ATARI USER CLUB

und Colour-Genie

USER CLUB 2530

BREMERHAVEN

MINI-INFO MINI-INFO MINI-INFO MINI-INFO MINI-INFO MINI-IN

MINI-INFO

OFNI-INIM OFNI-INIM OFNI-INIM OFNI-INIM OFNI-INIM OFNI-INI

3. JAHRGANG | 8. AUSGABE

Red.: Peter Spieß, Trugenhofenerstr. 27, 8859 Rennertshofen 1

---> URLAUBSAUSGABE <---

INTERNES

INTERNES VOM BETREUER

Liebe Clubmitglieder,

der Sommer geht (hoffentlich) seinem Höhepunkt entgegen und so trete ich ab 20.07.1985 meinen wohlverdienten Urlaub an. Bis einschließlich 01.09.1985 bin ich nicht erreichbar. Das soll natürlich nicht heißen, daß unser Briefträger auch keine Post austrägt. Beiträge für die kommenden Infos sind immer willkommen.

Das Urlaubsinfo ist leider etwas dünner als gewohnt ausgefallen. Ich bitte um Einsicht und hoffe, daß der ungewöhnliche Versandtermin nicht allzusehr angekreidet wird. Ab September gibt's dann das gewohnte Heft wieder.

Ich wünsche Euch allen fröhliche Urlaubs- (oder Arbeitstage) und verbleibe wie immer

Euer BitPit

$$$

Noch'n Hinweis: Zur Zeit wird in der heimischen Bastelstube der Druckerspooler aus der c't 6/85 aufgebaut und getestet. Er kann in vier verschiedenen Versionen bestückt werden: 8K, 16K, 32K und 64K. Da nur relativ wenige Bauteile benötigt werden, könnte man ihn preislich als echte Konkurrenz zu professionellen Geräten bezeichnen. Bei entsprechender Nachfrage läßt sich der Preis natürlich noch weiter senken. Wenn das Projekt zur Zufriedenheit funktioniert, folgt ein umfassender Testbericht.

P. Spieß

Geburtstagsecke

Im August können Geburtstag feiern:

Hans Bornschlegel	Dirk Hanss
Udo Jourdan	Hartmut Obermann
Alfred Punzet	Klaus Wolf
Waldemar Misioch	

Herzlichen Glückwunsch !!!

Der Joystick am TRS-80

Man hat die besten Spiele,spielt ab und zu damit,aber mal hört man doch auf.Warum??
Nun,die meisten denken sich weil sie keine Lust mehr haben.
Weit gefehlt.
Die Tastatur ist schuld,man will die Tastatur nicht ruinieren obwohl bestimte Tasten bestimmt schon etwas ausgeleiert sind.
Z.B. Die Pfeile und Space,nicht wahr?
Und warum?
Weil die meisten Spiele gerade mit diesen Tasten arbeiten.
Nun wie enlastet man diese Tasten wenn man weiter spielen will?
Man baut sie sich ausserhalb der Tastatur nocheinmal,und was hat man damit gewonnen??
Wenig,die Handhabung ist nicht gut,also etwas bequemeres.
Nun sehr bequem wäre ein Joystick.
Ja ja denkt man dann.A/D Wandler,Interface,Programmänderung...
Kostet alles in allem mehrere Märker + Joystick,aber um den kommte man ja sowieso nicht herum.Also dann gleich ein richtiger möglichst so,damit man ihn nicht umbauen muß.Also den weit verbreiteten ATARI-Joystick.
Damit wären wir beim Thema.

Einbau des Joysticks

Stückliste:
Wir brauchen
-einen ATARI-Joystick (natürlich)
-einen Passenden Stecker dazu ——— gibt's bei Jeyou!
-etwas Kabel
-einen Lötkolben
-evtl. ein Meßgerät
-diese Bauanleitung (klar!!!!!)
-ein wenig Zeit
-kein Wissen über Elektronik
sonst nichts

Nun denn,machen wir uns an die Arbeit.Diese Bauanleitung werden viele vielleicht schon kennen,aber manche auch noch nicht.
Diese Bauanleitung ist für den Anfänger und Laien gedacht.
Also ihr Profis,die auch einen Joystick haben wollen, vergeßt bitte alles,was Ihr über Elektronik wißt,weil es jetzt gleich einfach wird (zu einfach??)

