

oettle & reichler
datentechnik

Schießgrabenstr. 28
8900 Augsburg 1

Tel.: (0821) 15 46 32

R G B

RGB-Terminal Software

H A N D B U C H

Copyright (C) by DATENTECHNIK OETTLE + REICHLER, Augsburg
September 1984

Inhaltsverzeichnis

Einführung	3
Zeichenaufbau	3
Zeichen- und Befehlsausgabe	3
TVI950 Kompatibilität	4
Cursor Befehle	5
Löschfunktionen	6
Zoom und Bildschirmausschnitt	7
Softscroll	8
Charakter Darstellung	8
Farbanwahl	10
Farbparameter	11
Invert	11
Block Drawing	11
Komplement	12
Farbparameter Bitbelegung	13
Hinweise	14
Cursor Darstellung	15
Direkte RGB-Modul Programmierung	16
Übersicht und Voreinstellung der Kommandos	17

Kein Teil dieser Veröffentlichung darf reproduziert, vervielfältigt, gespeichert oder übersetzt werden, ohne die ausdrückliche schriftliche Zustimmung von DATENTECHNIK oettle & reichler. Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung einer Schaltung oder unserer Produkte dienen, ohne besondere Hinweise vorzunehmen. Für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen wird keine Haftung übernommen.

Einführung:

Das Softwarepaket RGB-TERM übernimmt die komfortable alphanumerische und grafische Bildschirmausgabe über das RGB-Modul. Neben den grafischen Möglichkeiten des Graphic Display Prozessors uPD 7220 stehen dem Anwender eine Vielfalt alphanumerischer Ausgabe-Befehle zur Verfügung, um flexibel einen übersichtlichen Bildschirmaufbau zu gestalten. Es können z.B. Grafiken mit Texten unterschiedlicher Größe, Schreibrichtung, Form und Farbe kommentiert werden. RGB-TERM ist in das Bios des CP/M Systems eingebunden und in Z-80 Code geschrieben. Dem Software-Paket ausgelagert ist die Definition der Control- und Escape Sequenzen im File TVI950.ASM, der standardmäßig dem Terminal TVI950 angepasst ist. Durch die Trennung ist eine Umdefinition der Zeichen leicht möglich und kann somit an bestehende Software angepasst werden.

Zeichenaufbau:

Bei ausgeschalteter Zoomfunktion (xl) werden 2040 Zeichen zu 85 x 24 Zeichen auf dem Bildschirm dargestellt. Die Zeichen sind aus einer 12x8 Punktmatrix mit echten Unterlängen aufgebaut. Daraus ergibt sich ein Bildaufbau aus 680 Punkten horizontal und 280 Linien vertikal. Diese recht hohe Zahl aktiver Zeilen und Spalten kann ein Nachjustieren der Bildlage im angeschlossenen Monitor notwendig machen. Dazu sind die Regler für vertikale und horizontale Amplitude auf kleiner Werte zu stellen. Um ein absolut flimmerfreies Bild zu gewährleisten, wurde eine Bildwechselfrequenz von 50 Hz (noninterlaced) gewählt. Manipulationen des Videorams finden nur in den Blanking-Phasen des Bildes statt. Sie sind daher für den Betrachter nicht bemerkbar.

Zeichen- und Befehlsausgabe:

Die im folgenden aufgeführten Befehle an das RGB-TERM sind über die BDOS Funktion 2 Console Output aufrufbar. Voraussetzung hierzu ist allerdings, daß die Console-Ausgabe auch wirklich über das RGB-Terminal abläuft. Dies muß mit dem Device-Kommando entsprechend eingestellt werden:

```
A>DEVICE CONOUT:=CRT
```

Die auszugebenden Charakter bzw. Steuerzeichen müssen sich im CPU-Register C befinden. Einfache Escape-Sequenzen lassen sich auch direkt im CCP aufrufen. Die gewünschte ESC-Sequenz ist wie

ein Programmname einzugeben, als Abschluß die Return-Taste. Der CCP sucht auf der Diskette nach dem vermeintlichen Programmnamen, findet ihn nicht und beantwortet dies mit der Ausgabe des Programmnamens auf die Console und einem Fragezeichen. Diese Ausgabe interpretiert das RGB-Termianl als entsprechende Esc-Sequenz und führt sie durch. Zudem besetzt die Möglichkeiten über 'ESC u' in den Local-Modus umzuschalten, d.h. die eingegebenen Zeichen werden direkt an das RGB-Terminal weitergeleitet

Die Befehle sind teilweise direkt über Control-Steuerzeichen 00h-1Fh oder über sogenannte Escape-Sequenzen (Esc = 1BH) gefolgt von mehreren Parametern aufrufbar. Zeichen größer 1Fh (20h-7Fh) werden als normale ASCII-Charakter interpretiert und sofort entsprechend dargestellt. Die Cursorposition bewegt sich dadurch um eine Position nach rechts bzw. am Bildschirmrand nach unten. Falls nötig findet am Bildschirmende automatisch ein Scrollen des Bildschirms statt.

