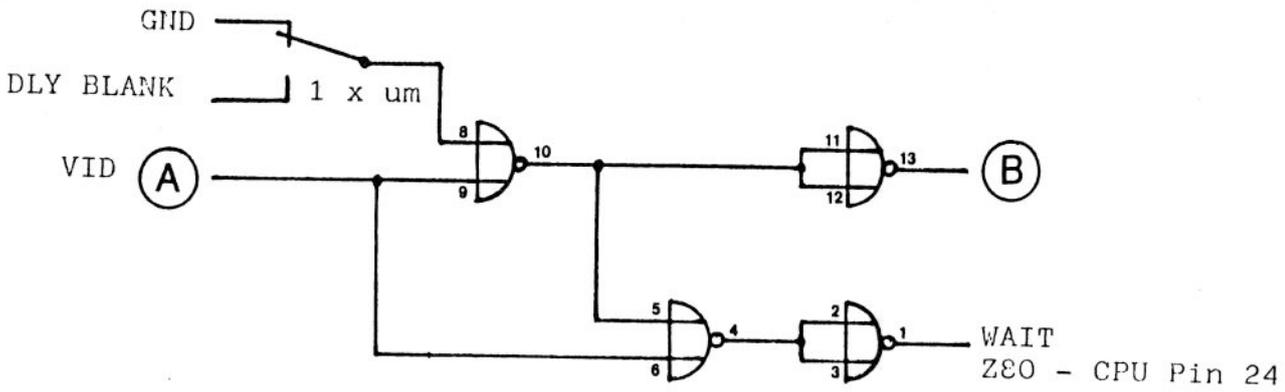


Abbildung 2

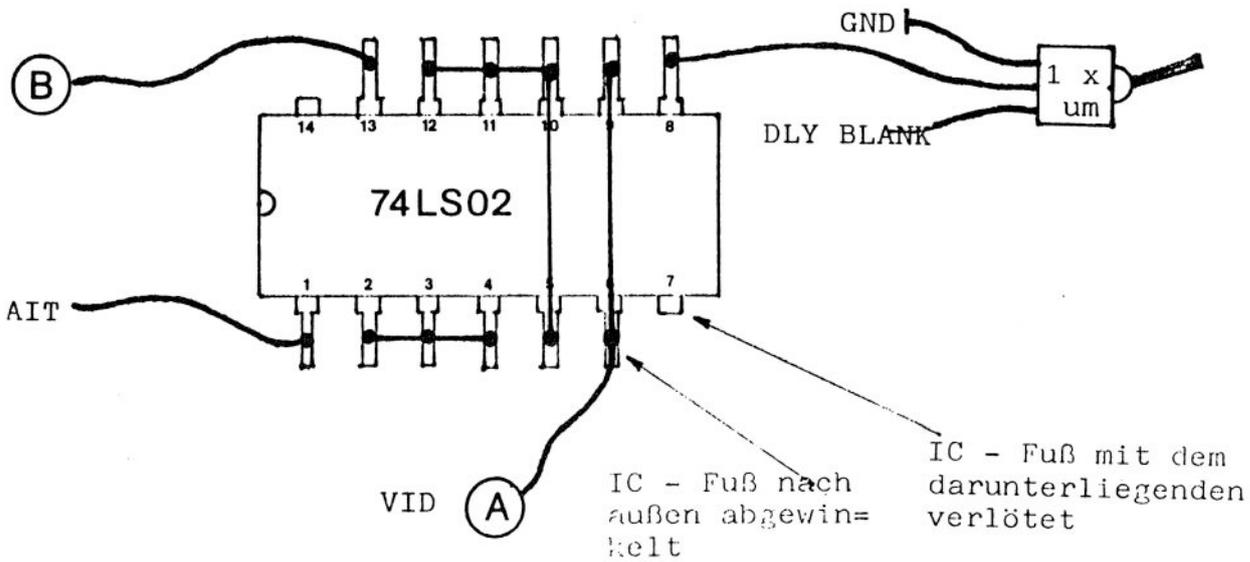


Abbildung 3

===== MITGLIEDER-ADRESSLISTE (ALPHABETISCH) =====

NAME	VORNAME	ADRESSE	WOHNORT	TELEFON
=====	=====	=====	=====	=====
BALLARIN	GREGOR	OWINGERSTR. 6	777 UEBERLINGEN	07551/63919
BAWIEDEMANN	KARL	PERETSHOFENERSTR. 7	8000 MUENCHEN 71	089/7913535
BERGBAUER	RUDOLF	PFEUFERSTR. 33	8000 MUENCHEN 70	089/7253469
BERGER	FRANZ	SCHUBERTSTR. 5	8037 DLCHING	08142/16876
BOEHLER	SEPP	MEMELWEG 21	7400 TUEBINGEN	07071/31825
BONENBERGER	PETER	WALDBLICKSTR. 15	7912 WEISSENHORN	07309/5570
BOVERMANN	KLAUS	OBERFOEHRINGERSTR. 107	8000 MUENCHEN 81	089/952239
BRANDES	HANS-DIETER	KOETNERHOLZWEG 47	3000 HANOVER 91	0511/2100547
BRUEBACH	ALFRED	IM KLEINEN FELDE 16	3430 WITZENHAUSEN 1	05542/4457
BUERGMAYR	MARKUS	MUENCHNERSTR. 22/2	8019 STEINHOERING	08094/1204
DEGENHARDT	JUERGEN	HILDEBRANDSTR. 34	3300 BRAUNSCHWEIG	0531/325700
DENZ	KLAUS	NELL.-SCHIERBERG 74	2846 NEUENKIRCHEN	05493/665
DUMKE	ANDREAS	PFANNMUELLERWEG 19	6100 DARMSTADT	06151/717700
EICKENBERG	GUSTAVO	JOHANN CLANZESTR. 43/W73	8000 MUENCHEN 70	089/7692251
EISENBERGER	KARL-HEINZ	GARTENSTR. 3	8011 GRASBRUNN 1	089/465621
ENDRES	MICHAEL	BRUCHSTR. 54	6920 SINSHEIM	07261/63666
FRANK	KLAUS	PERNERSTR. 30	8017 EBERSBERG	08092/23589
FRANZ	WOLFGANG	J.BAPTIST ZIMMERMANNSTR 4	8018 GRAFING	08092/5303
GEBERT	MANFRED	GAUTINGERSTR. 8	8031 GEISENBRUNN	
GIESELMANN	WILLHELM	AHRWEG 20	5142 HUECKELHOVEN	02433/85579
GRAESSLE	WILHELM	RACHELSTR. 34	8313 VILSBIBURG	08741/7450
GRENSING	WOLFGANG	HOMBERGER HOF	7776 OWINGEN	07551/62410
GREUBEL	KARL-HEINZ	OBENER WEG 9	8730 BAD KISSINGEN	0971/9380
GRIES	ULRICH	SILBERSTEINSTR. 92	1000 BERLIN 44	030/6253625
GROSSEGESSE	HANS JORDAN	WOLFRATSHAUSENER-STR. 68A	8000 MUENCHEN 70	089/7231905