Zuerst öffnen wir vorsichtig unseren Computer und suchen nach der Tastatur-Platine,das ist die,wo ganz wenig IC's und solche schwarzen Dinger drauf sind.
Wenn wir die haben suchen wir uns alle vier Pfeile heraus, und merken uns wo sie sind.Wir finden die Pfeile ganz einfach da die Platine auch auf der Rückseite beschriftet ist.
Jetzt suchen wir uns eine dieser seltsamen grünen Bahnen,auch Leiterbahnen genannt,die alle Pfeile miteinander verbindet.
D.h. Eine Leiterbahn,die von einem Lötfleck,eines Pfeiles genau zu einem Lötfleck eines anderen Pfeiles führt,wir markieren uns die Lötflecken,auch Pins genannt,oder schreiben sie uns auf.
Z.B. Pfeil hoch,oberer Pin
So suchen wir alle Pins auf den Pfeilen heraus (SPACE nicht vergessen) und löten an einen von diesen Pins ein nicht zu

kurzes Kabel.
An die anderen Pins löten wir auch ein Kabel,aber vorsicht
nicht an die Pins die wir uns aufgeschrieben haben.
Das war's bis jetzt.Jetzt bauen wir den Stecker ein.
Na wo soll er denn hin? Das Gehäuse muß auch noch zugehen.
Erst wenn wir damit fertig sind kann's weitergehen.
Ich warte solange.

```
Warte
Warte
Warte
Warte
Warte
Warte
```

Nanu schon fertig,bei mir ging's nicht so schnell.
Aber macht nichts.
Machen wir weiter.
Jetzt gelten folgende Bezeichnungen:
Das eine Kabel,daß wir nur an dem einen Pin angelötet haben
nennen wir jezt Masse kurz GND (von engl. Ground)
Das Kabel am Pin von HOCH heißt Hoch.
Das Kabel am Pin von TIEF heißt Tief.
usw.
Jetzt wirds interessant.
Am Stecker sind Nummern angebracht.
GND kommt an Pin 8 am Stecker
Space kommt an Pin 6
Hoch an Pin 1
Tief an Pin 2
Rechts an Pin 4
Links an Pin 3

Stecker rein,Computer zubauen und einschalten.
Stellt sich der normale Betrieb ein,stimmt bisher alles.
Stimmt etwas nicht,dann unter FEHLER nachschauen.
Alles klar,dann stecken wir den Joystick ein.
Dann geben wir das kleine Testprogramm ein und starten es.
Bringt es den gewünschten Erfolg,so steht dem Joystickspielchen
eigentlich nichts mehr im Wege.
Jetzt noch wie man den Joystick abfrägt:
Entweder mit PEEK(14400)

Hoch	=	8
Tief	=	16
Rechts	=	64
Links	=	32
Space	=	128

oder mit INKEY$ und den entsprechenden Abfragen.

Fehler:

Nun,stellt sich nicht der gewohnte Betrieb ein,oder das Test-
programm bringt einen Fehler,dann schalten wir den Computer
wieder aus,und öffnen ihn nochmals.
Wir überprüfen alle Lötstellen,ob sie auch guten Kontakt haben
oder sonstwie irgendetwas miteinander verbinden.
Haben wir keinen Fehler entdecken können,dann überprüfen wir
noch die Anschlüsse am Stecker.
Nachdem alles überprüft ist beginnen wir nochmals beim
Anschalten.Wenn trotzdem noch Fehler sind,so setzt euch bitte
mit mir in Verbindung.