TVI 950 - Kompatibilität:

Die im folgendem angegebenen Control bzw. Escape-Sequenzen sind die Standardcodes, die sich durch Modifikation des TVI950.ASM-Files jederzeit ändern lassen.

Es wurde versucht, ein handelsübliches Terminal mit der RGB-Software zu emulieren. Solche Emulationen sind leider nie 100 prozentig erfüllbar, da Funktionen eines abgesetzten Terminals (TVI950) in einem integrierten Terminal (RGB) nicht notwendig sind, andererseits sind Funktionen des RGB-Terminals auf TVI950 nicht erfüllbar. Die Kompatibilität erstreckt sich daher in der Regel nur auf die Editier- und Cursorpositionierkommandos. Zum Installieren von Anwenderprogrammen (z.B. WordStar) reicht diese weitgehende Übereinstimmung meist aus. Vorsicht ist bei den Funktionen Delete- und Insert-Line geboten, da diese vom RGB-Terminal nicht unterstützt werden. Die Gründe dafür liegen in der Struktur des uPD 7220, der im Graphic Mode diese Funktionen nicht erfüllt. Betreibt man dagegen den Controller im Alpha-Modus, so könnte man die leistungsfähigen Graphic-Funktionen des 7220 nicht benutzen.

Cursor Befehle:

Die Cursor Befehle dienen der Positionierung der augenblicklichen Schreibposition und ihrer Anzeige.

CHOME **^^ (1Eh)** Schreibposition linker oberer Bildschirmrand
Position (0,0)

CUP **^K (0Bh)** Schreibposition ein Zeichen nach oben (X-1,Y)

CRIGHT **^I (09h)** Schreibposition ein Zeichen nach rechts
^L (0Ch) (X,Y+1), 'Horizontal Tab HT'

CDOWN **^J (0Ah)** Schreibposition ein Zeichen nach unten
^V (16h) (X+1,Y), 'Linefeed LF'

CLEFT **^H (08H)** Schreibposition ein Zeichen nach links (X,Y-1), 'Backspace BS'

CRET **^M (0Dh)** Schreibposition zum Zeilenanfang (X,0),
'Carriage Return CR'

RNDCSR **Esc,=,x,y** Direkte Cursoradressierung auf die Position x für die Zeile (Row) und Spalte y (Column). Zu den gewünschten Werten für x und y ist jeweils der Hex-Wert 20h zu addieren. Die Eingabe für x und y hat in hex zu erfolgen und darf nicht über den Bildschirmrand hinaustreten. Im linken, oberen Bildschirmrand befindet sich die Zeichenposition (0,0), im rechten untern Rand die Position (23,84).

X = Row 20h - 37h (0 - 23)

Y = Clm 20h - 74h (0 - 84)

Beispiel: 1Bh, '=', 20h, 20h

Direkte Cursorpositionierung auf Position (0,0), entspricht Cursor home.

CRSBLK **-1, (FFh)** Dieses Steuerzeichen dient der Anzeige der Cursorposition durch Invertieren des Zeichens unter dem Cursor. Beim ersten Aufruf von CRSBLK wird das unter dem Cursor gelegene Zeichen invertiert dargestellt. Beim nächsten

Aufruf wird es wieder Invertiert, d.h. normal dargestellt. Durch entsprechende Verzögerungen im Aufruf der CRSBLK-Routine entsteht ein 'blinkender Cursor'. Folgt einem CRSBLK-Aufruf ein anders Steuerzeichen oder ein ASCII-Ausgabe-Zeichen so wird, falls nötig, die Invertierung des Cursor-Zeichens rückgängig gemacht, d.h. das Zeichen wird in jedem Fall wieder normal dargestellt.