HAIBLE	BERNHARD	SCHOENHUTWEG 5	7170 SCHWAEBISCH HALL	0791-43703
HANNE	BRUND	BIRKENSTR. 2	3014 LAATZEN 1	0511/867681
HERZOG	BENEDICT	STRASSBURGER STR. 77	2800 BREMEN 1	04221/344954
HESS	BERNHARD	KAZMAIRSTR. 30	8000 MUENCHEN 2	089/503125
HOFMANN	DIETER	OSTTOR 186	4400 MUENSTER	02501/13104
HOMBERGER	RUDOLF	ROEGGERSTR. 9	8900 AUGSBURG 21	0821/84173
HORNUNG	GUENTHER	KREUZBERGWEG 2	5568 DAUN	06592/1623
HUBER	HANS	HURTOEST 14	8225 TRAUNREUT	08669/5005
IMMERZ	PETER	WALTER-SCOTT-STR 4/312	8000 MUENCHEN	089/5701431
JANZ	KARSTEN	WEIMARERSTR. 30	5303 BORNHEIM 4	02227/1426
KART	RENATE	DEROYSTR. 6	8000 MUENCHEN 2	089/105983
KIRCHNER	PETER	BLUMENSTR. 11	8938 BUCHLOE	08241/2332
KOERBER	RUNAR	SPERLINGSTR. 14 B.WOERLE	8900 AUGSBURG	
KOSTHORB	ALFONS	DORFBAUERNGEHOEFT 50	4236 HAMINKELN 2	02052/4519
KRAML	KLAUS	SCHOENSTR. 20	8000 MUENCHEN 90	089/6518617
KRETSCHMAR	GUENTER	LEITENWEG 16	8190 WOLFRATSHAUSEN	08171/18457
KRONSCHNABL	KURT	VEILCHENWEG 5	8037 NEU-ESTING	08142/14469
LUECKEL	MANFRED	OSTERFELDERSTR. 13	4250 BOTTROP	02041/22324
MADER	MARTIN	SEBASTIAN-FRANCK-STR. 5	8850 DONAUWOERTH	0906/6673
MAIER	GERHARD	NEUBIBERGER STR. 58/2	8011 PUTZBRUNN	089/6015887
MAYRING	DR. LOTHAR	KARLSTR. 43/III	8000 MUENCHEN 2	089/595170
MILICZEK	KARL-HEINZ	HEITERWANGER STR. 46	8000 MUENCHEN 70	089/7602966
MODEL	KLAUS	YORCKSTR. 73	1000 BERLIN 61	030/7851837
NAGY	PAUL	BUCHENWEG 8	8192 GERETSRIED	08171/8245
NETZ	BERND	LAUINGERSTR. 10	8000 MUENCHEN 50	089/1491221
NEUBAUER	RAINER	TOELZERSTR. 131	8160 MIESBACH	080258949
NIEDERMEIER	BERND	HIRSCHBERGWEG 9	8011 KIRCHHEIM	089/9035731
NOSBUESCH	HUDA +	HEIDESTR. 82	5620 VELBERT 1	XXXXXX/54887
ORTHUBER	WOLFGANG	CHR.-PROBST STR. 16/1016	8000 MUENCHEN 40	089/3233263