Meine Adresse:
Jürgen Wagner
Espachweg 24
8951 Dösingen
08344/1333

Viel Spaß

3 8/85

```
10 CLS
20 PRINT"DRUECKEN SIE NACH OBEN"
30 I$=INKEY$:IFI$=""THEN30
40 IFASC(I$)=91ORASC(I$)=123THENPRINT"O.K.":GOTO60
50 PRINT"FEHLER":END
60 PRINT"DRUECKEN SIE NACH UNTEN"
70 I$=INKEY$:IFI$=""THEN70
80 IFASC(I$)=10THENPRINT"O.K.":GOTO100
90 PRINT"FEHLER":END
100 PRINT"DRUECKEN SIE NACH RECHTS"
110 I$=INKEY$:IFI$=""THEN110
120 IFASC(I$)=9THENPRINT"O.K.":GOTO140
130 PRINT"FEHLER":END
140 PRINT"DRUECKEN SIE NACH LINKS"
150 I$=INKEY$:IFI$=""THEN150
160 IFASC(I$)=8THENPRINT"O.K.":GOTO180
170 PRINT"FEHLER":END
180 PRINT"DRUECKEN SIE NACH FEUER"
190 I$=INKEY$:IFI$=""THEN190
200 IFI$=" "THENPRINT"O.K.":END
210 PRINT"FEHLER":END
```

LOTUS-Erweiterungen

..... deutsche Versionen zur Verfügung, die auf dem IBM-PC und kompatiblen Computersystemen mit einer speziellen Härte-erweiterung in der Lage sind, bis 8 Mega Watt Hauptspeicher für Programme und Informationen zu nutzen.

(gefunden von W. Reichelsdorfer in PC + PC-Soft Heft 7/85)

****

Nun aber etwas ernstes: Wer hat Interesse an einer MS-DOS oder PC-DOS Ecke in unserem Clubinfo ? Auf Grund der fortschreitenden Technologie dürfte dieser Themenbereich für uns in Zukunft nicht uninteressant sein.

Meinungen dazu bitte an die Betreuungsadresse oder an Wolfgang Reichelsdorfer direkt.

REM's unsichtbar

oder

Wie programmiert man REM's ohne REM und ohne Zeilennummer

Um von einem Programm ein ansprechendes Listing zu erhalten muß
man oft verwirrende Wege gehen,meistens über PEEK und POKE.
Aber es geht auch einfacher.
Zunächst einmal,wie muß ein entsprechendes Listing aussehen???
-Alle REM's sollten zwar dastehen,aber es sieht nicht gut aus,
wenn jede 3.Zeile ein REM am Anfang hat.
-Man sollte auch auf dem Bildschirm ein gutes Listing haben.
-Es sollte einfach zu machen sein.

REM's ohne REM und Zeilennummer

Ganz einfach.Man schreibt seine REM-Zeile ganz normal.
Drückt <ENTER> und geht in den EDIT Mode.
Mit dem Cursor auf die Stelle nach dem REM
Dann <I>nsert und <SHIFT><BACKSPACE>,bis der Cursor auf der
ersten Ziffer der Zeilennummer steht und <ENTER>
Bei LIST erscheint keine Zeilennummer und kein REM-Statement.

Aber

Die Bemerkung muß midestens so lang sein,daß man nichts mehr von
Zeilennumer und REM sieht,notfalls mit <SPACE> auffüllen.

Ist das jetzt einfach,und ohne PEEK und POKE.
Ja sogar der Drucker spielt mit er bringt ein Superlisting ohne
REM's heraus.
Zugegeben,die Methode ist kompliziert,aber wenn man nach jedem
REM sofort die Änderung macht,macht es eigentlich Spaß,weil man
ein Listing ohne Zeilennummern hat.
Ich glaube der Vorgang ist einfacher durchzuführen als zu
beschreiben.Als Abschluß noch ein paar Beispiele:

Printed by YOGI-Text 2.0 von Jürgen Wagner

```
10 REM Beispiele
20 REM Für REM's
30 REM vorher
```

Beispiele
Für REM's
vorher und nachher

S 8185

VIDHEX - Hexanzeige des Bildschirms mit der HRG

Die Sonderzeichen des Genie 3 (serienmäßig), 2 und 1 (nach Hardwareänderung) mit den ASCII-Codes 00-1F sehen zwar ganz putzig aus. Sieht man sie jedoch in einem Dump mit dem Debugger oder erscheinen sie auf andere Weise auf dem Bildschirm, so ist man gelegentlich ratlos. Sie sind nämlich in keiner Tabelle zu finden, so daß man ihre ASCII-Codes bei Bedarf nur erraten kann. Um diesem Mißstand abzuhelfen, entwarf ich das Programm VIDHEX/CMD, das mit Hilfe der HRG 1b anstelle der ASCII- und Sonderzeichen kleine zweistellige Hexzahlen anzeigt. Was dabei herauskommt, ist in dem HRG-Ausdruck am Ende des Listings zu sehen. Es handelt sich um eine Anzeige des Debuggers.