Löschfunktionen:

Folgende Befehle löschen Teile des Bildschirms oder den gesamten Video-Ram mit sehr hoher Geschwindigkeit.

INIT	Esc,l	Initialisierung RGB-Modul, löschen Bildschirm, Cursor home, Zoom x 1
CLRSCR	Esc,+ Esc,; ^Z (1Ah)	Löschen des Bildschirms, Cursor home
CLRLN	Esc,R Esc,E ^R	Löschen der Cursor Zeile
CLREND	Esc,T Esc,t ^W (17h)	Löschen Rest der Zeile ab augenblicklicher Cursorposition
CLREOP	Esc,Y Esc,y	Löschen Rest des Schirms ab augenblicklicher Cursorposition
SCRON	Esc,n	Wiedereinschalten des Bildschirms während eines Bildaufbaus
SCROFF	Esc,o	Ausalten des Bildschirms

Zoom und Bildschirmausschnitt:

Zur Vergrößerung des Bildes steht die Hardware Zoom Funktion HZOOM und die Software Zoom-Funktion GZOOM zur Verfügung. Ist die Hardware Zoom Funktion gesetzt, so verringert sich der Bildschirmausschnitt entsprechend. Mit den vier Pan-Funktion läßt sich der Bildschirmausschnitt entsprechend verschieben. Die HZOOM-Funktion läßt sich einsetzen um sehr kleine Details auf dem Bildschirm vergrößert darzustellen. Mit Hilfe der Pan-Funktionen kann der Anwender wie durch ein Fenster Ausschnitte der Darstellung heranholen. Die GZOOM-Funktion schreibt die Zeichen mit den entsprechenden Zoom-Faktoren direkt vergrößert in den Video-Ram. Daher sind unterschiedliche Vergrößerungsfaktoren pro Bild zulässig. Ein neues GZOOM-Kommando schreibt alle nachfolgenden Zeichen mit dem gewünschten Zoom-Faktor in den Video-Ram, bis ein neues GZOOM-Kommando mit verändertem Parameter folgt.

HZOOM **Esc,v,x** Die Zoom Funktion HZOOM stellt den Hardware-Zoomfaktor ein. Die Werte für x sind '0' entsprechend ausgeschalteter Zoomfunktion (x1), '1' (x2), '2' (x4) und '3' (x8). Andere Werte sind unzulässig. Der Hardwarezoom vergrößert das dargestellte Bild in x und y-Richtung unter entsprechender Verkleinerung des Bildschirmausschnitts.

PAN **Esc,' ',x** Bildschirmausschnitt verschieben. Dabei muß ein dritter Parameter x angegeben werden, der die Verschiebungsrichtung bestimmt:

PANUP: x = 'U'

PANDOWN: x = 'N'

PANRIGHT: x = 'J'

PANLEFT: x = 'H'

GZOOM **Esc,m,x** GZOOM stellt den Zoomfaktor ein, mit dem die dem GZOOM-Kommando folgenden Zeichen vergrößert dargestellt werden. Der sog. 'Graphik Charakter Zoom' hat bis zum nächsten GZOOM Kommando Gültigkeit. Der Bildschirmausschnitt wird von diesem Befehl nicht verändert, da die Zeichen vergrößert in den Video-Speicher

geschrieben werden. Gültige Werte für den GZOOM-Parameter x sind 'A' bis 'P' entsprechend Zoomfaktoren 1 bis 16.

Softscroll:

Die Scrollgeschwindigkeit des Bildschirms läßt sich in Stufen in zwei Stufen einstellen. Beim Scrollen findet ein augenschonender Softscroll statt, d.h. das Bild wird Linienweise nach oben geschoben. Dadurch entsteht eine fließende Bewegung des Bildes um eine Zeile nach oben.

SMOTHON **Esc,8** Einschalten der Softscroll-Funktion

SMOTHOFF **Esc,9** Auschalten des Softscrolls (default)

Charakter Darstellung:

Die unten aufgeführten Kommandos beeinflussen die Darstellung der im weiterem ausgegeben ASCII-Zeichen. Beliebige Kombination der Charakterdarstellungen sind möglich. So können z.B. Zeichen vergrößert in cursiver Schrift, unterstrichen und einer Zeichenrichtung nach oben ausgegeben werden.

INVOFF **Esc,d** Normale Zeichenfarbe auswählen, dient zum schnellen Umschalten zwischen verschiedenen Zeichenfarben. Bei monochromen Geräten wird die Invers-Darstellung aufgehoben.