\*\*\*\*\* MITGLIEDER-ADRESSLISTE (ALPHABETISCH) \*\*\*\*\*

NAME	VORNAME	ADRESSE	WOHNORT	TELEFON
====	=====	=====	=====	=====
PENTENRIEDER	FRANZ JOSEF	WILDMOOSSTR. 9	8130 STARNBERG-WANGEN	08151/89071
RAUCH	NORBERT	ERNST-HAECKEL-STR. 69 B	8000 MUENCHEN 50	089/8123081
REICHELSDORF	WOLFGANG	MARIENBADERSTR. 21	8858 NEUBURG/DONAU	08431/7846
RESSEL	JOSEF	EFFNERSTR. 75/C	8000 MUENCHEN 81	089/981408
RIEGER	LEONHARD	INN TALSTR. 4	8018 GRAFING	0892/5412
SAGNER	RAINER	AMSELWEG 10	8050 PULLING	08161/1546
SCHAARSCHMIT	BERNHARD	RAIFFEISENSTR. 62	8044 UNTERSCHLEISSHEIM	089/3101484
SHELLHORN	KURT	DONNERSBERGERSTR. 32	8000 MUENCHEN 2	089/165394
SCHICK	KLAUS	RHEINGAUSTR. 6	6238 HOFHEIM	06192/7500
SCHIER	REINHOLD	PAPPENHEIMSTR. 12	8000 MUENCHEN 2	089/194926
SCHITTENHELM	GERHARD	REUSSEN BACHSTR. 23	7778 MARKDORF	07544/3170
SCHLADEBACH	GERT	GERBERGASSE 1	7845 BUGGINGEN	07631/5379
SCHNEIDER	WOLFGANG	KRUENERSTR. 31	8000 MUENCHEN 70	089/7604120
SCHRAMM	VOLKER	PFRUENDESIEDLUNG 17	8311 GERZEN	08744/226
SCHUMMEL	MICHAEL	BREMERSTR. 143	2940 WILHELMSHAVEN	04421/25978
SCHWARM	HANS-MARTIN	ROLLNERSTR. 50	8500 NUERNBERG 10	0911/355820
SEIBOLD	RUDI	SEMPTWEG 2	8011 KIRCHHEIM	089/9037351
SEITZ	PETER	BONAMESSER STR. 69	6000 FRANKFURT 50	
SEIZMAIR	WINFRIED	FEHWIESENSTR. 10	8000 MUENCHEN 80	089/4313436.
SPIES	KARL	LUDWIG-STEUB-STR. 7	8025 UNTERHACHING	089/6115575
STAHL	VOLKER	POSSENHOFENERSTR. 39	8130 STARNBERG	08151/7215
SYLVIO	WALDAMERO	POSTFACH 402004	8000 MUENCHEN 40	089/5804184
THALMEIER	GREGOR	POSTFACH 08	8011 KIRCHSEEDON	08091/9085
TRAPPSCHUH	KURT	REINECKESTR. 6	8036 HERRSCHING	08152/2512
VOGEL	LOTHAR	GEORG-WALTER-STR. 5	3300 BRAUNSCHWEIG	0531/77273
VOGELSANG	MANFRED H.	POSTFACH 280	8316 FRONTENHAUSEN	08732/514
VOIGTS	FRIEDEMANN	ESCHENSTRASSE 4	8034 GERMERING	089/8414991
VOSS	ANDREAS	PICKELSTR. 19	8000 MUENCHEN 19	089/153784
WIMMER	FRANZ	RINGSTR. 20	8031 MAISACH	08142/13876
WINKLER	HERMANN	ASTALLERSTR. 6	8000 MUENCHEN 2	089/5024853
WIRTZ	WOLFGANG	SCHANDERLWEG 7	8000 MUENCHEN 82	089/4304324