Die Ladeadresse 3900 ist mit der serienmäßigen Hardware natürlich nicht zu realisieren. Das geht mit einer kleinen Zusatzplatine, die Helmut Bernhardt in c't vorstellte (RAM von 3900-3BFF). Wer sie nicht hat, braucht nur ORG auf irgendeine andere Adresse zu setzen (genau 256 Bytes müssen bis zum Himem noch mindestens frei sein).

Das Programm ist mit ZEUS/CMD erstellt. Für EDTASM müssen alle Labels auf max. 6 Zeichen gekürzt werden. Kleinbuchstaben sind nur in den Kommentaren erlaubt. DB muß in DEFB geändert werden, DW in DEFW. In einem DEFB-Statement dürfen die einzelnen Bytes auch nicht durch Komma getrennt in eine gemeinsame Zeile gepackt werden, sondern jedes Byte erhält eine eigene Zeile. Das alles ändert aber nichts an der Programmlogik, die im folgenden erläutert werden soll.

Unter allen Steuerzeichen ist (zumindest in der Direkteingabe über die Tastatur) ESC das überflüssigste (escape, Shift-Hochpfeil, ASCII 27 bzw. 1B). Es hat zudem den Vorteil, daß sich mit ESC nichts auf dem Bildschirm ändert, solange man nicht etwa mit dem Level-4-ROM arbeitet und ESC zusammen mit einer anderen Taste drückt. Deshalb wurde als Trigger für die Umwandlung des Bildschirms ESC ausgewählt. Hierzu erhielt der Tastaturtreiber einen kleinen Vorspann, der vor der normalen Tastaturabfrage prüft, ob Shift mit dem Hochpfeil gedrückt wurde. Dies geschieht im Programmsegment newdrv. Wird kein oder ein anderes Zeichen festgestellt, geht es weiter an 4516, dem Beginn des normalen Tastaturtreibers (G-DOS bzw. H-DOS).

Nach ESC erfolgt nun die Umwandlung. Hierzu wird zunächst der Bildschirm gerettet, denn er muß gelöscht werden, um die HRG-Anzeige nicht zu stören. Ein Puffer wird für das Kilobyte des Bildschirms gebraucht. Die HRG benutzt von jedem Byte ihres Speichers nur 6 Bits zur Anzeige. Die beiden höchstwertigen Bits bleiben unsichtbar. Was liegt näher, als dort den Bildschirm zu verstauen? Dazu wird zunächst der Videozeiger mit 3C00 geladen, dem Beginn des Screens. Der HRG-Zeiger kommt auf den Anfangswert 0000. In den verschachtelten Schleifen vidsav1 und vidsav2 wird jeweils der Akku mit dem Videobyte geladen und mit C0 undiert, um die übrigen Bits zu löschen. Dieses Viertelbyte im Akku wird nun über den Port 5 in die HRG geladen. Jetzt werden die beiden oberen Bits der Videostelle nach unten rotiert; die nächsten beiden Bits stehen an. Das ist ein bißchen fummelig und kostet auch ein paar Bytes Programmcode, aber 1 kB Bildschirmpuffer im RAM wäre ein vielfacher Verlust.

Anschließend wird der HRG-Speicher gelöscht. Weshalb das nötig ist, wird später erklärt. Seine internen Adressen gehen von 0000 bis 2FFF. Der Zeiger HL startet deshalb nach dem Puffern des Bildschirms mit 0400 (= 1 kB), wo der HRG-Zeiger jetzt gerade steht. Das MSB der HRG wird bei jedem Schleifendurchlauf auf 30, den ersten nicht mehr erlaubten Wert geprüft. Dies geschieht in der Schleife clear.