INVON **Esc,b** Zweite Zeichenfarbe anwählen. Schaltet NORM aus. Dient zum Hervorheben von Zeichen. Bei monochrom Einstellung werden die Zeichen invertiert dargestellt.

ATTRI **Esc,G,x** Character Attribute bestimmen. Der Parameter x bestimmt die gewünschte Art:

x = 0:	Löschen aller Attribute
x = 4:	Invers Video ON
x = 8:	Durchgezogen unterstreichen
x = 9:	Gestrichelt unterstreichen
x = A:	Cursive Schrift
x = B:	Gesperrte Schrift

Vereinfachte Beispiele:

*****	*****	*****	*****
*	*	*	*
*	*	*	*
*****	*****	*****	*****
*	*	*	*
*	*	*	*
*****	*****	*****	*****
	*****	***	***
Normal	Undscr	Undlin	Cursiv

COLOUR **Esc,w** Wechsel zwischen Farb- und Schwarz/Weiß-Darstellung. Im s/w-Modus werden die Zeichen schneller ausgegeben, da nur eine Farbe geschrieben werden muß. Ein Wechsel auf die zweite Zeichenfarbe im s/w-Modus stellt die Zeichen invertiert dar. Initialisierungswert ist der s/w-Modus.

GERMAN **Esc,z,x** Wahl des Zeichensatzes:

x = 0: Internationaler Standardzeichensatz nach ASCII

x = 1: Deutscher Zeichensatz. Von der Umschaltung sind die Umlaute ä, Ä, ü, Ü, ö, Ö, das scharfe 'ß' und das Paragraph-Zeichen betroffen.

ä <-> {	Ä <-> [
ü <-> }	Ü <->]
ö <->	Ö <-> \
ß <-> @	

DIR **Esc,x,x** Das Dir-Kommando legt die Zeichenrichtung fest. Die Werte für den Dir-Parameter x liegen zwischen 0 und 7:

- 0 Schreibrichtung nach unten
- 2 Schreibrichtung normal aufrecht rechts
- 4 Schreibrichtung nach oben
- 6 Schreibrichtung nach links umgekehrt

Vereinfachte Beispiele:

*****	*****	*	*
* *	*	* *	*
* *	*	* *	*
	*	*****	*****
	*		*
	*		*
	*		*****
Dir 0	Dir 2	Dir 4	Dir 6

Farbanwahl:

Zur Farbauswahl dienen vier Esc-Sequenzen die es gestatten, getrennt voneinander die Hintergrund-Farbe (z.B. Clear-Kommandos), die Darstellung des Cursors (CRSBLK-Kommando), die Normal-Schriftfarbe (NORM-Kommando) und die zweite Schriftfarbe (DOUBLE-Kommando) zu bestimmen.

CRSCOL	Esc,q,x,y,z	Darstellung des Cursors beim CRSBLK-Kommando
NRMCOL	Esc,r,x,y,z	Normal Schriftfarbe nach NORM-Kommando
DBLCOL	Esc,s,x,y,z	Zweite Schriftfarbe nach DOUBLE-Kom.
BAKCOL	Esc,*,x	Hintergrundfarbe definieren

Soll ein Zeichen auf das RGB-Modul ausgegeben werden, so werden im COLOUR-Modus (s. COLOUR-Kommando) bis zu drei getrennte Schreibbefehle durchgeführt. Im S/W-Modus hingegen findet beschleunigt ein Schreibbefehl mit festdefinierten Farbwerten statt. Die Farbparameter x,y,z entsprechen im Colour-Modus den drei Schreibbefehlen mit den jeweiligen Farbwerten an das RGB-Modul. Wird ein Zeichen ausgegeben, so wird Farbwert x angewählt

und das Zeichen in den Video-Speicher geschrieben. Danach folgt ein weiterer Schreibbefehl mit Farbwert y, schließlich Farbwert z. Beim Schreiben der Hintergrundfarbe BAKCOL wird nur die Angabe eines Farbwerts benötigt. Der angegeben Farbwert wird beim Löschen gesetzt, die restlichen Farben gelöscht.

Farbparameter:

Die Farbparameter wählen die gewünschte Farbe und die Darstellungsart der Charakter aus. Das Low-Byte der Parameter bestimmt die Farbebenen die modifiziert werden sollen. Im High-Byte des Parameters können bestimmte Optionen gesetzt werden, um z.B Hintergrundfarben zu bestimmen. Dabei sind beliebige Kombinationen an Farbebenen und Optionen zulässig.