Nach dem Löschen geht es im UP hexdisp weiter. Es wird jeweils ein Bildschirmzeichen geladen und mit Blank verglichen. Um den Bildschirm übersichtlich zu halten, wird ein Blank nicht mit der Hexzahl 20 ange-

zeigt. In diesem Fall wird das UP byte nicht angesprungen. Das würde ohne vorheriges Löschen der HRG bedeuten, daß die alten Codes dort erhalten blieben. Daher war zuvor die Löschung erforderlich.

Im UP byte wird die Videoadresse auf die HRG-Adresse umgerechnet. Hierzu braucht (zumindest für die oberste Dotzeile) nur 3C vom MSB subtrahiert zu werden. In diesem Falle wird mit 03 undiert, was auf dasselbe hinausläuft. Nacheinander werden nun die beiden Halbbytes in eine Ziffer umgerechnet und angezeigt:

Das obere Nibble wird zunächst durch 16 dividiert und damit ins untere geschoben. DE wird nun als Zeiger auf die Zeichensatztabelle chrtab geladen und das UP nibble angesprungen. Je nach Ziffer wird der Zeiger dort bis zu 16mal um fünf Stellen weitergerückt, denn jede Ziffer setzt sich aus fünf Dotmustern zusammen. Die jeweilige Stelle in der HRG wird im UP HRGadr adressiert und bei output das Byte ausgegeben. Für das nächste der fünf Bytes pro Ziffer muß der HRG-Zeiger HL um 1 kB erhöht werden. Das geht am einfachsten durch viermaliges Inkrementieren des MSB.

Um beide Hexziffern auf dem engen Raum einer einzigen Video-Anzeigestelle gut unterscheidbar zu machen, stehen sie untereinander und sind um zwei Dots in der Waagerechten gegeneinander verschoben. Das geschieht beim unteren Nibble (Einerstelle der Zahl) durch zweimaliges RLCA. Da das UP Nibble nicht unterscheidet, welche der beiden Ziffern gerade ansteht, werden die beiden RLCA immer abwechselnd abgearbeitet oder übersprungen. Dazu dient der relative Sprungbefehl JR displc. Die Sprungdistanz displc wird mit dem XOR-Befehl gleich zu Beginn des UP nibble ständig zwischen 00 und 02 (durch Verwendung von Labels variabel gehalten) hin- und hergeschaltet.

Wenn beide Ziffern angezeigt sind, geht es mit der nächsten Bildschirmstelle weiter, bis alle Bildschirmzeichen als Hexzahlen auf dem HRG-Screen stehen. Die HR-Graphik bleibt stehen, bis irgendeine Taste gedrückt wird. Jetzt muß der alte Bildschirm wieder restauriert werden. Die Bildschirmzeichen, die bisher geduldig im Puffer gewartet haben, werden zurückgeladen. Das geschieht im Prinzip wie das Puffern, nur eben umgekehrt: Die beiden oberen Bits aus einer HRG-Stelle werden aus dem Akku hinaus- und in ein Bildschirmbyte hineinrotiert. Damit ist endlich alles erledigt, mit RET geht es zurück ins Betriebssystem.

Die Befehle OUT (1),A und OUT (0),A zum Ein- und Wiederausschalten der HRG stehen an frühst- bzw. spätestmöglicher Stelle. Daher kann der User bei 1,77 MHz ungefähr 2-3 Sekunden lang ein wildes Schauspiel auf dem Screen beobachten. Gute Unterhaltung!

Arnulf Sopp

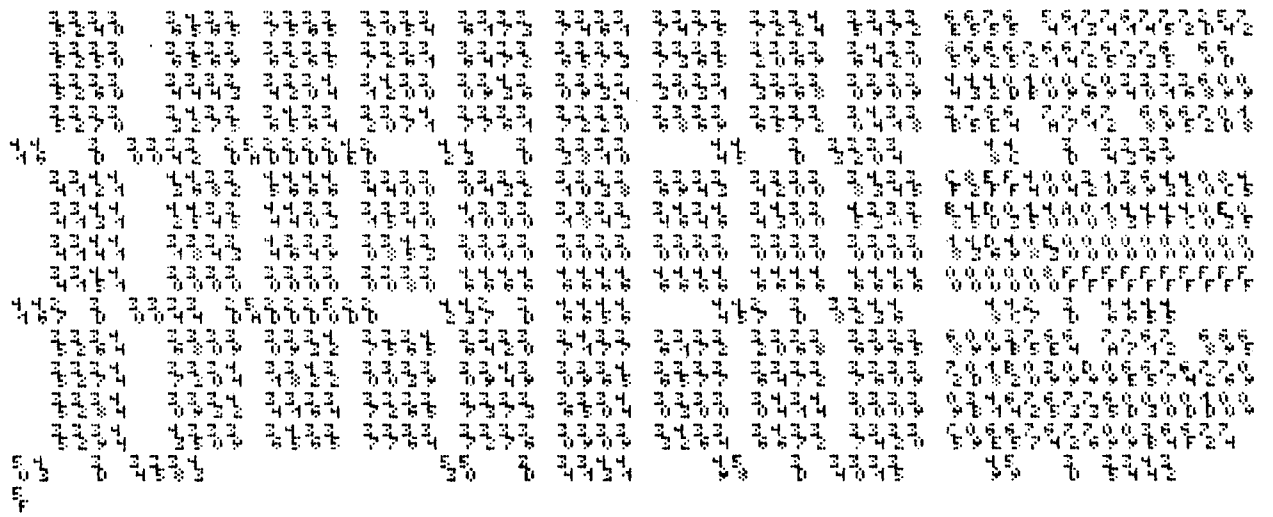
```
                    00001 ;=====================================================
                    00002 ;       Umwandlung der Bildschirmanzeige von ASCII-
                    00003 ;       Zeichen in Hexzahlen mit Hilfe der HRG 1b
                    00004 ;               (C) '85 by The HACKTORY
                    00005 ;=====================================================
                    00006
3900                00007         ORG     3900h           ;Lade- u. Einsprungsadr.
                    00008
                    00009 ;neuer Tastaturtreiber für Shift-Aufwärtspfeil
3900  3A4038        00010 newdrv  LD      A,(3840h)       ;Tastatur Steuerzeichen
3903  CB5F          00011         BIT     3,A             ;Aufwärtspfeil gedrückt?
3905  2804          00012         JR      Z,exit          ;falls nein
3907  3A8038        00013         LD      A,(3880h)       ;ja, Shift-Reihe
390A  B7            00014         OR      A               ;Shift gedrückt?
390B  CA1645        00015 exit    JP      Z,4516H         ;falls nicht Sh.-Hochpf.
                    00016
                    00017 ;nach Shift-Hochpfeil Bildsch. retten
390E  D301          00018         OUT     (1),A           ;HRG einschalten
3910  21003C        00019         LD      HL,3c00h        ;Bildschirmadresse
3913  55            00020         LD      D,L             ;DE <- 00xx, HRG-Adresse
3914  5D            00021         LD      E,L             ;DE <- 0000
3915  D5            00022         PUSH    DE              ;HRG-Adresse retten
3916  E5            00023         PUSH    HL              ;dto. Videoadresse
3917  01C004        00024 vidsav1 LD      BC,04c0h        ;4*2 Bits/Byte, Konst. C0
391A  CDA839        00025 vidsav2 CALL    HRGadr          ;HRG-Stelle adressieren
391D  7E            00026         LD      A,(HL)          ;Bildschirmzeichen
391E  A1            00027         AND     C               ;nur oberste 2 Bits
391F  D305          00028         OUT     (5),A           ;auf HRG ausgeben
3921  CB06          00029         RLC     (HL)            ;Zeichen 2 Bits aufrücken
3923  CB06          00030         RLC     (HL)
3925  10F3          00031         DJNZ    vidsav2         ;bis 1 Byte fertig
3927  23            00032         INC     HL              ;nächste Videostelle
3928  CB74          00033         BIT     6,H             ;Bildsch. überschritten?
392A  28EB          00034         JR      Z,vidsav1       ;falls noch nicht
                    00035
                    00036 ;HRG-Speicher ab nächster freier Stelle löschen
392C  CDA839        00037 clear   CALL    HRGadr          ;HRG-Stelle adressieren
392F  FE30          00038         CP      30h             ;Speicher überschritten?
3931  2805          00039         JR      Z,cleared       ;falls ja
3933  AF            00040         XOR     A               ;A <- 00
3934  D305          00041         OUT     (5),A           ;diese Stelle löschen
3936  18F4          00042         JR      clear           ;nächste Stelle
3938  E1            00043 cleared POP     HL              ;Bildschirmanfang
3939  E5            00044         PUSH    HL              ;für später retten
                    00045
                    00046 ;Bildschirm von ASCII- in Hexanzeige ändern
393A  E5            00047 hexdisp PUSH    HL              ;Videozeiger retten
393B  4E            00048         LD      C,(HL)          ;Bildschirmzeichen
393C  3E20          00049         LD      A,' '           ;Blank
393E  B9            00050         CP      C               ;Blank? (nicht verändern)
393F  77            00051         LD      (HL),A          ;diese Stelle löschen