Option 1 Invert:

Ist Bit 4 des Parameters gesetzt, so werden die Zeichen vor ihrer Ausgabe invertiert. Dies dient z.B. dem Hervorheben bestimmter Bildschirmteile.

Vereinfachtes Beispiel:

[illegible]

Normal

Invert

Option 2 Block Drawing:

Ist Bit 5 im Farbparameter gesetzt, so wird für diesen Schreibbefehl nicht das Äquivalent des zu zeichnenden ASCII-Charakters aus

dem Charakter-Eprom als Muster genommen, sondern ein einfacher 8x12 Block, indem alle Bits auf 1 gesetzt sind. In Verbindung mit der Invers-Funktion lassen sich somit bestimmte Farbebenen löschen, ander hingegen können gesetzt werden. In Verbindung mit der Komplement-Option (3) kann ein gesamtes Charakter komplementiert werden.

Option 3 Komplement:

Bit 6 im Farbparameter bestimmt die Art des WDAT-Cyclus im Schreibbefehl an den uPD 7220. Ist bit 6 auf 0 gesetzt, so wird der Inhalt des Video-Rams einfach mit dem neuen ASCII-Muster überschrieben (Replace). Ist Bit 6 auf 1 gesetzt, so werden diejenigen Stellen des alten Bitmusters invertiert, die im neuen Bitmuster auf 1 gesetzt sind (Complement).

Vereinfachtes Beispiel:

*****	*****	*****
*	*****	*** **
*	*****	*** *****
*****	*****	*** *****
*	*****	*** **
*	:	: *
*****	:	: ***** :
	:	:
	:.....	:.....
Inhalt des Video-Rams	Neues Bitmuster	Inhalt des Video-Rams nach Komplement

Farbparameter Bitbelegung:

Bit 0 = 0	Farbebene Blau wird beschrieben
Bit 1 = 0	Farbebene Grün wird beschrieben
Bit 2 = 0	Farbebene Rot wird beschrieben
Bit 3	keine Bedeutung, immer 0
Bit 4 = 1	Charakter werden vor dem Schreiben invertiert
Bit 5 = 1	Nicht das ASCII-Charakter, sondern ein 8x12 Block wird geschrieben
Bit 6 = 1	Charakter ersetzt nicht das bisherige Zeichen im Video-Ram, sondern komplementiert es
Bit 7	Keine Bedeutung, immer 0

Wird für einen der Farbparameter x,y,z ein Wert von OFEH gewählt, so wird der Schreibbefehl mit diesem Farbwert übersprungen und beim nächsten Farbwert fortgesetzt. Dies beschleunigt entsprechend die Ausgabe der Zeichen, da weniger Schreibbefehle gegeben werden müssen.

Bit 0 bis Bit 2 bestimmen die Farbebenen, die zum Schreiben freigegeben werden:

Bit 2: 1: 0:

0	0	0	rot/grün/blau	= weiß
0	0	1	rot/grün	= gelb
0	1	0	rot/blau	= purpur
0	1	1	rot	= rot
1	0	0	grün/blau	= indigo
1	0	1	grün	= grün
1	1	0	blau	= blau
1	1	1	-	= schwarz

Wird der Farbwert schwarz (Bit 0-2 auf 1) angewählt, so findet kein Schreibbefehl in den Video-Speicher statt, sein Inhalt wird somit nicht verändert.

Hinweise:

1. Nach Möglichkeit sollte jeder der drei Farbparameter x,y und z nur einen Farbwert (rot, grün oder blau) anwählen, da es sonst bei Farbwechseln (NORM, DOUBLE) zu unerwünschten Übertragungen eines Farbwerts in eine andere Farbebene kommen kann.
2. Sind Bit 4-6 des Farbparameters 0, so wird das Ascii-Charakter normal mit dem durch Bit 0-2 bestimmten Farbwert ausgegeben.
3. Beim Definieren der Cursor-Farbe (CRSCOL) sollte Bit 6 immer auf 1 stehen, da bei Komplement-Operationen der Inhalt des Video-Rams reproduzierbar ist, d.h. das Zeichen unter dem Cursor bleibt durch zweifache Anwendung des BLKCSR-Befehls erhalten.
4. Beim Definieren der Hintergrundfarbe (BAKCOL) haben Bit 4-6 keine Bedeutung.
5. Die Farbparameter werden nur im Farb-Modus berücksichtigt (s. COLOUR-Kommando). Im schwarz/weiß Modus wird mit erhöhter Geschwindigkeit ein festdefinierter Farbewert (gelb bei NORM, invertiert gelb bei DOUBLE, complement gelb bei BLKCSR, schwarz bei CLEAR) verwendet.