3940  C46539        00052         CALL    NZ,byte         ;kein Bl., 1 Byte umwand.
                    00053
                    00054 ;Zeichen umgewandelt oder Blank überspr.; nächst. Zeichen
3943  E1            00055         POP     HL              ;Bildschirmzeiger
3944  23            00056         INC     HL              ;nächste Stelle
3945  CB74          00057         BIT     6,H             ;Bildsch. überschritten?
3947  28F1          00058         JR      Z,hexdisp       ;nein, nächstes Byte
                    00059
                    00060 ;Nach Anzeige auf Taste warten, dann Bildschirm restaur.
3949  CD4900        00061         CALL    0049h           ;auf Tastendruck warten
394C  E1            00062         POP     HL              ;Videoadresse
394D  D1            00063         POP     DE              ;HRG-Adresse
394E  0604          00064 restor1 LD      B,4             ;4*2 Bits/Byte
3950  CDA839        00065 restor2 CALL    HRGadr          ;HRG-Stelle adressieren
3953  DB04          00066         IN      A,(4)           ;HRG-Byte holen
```

```
3955  07        00067          RLCA                    ;2 oberste Bits
3956  CB16      00068          RL      (HL)            ;in den Bildsch. laden
3958  07        00069          RLCA
3959  CB16      00070          RL      (HL)
395B  10F3      00071          DJNZ    restor2         ;bis 1 Videobyte fertig
395D  23        00072          INC     HL              ;nächste Bildschirmstelle
395E  CB74      00073          BIT     6,H             ;Bildsch. überschritten?
3960  28EC      00074          JR      Z,restor1       ;falls noch nicht
3962  D300      00075          OUT     (0),A           ;HRG ausschalten
3964  C9        00076          RET                     ;Tastaturtreiber verlass.
                00077
                00078 ;UP Hexanzeige: 1 Byte ändern
3965  7C        00079 byte     LD      A,H             ;MSB der Videoadresse
3966  E603      00080          AND     03              ;Adr. Vid. -> Adr. HRG
3968  57        00081          LD      D,A             ;neues MSB
3969  5D        00082          LD      E,L             ;HRG-MSB wie Video-MSB
396A  79        00083          LD      A,C             ;Videozeichen
396B  E6F0      00084          AND     0f0h            ;oberes Nibble
396D  0F        00085          RRCA                    ;ins untere schieben
396E  0F        00086          RRCA
396F  0F        00087          RRCA
3970  0F        00088          RRCA
3971  21AB39    00089          LD      HL,chrtab-5     ;vor Tab. f. Hexzeich.
3974  E5        00090          PUSH    HL              ;brauchen wir noch
3975  CD7C39    00091          CALL    nibble          ;oberes Nibble anzeigen
3978  79        00092          LD      A,C             ;alter Code
3979  E60F      00093          AND     0fh             ;unteres Nibble
397B  E1        00094          POP     HL              ;Tabellenzeiger
                00095
                00096 ;einzelnes Halbbyte in die HRG laden
397C  47        00097 nibble   LD      B,A             ;als Zähler i. d. Tabelle
397D  3A9639    00098          LD      A,(displc)      ;Sprungdistanz
3980  EE02      00099          XOR     output-displc-1 ;umschalten
3982  329639    00100          LD      (displc),A      ;neu laden
3985  04        00101          INC     B               ;wegen DE = Tabelle -5
3986  23        00102 seekchr  INC     HL              ;Zeiger nachstellen
3987  23        00103          INC     HL              ;über 5 Stellen, weil
3988  23        00104          INC     HL              ;5 Codes pro Zeichen
3989  23        00105          INC     HL
398A  23        00106          INC     HL
398B  10F9      00107          DJNZ    seekchr         ;bis Code gefunden
398D  0605      00108          LD      B,5             ;5 Dotzeilen pro Zeichen
398F  C5        00109 nibloop  PUSH    BC              ;Zähler retten
3990  CDA839    00110          CALL    HRGadr          ;HRG-Stelle adressieren
3993  1B        00111          DEC     DE              ;HRG-Zeiger korrigieren
3994  7E        00112          LD      A,(HL)          ;Dotzeile laden
3995  1800      00113          JR      $+2             ;variable Sprungdistanz
3996            00114 displc   EQU     $-1             ;hier Distanzbyte
3997  07        00115          RLCA                    ;lower Nibble verschieben
3998  07        00116          RLCA                    ;um 2 Dots
3999  4F        00117 output   LD      C,A             ;Dotzeile retten
399A  DB04      00118          IN      A,(4)           ;HRG-Byte mit Videocode
399C  B1        00119          OR      C               ;mit Dotzeile verknüpfen
399D  D305      00120          OUT     (5),A           ;Dotzeile in HRG laden
399F  14        00121          INC     D               ;im MSB um 1 kB erhöhen
39A0  14        00122          INC     D               ;für nächste Dotzeile
39A1  14        00123          INC     D
39A2  14        00124          INC     D
39A3  23        00125          INC     HL              ;nächster Code für Ziffer
39A4  C1        00126          POP     BC              ;Zähler restaurieren
39A5  10E8      00127          DJNZ    nibloop         ;bis Nibble angezeigt
39A7  C9        00128          RET                     ;zurück
                00129
                00130 ;UP, um die HRG-Adresse auszugeben
39A8  7B        00131 HRGadr   LD      A,E             ;LSB der HRG-Adresse
39A9  D302      00132          OUT     (2),A           ;auf Port ausgeben
```

```
39AB  7A        00133          LD      A,D             ;MSB
39AC  D303      00134          OUT     (3),A           ;dto.
39AE  13        00135          INC     DE              ;nächste HRG-Stelle
39AF  C9        00136          RET
                00137
                00138 ;Codetabelle für HRG-Zeichensatz
39B0  02        00139 chrtab   DB      2,5,5,5,2       ;0
39B5  04        00140          DB      4,6,5,4,4       ;1
39BA  03        00141          DB      3,4,2,1,7       ;2
39BF  07        00142          DB      7,4,2,4,7       ;3
39C4  04        00143          DB      4,5,7,4,4       ;4
39C9  07        00144          DB      7,1,3,4,3       ;5
39CE  06        00145          DB      6,1,7,5,2       ;6
39D3  07        00146          DB      7,4,2,1,1       ;7
39D8  02        00147          DB      2,5,2,5,2       ;8
39DD  02        00148          DB      2,5,7,2,1       ;9
39E2  02        00149          DB      2,5,7,5,5       ;A
39E7  03        00150          DB      3,5,3,5,3       ;B
39EC  06        00151          DB      6,1,1,1,6       ;C
39F1  03        00152          DB      3,5,5,5,3       ;D
39F6  07        00153          DB      7,1,3,1,7       ;E
39FB  07        00154          DB      7,1,3,1,1       ;F
                00155
                00156 ;neue Tastatur-Treiberadresse im DCB
4016            00157          ORG     4016h           ;und zwar hier
4016  0039      00158          DW      newdrv          ;Adresse
                00159
3900            00160          END     newdrv          ;dort Einsprung

00000 Fehler

HRGadr  39A8    byte    3965    chrtab  39B0    clear   392C    cleared 3938
displc  3996    exit    390B    hexdisp 393A    newdrv  3900    nibble  397C
nibloop 398F    output  3999    restor1 394E    restor2 3950    seekchr 3986
vidsav1 3917    vidsav2 391A
```



10 8185