Beispiel 1: Esc, 'r', 0, FEh, FEh

Im Norm-Modus (NORM-Kommando) werden die Zeichen durch einen einzigen Schreibbefehl dargestellt, da zwei Farbparameter auf OFEh liegen. Der erste Farbparameter selektiert alle drei Farbenen, die Zeichen werden normal (d.h. keine Invertierung, kein Komplement, kein 8x12 Block Drawing) dargestellt und erscheinen weiß auf schwarzem Hintergrund.

Beispiel 2: Esc, 's', 03h, 35h, 16h

Nach der Umschaltung auf die zweite Schreibfarbe durch das DOUBLE-Kommando (Esc,B) werden die auszugebenden Zeichen rot mit blauen Hintergrund dargestellt:

x: 0000.0011 = 03h Beim ersten Schreibbefehl wird das Ascii-Zeichen normal in die Rot-Ebene geschrieben.

y: 0011.0101 = 35h Der zweite Schreibbefehl löscht das Grünfeld unter dem Charakter, da ein 8x12 Block komplementiert (Bit 5,4 = 1) in die Grün-Ebene geschrieben wird.

z: 0001.0110 = 16h Im dritten Schreibkommando werden die Ascii-Zeichen invertiert in die Blau-Ebene geschrieben (Bit 4 = 1).

Beispiel 3: Esc, 'q', 63h, 66h, FEh

Zur Darstellung des Cursors nach einem CRSBLK-Kommando werden die Rot- und Blauanteile des unter dem Cursor befindlichen Charakters komplementiert:

x: 0110.0011 = 63h Die Rot-Ebene wird durch einen 8x12 Block (Bit 5 = 1) komplementiert (Bit 6 = 1).

y: 0110.0110 = 66h Die Blau-Ebene wird ebenso komplementiert

Beispiel 4: Esc, '*', 05h

Die Hintergrundfarbe des Bildschirms ist mit dieser Esc-Sequenz auf grün festgelegt. D.h. beim Bildschirmlöschen wird die gesamte Grün-Ebene gesetzt, die Rot- und Blau-Ebene gelöscht.

Cursor Darstellung:

CURFMT Esc, ., x Dieses Kommando beeinflusst die Darstellung des Cursors. Der Parameter x bestimmt dabei die Funktion:

- x = 0: Keine Cursor Darstellung
- x = 1: Blinkender Blockcursor
- x = 2: Stehender Blockcursor
- x = 3: Blinkender Unterstrich
- x = 4: Stehender Unterstrich

Sonstige Befehle:

LOCAL **Esc,U** Loakales Editieren ein, d.h. Zeichen von der Konsole werden direkt an das RGB-Modul übergeben. Die Wirkung der Funktionstasten ist direkt überprüfbar.

DUPLEX **Esc,u** Beenden des lokalen Editierens.

BELL **^G (07h)** Einschalten des Signaltons am Terminal.

Direkte RGB-Modul Programmierung:

Um die vielfältigen Graphik-Funktion des uPD 7220 (Kreis, Rechteck, Area-Fill, Vector Plott) dem Anwender direkt zugänglich zu machen, wurden folgende zwei Esc-Sequenzen implementiert.

CMDOUT **Esc,c,x** Kommando x an uPD 7220

PAROUT **Esc,p,x** Parameter x an uPD 7220

Zur Ausgabe eines Kommandos an den GDP dient die CMDOUT-Sequenz. Zuerst wird geprüft, ob der GDP bereit ist ein Kommando zu empfangen, danach erfolgt die Übergabe an den entsprechenden Port. Die dem Kommando evt. folgenden Parameter werden der PAROUT-Routine übergeben. Pro Parameter ist ein neuer Aufruf der PAROUT-Routine erforderlich. Eine detaillierte Beschreibung der möglichen Kommandos und ihre Parameter finden sie im Handbuch des RGB - Farbgrafik Video Controllers.

Der TVI950.ASM - File:

Sollte es notwendig sein die Control- und Escape-Sequenzen auf Grund bestehender Software ändern zu müssen, so geschieht dies mit Hilfe eines Editors durch Modifikation des mitgelieferten TVI950-Source-Files. Die Reihenfolge der einzelnen Kommandos darf nicht verändert werden, nur die einzelnen Steuercodes dürfen umbenannt werden. Doppelbezeichnungen sind zu vermeiden. Neue unbekannte Codes können natürlich nicht eingefügt werden. Nach der Modifikation ist der File zu assemblieren und mit dem übrigen Bios neu zu linkern.

```
;*****
```

```
; CONTROL CHARACTER DEFINITIONS
```

```
;*****
```

```
07          db 07h          ; Bell = ^G
08          db 08h          ; Cursor left = ^H
0C          db 0ch          ; Cursor right = ^L
16          db 16h          ; Cursor down = ^V
0B          db 0bh          ; Cursor up = ^K
1E          db 1eh          ; Cursor home = ^^
1A          db 1ah          ; Clear screen = ^Z
0D          db 0dh          ; Carriage return = ^M
0A          db 0ah          ; Line Feed = ^J
09          db 09h          ; Tab = ^I
17          db 17h          ; Clear to end of line = ^W
12          db 12h          ; Clear current line = ^R
```

```
;*****
```

```
; ONE LETTER ESCAPE SEQUENCES
```

```
;*****
```

```
000D 2B     db '+'          ; clr screen
000E 3B     db ';'          ; clr screen
000F 52     db 'R'          ; clear line
0010 45     db 'E'          ; clear line
0011 54     db 'T'          ; clear EOL
0012 74     db 't'          ; clear EOL
0013 59     db 'Y'          ; clear EOP
0014 79     db 'y'          ; clear EOP
0015 38     db '8'          ; smoth scroll on
0016 39     db '9'          ; smoth scroll off
0017 64     db 'd'          ; invers video off
0018 62     db 'b'          ; invers video on
0019 6E     db 'n'          ; screen on
001A 6F     db 'o'          ; screen off
001B 55     db 'U'          ; local edit
001C 75     db 'u'          ; full duplex edit
001D 6C     db 'l'          ; full init
001E 77     db 'w'          ; black & white / colour
```

```
;*****
```

```
; MULTI LETTER ESCAPE SEQUENCES
```

```
;*****
```



```

7A db 'z'           ; Characters, '0'=int., '1'=german
47 db 'G'           ; Attributes, more under character attributes
20 db ' '           ; Panning, see more under pan directions
3D db '='           ; Direct Cursor adressing, first row, then clm
76 db 'v'           ; Hardware zooming, from '0' (1x) to '3' (8x)
6D db 'm'           ; Software Zooming, from 'A' (1x) to 'P' (16x)
70 db 'p'           ; Parameter to GDP
63 db 'c'           ; Command to GDP
71 db 'q'           ; Set Cursor Colour, 3 parameters
72 db 'r'           ; Set Normal Colour, 3 para.
73 db 's'           ; Set reverse Colour, 3 para.
2A db '*'           ; Set Background Colour, 1 para.
78 db 'x'           ; Set drawing direction, from '0' to '7'
2E db '.'           ; Set cursor format, more under cursor fmt.

```

```

;*****

```

; CHARACTER ATTRIBUTE IDENTIFIER

```

;*****

```

```

                attrchr:      ; Attribute identifier
002D 30          db '0'      ; Clear all character attributes
002E 34          db '4'      ; Invers Video On
002F 38          db '8'      ; Underline with straight line
0030 39          db '9'      ; Underline with dotted line
0031 41          db 'A'      ; Print slanted character
0032 42          db 'B'      ; Space each 2nd character

```

```

;*****

```

; PANNING DIRECTIONS

```

;*****

```

```

                panchr:      ; Pan directions
0033 55          db 'U'      ; UP
0034 4E          db 'N'      ; DOWN
0035 4A          db 'J'      ; RIGHT
0036 48          db 'H'      ; LEFT

```

```

;*****

```

; CURSOR DISPLAY AND FORMAT

```

;*****

```

```

                curschr:     ; Define cursor format
0037 30          db '0'      ; no cursor dispaly
0038 31          db '1'      ; blinking block cursor
0039 32          db '2'      ; steady block cursor
003A 33          db '3'      ; blinking underline cursor
003B 34          db '4'      ; steady underline cursor

```